UDC 004.932.2:004.93'1

APPROACH TO THE AUTOMATIC CREATION OF AN ANNOTATED DATASET FOR THE DETECTION, LOCALIZATION AND CLASSIFICATION OF BLOOD CELLS IN AN IMAGE

Kovalenko S. M. – PhD, Associate Professor, Associate Professor of the Department of Software Engineering and Management Intelligent Technologies of the National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Kutsenko O. S. – Dr. Sc., Professor, Professor of the Department of System Analysis and Information-Analytical Technologiesof the National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Kovalenko S. V. – PhD, Associate Professor, Professor of the Department of System Analysis and Information-Analytical Technologiesof the National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

Kovalenko A. S. – Postgraduate student of the Department of System Analysis and Information-Analytical Technologies of the National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine.

ABSTRACT

Context. The paper considers the problem of automating the creation of an annotated dataset for further use in a system for detecting, localizing and classifying blood cells in an image using deep learning. The subject of the research is the processes of digital image processing for object detection and localization.

Objective. The aim of this study is to create a pipeline of digital image processing methods that can automatically generate an annotated set of blood smear images. This set will then be used to train and validate deep learning models, significantly reducing the time required by machine learning specialists.

Method. The proposed approach for object detection and localization is based on digital image processing methods such as filtering, thresholding, binarization, contour detection, and filling. The pipeline for detection and localization includes the following steps: The given fragment of text describes a process that involves noise reduction, conversion to the HSV color model, defining a mask for white blood cells and platelets, detecting the contours of white blood cells and platelets, determining the coordinates of the upper left and lower right corners of white blood cells and platelets, calculating the area of the region inside the bounding box, saving the obtained data, and determining the most common color in the image; filling the contours of leukocytes and platelets with said color; defining a mask for red blood cells; defining the contours of red blood cells; determining the coordinates of the upper left and lower right corners of red blood cells; calculating the area of the region within the bounding box; entering data about the found objects into the dataframe; saving to a .csv file for future use. With an unlabeled image dataset and a generated .csv file using image processing libraries, any researcher should be able to recreate a labeled dataset.

Results. The developed approach was implemented in software for creating an annotated dataset of blood smear images

Conclusions. The study proposes and justifies an approach to automatically create a set of annotated data. The pipeline is tested on a set of unlabelled data and a set of labelled data is obtained, consisting of cell images and a .csv file with the attributes "file name", "type", "xmin", "ymin", "xmax", "ymax", "area", which are the coordinates of the bounding box for each object. The number of correctly, incorrectly, and unrecognised objects is calculated manually, and metrics are calculated to assess the accuracy and quality of object detection and localisation.

KEYWORDS: computer vision, object detection, object localization, digital image processing, classification.

ABBREVIATIONS

CNN is a convolutional neural network;
CSV are comma separated values;
CV is a computer vision;
FN is a false negative;
FP is a false positive;
HSV are hue, saturation, value
RBC is a red blood cell;
RGB is a red, green, blue;
RNN is a recurrent neural networks
TP is a true positive;
WBC is a white blood cell.

NOMENCLATURE

$\delta(a,b)$ is a Kronecker delta;
σ is a standard deviation of the Gaussian distribution;
Acc ia accuracy of cell recognition;
B is a blue component in the RGB model;
C is a most common contour in the image;
Cnt is a contour, found in the image;
G_f is a Gaussian filter;
G is a green component in the RGB model;
$G(u,v)$ is a Gaussian distribution at position (u,v) ;
H is hue is a component in the HSV model;
*F*1 is a harmonic mean between precision and recall;
I is an image;
$I(x,y)$ is a value of a pixel in the image *I* at position (x,y);
$M(x,y)$ is a value of the pixel on the mask at position (x,y);
Pr is a precision of cell recognition;
R is a red component in the RGB model
Rec is a recall of recognition;
r is a blur radius;
S is a saturation is a component in the HSV model;
T is a threshold value;
V is a value component in the HSV model;
(x,y) are coordinates of the pixel in the image.

DOI 10.15588/1607-3274-2024-1-12

INTRODUCTION

With the development of machine learning models, this branch of artificial intelligence is increasingly penetrating various areas of our lives and is transforming many aspects of the modern world. This technology is reprogramming and modernizing various sectors, from industry and medicine to education and everyday life.

Artificial intelligence can effectively solve the tasks of detecting, localizing, classifying, and counting cells in an image, which is an important task in the field of medicine and biology [1]. It is an important stage in the process of counting the number of blood cells of different types. This procedure, in turn, is an important step in laboratory diagnostics and medical research, which provides valuable data for assessing the health status of patients and detecting various diseases.

Currently, convolutional neural networks and deep learning methods are used to identify and classify blood cells, which are essentially supervised learning methods and require large enough data sets for training and validation. The general pipeline for the machine learning process is shown in Fig. 1.

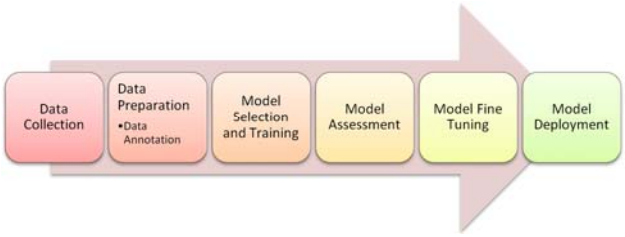


Figure 1 – General scheme of machine learning steps

Sometimes it is believed [2] that machine learning specialists spend the most time training the model, but a study [3] found that data scientists spend approximately 37.75% of their time preparing and cleaning data. In addition to data preparation and cleaning, interpretation of the results also remains important. Data visualization (12.99%) and demonstrating the value of data through reporting and presentation (16.20%) are also important and time-consuming steps to turn data into an actionable tool for answering critical questions. Working with models through model selection, training, and implementation takes approximately 26.44% of respondents' time (Fig. 2).

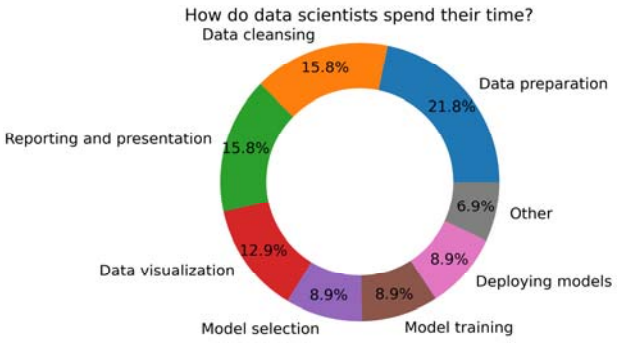


Figure 2 – Time management for machine learning specialists

Thus, the creation of datasets is in itself a rather time- and resource-consuming task. Such datasets are created manually using certain software, such as LabelIMG, VIA, VOTT, and others. Image annotation software has different functionality [4] and can significantly reduce the annotation time compared to simple graphic editors, but still the process of annotating images is a very long process.

Therefore, the automatic creation of annotated datasets for training blood cell classification models will help to significantly reduce the time spent by machine learning specialists.

The object of the study is the processes of digital image processing for the detection and localization of blood cells of various types in a blood smear image.

The subject of the study is methods and algorithms for digital image processing.

The purpose of the work is to increase the overall speed of the deep learning process for detecting, localizing, and classifying blood cells by automatically creating an annotated data set.

1 PROBLEM STATEMENT

The study aims to develop and validate an approach for automatically creating an annotated dataset for classifying, identifying, and localizing blood cells in images.

The result of the study is a pipeline for automatic creation of a labeled dataset based on methods of digital image processing and analysis, noise reduction, contour extraction, binarization, filtering, etc. Pipeline should provide the ability to detect and localize cells in blood smear images, taking into account their features and characteristics.

This also includes the development of software for detecting, isolating, analyzing, and localizing blood cells and the creation of a dataset. The output dataset will be defined as a set of unlabeled images and a .csv file containing the following features:

– "file_name" is the relative name of the file in the folder;

– "xmin", "ymin" are coordinates of the upper left corner of the bounding box;

– "xmax", "ymax" are coordinates of the lower right corner of the bounding box;

– "type" is cell type ("WBC" for leukocyte, "RBC" for erythrocyte, and "platelet");

– "area" is the area inside the bounding box.

With an unlabeled dataset and a generated .csv file using image processing libraries (e.g. PIL and/or opencv-python for Python programming language), any researcher should be able to recreate a labeled dataset.

The study focuses on improving the speed and efficiency of the dataset generation process, ensuring the representativeness of various cells and optimal image quality. The dataset obtained with the developed approach will be used to train and validate a machine learning model that will recognize blood cells in smear images with high accuracy and reliability.

DOI 10.15588/1607-3274-2024-1-12

2 REVIEW OF THE LITERATURE

Rapid assessment of the body's condition using less invasive laboratory tests has always been relevant. A regular clinical blood test is a fairly accurate and reliable method of assessing the patient's condition and making a diagnosis. It provides important data for assessing the state of health of patients and early detection of possible diseases.

Fig. 3 presents the main areas in which blood cell counting is used and, accordingly, their classification.

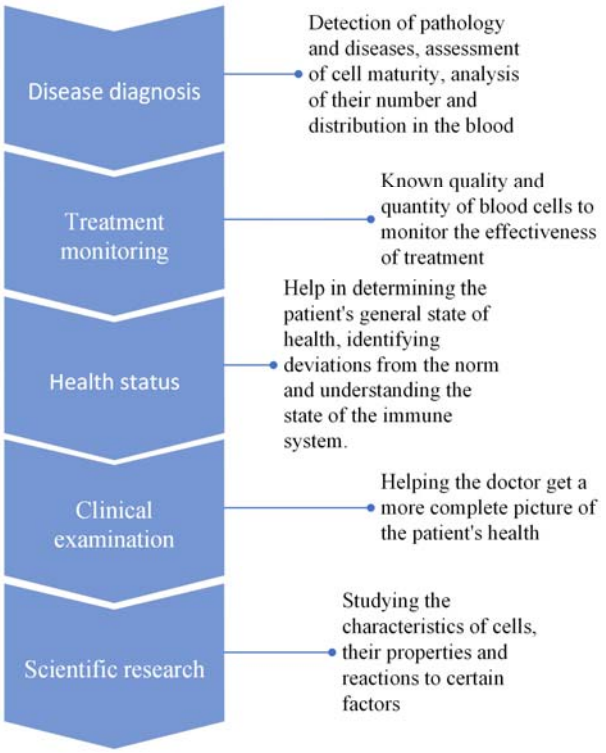


Figure 3 – Fields of application of blood cell classification

Thanks to advances in computer vision and machine learning, the use of automated systems for this task is becoming more common and affordable.

One of the most common tasks is to detect and recognize different types of cells, such as red blood cells (erythrocytes), red blood cells (leukocytes), and platelets.

As mentioned earlier, the classification and counting of blood cells of different types in the image is broad used in diagnosis and monitoring of diseases and treatment effectiveness, medical and scientific research, optimization of processes in laboratories.

There are several approaches to solving this problem. The first of them is the so-called traditional method [5, 6, 7], which combines image processing methods, such as noise reduction, contour extraction, edge detection, segmentation and filtering [8, 9], with machine learning or morphological analysis methods. This approach can be used both to detect different types of blood cells and to detect other biological objects [7]. These methods allow you to significantly speed up the process of object detection (in comparison with non-automated systems) and provide the ability to use them in real time. However, they depend on the quality of the input image and require preprocessing.

Another approach is based on the use of real-time object detection algorithms, such as YOLO (You Only Look Once) [10, 11] and/or SSD (Single Shot MultiBox Detector) [11]. This method has become quite popular in image classification in recent years because it has high accuracy and the ability to work with different, complex data, such as different resolutions and overlapping object detection. The main feature of this approach is to identify objects in one pass of the image through a neural network. For example, in the case of YOLO, an image is passed through a neural network at once, deriving predicted object boundaries and classes, instead of the traditional approach of first passing the image through an object detection model and then processing the results to determine object boundaries.

However, such methods require a large amount of labeled data for training, which is a separate task. This approach requires large computing power and has a large model size, which can lead to difficulties in storing, implementing and distributing these models. These difficulties can be circumvented by using mechanisms such as transfer of learning [12]. But the problem of having a set of labeled data remains. The size of the dataset can be reduced due to data augmentation [13], but the process of annotation different types of cells still remains time-consuming, especially in the case of determining different types of cells (erythrocytes, leukocytes, platelets).

In [2], an approach to the segmentation of blood cells based on image processing methods is proposed. In this study, it is proposed to apply a similar approach to the automatic creation of a set of annotated data for further application in the automatic detection different types of cells using artificial neural networks.

Machine learning models are usually divided into 3 large groups: unsupervised, supervised, and reinforcement learning models [14] (Fig. 4).

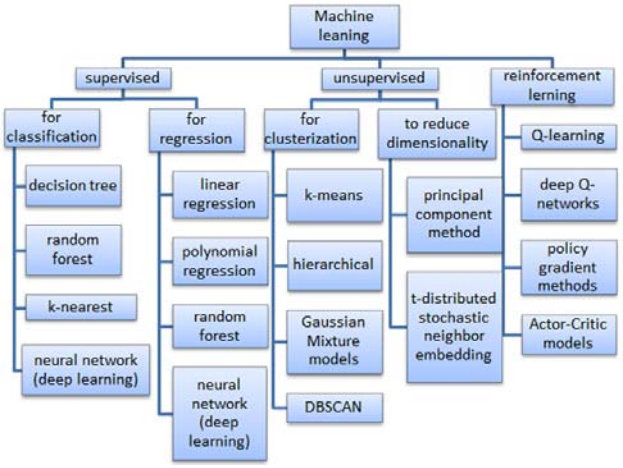


Figure 4 – Classification of machine learning models

DOI 10.15588/1607-3274-2024-1-12

One of the fields of artificial intelligence that uses machine learning is computer vision [15]. Computer vision is a branch of artificial intelligence that studies how computers can understand and analyze images and videos. To achieve this goal, various methods and algorithms are used that allow computers to "see" the world around them.

The main tasks of computer vision are [16] (Fig. 5)

1. Image classification: determining the class and type of the object in the image.

2. Object localization: determining the location of the object and selecting it (usually using a rectangular frame).

3. Object detection: determining the location (using frames) and the type and class of objects in the image.

4. Semantic segmentation of the object with selection of the recognized object using a contour and (sometimes) changing certain pixels of the object instead of defining the frame.

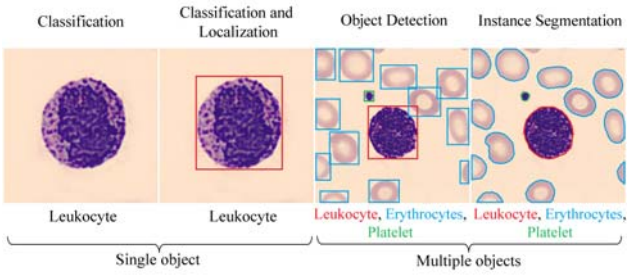


Figure 5 – Classes of machine vision problems

All of these tasks use supervised machine learning techniques to train models based on annotated data and use them in computer vision tasks.

One of the key methods in computer vision is Convolutional Neural Networks (CNN) [15, 17]. These networks are capable to detect patterns and features in images using filters that perform a convolution operation. CNNs are used for object classification, contour detection, face recognition, and many other tasks in computer vision.

Another important method is Recurrent Neural Networks (Recurrent Neural Networks – RNN) [17] that allow you to work with sequences of data, for example, with sequences of video frames. They are useful for analyzing the dynamics of actions or the movement of objects.

In addition, image processing methods are used, such as filtering, segmentation, detection of edges and contours, as well as feature extraction, which help in the recognition of objects in images [2].

Deep learning techniques [17, 18], in particular, are very popular in computer vision. They allow creating complex models that can automatically study the data presented to them, and identify complex patterns and features to solve recognition and classification tasks. Also, geometric modeling methods are used to describe the geometric features of objects in images, which help in solving problems of object recognition and measurement of their characteristics [6, 19].

In general, the combination of these methods and techniques in computer vision allows computers to recognize objects, detect images, classify data, and helps solve many tasks that require the analysis of visual information in real or virtual environments.

That is, different types of neural networks using machine learning methods used in computer vision to recognize objects, detect images, classify data and help solve many tasks that require the analysis of visual information in a real or virtual environment.

3 MATERIALS AND METHODS

To automatically create a set of annotated data, the following sequence of actions is proposed (Fig. 6):

1. Loading the image.
2. Noise reduction.
3. Conversion to HVS model.
4. Determining the mask for leukocytes and platelets.
5. Detection contours of leukocytes and platelets
6. Determining the coordinates of the upper left and lower right corners of leukocytes and platelets.
7. Calculation of the area of the region inside the bounding box.
8. Entering data about found objects into a dataframe with columns "file_name", "xmin", "ymin", "xmax", "ymax", "type", "area".
9. Determining of the most common color *C* in the image.
10. Filling the contours of leukocytes and platelets with color *C*.
11. Determining the mask for erythrocytes.
12. Detection the contours of erythrocytes.
13. Detection the coordinates of the upper left and lower right corners of erythrocytes.
14. Calculation of the area of the region inside the bounding box.
15. Entering data about found objects into the dataframe.
16. Saving in .csv file for later use.

To eliminate excess noise that interferes with the recognition of target elements in the image, a Gaussian filter is applied to reduce the sharpness of the image and make it more blurry. This effect is achieved by transforming image pixels. In order to calculate the degree of transformation, this filter uses a normal distribution for two measurements, which can be calculated by formula (1):

$$G(u,v) = \frac{1}{2\pi\sigma^2} e^{-\left(x^2+y^2\right)/\left(2\sigma^2\right)}, \qquad (1)$$

where r is a blur radius; $r = x^2 + y^2$; σ is the standard deviation of the Gaussian distribution.

The result of calculations according to this formula is an area of the image that has the form of concentrated circles with a normal distribution from the central point. Pixels with a distribution level that is not equal to 0 are used to create a convolution matrix that is used to analyze the input image. The distribution of the Gaussian value is

DOI 10.15588/1607-3274-2024-1-12

as follows: one of the pixels receives the largest weight, while the surrounding pixels, according to their distance from the central one, receive smaller weights. The convolution (core) of the mask is usually much smaller than the entire image. The mask is moved across the image and convolution is applied at each location in the image. The larger the core size, the lower the sensitivity of the detector to noise. The greater the value σ, the smoother the result with less noise. However, with more image smoothing, fewer edges will be detected by the detector.

The algorithm is sequentially implemented for each pixel. A 3×3 or 5×5 cores are often used. An example of a Gaussian filter with a standard deviation $\sigma = 1.4$, is calculated using the kernel:

$$G_f = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}.$$

The next step is to convert from the RGB color model to the HSV model. The HSV (Hue, Saturation, Value) model is a cylindrical color model used to represent colors in the form of three components: Hue, Saturation, and Value. This model is built on the basis of a transformation from the RGB (Red, Green, Blue) model.

Converting an image from RGB to HSV can simplify color processing and allow various color operations to be performed more efficiently [11]. Converting images from RGB to HSV can make a number of image processing tasks easier and allow us to better understand and work with colors.

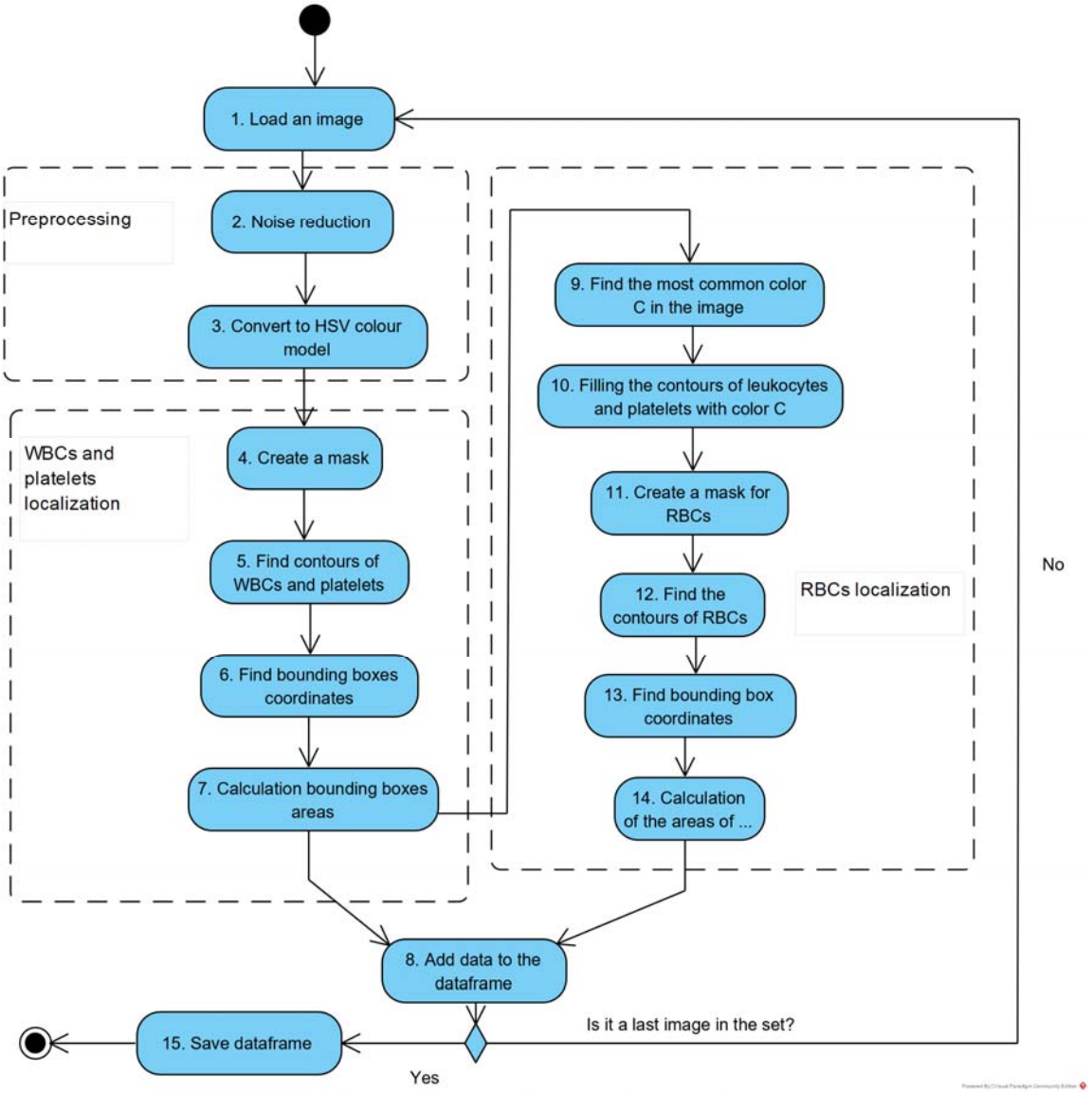


Figure 6 – Activity diagram of automatic creation of an annotated dataset

DOI 10.15588/1607-3274-2024-1-12

Converting from the RGB to HSV color model is often used to process images with color information for several reasons (Table 1).

Table 1 – Advantages of image processing in HSV

Advantage	Explanation
Easy color interpretation	In the HSV model, each parameter (hue, saturation, value) represents a specific color characteristic, which simplifies color interpretation. For example, the hue component defines the color tone, saturation defines the brightness of the color, and value defines the brightness of the light source.
Easier color processing	Switching from RGB to HSV can make it easier to process colors and perform a variety of color operations, such as changing hue, increasing or decreasing saturation, adjusting brightness, and more.
Operations with colors	Converting to HSV allows us to perform color-related operations, such as filtering by specific colors, finding color ranges for specific effects, and more.
Color segmentation	Since the HSV model is closer to how humans perceive colors, it can be useful for segmenting objects in an image by color. For example, to distinguish objects of a certain color or group of colors.
Brightness and contrast control	Changing the values in the HSV model allows for better control of image brightness and contrast, which can be useful for enhancing an image or increasing its clarity.

An HSV model can be represented as a cylindrical coordinate system, where hue is located on a circular disk, and saturation and value are displayed on the vertical and radial axes, respectively. This model makes it easier to understand and manipulate individual aspects of color, compared to the RGB model. This is how this model is built:

1. Hue is measured on the color wheel and expressed in degrees, where 0° corresponds to red, 120° to green, and 240° to blue. It is a quantitative value that defines a specific color in the spectrum.

2. Saturation is defined as the distance from the center of the color wheel to its edge. It indicates the "purity" or "vibrancy" of the color. Maximum saturation means a vibrant, rich color, while low saturation can lead to grayscale.

3. Value measures the brightness of a color. This is the relative brightness of a color from black to white. This is an important aspect because it determines how light or dark the color will be.

To convert from RGB to HSV, the following ratios are used (2): in the case of 8-bit and 16-bit images, R, G, and B are converted to the floating-point format and scaled to fit the 0 to 1 range:

$$V = \max(R,G,B),$$
$$S = \begin{cases} \dfrac{V - \min(R,G,B)}{V}, & \text{if } V \neq 0, \\ 0, & \text{otherwise}, \end{cases}$$
$$H = \begin{cases} 60(G-B)/(V-\min(R,G,B)), & \text{if } V = R, \\ 120 + 60(B-R)/(V-\min(R,G,B)), & \text{if } V = G, \\ 240 + 60(R-G)/(V-\min(R,G,B)), & \text{if } V = B, \\ 0, & \text{if } R = G = B. \end{cases} \quad (2)$$

If $H < 0$ then $H = H + 360$. On output $0 \le V \le 1$, $0 \le S \le 1$, $0 \le H \le 360$.

The values are then converted to the destination data type (8-bit images):

$$V = 255V, \quad S = 255S, \quad H = H/2 \text{ (to fit to } [0, 255]).$$

Obtaining a mask on an image can be mathematically written using threshold binarization, where the pixel value is compared to a certain threshold and a certain range of values is set to create the mask.

We have an image I. To get a mask M, where the white pixels correspond to contour areas, a threshold T must be determined, which will be used for binarization.

Mathematically, the binarization process can be written as (3):

$$M(x,y) = \begin{cases} 255, & \text{if } I(x,y) > T, \\ 0, & \text{otherwise}, \end{cases} \quad (3)$$

where $M(x,y)$ is the value of the pixel in the mask at the position (x,y); $I(x,y)$ is the value of a pixel in the image I in position (x,y), T is the threshold for binarization.

Specifying the threshold T for binarizing the image and create a mask is an important step that determines which pixels are considered part of the contour area. If the threshold value is too low, there may be a lot of "noise" or background detail in the contour mask. If the threshold value is too high, part of the contour may be lost.

Therefore, the choice of threshold depends on the specific image and its properties. There are several approaches to select a threshold:

1. Global threshold using statistical methods, such as the Otsu method, which automatically determines the threshold by maximizing the variance between pixel classes (background and object).

2. Adaptive threshold, where instead of one global threshold, you can use an adaptive threshold for different areas of the image, especially when the illumination in the image is uneven, i.e. use different thresholds for different parts of the image.

3. Experimental selection, when several threshold values are examined and the one that gives the best results for a specific task is chosen.

4. Using data knowledge, where knowledge of image characteristics or regions to be detected is used.

In our study, a combined approach was used to determine the threshold value, namely a combination of an experimental approach and knowledge of the data. It is known [20] that Leishman staining is used to stain blood on smears for microscopy. It is usually used to distinguish and identify leukocytes. It is based on a methanol mixture of "polychrome" methylene blue and eosin, which stain leukocytes and platelets purple, while erythrocytes remain pink. Based on this knowledge and experiment, the threshold value was determined as $T = 180$.

DOI 10.15588/1607-3274-2024-1-12

To detect and localize RBC, it is proposed to first "fill" the contours of leukocytes and platelets with the color that is most present in the image. The following algorithm is used for this purpose:

Let us denote *H*, *S*, and *V* as the sets of possible values of hue, saturation, and brightness values, respectively. Based on the features of the opencv-python library [21], which is used for image processing in this study, each of the values of *H*, *S* and *V* varies in the range [0, 255].

Let's calculate the amount of each color in the HSV space. For this, we will create a matrix $count\_hsv_{256\times256\times256}$ to count the number of pixels of each color. To do this, use the formula (4):

$$count\_hsv\left(H_i,S_j,V_k\right)= \sum_{x,y\in I}\delta\left(I_H\left(x,y\right)=H_i \,\&\, I_S\left(x,y\right)=S_j \,\&\, I_V\left(x,y\right)=V_k\right), \quad (4)$$

where $I_H\left(x,y\right)$, $I_S\left(x,y\right)$ and $I_V\left(x,y\right)$ are values of hue, saturation and brightness of a pixel $\left(x,y\right)$ in the image I; H_i, S_j, V_k are current values of hue, saturation and brightness; $i,j,k\in[0,255]$; $\delta\left(a,b\right)$ – the Kronecker delta (5):

$$\delta\left(a,b\right)=\begin{cases}1, \text{ if } a=b,\\ 0, \text{otherwise.}\end{cases} \quad (5)$$

Finding the most common combination of *H*, *S* and *V* is represented by formula (6):

$$C=\left\{H,S,V\right\}= \arg\max_{\left(H_i,S_j,V_k\right)\in H\times S\times V} count\_hsv\left(H_i,S_j,V_k\right). \quad (6)$$

The next step is to select the contours. There are several ways to accomplish this task. One of the most common is to use of mathematical morphology operations and find contours based on a mask. The contours can be obtained as the difference between the extended object and the (initial) object itself.

In turn, the dilation operator accepts two pieces of data as input. First, is the image to be expanded. The second is a (usually small) set of coordinate points known as a structural element. It is this structural element that determines the exact dilation effect on the input image. To calculate the dilation of the input image using this structuring element, we consider each background pixel in the input image in turn. For each background pixel, the structural element is superimposed over the input image so that the origin of the structural element coincides with the position of the input pixel. If at least one pixel in the structural element matches a foreground pixel in the image below it, then the input pixel is set to the foreground value. However, if all matching pixels in the image are background, the input pixel remains.

These mathematical operations help to extract contours in an image based on a binary mask.

To calculate the bounding box that wraps the contour, we need to find the minimum rectangle that joins all points of the contour.

Let *Cnt* be a contour found in the image consisting of an array of points with coordinates $\left(x_i,y_i\right)$, $i=1,2,\ldots,n$, hat form a line or a closed shape of an object. We have $Cnt=\left\{\left(x_1,y_1\right),\left(x_2,y_2\right)\right\}$.

Mathematically, a bounding box consists of coordinates $\left(x_1,y_1\right)$ of upper left corner and $\left(x_2,y_2\right)$ of lower right corner. That is, to find the bounding box, it is necessary to find the smallest and largest coordinate values *x* and *y* among all points on the contour. We have the value calculated by formula (7):

$$Cnt=\left\{\left(\min\left(x_i\right),\min\left(y_i\right)\right),\left(\max\left(x_i\right),\max\left(y_i\right)\right)\right\}, \quad i=1,2,\ldots,n. \quad (7)$$

We will store the data in a csv file with the following features:

"file_name" is the relative name of the file in the folder;

"xmin", "ymin" are the coordinates of the upper left corner of the bounding box;

"xmax", "ymax" are the coordinates of the lower right corner of the bounding frame;

"type" is the cell type (WBC for leukocyte, RBC for erythrocyte, and platelet)

"area" is the area inside the bounding box.

4 EXPERIMENTS

The developed pipeline was tested on images of the dataset [2, 22]. This dataset is released under the CC BY 4.0 license and consists of JPG images with dimensions of 360×363 pixels.

The Python programming language in the Jupyter notebook environment was used for image processing. Third-party libraries were also used:

– PIL and opencv-python for image processing;

– matplotlib, seaborn for plotting graphs and diagrams;

– pandas for creating and processing a set of structured data;

– numpy – for processing images as arrays of pixels.

Fig. 7 shows the transformations that blood smear images undergo after each step of the algorithm.

DOI 10.15588/1607-3274-2024-1-12

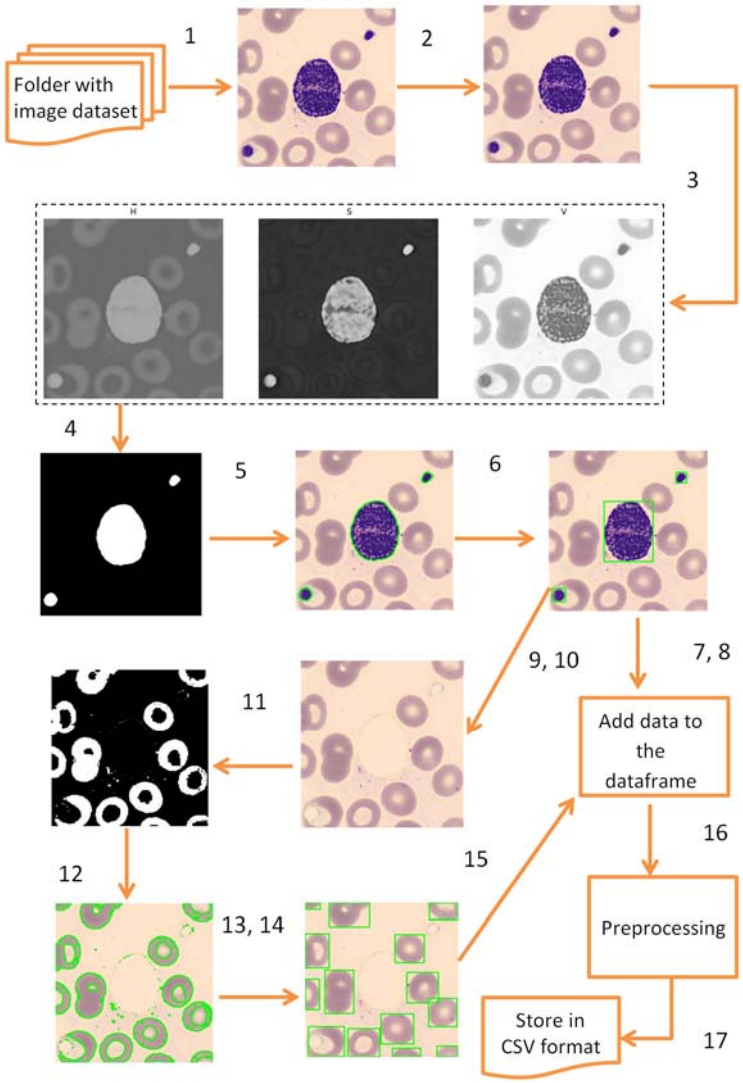


Figure 7 – Share of cells of each type of cells in the total number of detected objects

5 RESULTS

A total of 2159 RBCs, 177 WBCs, and 148 platelets were detected and localized, as shown in Fig. 8.

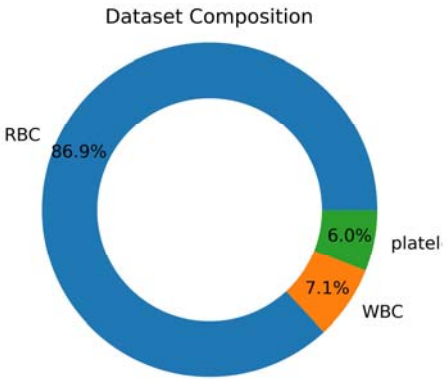


Figure 8 – Share of cells of each type of cells in the total number of detected objects

As a result, we receive a file in csv format, in which the coordinates of the bounding boxes, cell types on certain images are stored (Fig. 9).

	file_name	xmin	ymin	xmax	ymax	type	area
0	BA_206112.jpg	227	307	248	327	platelet	420
1	BA_206112.jpg	141	298	165	322	platelet	576
2	BA_206112.jpg	87	251	103	268	platelet	272
29	BA_208485.jpg	247	135	259	181	platelet	552
55	BA_209608.jpg	266	349	285	362	platelet	247

Figure 9 – Structure of the annotation dataframe

It took 9 minutes and 51 seconds to process 1000 images and create the resulting annotated dataset using the computer with Windows 10 operating system, 8GB of RAM, and an AMD Rizen 5 4500U processor using Python 3.10.0, pandas 1.3.5, and opencv-python 4.9.0.

DOI 10.15588/1607-3274-2024-1-12

6 DISCUSSION

To assess the quality of automatic annotation, bounding boxes were added to images and 160 images were processed manually.

The following cases were obtained during processing:

a) cells were recognized correctly;
b) incorrectly recognized leukocytes;
c) unrecognized leukocytes;
d) incorrectly recognized platelets;
e) unrecognized platelets;
f) incorrectly recognized erythrocytes;
g) unrecognized erythrocytes.

Of course, cases b)-g) can be present in different combinations.

Fig. 10 shows most common cases: a) all objects are recognized; b) the presence of incorrectly recognized erythrocytes; c) recognition of overlapping erythrocytes as one object; c) the presence of unrecognized erythrocytes.

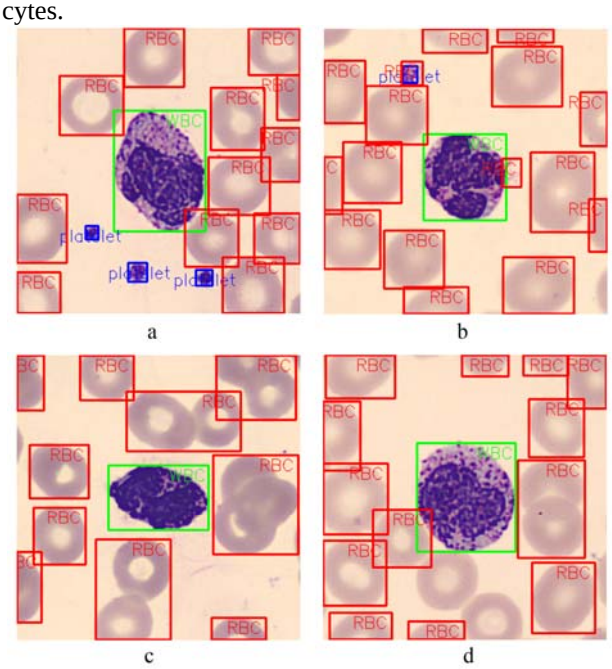


Figure 10 – Cases of cell recognition and localization

For further processing, the number of correct (True Positive, TP), incorrect (False Positive, FP) and unrecognized cells (False Negative, FN) of the considered types was calculated.

Let's define in more detail:

TP (True Positives) is the number of cells that are correctly recognized and assigned to a certain class;

FP (False Positives) is the number of cells that are recognized but incorrectly assigned to a certain class;

FN (False Negatives) is the number of cells that are not recognized as cells of a certain class.

The results are shown in Table 2.

Table 2 – Quality of recognition of blood cells

	Cell type		
	Leukocytes	Platelets	Erythrocytes
Correctly Recognized (TP)	160	127	2161
Incorrectly recognized (FP)	14	20	170
Not Recognized (FN)	1	13	178

To visualize the obtained results, let's build a heat map for the number of cells of each type and the number of correctly, incorrectly, and unrecognized blood cells (Fig. 11).

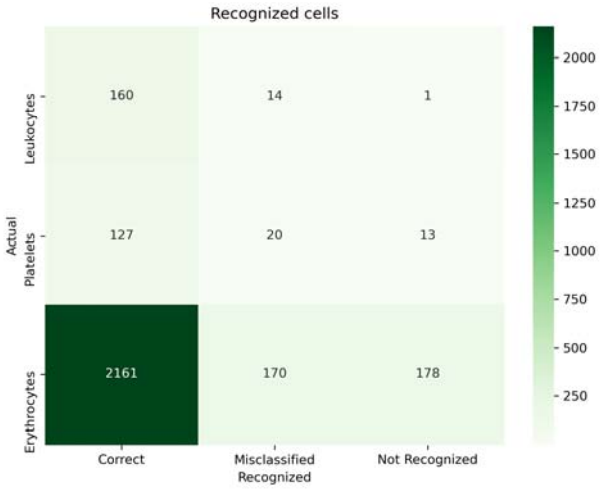


Figure 11 – Quality of cell recognition by type

To visually display the proportion of correctly, incorrectly, and unrecognized cells of each type in the total volume of cells, we will build a stacked chart (Fig. 12).

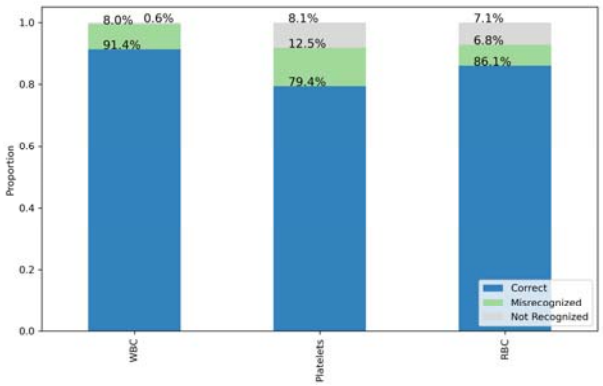


Figure 12 – Relative contribution of each cell type

Figure 12 shows that the number of correctly recognized white blood cells (WBCs) is 91.4% of the total, where 8% are misrecognized and only 0.6% WBCs are unrecognized. 79.4% of platelets are correctly recognized, 12.5% of the total are FPs, and 8.1% are not identified as platelets. TP red blood cells (RBCs) are 86.1%, 6.8% are FP, and 7.1% are FN.

The following metrics were calculated to assess the quality of object annotation:

DOI 10.15588/1607-3274-2024-1-12

– accuracy, which calculates the overall classification accuracy (7):

$$Acc = \frac{TP}{Total\ number}, \quad (7)$$

– precision, which estimates the percentage of objects that are correctly identified as a certain class (8):

$$Pr = \frac{TP}{TP + FP}, \quad (8)$$

– recall shows how many objects of a given class were correctly identified relative to all real objects of that class (9):

$$Rec = \frac{TP}{TP + FN}, \quad (9)$$

– F1 is the harmonic mean between precision and recall, which combines these two metrics into a single number and makes it possible to evaluate the effectiveness of the classification model, by choosing a balance between precision and recall (10):

$$F1 = 2\left(\frac{Pr \cdot Rec}{Pr + Rec}\right). \quad (10)$$

The values of the calculated metrics are given in the table. 3

Table 3 – Quality metrics of automatic recognition of blood cells

	Cell type		
	Leukocytes	Platelets	Erythrocytes
Accuracy	0.9143	0.7937	0.8613
Precision	0.9195	0.8639	0.9271
Recall	0.9938	0.9071	0.9239
F1	0.9552	0.8850	0.9255

The obtained metric values indicate a sufficiently high efficiency of recognition of leukocytes and erythrocytes and insufficient efficiency of platelet recognition, which is associated with a high number of incorrectly recognized ones and not recognized cells. In our opinion, this is due to an unsuccessful choice of the threshold value when constructing the mask.

In further research, we plan to investigate the impact of different approaches to creating binary masks and contour extraction to improve the quality of cell recognition and localization. Also, this approach does not handle overlapping erythrocytes well enough. Processing of such types of cells is also the object of further research.

CONCLUSIONS

In this paper, we propose an approach to automatically generate an annotated dataset for recognizing different types of blood cells in an image. Our method is based on the use of image processing techniques to detect, segment and localize blood cells. The approach has been evaluated on a blood cell microscopy dataset and shown to achieve high accuracy. Our method has several advantages over existing manual methods for creating annotated datasets. First, it is fully automated, which makes it more efficient and scalable. Second, it can be used to create annotated datasets for any type of blood cell.

The following results were obtained:

1. An approach for automatically creating a set of annotated data for further use for object recognition using supervised machine learning methods is proposed and substantiated.

2. The pipeline is tested on a set of unlabeled data.

3. An annotated dataset is obtained consisting of cell images and a .csv file with the features “file_name”, “type”, “xmin”, “ymin”, “xmax”, “ymax” which are bounding box coordinates for each object.

4. The number of correctly, incorrectly and unrecognized objects is counted and metrics were calculated to evaluate the accuracy.

5. Metrics for evaluating the quality of object detection and localization are obtained.

The scientific novelty of obtained results consists in the application of digital image processing methods to automatically create an annotated dataset.

The practical significance of the obtained results is a significant reduction in the time spent by machine learning specialists on creating an annotated dataset, which will positively affect the overall time spent on the model.

Prospects for further research are to extract overlapping cell contours and more accurate platelet recognition using other binarization and contour extraction algorithms.

REFERENCES

1. Chadha G. K., Srivastava A., Singh A., Gupta R., Singla D. An Automated Method for Counting Red Blood Cells using Image Processing. Procedia Computer Science, *Procedia Computer Science*, 2020, Vol. 167, pp. 769-778. DOI: 10.1016/j.procs.2020.03.408.
2. Kovalenko S. Kovalenko S., Mikhnova O., Kovalenko A., Pelikh D., Severin V. An Approach to Blood Cell Classification Based on Object Segmentation and Machine Learning, *IEEE 4th KhPI Week on Advanced Technology (KhPIWeek)*, 2023, pp. 1–6. DOI: 10.1109/KhPIWeek61412.2023.10312903.
3. 2022 State of Data Science by Anaconda [Electronic resource]. Access mode: https://www.anaconda.com/resources/whitepapers/state-of-data-science-report-2022.
4. Aljabri M., AlAmir M., AlGhamdi M., Abdel-Mottaleb M., Collado-Mesa F. Towards a better understanding of annotation tools for medical imaging: a survey, *Multimedia tools and applications*, 2022, Vol. 81(18), 202225877-25911. DOI: 10.1007/s11042-022-12100-1.
5. Kutsenko A. S., Megel Y. Y., Kovalenko S. V., Kovalenko S. M., Omiotek Z., Zhunissova U. An approach to quality evaluation of embryos based on their geometrical parameters, *Proc. SPIE 11176, Photonics Applications in Astron-*

DOI 10.15588/1607-3274-2024-1-12

omy, Communications, Industry, and High-Energy Physics Experiments, 2019. 111762G. DOI: 10.1117/12.2536420.
6. Megel Y., Kalimanova I., Rybalka A., Kovalenko S., and Kovalenko S. Automation of measurement of objects geometrical parameters, *Proceedings of International Scientific Symposium "Metrology and Metrology Assurance"*, 2017, pp. 225–259.
7. Burger W., Burge M. J. Digital Image Processing: An Algorithmic Introduction. Springer Nature, 2022, 927 p. DOI: 10.1007/978-1-4471-6684-9
8. Chityala R., Sridevi P. Image processing and acquisition using Python. CRC Press, 2020, 420 p. DOI: 10.1201/9780429243370.
9. Dey S. Python image processing cookbook: over 60 recipes to help you perform complex image processing and computer vision tasks with ease. Packt Publishing Ltd, 2020, 438 p. ISBN: 978-1-789537147.
10. Gaudenz B. Object Detection in 2024: The Definitive Guide. [Electronic resource]. Access mode: https://viso.ai/deep-learning/object-detection/.
11. Plataniotis K., Venetsanopoulos A. N. Color Image Processing and Applications. Springer Science & Business Media, 2000, 355 p. DOI: 10.1007/978-3-662-04186-4.
12. Zhuang F., Qi Z., Duan K., Xi D., Zhu Y., Zhu H., Xiong H., He Q. A Comprehensive Survey on Transfer Learning, *Proceedings of the IEEE*, 2021. Vol. 109, No. 1, pp. 43–76. DOI: 10.1109/JPROC.2020.3004555.
13. Mikołajczyk A. and Grochowsk M. Data augmentation for improving deep learning in image classification problem, *2018 International Interdisciplinary PhD Workshop (IIPhDW)*. Świnoujście, Poland, 2018, pp. 117–122. DOI: 10.1109/IIPHDW.2018.8388338.
14. Bonaccorso G. Machine Learning Algorithms – Second Edition. Packt Publishing Ltd., 2018, 522 p. ISBN 978-1-78934-799-9.
15. Szeliski R. Image-Based rendering, *Computer Vision. Texts in Computer Science*. Springer, Cham, 2022, pp. 681–722. DOI: 10.1007/978-3-030-34372-9_14.
16. Brownlee J. A gentle introduction to object recognition with deep learning, *Machine Learning Mastery 5* [Electronic resource]. Access mode: https://machinelearningmastery.com/object-recognition-with-deep-learning/.
17. Sinha R. K., Pandey R., Pattnaik R. Deep learning for computer vision tasks: a review, *arXiv preprint arXiv*:1804.03928, 2018. DOI: 10.26438/ijcse/v7i7.195201.
18. Lu L., Wang X., Carneiro G., Yang L. Deep learning and convolutional neural networks for medical imaging and clinical informatics. Berlin/Heidelberg, Germany, Springer International Publishing, 2019, 461 p. DOI: 10.1007/978-3-030-13969-8.
19. Megel Yu., Chaly I., Kovalenko S., Mikhnova O. Doslidzhennja dyhal'nyh ruhiv za dopomogoju ekspertnoi systemy na bazi obchysljuvalnogo intelektu [Study of Respiratory Movements Using an Expert System Based on Computational Intelligence], *Systemy obrobky informacii [System. Information Processing Systems]*, 2022, № 3 (170), pp. 41–46. DOI: 10.30748/soi.2022.170.05.
20. Tigner A., Ibrahim S. A., Murray I. V. Histology, White Blood Cell, *StatPearls Publishing, 2022 [Electronic resource]*. Access mode: https://www.ncbi.nlm.nih.gov/books/NBK563148.
21. OpenCV. Color conversions [Electronic resource]. Access mode: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html.
22. Acevedo A., Merino A., Alférez S., Molina Á., Boldú L. and Rodellar J. A dataset for microscopic peripheral blood cell images for development of automatic recognition systems, *Data in brief*, 2002, Vol. 30. 105474. DOI: 10.1016/j.dib.2020.105474.

Received 26.01.2024.
Accepted 28.02.2024.

UDC 004.932.2:004.93'1

ПІДХІД ДО АВТОМАТИЧНОГО СТВОРЕННЯ АНОТОВАНОГО ДАТАСЕТУ ДЛЯ ВИЯВЛЕННЯ, ЛОКАЛІЗАЦІЇ ТА КЛАСИФІКАЦІЇ КЛІТИН КРОВІ НА ЗОБРАЖЕННІ

Коваленко С. М. – канд. техн. наук, доцент, доцент кафедри програмної інженерії та інтелектуальних технологій управління Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

Куценко О. С. – д-р техн. наук, професор, професор кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

Коваленко С. В. – канд. техн. наук, доцент, професор кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

Коваленко А. С. – аспірант кафедри системного аналізу та інформаційно-аналітичних технологій Національного технічного університету «Харківський політехнічний інститут», Харків, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто проблему автоматизації створення анотованого набору даних для його подальшого використання в системі виявлення, локалізації та класифікації клітин крові на зображенні з використанням глибокого навчання. Об'єктом дослідження є процеси обробки цифрових зображень для виявлення та локалізації об'єктів.

Мета роботи – розробка пайплайну із послідовності методів обробки цифрових зображень для автоматичного створення анотованого набору зображень мазків крові з подальшим використанням для навчання та валідації моделей глибокого навчання, що має суттєво скоротити час спеціалістів з машинного навчання.

Метод. Запропонований підхід для виявлення та локалізації об'єктів базується на методах обробки цифрових зображень: методах фільтрації, порогової фільтрації, бінаризації, знаходження та заливки контурів тощо. Пайлайн по виявленню та локалізації складається з наступних кроків: приглушення шумів; перетворення в HVS кольорову модель; визначення маски для лейкоцитів та тромбоцитів; визначення контурів лейкоцитів та тромбоцитів; визначення координат верхнього лівого та правого нижнього кутів лейкоцитів та тромбоцитів; обчислення площі області всередині обмежувальної рамки; збереження

DOI 10.15588/1607-3274-2024-1-12

отриманих даних; визначення найпоширенішого кольору на зображенні; заливка цим кольором контурів лейкоцитів та тромбоцитів; визначення маски для еритроцитів; визначення контурів еритроцитів; визначення координат верхнього лівого та правого нижнього кутів еритроцитів; обчислення площі області всередині обмежувальної рамки; занесення до датафрейму даних про знайдені об'єкти; збереження в файлі .csv для подальшого використання.

Результати. Розроблений підхід був впроваджений у програмне забезпечення для створення анотованого набору даних зображень мазків крові.

Висновки. В дослідженні запропоновано та обґрунтовано підхід для автоматичного створення набору анотованих даних. Пайплайн протестовано на наборі нерозмічених даних та отримано набір розмічених даних, що складається з зображень клітин та файлу в форматі .csv, що має ознаки «назва файлу», «тип клітини», «xmin», «ymin», «xmax», «ymax», що є координатами обмежувальної рамки для кожного об'єкту. Підраховано кількість правильно, неправильно та нерозпізнаних об'єктів та розраховано метрики для оцінки точності та якості виявлення та локалізації об'єктів.

КЛЮЧОВІ СЛОВА: комп'ютерний зір, визначення об'єктів, локалізація об'єктів, обробка цифрових зображень, класифікація.

ЛІТЕРАТУРА

1. An Automated Method for Counting Red Blood Cells using Image Processing. Procedia Computer Science / [G. K. Chadha, A. Srivastava, A. Singh et al.] // Procedia Computer Science. – 2020. – Vol. 167. – P. 769–778. DOI: 10.1016/j.procs.2020.03.408.
2. An Approach to Blood Cell Classification Based on Object Segmentation and Machine Learning / [S. Kovalenko, S. Kovalenko, O. Mikhnova et al.] // IEEE 4th KhPI Week on Advanced Technology (KhPIWeek). – 2023. – P. 1–6. DOI: 10.1109/KhPIWeek61412.2023.10312903.
3. 2022 State of Data Science by Anaconda [Electronic resource]. – Access mode: https://www.anaconda.com/resources/whitepapers/state-of-data-science-report-2022.
4. Towards a better understanding of annotation tools for medical imaging: a survey / [M. Aljabri, M. AlAmir, M. AlGhamdi et al.] // Multimedia tools and applications. – 2022. – Vol. 81(18). – 202225877–25911. DOI: 10.1007/s11042-022-12100-1.
5. An approach to quality evaluation of embryos based on their geometrical parameters / [A. S. Kutsenko, Y. Y. Megel, S. V. Kovalenko et al.] // Proc. SPIE 11176, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments. – 2019. – 111762G. DOI: 10.1117/12.2536420.
6. Automation of measurement of objects geometrical parameters / [Y. Megel, I. Kalimanova, A. Rybalka, S. Kovalenko, and S. Kovalenko] // Proceedings of International Scientific Symposium "Metrology and Metrology Assurance". – 2017. – P. 225–259.
7. Burger W. Digital Image Processing: An Algorithmic Introduction / W. Burger, M. J. Burge. – Springer Nature, 2022. – 927 p. DOI: 10.1007/978-1-4471-6684-9.
8. Chityala R. Image processing and acquisition using Python / R. Chityala, P. Sridevi. – CRC Press, 2020. – 420 p. DOI: 10.1201/9780429243370.
9. Dey S. Python image processing cookbook: over 60 recipes to help you perform complex image processing and computer vision tasks with ease / S. Dey – Packt Publishing Ltd, 2020. – 438p. ISBN: 978-1-789537147.
10. Gaudenz B. Object Detection in 2024: The Definitive Guide. / B. Gaudenz [Electronic resource]. – Access mode: https://viso.ai/deep-learning/object-detection/.
11. Plataniotis K. Color Image Processing and Applications / K. Plataniotis, A. N. Venetsanopoulos. – Springer Science & Business Media, 2000. – 355 p. DOI: 10.1007/978-3-662-04186-4.
12. A Comprehensive Survey on Transfer Learning / [F. Zhuang, Z. Qi, K. Duan et al.] // Proceedings of the IEEE. – 2021. – Vol. 109, No. 1. – P. 43–76. DOI: 10.1109/JPROC.2020.3004555.
13. Mikołajczyk A. Data augmentation for improving deep learning in image classification problem / A. Mikołajczyk and M. Grochowsk // 2018 International Interdisciplinary PhD Workshop (IIPhDW). – Świnoujście, Poland, 2018. – P. 117–122. DOI: 10.1109/IIPHDW.2018.8388338.
14. Bonaccorso, G. Machine Learning Algorithms – Second Edition / G. Bonaccorso. – Packt Publishing Ltd.,2018. – 522 p. ISBN 978-1-78934-799-9.
15. Szeliski R. Image-Based rendering / R. Szeliski // Computer Vision. Texts in Computer Science. – Springer, Cham, 2022. – P. 681–722. DOI: 10.1007/978-3-030-34372-9_14.
16. Brownlee J. A gentle introduction to object recognition with deep learning. / J. Brownlee // Machine Learning Mastery 5 [Electronic resource]. – Access mode: https://machinelearningmastery.com/object-recognition-with-deep-learning/.
17. Sinha R. K. Deep learning for computer vision tasks: a review / R. K. Sinha, R. Pandey, R. Pattnaik // arXiv preprint arXiv:1804.03928, 2018. DOI: 10.26438/ijcse/v7i7.195201.
18. Deep learning and convolutional neural networks for medical imaging and clinical informatics / [L. Lu, X. Wang, G. Carneiro, L. Yang]. – Berlin/Heidelberg, Germany : Springer International Publishing, 2019. – 461 p. DOI: 10.1007/978-3-030-13969-8.
19. Дослідження дихальних рухів за допомогою експертної системи на базі обчислювального інтелекту / [Ю. Є. Мегель, І. В. Чалий, С. М. Коваленко, О. Д. Міхнова] // Системи обробки інформації. – 2022. – № 3 (170). – P. 41–46. DOI: 10.30748/soi.2022.170.05.
20. Tigner A. Histology, White Blood Cell // A. Tigner, S. A. Ibrahim, I. V. Murray // StatPearls Publishing, 2022 [Electronic resource]. – Access mode: https://www.ncbi.nlm.nih.gov/books/NBK563148
21. OpenCV. Color conversions [Electronic resource]. – Access mode: https://docs.opencv.org/3.4/de/d25/imgproc_color_conversions.html.
22. A dataset for microscopic peripheral blood cell images for development of automatic recognition systems / [A. Acevedo, A. Merino, S. Alférez et al.] // Data in brief. – 2002. – Vol. 30. – 105474. DOI: 10.1016/j.dib.2020.105474.

DOI 10.15588/1607-3274-2024-1-12

UDC 621.391

A RESEARCH OF THE LATEST APPROACHES TO VISUAL IMAGE RECOGNITION AND CLASSIFICATION

Lysechko V. P. – PhD, Professor, Professor of Transport Communication Department, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.

Sadovnykov B. I. – Postgraduate student of Transport Communication Department, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.

Komar O. M. – PhD, Associate Professor, Associate Professor of National Aviation University, Kyiv, Ukraine.

Zhuchenko O. S. – PhD, Associate Professor, Associate Professor of Transport Communication Department, Ukrainian State University of Railway Transport, Kharkiv, Ukraine.

ABSTRACT

Context. The paper provides an overview of current methods for recognizing and classifying visual images in static images or video stream. The paper will discuss various approaches, including machine learning, current problems of these methods and possible improvements. The biggest challenges of the visual image retrieval and classification task are discussed. The main emphasis is placed on the review of such promising algorithms as SSD, YOLO, R-CNN, an overview of the principles of these methods, network architectures.

Objective. The aim of the work is to analyze existing studies and find the best algorithm for recognizing and classifying visual images for further activities.

Method. Primary method is to compare different factors of algorithms in order to select the most perspective one. There are different marks to compare, like image processing speed, accuracy.

There are a number of studies and publications that propose methods and algorithms for solving the problem of finding and classifying images in an image [3–6]. It should be noted that most promising approaches are based on machine learning methods.

It is worth noting that the proposed methods have drawbacks due to the imperfect implementation of the Faster R-CNN, YOLO, SSD algorithms for working with streaming video. The impact of these drawbacks can be significantly reduced by applying the following solutions: development of combined identification methods, processing of edge cases – tracking the position of identified objects, using the difference between video frames, additional preliminary preparation of input data. Another major area for improvement is the optimization of methods to work with real-time video data, as most current methods focus on images.

Results. As an outcome of the current research we have found an optimal algorithm for further researches and optimizations.

Conclusions. Analysis of existent papers and researches has demonstrated the most promising algorithm for further optimizations and experiments. Also current approaches still have some space for further. The next step is to take the chosen algorithm and investigate possibilities to enhance it.

KEYWORDS: machine learning, computer vision, image processing, convolutional neural networks, visual image recognition, visual image classification, algorithms, telecommunication systems.

INTRODUCTION

The field of computer vision is a promising area for the development of visual image processing systems. An example of the implementation of such a system is the image classification system, namely the analysis of medical images, the solution of which opened up the possibility of developing systems for the automatic detection of pathology in patient images [1]. Another example is the process of production automation based on automated quality control of products based on photographs [2].

The problem of recognition and classification of images in a fixed image or video stream is complex and important for many potential and existing practical applications in various fields of activity, primarily in the operation of video surveillance systems as an element of a telecommunications system.

The object of study is the process of recognition and classification of object on the video.

The subjects of study are algorithms to detect and classify objects on the video or image.

The purpose of this work is to review and analyze existing methods and approaches for recognizing and classifying visual patterns in images in order to identify possible ways to improve their performance.

1 PROBLEM STATEMENT

The task of recognizing and classifying visual images in an image is a complex one, and its solution consists of several separate steps, as shown in Fig. 1.

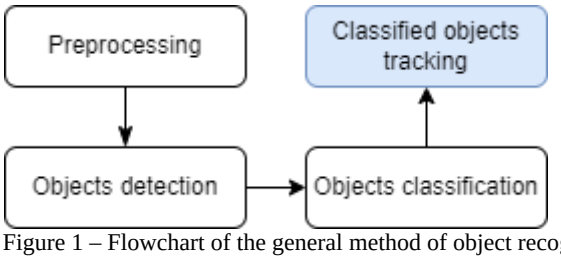


Figure 1 – Flowchart of the general method of object recognition and classification

From Figure 1, we can conclude that the method can be the same for images and video, since video is a set of images (frames) that change at a certain interval, which depends on the number of frames per second. That is, to perform the task with the input data in the form of video,

DOI 10.15588/1607-3274-2024-1-13

it is enough to search and classify objects on each frame. There is a possibility that the next frame contains few or many differences from the previous one, or no differences at all. This provides additional room for optimizations and accuracy improvement.

2 REVIEW OF THE LITERATURE

The first step is to process the input image [7]. This is a common practice in the field of computer vision, as the image usually has redundant data. First of all, these are different encodings, formats of image pixels, which make sense in everyday use, but have no impact in the case of analysis. It makes sense to convert all images to one specific format for the algorithm to work with them.

The next component to be removed is redundant color data. In many cases, converting an image to grayscale improves the accuracy of the algorithms. Simplifying the color model also provides a number of benefits in terms of resource usage – the computations become simpler, so less CPU time is consumed and memory is used more economically due to less color data.

The last possible processing is physical transformations of the image, such as resizing, cropping, flipping, and mirroring. A certain constant image size that the algorithm works with greatly simplifies implementation and adds versatility, so it is advisable to resize all input images to a certain format. It should be noted that this operation may lead to the loss of some data, so it is necessary to use the most accurate algorithms, which may lead to some deterioration in execution time.

"Cropping" the image is not appropriate in this case because the algorithm is aimed at finding objects, i.e. it is not known whether the area to be cropped contains an object or part of it.

Flipping and reflecting the input image significantly increases the amount of computation required, as each transformation is a new image to process. This step can improve accuracy, but significantly degrade speed, which can be critical for an algorithm that potentially needs to work in real time.

The next step after image preprocessing is to search for potential objects in the image. The algorithm chosen for this task must satisfy a number of criteria, such as sufficient accuracy, the ability to work with real-time data, work with images of low quality and size, and resistance to noise, changes in angles and ambient light [8].

Classical computer vision algorithms or machine learning methods are potential approaches to solving this problem.

Computer vision algorithms consist of a set of mathematical operations and transformations of image pixel data, and are usually unchanged regardless of the input data. Any modifications to suit specific input data or environments must be done manually. This significantly reduces the flexibility of the solution as it potentially needs to be adjusted to work optimally with different data sets. It is possible that there will be several parallel implementations of the algorithm for slightly different input data. The lack of self-adaptation is a disadvantage of classical algorithms and can be significant in solving the problem of object detection and classification, as it can significantly complicate the practical use of methods based on these algorithms.

An alternative approach in terms of solution architecture is to use machine learning methods instead of classical computer vision algorithms, as the focus shifts to network building rather than image data operations. The main advantage of machine learning methods is their flexibility. They are able to adapt to the input data on their own, which greatly improves their practical use. That is, we have one neural network architecture that can adjust its weights according to the input data. This reduces the number of edge cases, and the algorithm can find logical connections in the data on its own. Thus, deep learning algorithms are a good candidate for solving the problem of finding and classifying objects in an image.

Convolutional neural networks are a special class of neural networks for image processing [9]. These networks consist of interconnected layers of neurons. The main goal of this architecture is to simulate the processes that occur in the human brain when analyzing images. Convolutional neural networks are used to solve such tasks as face recognition, image classification, search for anomalies in medical images, etc. This type of network automatically extracts important image features in numerical form from pixel data and makes predictions based on them. Which image features will be extracted depends on the input data and the task set at the model training stage. Each layer of the convolutional model works with one level of image features or details, for example, the first layers work with low-level details such as borders or texture. Each subsequent layer of the model works with more abstract features.

Searching for objects in an image and classifying them can be considered as two separate tasks, which is shown in the flowchart of the general method. The result of the search is information about the coordinates of the object in the image, while the result of the classification is the type or class of the object. Thus, you first need to find the position of the objects, and then classify them.

Modern machine learning algorithms are divided into two types depending on whether they perform search and classification together or separately. Thus, there are one-pass detectors that find objects in an image and determine their class in one cycle of image analysis and two-pass detectors that first analyze the image for the presence and position of objects and then classify the found objects. Good examples of one-pass models are YOLO, SSD, and for two-pass models, the R-CNN family of models. In any case, these algorithms simultaneously find the position of the objects and their type, i.e., they perform steps 2 and 3 from Figure 1.

Currently, there are several modifications of the R-CNN algorithm, namely Fast R-CNN and Faster R-CNN [10–12]. The main purpose of the modifications was to speed up the algorithm and improve accuracy, since the original version of R-CNN processed one image for about 40 sec. [13]. This level of speed is not enough for real-

DOI 10.15588/1607-3274-2024-1-13

time data processing, so the next modification of Fast R-CNN reduced this time to about 2.5 seconds. Further improvements in the next iteration of the network, called Faster R-CNN, made it possible to reduce the image processing time even further, to about 0.2 seconds per image [14].

In the following, it makes sense to consider only the latest iteration of the algorithm, namely Faster R-CNN, since it is the latest development of this family of algorithms and has the highest speed, which is a significant indicator for the task of object recognition and classification in streaming video.

Let's look at the general architecture of the Faster R-CNN neural network in Fig. 2.

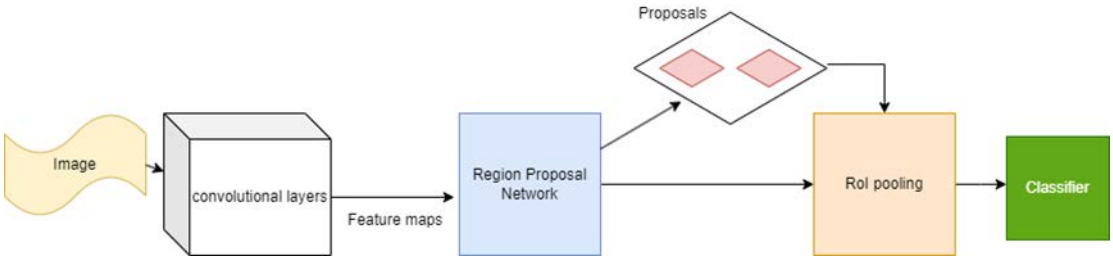


Figure 2 – structure of the Faster R-CNN network

Figure 2 shows that the network consists of two main modules. The first module is a deep convolutional network that provides suggestions for possible regions of the image that contain objects and is abbreviated as RPN (region proposal network). This optimization is the result of a study that demonstrated that based on the features extracted by the convolutional neural network, it is possible to make assumptions about regions containing objects and that working with pixel data directly is not required. RPN is an important addition to the model, as it allows to use GPU resources to search for potential regions of the image with objects. In previous iterations, the algorithm for finding potential regions was executed on the CPU. The Selective search algorithm was used for this task. Despite the fact that it is a greedy algorithm that combines groups of pixels based on low-level image features, it is slower than RPN, so these changes give a significant increase in the efficiency of the neural network as a whole.

The number of regions proposed by RPN is on average less than that of the selective search algorithm, 2000 vs. 300 on the PASCAL VOC 2007 dataset, which significantly reduces the number of detector calls and leads to a reduction in overall image processing time.

The second module is the Fast R-CNN detector, which processes the provided regions and draws conclusions about the presence and type of objects on them. An important architectural solution is the use of shared convolutional layers in the network that provides the regions and the detector. This reduces the requirements for the required amount of RAM and somewhat simplifies the overall network architecture. The disadvantage of this solution is additional complexity during training, since the same convolutional layers must be used for two different components.

Let us consider the general steps of Faster R-CNN. First, the input image, which has already passed the preprocessing stage, is transferred to the convolutional neural network to search for regions with objects on them. Regions, namely rectangles around potential objects, are determined using features calculated by convolutional layers, rather than pixel data per line as in previous iterations of the algorithm. Next, a smaller feature map is extracted from the calculated features for the regions and passed to the Faster R-CNN detector, which in turn de termines the presence of the object, class, and corrects the boundaries occupied by the object in the image.

From the network architecture and algorithm, it follows that search and classification are actually performed in two steps, so this algorithm is classified as a two-pass detector. Taking into account the speed of Faster R-CNN, this model is close to the possibility of real-time image processing, but still inferior to single-pass detectors [14]. When working with streaming video, additional processing optimizations are possible, so the algorithm may work faster with them. An important advantage of Faster R-CNN is its high accuracy and the ability to recognize small objects.

Another promising algorithm for the task of searching and classifying objects in an image is YOLO (You Only Look Once) [15]. Similar to R-CNN, there are many iterations and variations of this algorithm. One of the most recent versions of the model is YOLOv8 [16].

Unlike R-CNN, YOLO is not a complex algorithm consisting of several separate and interchangeable parts, but a monolithic model that performs the task of finding and classifying an object in an image. From the first to the eighth version of the model, a very significant number of changes took place. Due to the structure of the model, the main changes were in the framework model, for example, changing the Darknet24 framework model to the Darknet53 model in the third version, changes in the model training algorithm, input data, training parameters, and adjustments to the overall architecture of the model.

The main feature of the YOLO family of models is a constant focus on balancing speed and accuracy in order to provide sufficient efficiency for real-time operation without significant loss of accuracy. This balance changes slightly between different iterations of the model. The latest versions of the algorithm have several versions with different balances between these characteristics to opti-

DOI 10.15588/1607-3274-2024-1-13

mize performance on different types of instruments. The lightweight models are optimized for use on embedded devices and have the lowest accuracy in order to work in resource-constrained environments, while larger models require more resources but have higher accuracy.

The first step of the model is to divide the image into small cells of size S x S, where S depends on the model configuration. For each cell, the confidence that an object is located there and the boundaries of that object are calculated. The confidence also reflects how accurately the boundaries are calculated. Also, each cell represents a specific class of object inside. Only one class is calculated for a cell. This is a disadvantage of the model that if several small objects are in a cell, only one will be found and classified. Next, cells of the same type are merged to form the final boundaries and position of the found object, as shown in Fig. 3.

Figure 4 shows that according to the algorithm, objects are detected and classified in one pass through the image, so this algorithm is considered a single-pass detector. According to well-known studies, the speed of the latest versions of YOLO algorithms, including YOLOv8, exceeds 100 frames per second, so they can be used for real-time image processing [17].

SSD (Single Shot Detector) is an algorithm for recognizing and classifying visual images in an image that uses convolutional neural networks [18]. The main goal of developing the algorithm was to increase the speed of operation compared to the previous advanced YOLO algorithm and improve the recognition accuracy. The goal was to achieve an accuracy similar to two-pass detectors such as R-CNN or Faster R-CNN and to allow the algorithm to work in real time.

Consider the architecture of the SSD model in Fig. 4.

Figure 4 shows that the algorithm consists of two main parts. The first part is the main deep neural network, which is used to calculate image features. To calculate features, it is possible to use a trained classifier model. To do this, the top classifier layer of the network is removed to access the feature maps. The initial SSD implementation uses the VGG16 network without the classifier layer [19].

The next part is several convolutional SSD layers that process the image feature maps to find object boundaries and classify them.

The SSD model is also a single-pass detector like YOLO, so there are a number of similar steps in their algorithms. First, SSD divides the input image into cells, and each cell is responsible for searching for an object in that area of the image. The search is a calculation by each cell of the probability of finding an object of a particular class in that region. Then the boundaries of the found objects are formed. Unlike YOLO, image boundaries are formed using possible object boundaries, or anchor

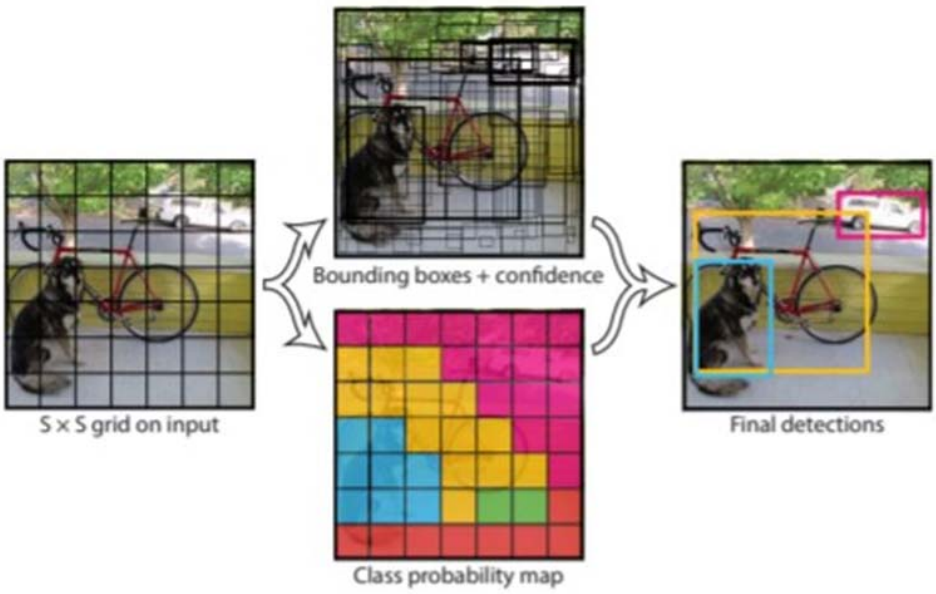


Figure 3 – YOLO algorithm flow

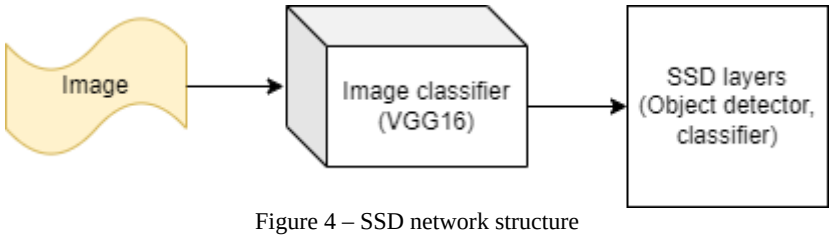


Figure 4 – SSD network structure

DOI 10.15588/1607-3274-2024-1-13

boundaries, determined at the training stage. Each anchor boundary has its own size, shape, and position inside the cell. Thus, the boundary that has the largest common area with a potential object is considered its boundary, and the object type is determined using this boundary. It is important to note that feature maps in different layers of the convolutional network are analyzed. That is, the size of feature maps changes, which leads to a change in the size of the image region described by each feature. Thus, the size of the cells into which the image is divided is different, while the size of the anchor cells remains fixed, which allows you to find objects of different sizes. Due to the anchor boundaries, SSD can find several objects in one cell, unlike YOLO. After that, for each boundary, the probability of finding an image of a particular class in it is displayed.

A critically small object is a visual object whose set can be located in a single cell into which the algorithm divides the image, i.e., one cell can contain 2 or more critically small objects. The main advantage of SSD is the ability to recognize critically small objects. According to a well-known study, the SSD model working with input images of 300 x 300 pixels has an accuracy of 3% higher than the Faster R-CNN, which worked on the basis of the VGG16 model [18]. The processing speed of the SSD300 was 59 frames per second, which is sufficient for real-time work.

The last optional step is to track the positions of the detected and classified objects. This step is only possible for video, and will allow you to avoid using the search algorithm for each frame. The main advantage of this approach is resource savings due to the use of a faster computing algorithm for tracking images on streaming video. Potential image tracking algorithms will be discussed in more detail in the following works.

3 MATERIALS AND METHODS

Let's consider a few metrics to setup a common performance measurement system for different algorithms.

The first crucial part of the neural network algorithm's efficiency is accuracy. Usually it displays how many correct answers network gave. In the current case it can be a combination of the correct class labels for detected objects and correctness of detected objects positions and borders.

Another important aspect is the time consumption for detection and classification. There can be plenty of options like time spend per frame, but the best one for the current problem is frame processed per second metric. We want to use an algorithm in real time processing, the primary characteristic for real time video is an amount of frames per second, in order to map algorithm's performance to real world, we can calculate amount of frames algorithm process per second. So it's straightforward to make a conclusion, can it process real time video data or not if we use an FPS as an efficiency metric.

4 EXPERIMENTS

Let's gather experimental data related to discussed algorithms.

We will use two primary metrics discussed previously: accuracy and FPS.

Figure 5 shows a diagram of the accuracy of the considered algorithms for searching and classifying objects.

It can be seen that the YOLO algorithm provides the ability to process images in real time with an accuracy lower by about 1.26 times relative to the slower Faster R-CNN algorithm.

Figure 6 shows a comparative speed chart of the methods.

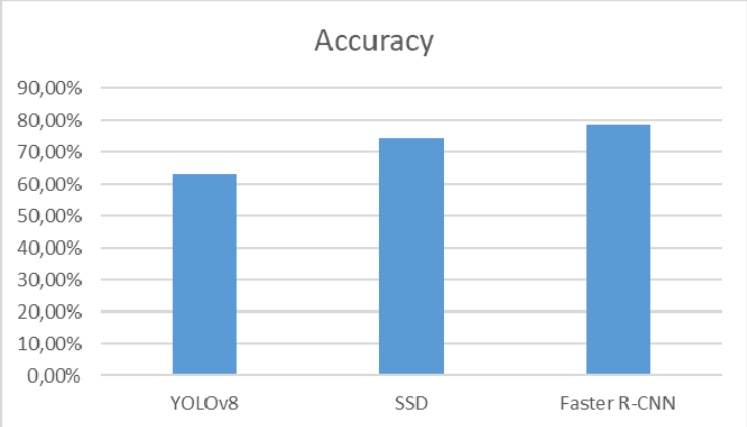


Figure 5 – Diagram of the accuracy of algorithms

DOI 10.15588/1607-3274-2024-1-13

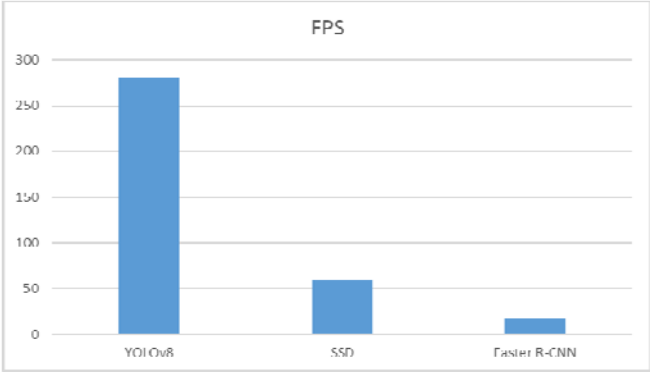


Figure 6 – Algorithms speed chart

5 RESULTS

As a result of the analysis of modern methods for searching and classifying visual images in an image, three promising neural network models were selected for further experimental research on the possibilities of optimizing them for more efficient work with streaming video data.

From the graphs shown in Figures 5 and 6, the Faster R-CNN method has a higher accuracy of approximately 80%, but insufficient speed for real-time work, 18 times lower than YOLO. SSD has a speed 5 times lower than YOLO, but sufficient for real-time operation and accuracy similar to Faster R-CNN.

Thus, for real-time work, it is advisable to choose the SSD algorithm, as it has a better balance between accuracy and speed. In addition, it makes sense to experiment with other algorithms, since accuracy and speed depend on the dataset and the task, it is possible that the algorithms will demonstrate different characteristics for the current task.

REFERENCES

1. Yue W., Liu S., Li Y. An Efficient Pure CNN Network for Medical Image Classification, *Applied Sciences*, 2023, No. 13(16), P. 9226.
2. Cui W., Zhang Y., Zhang X., Li L., Liou F. Metal Additive Manufacturing Parts Inspection Using Convolutional Neural Network, *Applied Sciences*, 2020, No. 10(2), P. 545.
3. Lysechko V. P., Syvolovskyi I. M., Shevchenko B. V. et al. Research of modern NoSQL databases to simplify the process of their design, *Academic journal: Mechanics Transport Communications*, 2023, Vol. №21, Issue 2, article №2363, pp. 234–242
4. Lysechko V. P., Zorina O. I., Sadovnykov B. I. et al. Experimental study of optimized face recognition algorithms for resource – constrained, *Academic journal: Mechanics Transport Communications*, 2023, Vol. №21, Issue 1, article №2343, pp. 89–95.
5. Mohana, Ravish Aradhya H. V Design and Implementation of Object Detection, Tracking, Counting and Classification Algorithms using Artificial Intelligence for Automated Video Surveillance Applications, Conference, *24th International Conference on Advanced Computing and Communications*, 2022, pp. 56–60.
6. Feroz A., Sultana M., Hasan R. et al. Object Detection and Classification from a Real-Time Video Using SSD and YOLO Models, *Computational Intelligence in Pattern Recognition*, 2021, 405 p.
7. Seker M., Köylüoğlu Y., Celebi A., Bayram B. Effects of Open-Source Image Preprocessing on Glaucoma and Glaucoma Suspect Fundus Image Differentiation with CNN [Electronic resource], 2021, Access mode: https://doi.org/10.21203/rs.3.rs-1695441/v1.
8. Sadovnykov B., Zhuchenko O., Perets K. Overview of state-of-the-art image object detection and classification approaches, *Collection of scientific papers of UkrDUZT International scientific and technical conference "Development of scientific and innovative activity in transport"*, 2023, Issue 177. Kharkiv, UkrDUZT, pp. 46–48.
9. Sharada K., Alghamdi W., Karthika K., Alawadi A., Nozima G., Vijayan V. Deep Learning Techniques for Image Recognition and Object Detection, *E3S Web of Conferences* 2023, Vol. 399, Article Number 04032, pp. 234–243.
10. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 86–114.
11. Girshick R. Fast R-CNN, *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 112–123.
12. Ren S., He K., Girshick R. et al. Faster R-CNN: Towards real-time object detection with region proposal networks, *Neural Information Processing Systems (NIPS)*, 2015.
13. Mijwil M., Aggarwal K., Doshi R. et al. The Distinction between R-CNN and Fast R-CNN in Image Analysis: A Performance Comparison, *Asian Journal of Applied Sciences*, 2022, No. 10(5), pp. 429–437.
14. Dong W. Faster R-CNN and YOLOv3: a general analysis between popular object detection networks, *Journal of Physics Conference Series*, 2023, No. 2580(1), № 012016.
15. Redmon J., Divvala S., Girshick R., Farhadi A. You Only Look Once: Unified, Real-Time Object Detection, *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, 118 p.
16. Terven J., Cordova-Esparza D. A comprehensive review of YOLO: from YOLOv1 and beyond, *arXiv: 2304.00501*, 2023, 125 p.
17. Li Y., Fan Q., Huang H. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition, *MDPI Innovative Urban Mobility*, 2023, pp. 35–45.
18. Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C., Berg A., SSD: Single Shot MultiBox Detector, *arXiv 1512.02325*, 2016.
19. Simonyan K., Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition, *arXiv:1409.1556*, 2014, pp 202–212.

Received 05.01.2024.
Accepted 28.02.2024.

DOI 10.15588/1607-3274-2024-1-13

УДК 621.391

ДОСЛІДЖЕННЯ НОВІТНІХ ПІДХОДІВ ДО РОЗПІЗНАВАННЯ ТА КЛАСИФІКАЦІЇ ВІЗУАЛЬНИХ ЗОБРАЖЕНЬ

Лисечко В. П. – канд. техн. наук, професор, професор кафедри транспортних комунікацій, Український державний університет залізничного транспорту, Харків, Україна.

Садовников Б. І. – аспірант кафедри транспортних комунікацій, Український державний університет залізничного транспорту, Харків, Україна.

Комар О. М. – канд. техн. наук, доцент, доцент Національного авіаційного університету, Київ, Україна.

Жученко О. С. – канд. техн. наук, доцент, доцент кафедри транспортних комунікацій, Український державний університет залізничного транспорту, Харків, Україна.

АНОТАЦІЯ

Актуальність. У статті представлено огляд сучасних методів розпізнавання та класифікації візуальних образів на статичних зображеннях або у відеопотоці. Будуть розглянуті різні підходи, включаючи машинне навчання, поточні проблеми цих методів та можливі вдосконалення. Обговорюються найбільші проблеми пошуку та класифікації візуальних зображень. Основний акцент зроблено на огляді таких перспективних алгоритмів, як SSD, YOLO, R-CNN, огляді принципів роботи цих методів, мережевих архітектур.

Мета. Метою роботи є аналіз існуючих досліджень та пошук найкращого алгоритму розпізнавання та класифікації візуальних зображень для подальшої діяльності.

Метод. Основним методом є порівняння різних факторів алгоритмів з метою вибору найбільш перспективного. Існують різні показники для порівняння, такі як швидкість обробки зображень, точність.

Існує ряд досліджень та публікацій, в яких пропонуються методи та алгоритми розв'язання задачі пошуку та класифікації образів на зображенні [3–6]. Слід зазначити, що найбільш перспективні підходи базуються на методах машинного навчання.

Варто зазначити, що запропоновані методи мають недоліки, пов'язані з недосконалою реалізацією алгоритмів Faster R-CNN, YOLO, SSD для роботи з потоковим відео. Вплив цих недоліків можна суттєво зменшити шляхом застосування наступних рішень: розробка комбінованих методів ідентифікації, обробка крайніх випадків – відстеження положення ідентифікованих об'єктів, використання різниці між відеокадрами, додаткова попередня підготовка вхідних даних. Іншим важливим напрямком вдосконалення є оптимізація методів для роботи з відеоданими в реальному часі, оскільки більшість сучасних методів орієнтовані на зображення.

Результати. В результаті проведеного дослідження було знайдено оптимальний алгоритм для подальших досліджень та оптимізацій.

Висновки. Аналіз існуючих робіт та досліджень показав найбільш перспективний алгоритм для подальших оптимізацій та експериментів. Також існуючі підходи все ще мають певний простір для розвитку. Наступним кроком є робота над обраним алгоритмом та дослідження можливостей його вдосконалення.

КЛЮЧОВІ СЛОВА: машинне навчання, комп'ютерний зір, обробка зображень, згорткові нейронні мережі, розпізнавання візуальних образів, класифікація візуальних образів, алгоритми, телекомунікаційні системи.

ЛІТЕРАТУРА

1. An Efficient Pure CNN Network for Medical Image Classification / W. Yue, S. Liu, Y. Li // Applied Sciences. – 2023. – No.13(16). – P. 9226.
2. Metal Additive Manufacturing Parts Inspection Using Convolutional Neural Network/ [W.Cui, Y. Zhang, X. Zhang et al.] // Applied Sciences. – 2020. – No. 10(2). – P. 545.
3. Research of modern NoSQL databases to simplify the process of their design/ [V. P. Lysechko, I. M. Syvolovskyi, B. V. Shevchenko et al.] // Academic journal: Mechanics Transport Communications. – 2023. – Vol. №21, Issue 2, article №2363. – P. 234–242.
4. Experimental study of optimized face recognition algorithms for resource – constrained / [V. P. Lysechko, O. I. Zorina, B. I. Sadovnykov et al.] // Academic journal: Mechanics Transport Communications. – 2023. – Vol. №21, Issue 1, article №2343. – P. 89–95
5. Design and Implementation of Object Detection, Tracking, Counting and Classification Algorithms using Artificial Intelligence for Automated Video Surveillance Applications, Conference / [Mohana, H. V Ravish Aradhya] //24th International Conference on Advanced Computing and Communications. – 2022. – P. 56–60.
6. Object Detection and Classification from a Real-Time Video Using SSD and YOLO Models / [A. Feroz, M. Sultana, R. Hasan et al.] // Computational Intelligence in Pattern Recognition. – 2021. – 405 p.
7. Effects of Open-Source Image Preprocessing on Glaucoma and Glaucoma Suspect Fundus Image Differentiation with CNN [Electronic resource]/ [M. Seker, Y. Köylüoğlu, A. Celebi, B. Bayram]. – 2021 – Access mode: https://doi.org/10.21203/rs.3.rs-1695441/v1.
8. Sadovnykov B. Overview of state-of-the-art image object detection and classification approaches / B. Sadovnykov O. Zhuchenko, K. Perets // Collection of scientific papers of UkrDUZT International scientific and technical conference "Development of scientific and innovative activity in transport". –2023. – Issue 177. – Kharkiv, UkrDUZT. – P. 46–48.
9. Deep Learning Techniques for Image Recognition and Object Detection/ [K. Sharada, W. Alghamdi, K. Karthika et al.] // E3S Web of Conferences. – 2023. – Vol. 399, Article Number 04032. – P. 234–243.
10. Rich feature hierarchies for accurate object detection and semantic segmentation / [R. Girshick, J. Donahue, T. Darrell, J. Malik] // IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2014. – P. 86–114.
11. Girshick R. Fast R-CNN / R. Girshick // IEEE International Conference on Computer Vision (ICCV). –2015. – P. 112–123.
12. Faster R-CNN: Towards real-time object detection with region proposal networks/ [S. Ren, K. He, R. Girshick et al] // Neural Information Processing Systems (NIPS). – 2015.
13. The Distinction between R-CNN and Fast R-CNN in Image Analysis: A Performance Comparison/ [M. Mijwil, K. Aggarwal, R. Doshi et al.] // Asian Journal of Applied Sciences 2022. – No. 10(5). – P. 429–437.
14. Dong W. Faster R-CNN and YOLOv3: a general analysis between popular object detection networks / W. Dong // Journal of Physics Conference Series. – 2023. – No. 2580(1), № 012016.
15. You Only Look Once: Unified, Real-Time Object Detection / [J. Redmon, S. Divvala, R. Girshick, A. Farhadi] // IEEE Conference on Computer Vision and Pattern Recognition. – 2016. – 118 p.
16. Terven J. A comprehensive review of YOLO: from YOLOv1 and beyond / J. Terven, D. Cordova-Esparza // arXiv: 2304.00501. – 2023. – 125 p.
17. Li Y. A Modified YOLOv8 Detection Network for UAV Aerial Image Recognition/ Y. Li, Q. Fan, H. Huang // MDPI Innovative Urban Mobility. – 2023. – P. 35–45.
18. SSD: Single Shot MultiBox Detector // [W. Liu, D. Anguelov, D. Erhan et al.] // arXiv 1512.02325. – 2016.
19. Simonyan K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // arXiv:1409.1556. – 2014. – P. 202–212.

DOI 10.15588/1607-3274-2024-1-13

UDC 004.912

UA-LLM: ADVANCING CONTEXT-BASED QUESTION ANSWERING IN UKRAINIAN THROUGH LARGE LANGUAGE MODELS

Syromiatnikov M. V. – Post-graduate student of the Department of Software Engineering, Odesa Polytechnic National University, Odesa, Ukraine.

Ruvinskaya V. M. – PhD, Professor of the Department of Software Engineering, Odesa Polytechnic National University, Odesa, Ukraine.

ABSTRACT

Context. Context-based question answering, a fundamental task in natural language processing, demands a deep understanding of the language's nuances. While being a sophisticated task, it's an essential part of modern search systems, intelligent assistants, chatbots, and the whole Conversational AI field. While English, Chinese, and other widely spoken languages have gathered an extensive number of datasets, algorithms, and benchmarks, the Ukrainian language, with its rich linguistic heritage and intricate syntax, has remained among low-resource languages in the NLP community, making the Question Answering problem even harder.

Objective. The purpose of this work is to establish and benchmark a set of techniques, leveraging Large Language Models, combined in a single framework for solving the low-resource problem for Context-based question-answering task in Ukrainian.

Method. A simple yet flexible framework for leveraging Large Language Models, developed as a part of this research work, enlights two key methods proposed and evaluated in this paper for dealing with a small amount of training data for context-based question-answering tasks. The first one utilizes Zero-shot and Few-shot learning – the two major subfields of N-shot learning, where N corresponds to the number of training samples, to build a bilingual instruction-based prompt strategy for language models inferencing in an extractive manner (find an answer span in context) instead of their natural generative behavior (summarize the context according to question). The second proposed method is based on the first one, but instead of just answering the question, the language model annotates the input context through the generation of question-answer pairs for the given paragraph. This synthetic data is used for extractive model training. This paper explores both augmentation-based training, when there is some annotated data already, and completely synthetic training, when no data is available. The key benefit of these two methods is the ability to obtain comparable prediction quality even without an expensive and long-term human annotation process.

Results. Two proposed methods for solving the low-to-zero amount of training data problem for context-based question-answering tasks in Ukrainian were implemented and combined into the flexible LLM experimentation framework.

Conclusions. This research comprehensively studied OpenAI GPT-3.5, OpenAI GPT-4, Cohere Command, and Meta LLaMa-2 language understanding capabilities applied to context-based question answering in low-resource Ukrainian. The thorough evaluation of proposed methods on a diverse set of metrics proves their efficiency, unveiling the possibility of building components of search engines, chatbot applications, and standalone general-domain CBQA systems with Ukrainian language support while having almost zero annotated data. The prospect for further research is to extend the scope from the CBQA task evaluated in this paper to all major NLU tasks with the final goal of establishing a complete benchmark for LLMs' capabilities evaluation in the Ukrainian language.

KEYWORDS: large language model, question-answering, few-shot learning, generative annotation.

ABBREVIATIONS

API is an Application Programming Interface;

BERT is a Bidirectional Encoder Representations from Transformers;

BLEU is a Bilingual Evaluation Understudy;

CBQA is a context-based question answering;

CoT is a Chain-of-Thought;

DeBERTa is a Decoding-enhanced BERT with disentangled attention;

EM is an Exact Match;

FLAN is a Fine-tuned Language Net;

GPT is a Generative Pre-trained Transformer;

LLaMA is a Large Language Model Meta AI;

LLM is a Large Language Model;

LM is a Language Model;

MLM is a Masked Language Modeling;

MT is a Machine Translation;

MVP is a Minimum Viable Product;

NLP is a Natural Language Processing;

NLTK is a Natural Language Toolkit;

NLU is a Natural Language Understanding;

PaLM is a Pathways Language Model;

QA is a Question-Answering;

RLHF is a Reinforcement Learning from Human Feedback;

RoBERTa is A Robustly Optimized BERT Pretraining Approach;

ROUGE is a Recall-Oriented Understudy for Gisting Evaluation;

SQuAD is a Stanford Question Answering Dataset;

TrecQA is a Text Retrieval Conference Question Answering;

ULMFiT is a Universal Language Model Fine-tuning.

NOMENCLATURE

A is a contiguous span of words from context C that is the most acceptable answer to question Q;

A_i is a possible answer span from context C;

C is a context represented as a sequence of words;

I_{LANG} is a language used for writing the task instruction included in the prompt;

k is a number of possible answers to question Q;

L is a number of layers of the neural network;

m is a length of question Q;

N is a number of context-question-answer triplets included in the prompt;

n is a length of context C;

DOI 10.15588/1607-3274-2024-1-14

P is a number of parameters of the neural network;

Q is a question represented as a sequence of words;

q_i is an i-th word of question Q;

$S_{POSITIVES}$ is a number of matching pairs where both predicted and true answers are not empty.

T_{GEN_NEG} is a number of generated question-answer pairs with unanswerable questions used for training;

T_{GEN_POS} is a number of generated question-answer pairs with answerable questions used for model training;

TOK_{TOTAL} is a total number of tokens used for both input and generated output;

$T_{UA\text{-}SQUAD}$ is a number of training samples from the UA-SQuAD dataset used for model training;

w_i is an i-th word of context C.

INTRODUCTION

In a century defined by the persistent influx of information and the ever-expanding digital landscape, the ability to extract relevant knowledge from vast repositories of text data has become increasingly paramount. CBQA, a fundamental task in the field of natural language processing, plays a pivotal role in addressing this need by enabling machines to comprehend human language and provide precise answers to user queries within a given context. Nowadays, search engines leverage CBQA systems like Google Quick Answer to provide users with precise and contextually relevant answers to their queries. Additionally, virtual assistants like Siri and Alexa employ context-based QA to facilitate natural language interactions, allowing users to ask questions and receive informative responses.

As the demand for efficient and accurate information retrieval continues to grow, the development of robust and sophisticated context-based QA systems remains a critical pursuit. The ability to emulate human-like comprehension, going beyond mere keyword matching, has always been a tough challenge for machines. However, recent advancements in NLP have ushered in a new era of possibility. The application of the attention mechanism to encoder-decoder architecture, which unlocked deeper context understanding for tasks with long text sequences, like machine translation, can be viewed as a starting point of the "golden age of NLP". Subsequent releases of Transformer architecture and its variations like BERT, RoBERTa, and DeBERTa pushed us as close as possible to human performance on most NLU tasks, including QA [1]. Being pretrained on giant text corpora, these encoder language models require just a little fine-tuning to demonstrate reasonable performance [2]. However, this strategy doesn't work well for low-resource languages [3, 4], including Ukrainian, as they were underrepresented during the pre-training stage, and in most cases, there is just a little-to-zero amount of training samples for fine-tuning.

With a scale from millions to tens of billions of parameters, we are now at the forefront of NLP advances – generative LLMs like GPT and PaLM, which have demonstrated unparalleled capabilities in understanding, generating, and processing human language across a multitude of languages and domains [5]. The ability to follow human instructions, multitasking, and orders of magnitude larger pre-training corpora turned this class of models into a perfect candidate to deal with the low-resource NLU tasks and languages.

While the known methods for CBQA require a substantial amount of annotated data for training, which is not available for most low-resource languages, including Ukrainian, the generalization capabilities encapsulated in LLMs as a result of pre-training with billions of words give them the ability to solve the vast majority of NLU problems, including CBQA, with just a task description and few training samples (few-shot learning) or even without them (zero-shot learning) completely [6]. Moreover, the ability to follow complex instructions combined with high generalization unveils the possibility of using these generative models for data annotation.

The object of study is the process of automatic context-based question answering with the neural network for the low-resource language.

The subjects of study are zero- and few-shot context-based question answering and data annotation with generative LLMs for the low-resource Ukrainian language.

The purpose of the work is to establish and benchmark a set of techniques, leveraging large language models, combined in a single framework for solving the low-resource problem for context-based question-answering task in Ukrainian. This endeavor includes a quantitative objective of reducing the required number of training examples while minimizing any decrease in quality, thus enhancing the feasibility and accessibility of CBQA in low-resource linguistic contexts.

1 PROBLEM STATEMENT

Let C represent a context consisting of a sequence of words $C = (w_1, w_2, \ldots, w_n)$, and Q denote a question represented as a sequence of words $Q = (q_1, q_2, \ldots, q_m)$. The task of context-based question answering can be formalized as follows: find the answer A in context C such that A is the most contextually relevant and correct response to the question Q. Mathematically, we aim to find A as:

$$A = \underset{A_i,\, i \in [1,\, k]}{\arg\max}\ P(A_i \mid Q, C).$$

$P(A_i|Q,C)$ represents the probability that the span A_i is the correct answer to the question Q given the context C.

This problem involves modeling the conditional probability distribution $P(A_i|Q,C)$ using advanced language models and machine learning techniques to accurately and contextually answer a wide range of questions within the given context. The challenge lies in identifying the correct answer span that maximizes this probability, considering the nuances of natural language and context.

2 REVIEW OF THE LITERATURE

Early research in context-based QA primarily focused on rule-based approaches [7] and information retrieval techniques [8]. Systems like IBM's DeepQA, which pow-

DOI 10.15588/1607-3274-2024-1-14

ered Watson, showcased the potential of structured data and knowledge graphs for answering questions, particularly in trivia-style competitions. While these early systems achieved remarkable milestones, they were constrained by their inability to handle the breadth and depth of human language variation and context.

The advent of machine learning techniques, particularly supervised and semi-supervised approaches, marked a significant shift in the QA landscape. Researchers began to explore methods for extracting features from text, creating labeled datasets, and training models to predict correct answers to questions [9]. Notable examples include the development of the TrecQA track and the emergence of datasets like SQuAD dataset, which laid the foundation for benchmarking QA systems.

One significant disadvantage of existing machine learning approaches was their isolated training process requiring a lot of training samples for reasonable performance [10] and generalization ability. The second rise of language models, like ULMFiT, empowered by deep neural networks, popularized domain adaptation and transfer learning – techniques aimed at pre-training of model on a giant amount of textual data to build a foundation for task-specific training with a higher generalization and less annotated examples required for the latter [11].

The subsequent introduction of encoding Transformer-based LMs, such as BERT and its descendants, RoBERTa and DeBERTa, catalyzed a quantum leap in most NLU tasks. These models, pre-trained on vast text corpora, demonstrated the ability to understand human language at an unprecedented scale. With the release of decoding or generative Transformer-based LMs like GPT, the question-answering was divided into two parts:

– extractive QA, where the encoding model tries to answer the question by predicting the most relevant span in the context;

– generative QA, where the model generates an answer to the given question; the context here is not mandatory since large generative models may retain the knowledge from the training stage.

Evaluations demonstrate that while extractive encoders like RoBERTa and DeBERTa perform better for rare terms or domains, generative decoders (GPT) and encoder-decoders (T5) are beneficial for long contexts [12]. For extractive models, the drawback is the required context, and for generative models, it's their hallucination – a tendency to output text sequence that may be structured correctly but wrong from the factuality side [13, 14].

While the fine-tuning stage with a reasonable amount of training data was initially required for both extractive and generative models, scaling from millions to billions and from billions to tens or hundreds of billions of parameters helps to mitigate this problem for the latter. Recent LLMs like PaLM 2 and LLaMA 2 outperform fine-tuned extractive QA models in a single-shot manner on the TriviaQA and BoolQ benchmarks [15, 16].

However, the bigger the language model is, the easier it is for her to hallucinate – generate incorrect facts and harmful information, or not follow the instructions given due to overfitting and a massive amount of controversial data in the training corpora. There are two main strategies to increase the model controllability: prompt engineering and instruction-based fine-tuning.

Prompt engineering generalizes a set of techniques that help to increase language model steerability without adjusting its weights, just through manipulating input (prompt) structure. Widespread techniques are instruction prompting and chain-of-thought. Instruction prompting augments the input sequence, usually a zero-shot or few-shot [17], with natural language instructions, so for example, instead of feeding just a context and a question for the CBQA task, one will also add "Extract the answer on the question using the context below" to the input. Chain-of-thought prompting is a more descriptive way to insert instructions into the input: instead of writing an abstract instruction, it uses one or multiple examples (one- or few-shot learning) to demonstrate how to derive a solution with a sequence of steps (thoughts). While complex tasks like math problems significantly benefit from CoT [18], it brings little to no gain for standard NLU tasks.

In contrast to prompt engineering, instruction-based fine-tuning connects natural language instructions with a diverse set of tasks to increase steerability by adjusting language model weights. FLAN LLM demonstrated that instruction tuning substantially improves zero-shot performance on unseen tasks [19]. Also, a more sophisticated strategy for instruction-based fine-tuning called Reinforcement Learning from Human Feedback, where human evaluations of language model predictions are used for its tuning to minimize bias and harmful generations, adapt it for the chat environment and zero-shot prompting [20]. The list of RLHF-powered models includes Gopher, InstructGPT, and well-known ChatGPT.

The key drawbacks of instruction-based fine-tuning compared to prompt engineering are a greater risk of overfitting, as the model becomes more tailored to the specific instructions or examples used during fine-tuning, potentially limiting its generalizability to a broader range of tasks, a more extensive and labor-intensive data collection process, and the higher computational power required for tuning model with billions of parameters. At the same time, prompt engineering, a more streamlined and efficient approach is mostly limited to high-resource languages as recent research on the multilingual capabilities of LLMs has indicated a significant decrease in quality when applied to low-resource languages [21].

3 MATERIALS AND METHODS

We propose two methods, extending the instruction prompting, to solve the data problem for the CBQA task in the low-resource Ukrainian language: N-shot bilingual instruction prompting and generative data annotation for extractive model training, both using LLMs.

In the proposed N-shot bilingual instruction prompting (Fig. 1), the English language is used for writing natural language instruction for the CBQA task and keywords, specifying the start of context, question, and answer written in Ukrainian. The resulting prompt consists of instruc-

DOI 10.15588/1607-3274-2024-1-14

tion, N annotated examples, where N>=0, and context with the question to be answered by the model. The motivation for using English as the primary language for instructions is based on the fact, that despite pre-training and fine-tuning corpora for LLMs include millions of texts, in most cases, they are dominated by English, and low-resource languages, including Ukrainian, are still underrepresented. For benchmarking purposes, we also evaluated monolingual instruction prompting with instructions and keywords written in Ukrainian. For N-shot learning, we explored both zero-shot and few-shot cases.

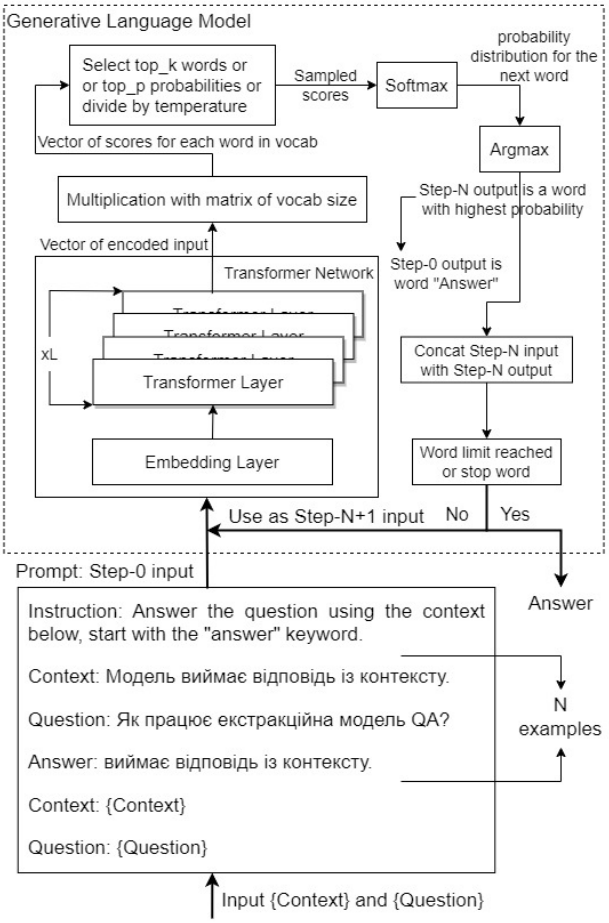


Figure 1 – Visualization of N-shot bilingual instruction prompting method

For simplicity, in Figure 1 above, a “word” term was used to demonstrate a language model’s decoding stage. However, most LMs work with tokens instead, where the token may be a complete word or just a part of it, depending on the word frequency in the training corpora. The word-to-token ratio depends on the language and the tokenization algorithm, but in general, a text with 750 words in English will be encoded into 1000 tokens, while the same number of words in Ukrainian and other low-resource languages may result in 1500–3000 tokens.

The question-answering with a generative language model, demonstrated in Figure 1, follows the standard autoregressive generation process iteratively predicting the next word in a sequence by leveraging the context of previous words. On each prediction step the model utilizes its knowledge acquired during training to estimate the probability distribution over the vocabulary by crafting the context representation of the input sequence and multiplying it with the weight matrix of vocab size. An important sampling step with top-K words, top-P probabilities, or softmax temperature strategies allows to parametrize and control creativity and diversity during the next word selection.

The second proposed method – a generative data annotation for extractive model training (Fig. 2), is based on the first one but introduces three key changes:

1. Gather texts from neutral domains like news portals or encyclopedias.

2. Instead of just answering the question with the given context, the LLM’s task is to annotate these gathered contexts, i.e. generate question-answer pair for each, where the answer span must be entirely presented in the context.

3. Depending on the available amount of data, use these generated annotations from the previous step either as the complete training set or to augment the existing dataset for the extractive QA model training.

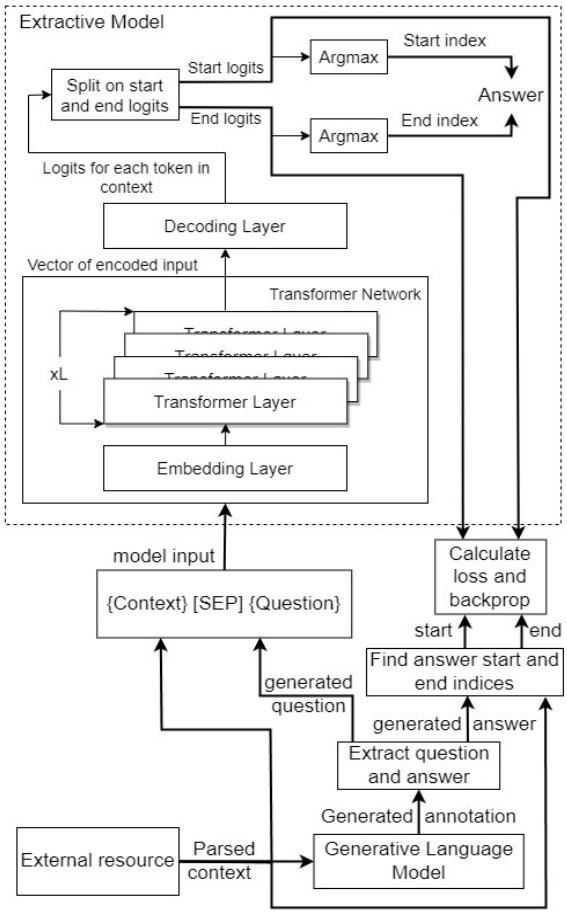


Figure 2 – Visualization of a generative data annotation method for extractive model training

DOI 10.15588/1607-3274-2024-1-14

In contrast to generative LLMs, where this can be achieved simply by adjusting the prompt, to adapt the encoding LM for any downstream task like context-based question answering, it is required to place a specific decoding layer on top of pre-trained LM for projecting encoding sequence context into desired dimension. In Figure 2, one can see the decoding layer placed after the stacked encoding transformer layers, which is just a linear projection. The dot product of sequence context with this linear layer results in two logits for each input token: start and end score. The higher the score is, the more confident the model is that this token is the start or the end of the answer. Therefore, the argmax operation applied to lists of start and end positions logits returns boundaries for the predicted answer. This approach for solving CBQA tasks with encoding Transformer LMs was first introduced with the BERT model release and later became the de facto standard for solving any machine reading comprehension task with transformers.

The direct usage of LLM for predicting answers proposed in the first method may be beneficial in the short-term due to implementation and serving simplicity: the ability to use one of the existing providers and almost zero requirement for annotated data. However, the inability to provide a completely deterministic behavior for LLM's inference due to hardware limitations and their hallucination may be inappropriate for some CBQA use cases, for which the answer's span precision is crucial. The proposed method of generative data annotation for extractive model training helps to avoid the issues described above by applying LLM not for the inference but for the extractive model's training stage. With this data level knowledge distillation, the extractive model (student) learns to mimic only the correct behavior of LLM (teacher) while being completely deterministic and cheaper due to orders of magnitude smaller size.

The following generative LLMs were used for the evaluation of the proposed methods: OpenAI GPT-3.5 Turbo, OpenAI GPT-4 and GPT-4 Turbo, Cohere Command, and Meta LLAMA 2 Chat. We followed two essential rules for the language model selection process:

– language support: the model has to support both English and Ukrainian languages;

– publicly available, the LLM has to be available for everyone either in the form of API or an open-sourced checkpoint to make results reproduction possible.

Let's briefly review each of the selected generative language models:

1. GPT-3.5 Turbo, also known as ChatGPT – language model from OpenAI, based on GPT-3 and optimized for conversations and instructions with RLHF fine-tuning while retaining most of the knowledge seen during pre-training. The model hyperparameters were not disclosed, but the original GPT-3 consists of 96 transformer decoder layers with a hidden size of 12288, totaling 175 billion parameters [22]. GPT-3.5 Turbo is available through OpenAI API in two variations: with a maximum input of 4098 and 16386 tokens.

2. GPT-4 – the latest generation LLM from OpenAI. GPT-4 outperforms GPT-3.5, demonstrating state-of-the-art results on NLU tasks and professional and academic exams [23], as well as achieving human-level performance on many of them [5]. The architecture details of GPT-4 were not disclosed, but according to rumors, the LLM leverages the Mixture-of-Experts technique with 8 GPT-3 expert models, resulting in approximately 1.7 trillion parameters. Standard GPT-4 is accessible through OpenAI API with a maximum input length of 8192 and 32768 tokens, while GPT-4 Turbo – the latest model released by OpenAI with improved instruction following, supports context of up to 128000 tokens. In this research, both standard and turbo versions were evaluated.

3. Command – a generative model built by Cohere and optimized for instruction-like prompts. Cohere provides access through API for the two versions of this model: command with 52.4 billion parameters and command-light with only 6.1 billion parameters. In this paper, the standard version is evaluated as it demonstrated reasonable quality on QA tests of the HELM benchmark [24].

4. LLaMA 2 – is a second generation of the open-source LLM family by Meta. Three different base models with 7, 13, and 70 billion parameters were pre-trained on 2 trillion tokens with a context size of 4096 [16]. Each of these models also has a fine-tuned version for chat (LLaMA Chat) and code (Code LLaMA). A version with 13 billion parameters fine-tuned for chat on over 1 million human annotations was selected for this research as it's an optimal trade-off between model size and performance.

In addition to generative LLMs, the following encoding LMs were used to build extractive models as a part of the generative data annotation for extractive model training and to establish a baseline:

1. XLM-RoBERTa – multilingual version of RoBERTa model by Meta pretrained on 2.5TB of text containing 100 languages in a self-supervised fashion with MLM objective [4]. More precisely, this language model based on encoding Transformer architecture was trained to predict masked words in the input sequence leveraging bidirectional context. For this research, the smaller version of XLM-RoBERTa with only Ukrainian and English tokens kept in the embedding layer was used.

2. DeBERTa – a family of Transformer-based encoding models by Microsoft demonstrating state-of-the-art performance and surpassing human performance on the SuperGLUE benchmark [25]. The usage of disentangled attention and enhanced mask decoder allows DeBERTa to outperform BERT and RoBERTa on most NLU tasks, including CBQA [25]. The third version of multilingual DeBERTa improved by ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing [26] was selected for this research.

For prediction evaluation, the following string-based metrics were selected:

1. SQuAD F1 – the harmonic mean of precision and recall adapted for SQuAD, a machine reading comprehension benchmark. The precision of the question-answering models is determined by the ratio of the number of cor-

DOI 10.15588/1607-3274-2024-1-14

rectly predicted words of the answer to the total predicted number of words of the answer and recall – the ratio of correctly predicted words of the answer to the total number of words in the true answer [27].

2. SQuAD Exact match – another metric used in the SQuAD benchmark along with F1. EM measures the exact match between strings on character level: EM = 1 for the exact match of the predicted answer span with the ground truth, and 0 in all other cases. For negative annotations with an empty string as a ground truth, any predicted text will result in EM = 0, even if it is a single character.

3. Any match F1 – similar to classification or detection F1 measure. This metric was used to measure LLM's hallucination and steerability: Any is true positive if there is a ground truth answer and the model predicted any span from the context, or if there is an empty true answer and the prediction is empty as well.

3. Partial match F1 – a stricter subset of Any metric, for which a true positive case happens only when there is an index-based overlap between the predicted answer span and ground truth.

4. BLEU – a metric initially implemented for evaluating MT systems by measuring the n-gram similarity between the translated text and high-quality reference translation [28]. While having an obvious drawback when applied for MT evaluation due to comparing tokens and not meaning, BLEU fits well for CBQA, allowing to evaluate LLM's predictions with augmentations or hallucinations.

5. ROUGE – a metric inspired by BLEU and used to score summarization algorithms. While both BLEU and ROUGE use n-grams to calculate similarity, the former is precision-oriented as it measures the number of n-grams from prediction appearing in the reference text, and the latter is recall-oriented instead, measuring how many n-grams from reference are presented in the prediction [29].

In order to simplify the LLM's inferencing and evaluation process for CBQA and make it flexible, the following experimentation framework named "UA-LLM" was designed and implemented (Fig. 3). The framework consists of 5 modules written in Python chained with OmegaConf-based config module by Hydra configuration framework.

Hydra's key features are the ability to dynamically create a hierarchical configuration by composition and override it through config files and the command line [30], as well as automatic recursive instantiation of Python objects during the runtime based on the config provided.

The entry point defined in main, accessible from the command line, expects a path to the OmegaConf file in YAML format with task config. The task module contains abstract classes with logic for prediction, evaluation, and annotation, along with their implementation for CBQA. Each task requires data reader and writer objects from the Data module and the model object from the LLM module. For each iteration of its run, the task gets the portion of data, combines it with the preconfigured prompt, and then feeds the obtained input to get the generated prediction. Depending on the task, the reader is used to load input data or predictions from an arbitrary source, and the writer is used to output predictions, annotations, or evaluation results. CSV tabular format reader and writer were implemented for CBQA experimentation. The data module also supports streaming for reading and writing to deal with large files.

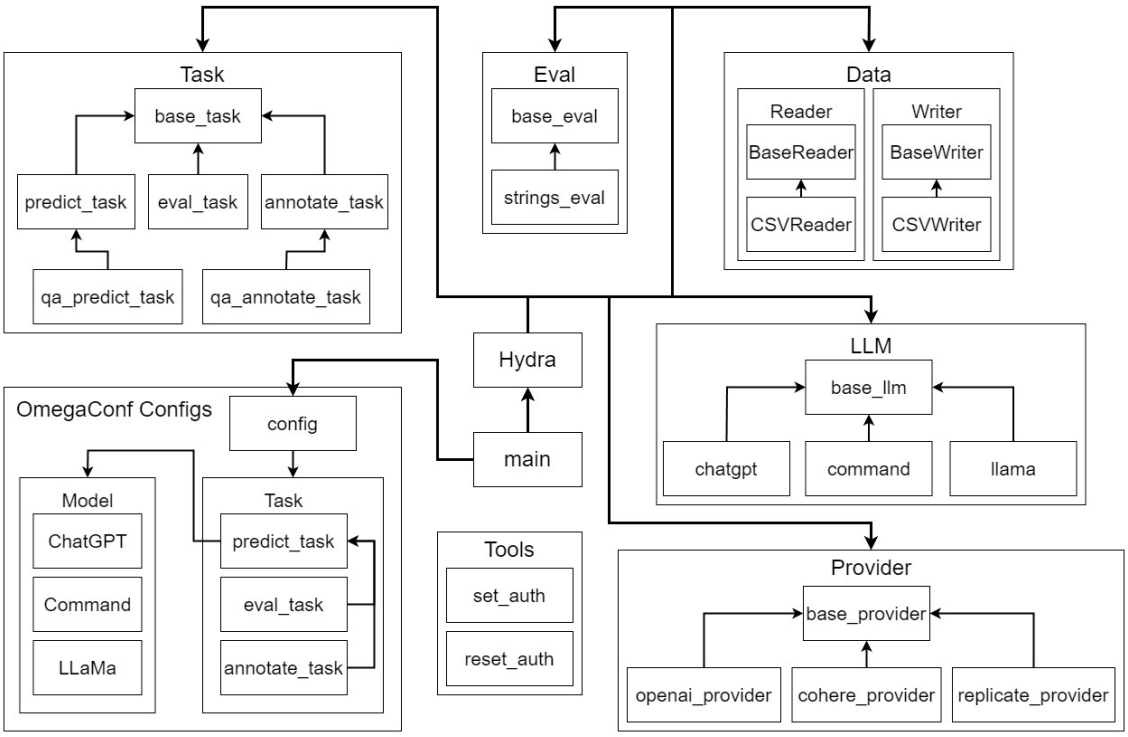


Figure 3 – Architecture diagram of the experimentation framework "UA-LLM"

DOI 10.15588/1607-3274-2024-1-14

Each model from the LLM module is just a simple wrapper for the provider object, which performs all low-level operations like establishing a connection to the model server, making requests, processing errors, and collecting session statistics. The framework contains providers for interaction with models accessible through API from OpenAI, Cohere, and Replicate. The eval module is a place for metrics grouped by evaluation level that could be used for prediction scoring. While strings_eval implements all metrics for strings comparison listed in the section above, one can select a subset with only desired metrics included by adjusting the argument in the task config.

Though the implemented framework for LLM experimentations primarily focuses on CBQA, it's easy to adapt it for any other NLU task since its decomposed architecture with most independent components provides high flexibility and generalization.

4 EXPERIMENTS

The localized version of the SQuAD 2.0 dataset was selected for methods evaluation. Stanford Question Answering Dataset (SQuAD) is a CBQA dataset consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable [31]. The second version combines 100,000 questions from version 1.1 with more than 50,000 unanswerable questions written by annotators as well. However, work on the Ukrainian version of this dataset is still in progress, and so UA-SQuAD contains only 13,859 question-answer pairs, among which 2,927 are unanswerable questions [32].

This dataset with almost 14 thousand examples was grouped by 2620 unique contexts, and then these groups were randomly split into train/validation/tests sets with 80%/10%/10% proportion resulting in 11,173 pairs for train and 1,343 – validation and test.

For the N-shot bilingual instruction prompting method experiments, 0-shot, 2-shot, and 4-shot cases were selected. The annotated examples for few-shot prompting were randomly sampled from the train set. Only the temperature hyperparameter of LLM's generation process was tuned, and the set of parameters for each model was determined based on the manual evaluation of 100 predictions with a focus on steerability maximization and hallucination minimization. Evaluation with a temperature equal to 0 was prioritized for each model as it provides almost complete deterministic behavior. In addition to evaluations of bilingual instructions, monolingual instructions were evaluated to establish a baseline.

The complete set of successful experiments for N-shot bilingual instruction prompting method evaluation is listed in Table 1. The pricing columns correspond to the model request price for text processing and generation.

Table 1 – Parameters for N-shot bilingual instruction prompting method evaluation
* – OpenAI has not disclosed hyperparameters for GPT-3.5-turbo, GPT-4, and GPT-4-turbo models

Id	Model	Architecture			Prompt			Pricing, $ per 1000 tokens	
		L	*P*, bn	Context	*N*	Instruction language	Temperature	Input tokens	Output tokens
1	LLaMA-2-Chat	40	13	4096	0	UA	0.1	0.003	0.003
2	LLaMA-2-Chat	40	13	4096	0	UA	0.5	0.003	0.003
3	LLaMA-2-Chat	40	13	4096	0	UA	0.9	0.003	0.003
4	LLaMA-2-Chat	40	13	4096	0	EN	0.1	0.003	0.003
5	LLaMA-2-Chat	40	13	4096	0	EN	0.5	0.003	0.003
6	LLaMA-2-Chat	40	13	4096	0	EN	0.9	0.003	0.003
7	Command	N/A	52.4	4096	0	UA	0.0	0.015	0.02
8	Command	N/A	52.4	4096	0	EN	0.0	0.015	0.02
9	Command	N/A	52.4	4096	2	UA	0.0	0.015	0.02
10	Command	N/A	52.4	4096	2	EN	0.0	0.015	0.02
11	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	UA	0.0	0.0015	0.002
12	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	UA	0.5	0.0015	0.002
13	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	UA	0.9	0.0015	0.002
14	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	UA	0.0	0.0015	0.002
15	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	UA	0.5	0.0015	0.002
16	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	UA	0.9	0.0015	0.002
17	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	4	UA	0.0	0.0015	0.002
18	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	4	UA	0.5	0.0015	0.002
19	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	EN	0.0	0.0015	0.002
20	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	EN	0.5	0.0015	0.002
21	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	0	EN	0.9	0.0015	0.002
22	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	EN	0.0	0.0015	0.002
23	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	EN	0.5	0.0015	0.002
24	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	2	EN	0.9	0.0015	0.002
25	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	4	EN	0.0	0.0015	0.002
26	GPT-3.5-turbo-06.13	up to 96*	up to 175*	4096	4	EN	0.5	0.0015	0.002
27	GPT-4	N/A	1700*	8192	0	UA	0.0	0.03	0.06
28	GPT-4	N/A	1700*	8192	0	EN	0.0	0.03	0.06
29	GPT-4-turbo-preview	N/A	N/A	128000	0	UA	0.0	0.01	0.03
30	GPT-4-turbo-preview	N/A	N/A	128000	0	EN	0.0	0.01	0.03

DOI 10.15588/1607-3274-2024-1-14

Few-shot experiments with LLAMA-2-Chat were not included in the final table since with N>0 the model stopped to follow instructions and hallucinated in all test generations, making answer extraction impossible. In addition, limited experiments were conducted with Command, GPT-4, and GPT-4 Turbo models due to usage limits and significant inference pricing.

English and Ukrainian instructions used in both methods were written with a focus on coherence and clarity and were not optimized for any observed LLM. For few-shot instructions, examples were sampled randomly from the training set. All LLM prompts were fed as a single message, without splitting them into system prompt and user prompt, since it did not affect prediction quality for the GPT and Command models and demonstrated higher steerability for the LLaMA model. Prompts used for both methods are listed on the paper's page[1] in the GitHub repository.

The best model and set of parameters in terms of quality and pricing determined from evaluations of the N-shot bilingual instruction prompting method in the results section was later selected as an annotator for generative data annotation for the extractive model training method.

External contexts for the data annotation process were crawled from the TSN news portal. Since news texts usually consist of multiple paragraphs or media objects, collected texts were preprocessed to remove unknown symbols, common phrases like "Read more" or "Similar articles", multiple spaces, and join paragraphs in case there are multiple of them. Some of these texts were then filtered out due to being shorter than 120 or longer than 350 tokens, as the maximum sequence length for extractive models was established at 384 tokens, for which a bit more than 30 tokens were reserved for questions. Finally, 2000 contexts for annotation were randomly sampled from the resulting amount.

The prompt from the best experiment was adapted for the data annotation task: instead of answer extraction, the request became to generate five question-answer pairs for the given context, where four pairs are positive – questions are answerable, and the last one, negative, should be unanswerable. Given the 2000 contexts for annotation, this should result in 10,000 question-answer pairs ideally, however, the unavoidable hallucinations like word order change in answer span or plurality and inflection modifications make the annotation postprocessing pipeline filter some pairs.

Training configs for extractive models used to establish a baseline and those used for generative data annotation for the extractive model training method evaluation are demonstrated in Table 2. For this method, both training only on generated annotations and augmentation-based training approaches with adding generated positive question-answer pairs were evaluated. All extractive models were trained on a single GPU for three epochs with a batch size of 12, a learning rate of 3e-05, a max sequence length of 384, a dropout rate of 0.1, and an Adam optimizer.

Table 2 – Parameters for extractive models training

Id	Model	P, mln	$T_{\text{UA-SQUAD}}$	$T_{\text{GEN\_POS}}$	$T_{\text{GEN\_NEG}}$
31	XLM-RoBERTa	110	11173	0	0
32	DeBERTa	278	11173	0	0
33	XLM-RoBERTa	110	0	6156	1724
34	DeBERTa	278	0	6156	1724
35	XLM-RoBERTa	110	11173	6156	0
36	DeBERTa	278	11173	6156	0

For augmentation-based experiments, where both $T_{\text{UA-SQUAD}}$ and $T_{\text{GEN\_POS}}$ were used for the extractive model training, the final train set was created by concatenation of these two sets with subsequent random shuffling. All experiments, including those with $T_{\text{UA-SQUAD}}$ equal to 0, used the validation set of UA-SQUAD dataset. The intermediate evaluation step aimed at filtering unpromising parameter combinations [33] was based on N-shot bilingual instruction prompting and resulted in only generated positives being used for augmentation.

5 RESULTS

Along with quality, the pricing aspect of LLM utilization plays an important role in this research as it defines whether proposed methods for solving the low-resource problem of language could be reproduced and adopted. Table 3 demonstrates the costs of the N-shot bilingual instruction prompting method for each unique set of parameters. In the table, $\text{TOK}_{\text{TOTAL}}$ and price columns were calculated as the sum of values for each of all 1343 examples from the test set. The detailed stats for the temperature parameter weren't included in the table since it affects only the length of the generated answer, so its influence on the total price is close to zero compared to the instruction language parameter I_{LANG} and the number of training examples N added to the prompt.

Table 3 – Influence of N-shot bilingual instruction prompting method parameters on inference pricing

Model	N	I_{LANG}	$\text{TOK}_{\text{TOTAL}}$	Price, $
LLaMA-2-Chat	0	UA	706272	2.12
LLaMA-2-Chat	0	EN	524563	1.58
Command	0	UA	1227015	18.58
Command	0	EN	794381	12.02
Command	2	UA	2447285	36.77
Command	2	EN	2025534	30.46
GPT-3.5-turbo-06.13	0	UA	968301	1.47
GPT-3.5-turbo-06.13	0	EN	676416	1.03
GPT-3.5-turbo-06.13	2	UA	1976186	2.97
GPT-3.5-turbo-06.13	2	EN	1686699	2.54
GPT-3.5-turbo-06.13	4	UA	3663482	5.51
GPT-3.5-turbo-06.13	4	EN	3372538	5.07
GPT-4	0	UA	961822	29.42
GPT-4	0	EN	679322	21.01
GPT-4-turbo-preview	0	UA	968681	10.01
GPT-4-turbo-preview	0	EN	684890	7.19

Results of the quality evaluation for the N-shot bilingual instruction prompting method are demonstrated in Table 4.

1 - https://github.com/NLPForUA/UA-LLM/blob/main/paper/README.md

DOI 10.15588/1607-3274-2024-1-14

Table 4 – Results of N-shot bilingual instruction prompting method evaluation, numbers in bold represent the highest metric score per model

Id	Model	SQuAD Metrics					N-gram Metrics			Index-based Metrics	
		EM	F1	Pos EM	Pos F1	Neg F1	BLEU-1	ROUGE-1	$S_{POSITIVES}$	Partial F1	Any F1
1	LLaMA-2-Chat	13.70	25.89	16.70	31.72	0.79	0.24	0.28	1089	0.28	0.30
2	LLaMA-2-Chat	13.48	25.95	16.33	31.69	**1.19**	0.25	0.29	1089	0.27	0.30
3	LLaMA-2-Chat	11.39	22.47	13.94	27.59	0.40	0.21	0.25	1089	0.24	0.26
4	LLaMA-2-Chat	**24.57**	**41.05**	**30.09**	**50.40**	0.79	**0.41**	**0.46**	1090	**0.48**	**0.52**
5	LLaMA-2-Chat	23.98	40.03	29.36	49.14	0.79	0.40	0.45	1090	0.47	0.51
6	LLaMA-2-Chat	23.38	38.35	28.53	46.98	**1.19**	0.39	0.43	1090	0.45	0.49
7	Command	26.43	40.13	**31.47**	**48.34**	4.74	0.42	0.48	1048	0.60	0.73
8	Command	27.48	39.09	29.63	43.94	18.18	0.45	0.51	910	0.58	0.74
9	Command	29.56	40.99	26.70	40.78	**41.90**	0.42	0.49	879	0.56	0.68
10	Command	**32.02**	**44.76**	**31.47**	47.17	34.39	**0.46**	**0.54**	924	**0.62**	**0.76**
11	GPT-3.5-turbo-06.13	36.04	54.34	34.40	56.96	43.08	0.50	0.57	1035	0.59	0.63
12	GPT-3.5-turbo-06.13	35.96	54.43	34.86	57.61	40.71	0.51	0.58	1038	0.59	0.63
13	GPT-3.5-turbo-06.13	32.91	51.87	31.19	54.55	40.32	0.47	0.54	1040	0.55	0.60
14	GPT-3.5-turbo-06.13	45.27	61.15	44.31	63.88	49.41	0.58	0.63	1011	0.65	0.69
15	GPT-3.5-turbo-06.13	43.34	59.67	42.57	62.69	46.64	0.57	0.63	1009	0.64	0.68
16	GPT-3.5-turbo-06.13	42.37	58.90	41.01	61.38	48.22	0.55	0.60	1010	0.63	0.66
17	GPT-3.5-turbo-06.13	44.30	60.78	43.12	63.42	49.41	0.57	0.62	1013	0.63	0.67
18	GPT-3.5-turbo-06.13	43.26	59.33	42.48	62.28	46.64	0.57	0.62	1005	0.63	0.66
19	GPT-3.5-turbo-06.13	40.95	57.85	36.79	57.61	**58.89**	0.55	0.61	979	0.66	0.71
20	GPT-3.5-turbo-06.13	40.58	57.96	36.88	58.30	56.52	0.55	0.62	981	0.65	0.70
21	GPT-3.5-turbo-06.13	38.42	56.30	34.77	56.80	54.15	0.54	0.60	974	0.62	0.66
22	GPT-3.5-turbo-06.13	46.01	62.85	**47.52**	**68.26**	39.53	**0.61**	0.66	1054	**0.72**	**0.76**
23	GPT-3.5-turbo-06.13	45.72	62.30	47.61	68.05	37.55	**0.61**	0.66	1050	**0.72**	**0.76**
24	GPT-3.5-turbo-06.13	43.48	60.62	44.77	65.88	37.94	0.58	0.64	1050	0.69	0.73
25	GPT-3.5-turbo-06.13	**46.24**	**63.36**	45.69	66.79	48.62	**0.61**	**0.67**	1027	0.71	0.75
26	GPT-3.5-turbo-06.13	**46.24**	62.81	46.06	66.47	47.04	0.60	0.65	1035	0.70	0.75
27	GPT-4	55.47	72.76	**52.57**	**73.88**	67.98	0.68	0.74	1061	0.87	0.89
28	GPT-4	52.94	71.35	49.45	72.13	67.98	0.66	0.72	1059	0.87	0.89
29	GPT-4-turbo-preview	**58.08**	**74.64**	51.47	71.88	86.56	**0.71**	**0.77**	1003	**0.88**	**0.90**
30	GPT-4-turbo-preview	55.92	72.73	48.44	69.15	**88.14**	0.70	**0.77**	960	0.87	**0.90**

The id column from the results table above corresponds to the same column from Table 1 with experiment parameters. Pos and Neg in the SQuAD metrics section are abbreviations for positive or answerable questions and negative or unanswerable questions. BLEU-1 and ROUGE-1 measure unigram similarity between the prediction and true answer using the NLTK's tokenizer to split sequences on words during the preprocessing. These N-gram metrics only support cases when both prediction and true answer are non-empty strings. The total number of these positive pairs for each evaluation is demonstrated in $S_{POSITIVES}$ column. All evaluations were performed on the test set with 262 unique contexts and 5.12 questions on average per context, reaching 1343 question-answer pairs in total, 1090 of which are positive and 253 – negative.

One of the best sets of parameters in terms of demonstrated quality and prediction price – 2-shot bilingual prompting for GPT-3.5 Turbo model with instruction in English and temperature parameter equal to zero (Id 22 in Tables 1 and 4) was chosen for generative data annotation. The generative annotation process for 2000 external contexts resulted in 9972 generated question-answer pairs, 2092 of which were filtered out due to the answer span not being presented in the context. Among the 7880 valid pairs, 6156 were positive and 1724 – negative. The total price for the annotation with 2.8 million tokens used for input and 526 thousand output tokens is $5.79.

The complete results of the evaluation of baseline extractive models trained on the UA-SQuAD dataset, as well as those trained entirely on or augmented by generated annotations, are demonstrated in Table 5.

Table 5 – Results of extractive models evaluation, numbers in bold represent the highest metric score per model

Id	Model	SQuAD Metrics					N-gram Metrics			Index-based Metrics	
		EM	*F1*	Pos EM	Pos F1	Neg F1	BLEU-1	ROUGE-1	$S_{POSITIVES}$	Partial F1	Any F1
31	XLM-RoBERTa	53.83	66.56	52.39	68.06	**60.08**	0.69	0.75	976	0.83	0.89
32	DeBERTa	59.05	**72.91**	57.06	74.15	**67.59**	**0.73**	**0.79**	1017	**0.88**	0.92
33	XLM-RoBERTa	35.82	48.67	36.97	52.81	30.83	0.56	0.62	899	0.73	0.80
34	DeBERTa	39.61	53.33	41.10	58.00	33.20	0.63	0.69	889	0.76	0.81
35	XLM-RoBERTa	**54.58**	**67.30**	**53.39**	**69.06**	59.68	**0.70**	**0.76**	983	**0.84**	**0.90**
36	DeBERTa	**60.31**	72.80	**58.81**	**74.19**	66.80	**0.73**	0.78	1023	0.87	**0.93**

DOI 10.15588/1607-3274-2024-1-14

The id column from the extractive models' evaluation table above corresponds to the column with the same name from Table 2 with parameters for extractive models training. In addition, for extractive models evaluation, the same test set and metrics as for generative language models (Table 4) were used, so the results in these two tables are comparable.

6 DISCUSSION

In this section, we present a detailed discussion of the results obtained from our experimental evaluations of proposed methods for context-based question answering: N-shot bilingual instruction prompting and generative data annotation for extractive model training. Our study aimed to investigate the efficacy of these novel approaches in enhancing the performance of low-resource QA where only little to no training data is available.

Our results demonstrate that N-shot bilingual instruction prompting can indeed be an effective strategy for improving the LLM's capabilities for solving CBQA tasks in low-resource languages, particularly in Ukrainian. For all language models, except GPT-4 and GPT-4 Turbo, zero-shot bilingual prompts with instruction written in high-resource English demonstrate a significant increase for all evaluated metrics compared to the same parameters set with instruction in Ukrainian.

The biggest gain from applying a bilingual prompt was achieved for the smallest generative LM evaluated – LLaMA-2-Chat with only 13 billion parameters: a comparison of evaluations from Table 4 with id 1 (monolingual) and 4 (bilingual) demonstrate a 10.87 and 15.16 points increase of SQuAD's EM and F1 respectively for the latter, as well as up to 1.7 times higher scores for BLEU, ROUGE, Partial and Any F1. Such a vast difference could be explained by the relatively small model size and low Ukrainian language presence of 0.07% in the training corpora since the absolute gain decreases for the larger models, like Command and GPT-3.5 Turbo, with orders of magnitude more parameters. Cohere Command with zero-shot instruction prompting demonstrates similar performance for monolingual and bilingual cases, with the former being better for answerable questions and the latter – for unanswerable. At the same time, the same setup for GPT-3.5 Turbo (id 19 in Table 4) allows to achieve SQuAD EM of 40.58 or 13.6% increase, SQuAD F1 of 57.85 or 6.5% increase, as well as up to 10% growth for n-gram metrics BLEU and ROUGE and up to 12% – for index-based Partial F1 and Any F1. The best zero-shot CBQA performance across all evaluated LLMs was demonstrated by OpenAI GPT-4 Turbo with monolingual prompting (id 29): 2 points higher scores than the ones by the proposed bilingual approach (id 30) on average for SQuAD metrics. The reason behind this is that GPT-4 Turbo is a significantly bigger LLM and better optimized for instructions and multilingualism compared to GPT-3.5 Turbo [34].

One noteworthy observation from our experiments is the influence of the size of the N-shot prompt on model performance. We found that as the number of example question-answer pairs in the second language increased, so did the model's ability to handle questions in that language. However, the rate of improvement tended to diminish beyond a certain point, suggesting that a modest-sized N-shot prompt can yield substantial benefits without overwhelming the model. From Table 4 with the evaluation results, it can be seen that N-shot bilingual prompting demonstrates the best scores for the Command model (id 10) with N=2, while the highest score for GPT-3.5 Turbo is the one with N=4. In addition, it was observed that few-shot monolingual prompting with instruction in Ukrainian also helps to increase the quality of predictions: two-shot prompt (id 14) improved SQuAD EM from 36.04 to 45.27 (25.6% gain) and F1 from 54.34 to 61.15 points (12.5% gain), BLEU and ROUGE were also increased by 16% and 10.5% correspondingly.

The best GPT-3.5 Turbo scores achieved with bilingual prompt and N=4 (id 24) are 46.24 and 62.81 for SQuAD EM and F1, with a relative increase of 28.3% and 16.6% over the monolingual zero-shot baseline (id 11). While the 4-shot bilingual instruction prompting demonstrates the highest scores among all GPT-3.5 Turbo evaluations, the difference for all metrics is no more than 1% between N=2 (id 22) and N=4, illustrating our observation around the quality improvement decay with an increase of N beyond the certain point.

Comparing the results of LLMs evaluation with extractive models (Table 5), one can see that both monolingual and bilingual instruction prompts allow GPT-4 and GPT-4 Turbo to outperform the baseline XLM-RoBERTa model (id 31) with more than 11 thousand annotated examples used for training in contrast to 0 samples used for the GPT. Moreover, GPT-4 Turbo with zero-shot prompting surpasses baseline DeBERTa's (id 32) SQuAD F1 score (74.64 vs 72.91) and strongly outperforms its Negative F1 (86.56 vs 67.59). While the GPT-4 and GPT-4 Turbo are prohibitively expensive, an order of magnitudes cheaper model, the GPT-3.5 Turbo with 4-shot bilingual instruction prompting allows to achieve 95.2% and 86.9% of baseline XLM-RoBERTa and DeBERTa models performance according to SQuAD F1.

The results of the evaluation of generative data annotation for extractive model training presented in Table 5 demonstrate that with just 2000 general-domain texts and $5.79, it is possible to obtain annotations sufficient to train XLM-RoBERTa or DeBERTa, achieving 73.1% (ids 33–34) of SQuAD F1 score of the same model trained with significantly more expensive human-annotated data (ids 31–32). Also, usage of the generative annotation for training data augmentation increased XLM-RoBERTa's scores (id 35) by 1 point for all metrics except for Negative F1, compared to the baseline result (id 31), while for DeBERTa (id 36), only 1–2 points gain for SQuAD EM, Positive F1, and Any F1 was observed.

The advantage of the N-shot bilingual instruction prompting method proposed in this paper for solving CBQA tasks in low-resource language is the possibility of achieving up to 95% of the extractive model's quality trained on tens of thousands of annotated samples with

DOI 10.15588/1607-3274-2024-1-14

just 2 to 4 examples, thus solving the problem of close-to-zero amount of annotated data. The drawback of this method, however, is its high price: serving LLM with tens or hundreds of billions of parameters is enormously more expensive than inferencing an extractive model with only one or two hundred million parameters. Additionally, most LLMs available through API are billed per input and output text lengths.

The advantage of the generative data annotation for the extractive model training method proposed in this research is that it helps to achieve up to 73.1% of the prediction quality of the extractive model trained with significantly more expensive human annotations while using only cheap generative annotations. In addition, combining generative and human annotations for extractive model training slightly increases its overall performance. However, the key benefit of this method is that it takes the best of two worlds: distilling LLM's knowledge into the extractive model through training on generative annotations makes it possible to achieve MVP solutions with reasonable quality relatively fast while saving on the inference pricing due to expensive LLM being used only for data annotation. At the same time, the key disadvantage of this method is the lack of ability to evaluate the correctness of annotations generated by LLM without additional spending on human annotators.

CONCLUSIONS

The scientific novelty of the obtained results is that the N-shot bilingual instruction prompting method for solving CBQA tasks in low-resource Ukrainian language with a low-to-zero amount of training data is first proposed. This method, compared to existing, utilizes large language models in a zero- or few-shot manner with instruction written in high-resource English to increase the model steerability and prediction quality, taking advantage of the fact that most large language models during the training stage have seen more text in English than in all other languages combined.

Based on the first method, a method of generative data annotation for the extractive model training was also proposed as a solution for low-resource CBQA. In contrast to existing solutions with only question or answer generation, this method leverages LLM as a data annotator to generate complete question-answer pairs for the given contexts, allowing the extractive model to be trained without annotated data.

The application of the N-shot bilingual instruction prompting method with the GPT-3.5 Turbo model and four examples (N=4) achieved a 28.3% (46.24 vs 36.04) and 16.6% (62.81 vs 54.34) relative increase for SQuAD EM and F1 metrics, a 22% (0.61 vs 0.50) and 17.5% (0.67 vs 0.57) relative gain for unigram BLEU and ROUGE, as well as 20.3% (0.71 vs 0.59) and 19% (0.75 vs 0.63) improvement for Partial and Any F1 compared to baseline zero-shot (N=0) monolingual method. Also, this method solves the key problem of CBQA tasks in low-resource languages, the lack of annotated training data, retaining 95.2% and 86.9% of baseline XLM-RoBERTa and DeBERTa models prediction quality according to SQuAD F1 metric, while using 2793.3 times less annotated examples (4 instead of 11173).

Training extractive models on generated annotations achieved 73.1% of the practical ceiling of these models trained on human annotations, spending only $5.79 on the labeling process. Moreover, combining these generative annotations with the existing training set increased XLM-RoBERTa's results by 1 point for SQuAD, N-gram, and index-based metrics.

Lastly, to the best of our knowledge, this paper provides the first comprehensive study and evaluation of publicly available large language models' capabilities in Ukrainian context-based question answering.

The practical significance is that the open-source experimentation framework implementing the two methods proposed in this paper for solving context-based question-answering tasks with large language models is developed. The proposed methods formed the basis of the developed program system for the fully automated development of task-agnostic language models.

In addition, the conducted experiments and their evaluation results demonstrate the possibility of applying these methods for building components of search engines and intelligent chatbot applications, as well as standalone general-domain CBQA systems with Ukrainian language support and a low-to-zero amount of annotated data for training.

The limitations of this research should be duly acknowledged. First, the study primarily focuses on a single context-based question-answering dataset and one specific low-resource language, Ukrainian. While the findings provide valuable insights, they may not be universally applicable to other low-resource languages or fully encompass the diverse range of real-world contexts, questions, and answers.

Second, the evaluation metrics used to assess the effectiveness of N-shot bilingual instruction prompting and generative data annotation for the extractive model training methods may not capture all aspects of performance, and further research into more nuanced evaluation methodologies is warranted.

The prospect for further research is to broaden the investigation scope beyond the context-based question-answering task, which has been the primary focus of this research. The ultimate objective is to create a comprehensive benchmark that thoroughly assesses the capabilities of large language models within the context of the Ukrainian natural language understanding and generation, providing a holistic perspective on their performance and adaptability.

ACKNOWLEDGEMENTS

We would like to extend our heartfelt gratitude to the OpenAI team for providing us with early access to the GPT family of models. We also appreciate the contributions of Meta AI for making the LLaMA model family openly accessible.

DOI 10.15588/1607-3274-2024-1-14

REFERENCES

1. Rajpurkar P. The Stanford Question Answering Leaderboard [Electronic resource]. Access mode: https://rajpurkar.github.io/SQuAD-explorer/
2. Devlin J., Chang M., Lee K. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, 2–7 June 2019, proceedings. Stroudsburg, Association for Computational Linguistics, 2019, Vol. 1, pp. 4171–4186. DOI: 10.18653/V1/N19-1423
3. Conneau A., Lample G. Cross-lingual language model pretraining, *33rd International Conference on Neural Information Processing Systems*, Vancouver, 8–14 December 2019, proceedings. New York, Curran Associates Inc., 2019, pp. 7059–7069.
4. Conneau A., Khandelwal K., Goyal N. et al. Unsupervised Cross-lingual Representation Learning at Scale, *58th Annual Meeting of the Association for Computational Linguistics*, Online, 5–10 July 2020, proceedings. Stroudsburg, Association for Computational Linguistics, 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747
5. Bubeck S., Chandrasekaran V., Eldan R. et al. Sparks of Artificial General Intelligence: Early experiments with GPT-4. ArXiv preprint, 2023, Vol. 2303.12712.
6. Kojima T., Gu S., Reid M. et al. Large language models are zero-shot reasoners, *36th Annual Conference on Neural Information Processing Systems*, New Orleans, November 28 – December 9, 2022, proceedings. New York, Curran Associates Inc., 2022, Vol. 35, pp. 22199–22213.
7. Riloff E., Thelen M. A rule-based question answering system for reading comprehension tests, *2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding systems*, Seattle, 4 May 2000, proceedings. Stroudsburg, Association for Computational Linguistics, 2000, Vol. 6, pp. 13–19. DOI: 10.3115/1117595.1117598
8. Radev D., Fan W., Qi H. et al. Probabilistic question answering on the web, *11th international conference on World Wide Web*, Honolulu, 7-11 May 2002, proceedings. New York, Association for Computing Machinery, 2022, pp. 408–419. DOI: 10.1145/511446.511500
9. Radev D., Prager J., Samn V. Ranking suspected answers to natural language questions using predictive annotation, *6th conference on Applied natural language processing*, Seattle, 29 April 2000, proceedings. Stroudsburg, Association for Computational Linguistics, 2000, pp. 150–157. DOI: 10.3115/974147.974168
10. Ruder S., Peters M., Swayamdipta S. et al. Transfer Learning in Natural Language Processing, *17th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, 2–7 June 2019, proceedings. Stroudsburg, Association for Computational Linguistics, 2019, Tutorial Abstracts, pp. 15–18. DOI: 10.18653/v1/N19-5004
11. Howard J., Ruder S. Universal Language Model Fine-tuning for Text Classification, *56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, 15–20 July 2018, proceedings. Stroudsburg, Association for Computational Linguistics, 2018, Vol. 1, pp. 328–339. DOI: 10.18653/v1/P18-1031
12. Luo M., Hashimoto K., Yavuz S. et al. Choose Your QA Model Wisely: A Systematic Study of Generative and Extractive Readers for Question Answering, *Decoupling Logic from Knowledge: 1st Workshop on Semiparametric Methods in NLP*, Dublin, 27 May 2022, proceedings. Stroudsburg, Association for Computational Linguistics, 2022, pp. 7–22. DOI: 10.18653/v1/2022.spanlp-1.2
13. Guerreiro N., Voita E., Martins A. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation, *17th Conference of the European Chapter of the Association for Computational Linguistics*, Dubrovnik, 2–6 May 2023, proceedings. Stroudsburg, Association for Computational Linguistics, 2023, pp. 1059–1075. DOI: 10.18653/v1/2023.eacl-main.75
14. Zheng S., Huang J., Chang K. C. Why Does ChatGPT Fall Short in Providing Truthful Answers? ArXiv preprint, 2023, Vol. 2304.10513.
15. Anil R., Dai A. M., Firat O. et al. PaLM 2 Technical Report. ArXiv preprint, 2023, Vol. 2305.10403.
16. Touvron H., Martin L., Stone K. et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. ArXiv preprint, 2023, Vol. 2307.09288.
17. Ye S., Hwang H., Yang S. et al. In-Context Instruction Learning. ArXiv preprint, 2023, Vol. 2302.14691.
18. Wei J., Wang X., Schuurmans D. et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models, *36th Annual Conference on Neural Information Processing Systems*, New Orleans, November 28 – December 9, 2022, proceedings. New York, Curran Associates Inc., 2022, Vol. 35, pp. 24824–24837.
19. Wei J., Bosma J., Zhao M. et al. Finetuned Language Models Are Zero-Shot Learners, *International Conference on Learning Representations*, Online, 25–29 April 2022, proceedings. ArXiv, 2022, Vol. 2109.01652.
20. Ruis L., Khan A., Biderman S. et al. Large language models are not zero-shot communicators, *37th Annual Conference on Neural Information Processing Systems*, New Orleans, 10–16 December 2023, proceedings. San Diego, NeurIPS, 2023.
21. Bang Y., Cahyawijaya S., Lee N. et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. ArXiv preprint, 2023, Vol. 2302.04023.
22. Brown T., Mann B., Ryder N. et al. Language models are few-shot learners, *34th Annual Conference on Neural Information Processing Systems*, Vancouver, 6–12 December, 2020, proceedings. New York, Curran Associates Inc., 2020, Vol. 33, pp. 1877–1901.
23. OpenAI. GPT-4 Technical Report. ArXiv preprint, 2023, Vol. 2303.08774.
24. Liang P., Bommasani R., Lee T. et al. Holistic Evaluation of Language Models, *Transactions on Machine Learning Research*, 2023. ArXiv, Vol. 2211.09110.
25. He P., Liu X., Gao J. et al. DeBERTa: Decoding-enhanced BERT with Disentangled Attention, *International Conference on Learning Representations*, Online, 3–7 May, 2021, proceedings. ArXiv, 2021, Vol. 2006.03654.
26. He P., Gao J., Chen W. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing, *International Conference on Learning Representations*, Online, 3–7 May, 2023, proceedings. ArXiv, 2023, Vol. 2111.09543.
27. Rajpurkar P., Zhang J., Lopyrev K. et al. SQuAD: 100,000+ Questions for Machine Comprehension of Text, *2016 Conference on Empirical Methods in Natural Language Processing*, Austin, 1–5 November 2016, proceedings. Stroudsburg, Association for Computational Linguistics, 2016, pp. 2383–2392. DOI: 10.18653/v1/D16-1264
28. Papineni K., Roukos S., Ward T. et al. Bleu: a Method for Automatic Evaluation of Machine Translation, *40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, 6–12 July 2002, proceedings. Stroudsburg, Association for Computational Linguistics, 2002, pp. 311–318. DOI: 10.3115/1073083.1073135
29. Lin C. ROUGE: A Package for Automatic Evaluation of Summaries, *Text Summarization Branches Out: ACL-04 Workshop*, Barcelona, 25–26 July 2004, proceedings. Stroudsburg, Association for Computational Linguistics, 2004, pp. 74–81.

DOI 10.15588/1607-3274-2024-1-14

30. Yadan O. Hydra – A framework for elegantly configuring complex applications [Electronic resource]. Access mode: https://github.com/facebookresearch/hydra.
31. Rajpurkar P., Jia R., Liang P. Know What You Don't Know: Unanswerable Questions for SQuAD, *56th Annual Meeting of the Association for Computational Linguistics*, Melbourne, 15–20 July 2018, proceedings. Stroudsburg, Association for Computational Linguistics, 2018, Vol. 2, pp. 784–789. DOI: 10.18653/v1/P18-2124
32. Ivanyuk-Skulskiy B., Zaliznyi A., Reshetar O. et al. ua_datasets: a collection of Ukrainian language datasets [Electronic resource]. Access mode: https://github.com/fido-ai/ua-datasets.
33. Krisilov V., Komleva N. Analysis and evaluation of competence of information sources in problems of intellectual data processing, *Problemele energeticii regionale*, 2019, Vol. 40, Issue 1, pp. 91–104. DOI: 10.5281/zenodo.3239185.
34. Ahuja K., Diddee H., Hada R. et al. MEGA: Multilingual Evaluation of Generative AI. ArXiv preprint, 2023, Vol. 2303.12528.

Received 18.12.2023.
Accepted 29.01.2024.

УДК 004.912

UA-LLM: ПОКРАЩЕННЯ ВІДПОВІДІ НА ЗАПИТАННЯ ЗА КОНТЕКСТОМ УКРАЇНСЬКОЮ МОВОЮ З ВИКОРИСТАННЯМ ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ

Сиром'ятніков М. В. – аспірант кафедри Інженерії програмного забезпечення Національного університету «Одеська Політехніка», Одеса, Україна.
Рувінська В. М. – канд. техн. наук, професор кафедри Інженерії програмного забезпечення Національного університету «Одеська Політехніка», Одеса, Україна.

АНОТАЦІЯ

Актуальність. Відповідь на запитання за контекстом, фундаментальне завдання обробки природної мови, вимагає глибокого розуміння мови. Будучи складною задачею, вона є невід'ємною частиною сучасних пошукових систем, інтелектуальних помічників, чат-ботів і всієї сфери розмовного штучного інтелекту. У той час як англійська, китайська та інші широко поширені мови налічують велику кількість наборів даних, алгоритмів і тестів, українська – з її багатою лінгвістичною спадщиною та складним синтаксисом залишається серед малоресурсних мов, що ще більше ускладнює задачу відповіді на запитання за контекстом.

Мета роботи. Мета роботи полягає у розробці та оцінюванні методів на базі великих мовних моделей, об'єднаних у фреймворк для вирішення проблеми низькоресурсності задачі відповіді на запитання за контекстом в українській мові.

Метод. Простий, але гнучкий фреймворк для використання великих мовних моделей, розроблений в рамках цієї дослідницької роботи, висвітлює два ключові методи для вирішення проблеми даних у задачі відповіді на запитання за контекстом, запропоновані та оцінені в цій статті. Перший метод використовує Zero-shot і Few-shot learning – дві основні гілки N-shot learning, де N відповідає кількості тренувальних прикладів, для побудови двомовної стратегії підказок на основі інструкцій для роботи з мовними моделями у екстрактивний спосіб (пошук сегменту відповіді у контексті) замість їхньої природної генеративної поведінки (генерація відповіді на основі контексту). Другий запропонований метод базується на першому, але замість простої відповіді на запитання мовна модель розмічає вхідний контекст шляхом генерації пар запитання-відповідь. Отримані синтетичні дані використовуються для тренування екстрактивної моделі. У цій статті розглядається як навчання на основі аугментації даних, коли вже є деякі розмічені дані, так і повністю синтетичне навчання, коли дані відсутні. Ключовою перевагою запропонованих методів є можливість отримати якість передбачень на рівні натренованих екстрактивних моделей навіть без дорогого та довготривалого процесу розмітки даних людьми.

Результати. Два запропонованих методи для розв'язання проблеми недостатньої кількості тренувальних даних у задачі відповіді на запитання за контекстом для української мови було реалізовано та об'єднано в гнучкий фреймворк для роботи з великими мовними моделями .

Висновки. Дана робота демонструє результати всеосяжного дослідження рівня розуміння мови моделями OpenAI GPT-3.5, OpenAI GPT-4, Cohere Command і Meta LLaMa-2 на прикладі вирішення задачі відповіді на запитання за контекстом для низькоресурсної української мови. Ретельна оцінка запропонованих методів за різноманітним набором показників доводить їх ефективність, розкриваючи можливість побудови компонентів пошукових систем, інтелектуальних чат-ботів та автономних систем відповіді на запитання з підтримкою української мови та близькою до нуля кількістю розмічених тренувальних даних. Перспектива подальших досліджень полягає у розширенні сфери застосування від завдання відповіді на запитання за контекстом, розглянутого у цій статті, до усіх основних задач розуміння природної мови з кінцевою метою встановлення повного тесту для оцінювання можливостей великих мовних моделей в українській мові.

КЛЮЧОВІ СЛОВА: велика мовна модель, відповідь на запитання, few-shot learning, генеративна розмітка даних.

ЛІТЕРАТУРА

1. Rajpurkar P. The Stanford Question Answering Leaderboard [Electronic resource] / P. Rajpurkar. – Access mode: https://rajpurkar.github.io/SQuAD-explorer/.
2. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / [J. Devlin, M. Chang, K. Lee et al.] // Human Language Technologies : 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, 2–7 June 2019 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2019. – Vol. 1. – P. 4171–4186. DOI: 10.18653/V1/N19-1423
3. Conneau A. Cross-lingual language model pretraining / A. Conneau, G. Lample // Neural Information Processing Systems : 33rd International Conference, Vancouver, 8–14 December 2019 : proceedings. – New York : Curran Associates Inc., 2019. – P. 7059–7069.
4. Unsupervised Cross-lingual Representation Learning at Scale / [A. Conneau, K. Khandelwal, N. Goyal et al.] // 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2020. – P. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747
5. Sparks of Artificial General Intelligence: Early experiments with GPT-4 / [S. Bubeck, V. Chandrasekaran, R. Eldan et al.] // ArXiv preprint. – 2023. – Vol. 2303.12712.
6. Large language models are zero-shot reasoners / [T. Kojima, S. Gu, M. Reid et al.] // Advances in Neural Information Processing Systems : 36th Annual Conference on Neural Informa-

DOI 10.15588/1607-3274-2024-1-14

tion Processing Systems, New Orleans, November 28 – December 9, 2022 : proceedings. – New York : Curran Associates Inc., 2022. – Vol. 35. – P. 22199–22213.
7. Riloff E. A rule-based question answering system for reading comprehension tests / E. Riloff, M. Thelen // Reading comprehension tests as evaluation for computer-based language understanding systems : 2000 ANLP/NAACL Workshop, Seattle, 4 May 2000 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2000. – Vol. 6. – P. 13–19. DOI: 10.3115/1117595.1117598
8. Probabilistic question answering on the web / [D. Radev, W. Fan, H. Qi et al.] // 11th international conference on World Wide Web, Honolulu, 7–11 May 2002 : proceedings. – New York : Association for Computing Machinery, 2022. – P. 408–419. DOI: 10.1145/511446.511500
9. Radev D. Ranking suspected answers to natural language questions using predictive annotation / D. Radev, J. Prager, V. Samn // 6th conference on Applied natural language processing, Seattle, 29 April 2000 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2000. – P. 150–157. DOI: 10.3115/974147.974168
10. Transfer Learning in Natural Language Processing / [S. Ruder, M. Peters, S. Swayamdipta et al.] // Human Language Technologies : 17th Annual Conference of the North American Chapter of the Association for Computational Linguistics, Minneapolis, 2–7 June 2019 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2019. – Tutorial Abstracts. – P. 15–18. DOI: 10.18653/v1/N19-5004
11. Howard J. Universal Language Model Fine-tuning for Text Classification / J. Howard, S. Ruder // 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 15–20 July 2018 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2018. – Vol. 1. – P. 328–339. DOI: 10.18653/v1/P18-1031
12. Choose Your QA Model Wisely: A Systematic Study of Generative and Extractive Readers for Question Answering / [M. Luo, K. Hashimoto, S. Yavuz et al.] // Decoupling Logic from Knowledge : 1st Workshop on Semiparametric Methods in NLP, Dublin, 27 May 2022 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2022. – P. 7–22. DOI: 10.18653/v1/2022.spanlp-1.2
13. Guerreiro N. Looking for a Needle in a Haystack: A Comprehensive Study of Hallucinations in Neural Machine Translation / N. Guerreiro, E. Voita, A. Martins // 17th Conference of the European Chapter of the Association for Computational Linguistics, Dubrovnik, 2–6 May 2023 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2023. – P. 1059–1075. DOI: 10.18653/v1/2023.eacl-main.75
14. Zheng S. Why Does ChatGPT Fall Short in Providing Truthful Answers? / S. Zheng, J. Huang, K. C. Chang // ArXiv preprint. – 2023. – Vol. 2304.10513.
15. PaLM 2 Technical Report / [R. Anil, A. M. Dai, O. Firat et al.] // ArXiv preprint. – 2023. – Vol. 2305.10403.
16. Llama 2: Open Foundation and Fine-Tuned Chat Models / [H. Touvron, L. Martin, K. Stone et al.] // ArXiv preprint. – 2023. – Vol. 2307.09288.
17. In-Context Instruction Learning / [S. Ye, H. Hwang, S. Yang et al.] // ArXiv preprint. – 2023. – Vol. 2302.14691.
18. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models / [J. Wei, X. Wang, D. Schuurmans et al.] // Advances in Neural Information Processing Systems : 36th Annual Conference on Neural Information Processing Systems, New Orleans, November 28 – December 9, 2022 : proceedings. – New York : Curran Associates Inc., 2022. – Vol. 35. – P. 24824–24837.
19. Finetuned Language Models Are Zero-Shot Learners / [J. Wei, J. Bosma, M. Zhao et al.] // International Conference on Learning Representations, Online, 25–29 April 2022 : proceedings. – ArXiv, 2022. – Vol. 2109.01652.
20. Large language models are not zero-shot communicators / [L. Ruis, A. Khan, S. Biderman et al.] // Neural Information Processing Systems : 37th Annual Conference, New Orleans, 10–16 December 2023 : proceedings. – San Diego: NeurIPS, 2023.
21. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity / [Y. Bang, S. Cahyawijaya, N. Lee et al.] // ArXiv preprint. – 2023. – Vol. 2302.04023.
22. Language models are few-shot learners / [T. Brown, B. Mann, N. Ryder et al.] // Neural Information Processing Systems : 34th International Conference, Vancouver, 6–12 December, 2020 : proceedings. – New York : Curran Associates Inc., 2020. – Vol. 33. – P. 1877–1901.
23. OpenAI. GPT-4 Technical Report / OpenAI // ArXiv preprint. – 2023. – Vol. 2303.08774.
24. Holistic Evaluation of Language Models / [P. Liang, R. Bommasani, Lee. T et al.] // Transactions on Machine Learning Research. – 2023. ArXiv. – Vol. 2211.09110.
25. DeBERTa: Decoding-enhanced BERT with Disentangled Attention / [P. He, X. Liu, J. Gao et al.] // International Conference on Learning Representations, Online, 3–7 May, 2021 : proceedings. – ArXiv, 2021. – Vol. 2006.03654.
26. He P. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing / P. He, J. Gao, W. Chen // International Conference on Learning Representations, Online, 3–7 May, 2023 : proceedings. – ArXiv, 2023. – Vol. 2111.09543.
27. SQuAD: 100,000+ Questions for Machine Comprehension of Text / [P. Rajpurkar, J. Zhang, K. Lopyrev et al.] // 2016 Conference on Empirical Methods in Natural Language Processing, Austin, 1–5 November 2016 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2016. – P. 2383–2392. DOI: 10.18653/v1/D16-1264
28. Bleu: a Method for Automatic Evaluation of Machine Translation / [K. Papineni, S. Roukos, T. Ward et al.] // 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, 6–12 July 2002 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2002. – P. 311–318. DOI: 10.3115/1073083.1073135
29. Lin C. ROUGE: A Package for Automatic Evaluation of Summaries / C. Lin // Text Summarization Branches Out: ACL-04 Workshop, Barcelona, 25–26 July 2004 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2004. – P. 74–81.
30. Yadan O. Hydra – A framework for elegantly configuring complex applications [Electronic resource] / O. Yadan. – Access mode: https://github.com/facebookresearch/hydra.
31. Rajpurkar P. Know What You Don't Know: Unanswerable Questions for SQuAD / P. Rajpurkar, R. Jia, P. Liang // 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, 15–20 July 2018 : proceedings. – Stroudsburg : Association for Computational Linguistics, 2018. – Vol. 2. – P. 784–789. DOI: 10.18653/v1/P18-2124
32. Ivanyuk-Skulskiy B. ua_datasets: a collection of Ukrainian language datasets [Electronic resource] / [B. Ivanyuk-Skulskiy. A. Zaliznyi, O. Reshetar et al.]. – Access mode: https://github.com/fido-ai/ua-datasets.
33. Krisilov V. Analysis and evaluation of competence of information sources in problems of intellectual data processing / V. Krisilov, N. Komleva // Problemele energeticii regionale. – 2019. Vol. 40, Issue 1. – P. 91–104. DOI: 10.5281/zenodo.3239185.
34. MEGA: Multilingual Evaluation of Generative AI / [K. Ahuja, H. Diddee, R. Hada et al.] // ArXiv preprint. – 2023. – Vol. 2303.12528.

DOI 10.15588/1607-3274-2024-1-14

УДК 681.326

ІНТЕЛЕКТУАЛЬНИЙ КОМП'ЮТИНГ В ПАМ'ЯТІ

Хаханов В. І. − д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Абдуллаєв В. Х. – канд. техн. наук, доцент кафедри комп'ютерної інженерії, Азербайджанський державний університет нафти та промисловості, Азербайджан.

Чумаченко С. В. – д-р техн. наук, професор, завідувач кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Литвинова Є. І. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

Хаханова І. В. – д-р техн. наук, професор кафедри автоматизації проектування обчислювальної техніки, Харківський національний університет радіоелектроніки, Україна.

АНОТАЦІЯ

Актуальність. Оброблені великі дані мають соціальне значення для розвитку суспільства та промисловості. Інтелектуальна обробка великих даних є умовою створення колективного розуму соціальної групи, компанії, держави та планети в цілому. При цьому економіка великих даних (Data Economy) виходить на перше місце в оцінці механізмів обробки, оскільки дуже важливими є два параметри: швидкодія обробки даних та енерговитрати. Тому механізми, орієнтовані на паралельну обробку великих даних усередині центру зберігання даних, будуть завжди затребувані на IT-ринку.

Мета. Мета дослідження – підвищення економіки великих даних (Data Economy) завдяки аналізу даних як адрес таблиці істинності для ідентифікації патернів виробничих функціональностей на основі метрики подібності-відмінності.

Метод. Пропонуються архітектури Intelligent computing для управління кіберсоціальними процесами на основі моніторингу та аналізу великих даних. Пропонується обробка великих даних, як адрес таблиці істинності, для вирішення завдань ідентифікації, кластеризації, класифікації патернів соціальних та виробничих процесів. Пропонується сімейство автоматів для аналізу великих даних, як адрес. Розглядається таблиця істинності як розумна форма явних структур даних, що мають корисну константу – стандартний порядок прямування адрес. Мета обробки великих даних – зробити їх структурованими за допомогою таблиці істинності для подальшої ідентифікації до ухвалення актюаторних рішень. Таблиця істинності розглядається як механізм паралельної структуризації та пакування великих даних у її стовпці для визначення їх подібності-відмінності та еквівалентування даних за однаковими адресами. Подання даних, як адрес, пов'язане з унітарним кодуванням патернів двійковими векторами на знайденому універсумі примітивних даних. Механізм орієнтований на безпроцесорну обробку даних на основі read-write транзакцій за технологією in-memory комп'ютингу з суттєвою економією часу та енергії. Метрика обробки великих даних на таблиці істинності – це паралелізм, технологічна простота та лінійна обчислювальна складність. Платою за такі переваги є експоненційні витрати пам'яті зберігання явних структурованих даних.

Результати. Запропоновано паралельні алгоритми in-memory комп'ютингу для економічних механізмів перетворення великих неструктурованих даних, як адрес, корисні структуровані дані. Запропоновано архітектуру in-memory computing із глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес. Вона включає структуру матричного аналізу великих даних для визначення подібності між векторами, які надходять на входи матричного секвенсора. Векторний аналіз даних перетворюється на матричний комп'ютинг для обробки великих даних. Швидкодія паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів. Розроблено метод ідентифікації патернів ключовими словами. Він характеризується використанням унітарно-кодованих компонент даних для синтезу таблиці істинності бізнес-процесу. Це дозволяє застосовувати read-write транзакції для паралельної обробки великих даних, як адрес.

Висновки. Наукова новизна полягає у розробці наступних інноваційних рішень: 1) запропоновано нову векторно-матричну технологію паралельної обробки великих даних, як адрес, що характеризується використанням read-write транзакцій на матричній пам'яті без використання процесорної логіки; 2) запропоновано архітектуру in-memory computing з глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес; 3) запропоновано метод ідентифікації патернів ключовими словами, який характеризується використанням унітарно-кодованих компонентів даних для синтезу таблиці істинності бізнес-процесу, що дає можливість використовувати транзакцію read-write для паралельної обробки великих даних, як адрес. Практична значимість дослідження полягає в тому, що будь-яке завдання штучного інтелекту (подібність-відмінність, класифікація-кластеризація та розпізнавання, ідентифікація образів) можна технологічно просто та ефективно вирішувати за допомогою таблиці істинності (або її похідних) та унітарно кодованих великих даних. Перспективи дослідження пов'язані з імплементацією цієї технології моделювання цифрових пристроїв на ринку EDA.

КЛЮЧОВІ СЛОВА: Intelligent Computing, Cloud, fog, and edge computing, Big data computing, In-memory computing, Cyber social compiting, Hadoop Map-Reduce техніка, великі дані, як адреси, таблиця істинності, логічний вектор, подібність-відмінності, еквівалентність даних, універсум примітиви, патерни як двійковий вектор.

АБРЕВІАТУРИ

EDA – Electronic Design Automation;
VVV – volume, velocity, variety;
AI – Artificial Intelligence;
NN – Nueral Network;
QC – Quantum Computing;
OCR (optical character recognition) – оптичне розпізнавання символів;

DOI 10.15588/1607-3274-2024-1-15

GMM – модель гаусової суміші;
LR – логістична регресія;
RFC – класифікатор довільного лісу;
TF-IDF (TF – term frequency, IDF – inverse document frequency) – статистична міра, яка використовується для оцінки важливості слова в контексті документа;
HDFS – розподілена файлова система Hadoop;
MDV – матриця дедуктивних векторів;
MIV – матриця вхідних векторів;
MOV – матриця формування стовпців;
HPC (High-performance computing) – система високопродуктивних обчислень;
TI – таблиця істинності.

НОМЕНКЛАТУРА

D – дедуктивна матриця;
*D*1, *D*2 – відмінності щодо кожного входу секвенсора;
*S*12 – подібність між входами;
T – матриця вхідних даних (Data as Address),
F – матриця функціональності (Deductive Matrix L),
M – матриця-результат аналізу (Result Matrix);
U – універсум примітивів;
Uj – рядок універсуму примітивів;
Mj – стовпчик матриці аналізу примітивів;
F – несправність,
T – тест;
L – логічний вектор;
Z – якості розпізнавання кейсів;
Qi – інтегральна оцінка якості розпізнавання кейсів
Y – середня оцінка якості розпізнавання кейсів;
Li – ідеальний кейс (патерн);
Tj – актуальний кейс (патерн);
n – кількість входів схеми або функціональності;
*b*1, *b*2 – компоненти паттерна;
X – кількість входів двигуна;
Xi – екран;
Yi – кількість виходів двигуна;
k – потужність універсуму;
P – число патернів чи екранів;
Q =*kP*/*x* – швидкодія векторної технології обробки великих даних, як адрес;
N=*P*/*x* – апаратурні витрати (кількість восьмивходових елементів) на реалізацію каскадного секвенсора.

ВСТУП

Цікаві науково-практичні результати завжди виходять на стику наук шляхом інтеграції сучасних моделей, методів, алгоритмів та технологій. Так створюються нові наукові напрями, які вирішують практичні завдання для IT ринку. Очікуване об'єднання трьох просторів – соціального, фізичного, інформаційного – ініціювало створення нової парадигми Intelligent Computing (Інтелектуальний комп'ютинг) [1] для вирішення завдань моніторингу та управління кібер-соціально-фізичними процесами та створення мозку людства.

Intelligent Computing – галузь знань, що займається теорією, практикою та економікою детермінованого та ймовірнісного (AI) комп'ютингу за метрикою час-ресурси-якість для управління процесами та явищами на основі моніторингу оцифрованих та розумно-пов'язаних між собою соціального, фізичного та інформаційного простору. Intelligent Computing [1] – нова парадигма обчислювальних архітектур, яка з'єднує людину з комп'ютером, традиційні обчислення з перцептивним, когнітивним та автономним інтелектом та сприяє цифровій революції в епоху великих даних, штучного інтелекту та Інтернету речей за допомогою нових обчислювальних теорій, архітектур, методів, систем та додатків. Intelligent Computing [1] включає в тому числі: Cloud, fog, and edge computing, Big data computing, In-memory computing, Design and test computing [24], Cyber social computing, Generic intelligence, Data intelligence, Analog computing, Graph computing , Artificial neural network, Fuzzy systems, Evolutionary computation, Perceptual intelligence, Cognitive intelligence, Natural language processing, Causal inference, Autonomous intelligence, Brain-computer Interface, High-performance computing, Quantum computing, Photonic computing, Biocomputing, Biocomputing computing, Intelligent computing for society, economy and governance.

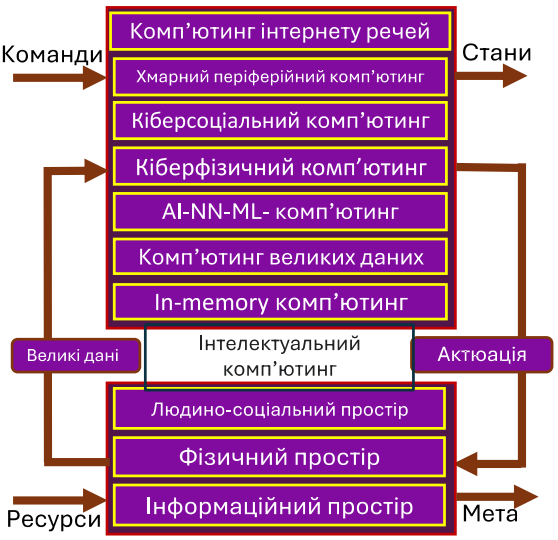


Рисунок 1 – Інтелектуальний комп'ютинг для трьох просторів

На рис. 2 позначено: Enterprise Systems – корпоративні системи; Smart IoT device – розумний пристрій IoT; Gateway – шлюз; CLOUD – ХМАРА: High computational resources – високо обчислювальні ресурси; Long-term storage – Тривале зберігання Global predictive models – глобальні прогнозні моделі; Global network management – управління глобальною мережею; FOG – ТУМАН: Distributed storage –

DOI 10.15588/1607-3274-2024-1-15

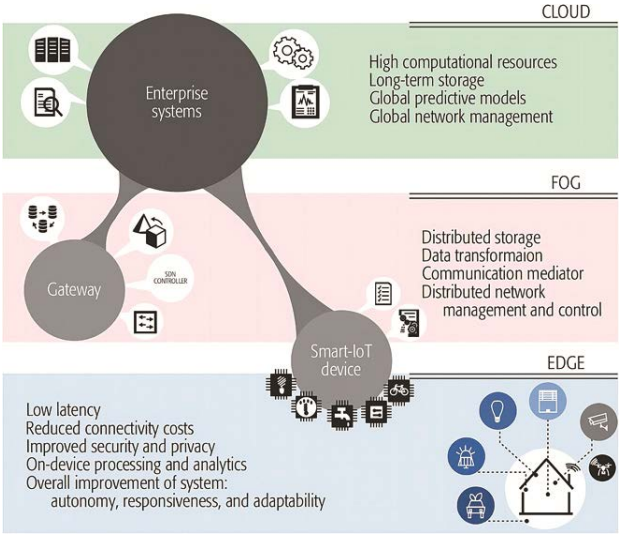


Рисунок 2 – Cloud, fog, and edge computing глобальна архітектура Intelligent Computing [1]

розподілене сховище; Data transformation – перетворення даних; Communiccation mediator – комунікаційний посередник; Distributed network management and control – управління та контроль розподіленої мережі; EDGE – КРАЙ: Low latency – низька затримка; Reduced connectivity costs – зменшені витрати на підключення; Improved security and privacy – покращена безпека та конфіденційність; On-device processing and analitycs – обробка та аналітика на пристрої; Overall improvement of system (autonomy, responsiveness, and adaptability) – загальне покращення системи (автономність, відклик, адаптивність).

Big Data – великі дані – кількість та форма інформації, що важко сприймаються свідомістю людини. Метрика великих даних VVV: volume, velocity, variety (обсяг, швидкість генерування та різноманітність). Data Science – наука, що вивчає життєвий цикл та форми даних з метою отримання актуальної інформації для прийняття рішення. Використовуються підходи: дискретна математика та статистика, штучний інтелект та хмарні обчислення для аналізу великих обсягів даних під час вирішення завдань класифікації, регресії, кластеризації. Ключові інструменти: R, Python, Apache Hadoop, MapReduce, Apache Spark, NoSQL Databases, Cloud computing, GitHub.

Machine learning – алгоритми пошуку закономірностей у вхідних даних без програмування на основі розумних механізмів їх структуризації з метою розпізнавання патернів та прийняття рішень. «Ніхто машину не навчає». Висловлювання типу «я навчаю комп'ютер» або «ми навчили нейромережу», звучать щонайменше наївно, на думку авторитетів у цій галузі, Daniel Faggella [3]. Відношення ієрархічного порядку між механізмами AI-computing: Data Science ⊢Machine Learning⊢ (AI, NN, QC) ⊢Smart Data structure ⊢Big Data. Структурна формула дослідження поєднує такі компоненти Intelligent Computing: Cloud-edge computing, Machine learning, Design and test computing [2], Cyber social computing, In-memory computing, Big data computing. Мета – зробити стійкими та моральними всі процеси та явища у кіберсоціальному просторі (рис. 3).

Архітектурно дослідження подано моделлю фон Неймана для хмарно-термінальної взаємодії механізмів федеративно-прискореного машинного навчання, що дозволяє керувати процесами на основі моніторингу даних, датчиків, людей та просторів. Щоб заощадити на часі і на витратах енергії, було запропоновано локально не використовувати архітектуру фон Неймана, а перейти на енергозберігаючу обробку великих даних у пам'яті, де вони зберігаються, використовуючи read-write транзакції, без потужної системи команд універсального процесора, які замінюють логічні вектори пам'яті.

Розумні (пов'язані) структури даних дозволяють без програмування вирішувати завдання структуризації корисної інформації шляхом суперпозиції таблиць та матриць, векторів. Ця ідея є основною в механізмах технічної діагностики, машинного навчання, включаючи нейронні мережі як тип розумних структур даних.

Добуток складності структур даних на обчислювальну складність алгоритму є постійною величиною (рис. 4).

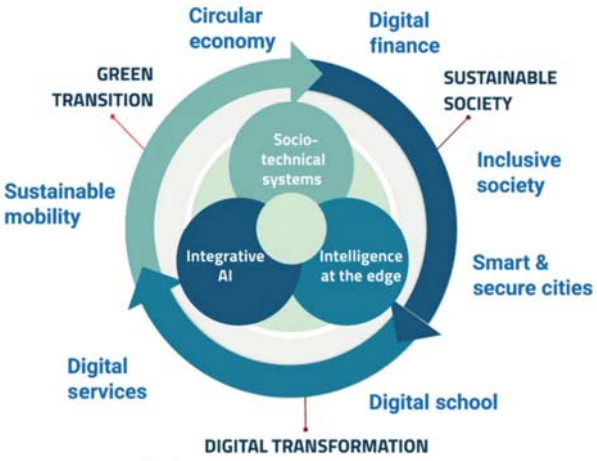


Рисунок 3 – Компоненти цифрового суспільства [1]: Socio-technical systems – соціо-технічна система; Intelligence at the edge – Інтелект на межі; Integrative AI – Інтегративний ШІ; Green Transition – зелений перехід; Sustainable Society – стале суспільство; Digital Transformation – цифрова трансформація; Digital Finance – цифрові фінанси; Inclusive socity – Інклюзивне суспільство; Smart & secure cities – Розумне та захищене суспільство; Digital school – цифрова школа; Digital services – цифрові послуги; Sustainable mobility – стійка мобільність; Circular economy – циркулярна економіка

DOI 10.15588/1607-3274-2024-1-15

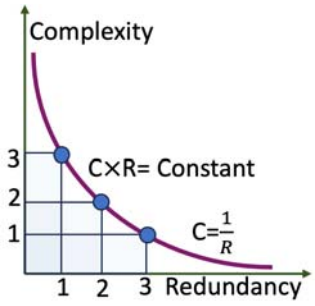


Рисунок 4 – Відношення між надмірністю структур даних та складністю алгоритму їх обробки: Complexity (C) – складність; Constant – константа; Redundancy (R) – надмірність

Тому платою за найпростіші алгоритми є експоненційні витрати пам'яті. Такі витрати будуть виправдані, якщо структура даних містить у явному вигляді рішення комбінаторних завдань. Найбільш універсальною моделлю комп'ютингу на вирішення всіх комбінаторних завдань є таблиця істинності, якій понад сто років. Її явні структури даних адекватно описують такі цікаві об'єкти: логічну функціональність, графову структуру, діаграму Хасе, бінарні діаграми, двійкове дерево рішень. Таблиця істинності перетворює експоненційну складність алгоритму розв'язання комбінаторної задачі на лінійну. Крім того, таблиця істинності моделює життєвий цикл великих даних, які подорожують таблицею істинності від его (відмінності) до самості (подібності). Питання лише у тому, як підготувати великі дані їхньої обробки на таблиці істинності, як адрес.

Чому цього ніхто не робив раніше? 1. Таблиця істинності (ТІ) лякає дослідників своєю експоненційною розмірністю. 2. Мало хто звертав увагу на комбінаторику явних адрес ТІ для вирішення перебірних завдань. 3. Найголовніше – це представляти великі дані, як адреси ТІ, – простий хід у виконанні, але виявляється не дуже простим у розумінні. 4. Властивість адресації, закладена у ТІ, вважають дослідники, закриває ворота для паралельної обробки інформації, хоча це зовсім не так. 5. Важко користуватися ТІ, наприклад, на 20 змінних. 6. Мало хто асоціює логічний вектор з компактним записом ТІ, на якій вирішуються практично всі завдання синтезу та аналізу оцифрованих процесів та явищ, включаючи кіберсоціальні та кіберфізичні. 7. Важко було припустити ТІ як структуру, якою подорожують великі дані від різниці до подібності, від его до самості по Карлу Янгу і навпаки, ідентифікуючи життєвий цикл даних. 8. Незрозуміло, чому не використовувалися адреси ТІ для ідентифікації та моделювання комбінацій несправностей та помилок у кіберфізичних та кіберсоціальних системах.

Вже утворився клуб вчених та підприємців (14 осіб) у галузі ІТ-індустрії, які мають нагороди Global-IT, що є аналогом Нобелівської премії. Практично всі вони у своїй інавгураційній промові заявляли, що майбутнє планети за комп'ютерними рішеннями в галузі моніторингу та управління суспільством. На цьому шляху людство має пройти такі кроки: 1) створення мозку людства до 2050; 2) безпілотне вирішення проблеми транспорту до 2030 року; 3) генерація запчастин людського організму до 2100; 4) вирішення проблеми харчування до 2040 року; 5) Intelligent Computing загального моніторингу та морального управління суспільством до 2080 року. Тут має бути добра воля політиків та керівників усіх країн та провідних компаній планети. Перший крок уже зроблено – IEEE-товариство легалізувало використання систем штучного інтелекту для рутинної роботи оформлення наукових ідей, досліджень та статей. Роботи ведуться зі створення стандарту ідентифікації оригінальних ідей у текстах та визначення валідності рівнів плагіаризму. Розумні контракти-програми, які пропонувалися в блокчейн комп'ютингу для управління соціальними групами, будуть замінюватись розумними структурами даних без програмування, які формують ML-механізми для моніторингу та управління соціальними групами, університетами, компаніями та державними структурами.

Об'єкт дослідження – in-memory intelligente комп'ютинг, який знижує енергетичні та годинні витрати під час обробки великих даних.

Предмет дослідження – in-memory аналіз елементів або цифрових схем будь-якої розмірності за допомогою read-write транзакцій на логічних векторах.

Мета дослідження – підвищення економіки великих даних (Data Economy) завдяки аналізу даних, як адрес таблиці істинності, для ідентифікації патернів виробничих функціональностей на основі метрики подібності-відмінності.

1 ПОСТАНОВКА ЗАДАЧІ

Розглядається задача побудови логічного вектора бізнес-функціональності для моніторингу та управління бізнес-процесом. Модель синтезу логічного вектора L подана адресним відношенням між екранами T і таблицею істинності F: $L=F(T)$. Розв'язок задачі полягає в обробці великих даних, що знаходяться на екранах, як адресів таблиці істинності, що формує відповідну кількість ключових слів, з яких складається логічний вектор або ідентифікатор групи екранів для формування шаблону або логічної функціональності. Для представлення ключових слів адресами таблиці істинності необхідно їх унітарно закодувати на множині дозволених символів, які формують універсум. Задача побудови логічного вектора на таблиці істинності має лінійну складність, завдяки експоненціальному характеру надлишковості структури даних таблиці істинності. Формальне рівняння синтезу логічного вектора функціональності бізнес-процесу подано співвідношенням наступних параметрів: $L=F(X)$, L – логічний вектор

DOI 10.15588/1607-3274-2024-1-15

функціональності, F – таблиця істинності, *X* – кінцеве число змінних-екранів, на яких задана таблиця істинності та логічний вектор.

Нехай задано потік екранів *X*={*X1, X2,…, Xn*}, які мають OCR-дані, представлені двома множинами: 1) headers – заголовки додатків та активних полів екрану; 2) keydata – дані, що заносяться в активні поля екрану. Практичне завдання побудови логічної моделі бізнес-процесу полягає у формуванні з ключових даних, що належать множині {headers, keydata}, ідентифікаторів екрану. Сукупність патернів формує логічну функцію бізнес-процесу. При цьому патерн – будь-яка просторово-часова послідовність елементів, що відтворюється, об'єднана смисловою метрикою. Патерн або кейс є реалізація бізнес-функціональності, яка залишає слід у вигляді даних. Ідентифікація патерну виходить шляхом стиснення великих даних у стовпцях таблиці істинності.

Ідея дослідження проста: для перетворення експоненційного алгоритму на лінійний, потрібно використовувати експоненційні структури явних даних, ідеальною формою яких є ТІ. Для обробки великих даних, як адрес ТІ, необхідно виконати унітарне кодування даних на універсумі примітивів. Отриману матрицю двійкових векторів перетворювати на адреси у вікні спостереження, що дорівнює кількості змінних у ТІ, та запакувати на ТІ. Комбінаторні властивості структури ТІ формують по стовпцях еквівалентні множини даних при лінійних витратах часу. На ТІ вирішуються такі завдання аналізу великих даних: 1) еквівалентування даних по стовпцям адресам ТІ; 2) визначення подібності-відмінності за даними у патернах; 3) пошук та ідентифікація патернів у великих даних; 4) пошук даних за заданими шаблонами; 5) побудова логічної моделі процесу чи явища за допомогою унітарного кодування патернів двійковими векторами на універсумі примітивів; 6) валідація кіберсоціального процесу та пошук актюаторних станів та помилок у ньому; 7) побудова тесту моделі кіберсоціального процесу.

Поетапно схему синтезу ком'ютингу можна подати такими задачами (рис. 5): 1) використання властивостей ТІ для ідентифікації екранів, еквівалентування ключових даних; 2) знаходження патернів виробничих функціональностей на основі подібності-відмінності логічних векторів, що формують фрагменти кейсу; 3) кластеризація бізнес-потоку великих даних щодо виробничих функціональностей; 4) ідентифікація виробничих функціональностей та кейсів у бізнес-потоці даних; 5) побудова автоматів пошуку подібності-відмінності даних у патернах бізнес-потоку; 6) синтез логічного вектора виробничого процесу; 7) синтез логічних векторів для моніторингу та управління виробничим процесом.

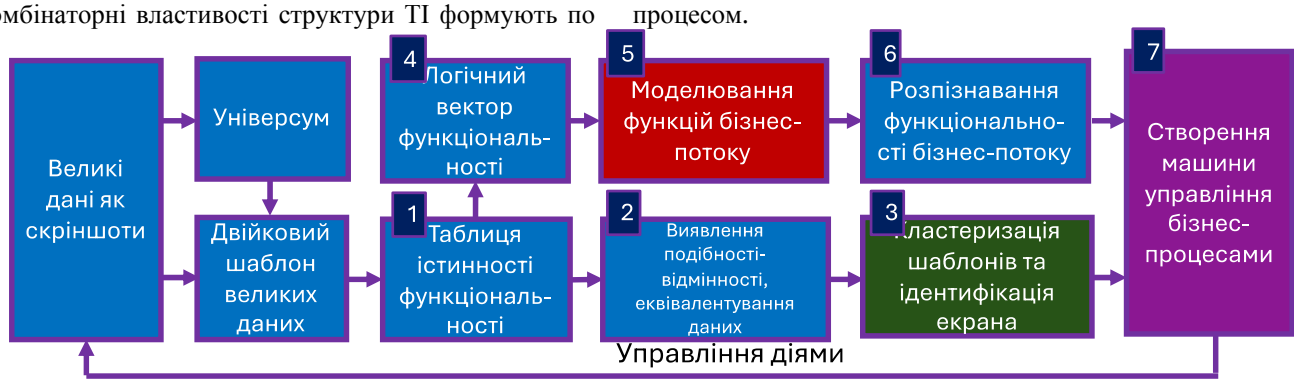


Рисунок 5 – Схема синтезу комп'ютингу бізнес-процесу

2 ОГЛЯД ЛІТЕРАТУРИ

Цикл поліпшення структур даних, алгоритмів та механізмів становить інтерес з погляду економії ресурсів та часу на обробку великих даних. Підтвердженням сказаного може бути велика кількість публікацій у IEEE Xplore за останні 10 років, присвячених темі Data Economy. Спостерігається стійкий тренд зниження карт-вузлів в архітектурі Map Reduced під час обробки великих обсягів даних [4–8]. Великі дані прагнуть використання одного data-центру з великим обсягом пам'яті для зберігання інформації. Це насамперед необхідно для зниження витрат енергії і часу обробки даних рахунок зменшення транзакцій передачі між обчислювальними центрами.

Проведені експерименти показують, що скорочення карт має велику цінність при обробці величезних обсягів даних у дуже великих кластерах. Використовуються також цікаві доповнення до Map Reduced, такі як Наївний Байєс, J48 та Random Forest, K-Means модель гаусової суміші (GMM), логістична регресія (LR) та класифікатор випадкового лісу (RFC) для ефективного майнінгу всередині величезних обсягів даних, які також економлять енергію. Показано, що Spark-технологія у 16 разів швидше, ніж обчислення Hadoop Map-Reduce для обробки великих даних і більш ефективна з точки зору економії енергії. Розглядаються механізми [9] прискорення ітеративних алгоритмів машинного навчання, реалізовані в Hadoop Map-Reduce (Stock) та Apache Spark на основі графічного процесора. Тут все добре зі швидкодією, але погано з енерговитратами. Розглянуто три алгоритми класифікації [10]: класифікатор *k*-NN, класифікатор центроїдів та

DOI 10.15588/1607-3274-2024-1-15

наївний байєсівський класифікатор. Емпіричні результати показують, що класифікатор Centroid є найбільш точним у цьому випадку з точністю до 95% порівняно з *k*-NN, точність якого становить 92%, і наївним класифікатором байєсівським з точністю 91,5%. Пропонується динамічне налаштування слотів [11] у процесі обробки даних – Dynamic Hadoop SlotAllocation, що підвищує завантаження кожного обчислювального вузла. Пропонується метод [12] передачі величезної бази даних від місця зберігання до вузла обчислення та індексування для розподілу запитів, масштабованості та продуктивності в гетерогенних середовищах. Результати свідчать, що запропонована робота скорочує час обробки даних на 30%. Наведено систему [13] обробки даних для автоматичної класифікації на основі важливих медичних термінів з використанням TF-IDF та тематичного моделювання для прийняття валідних діагнозів. Розглядається використання додаткової обчислювальної потужності [14] для зниження комунікаційного навантаження у розподілених обчисленнях, що позитивно впливає на швидкодію та зниження витрат енергії. Більш конкретно, розглядається загальна структура розподілених обчислень, заснована на структурах, що часто використовуються, таких як MapReduce, де загальні обчислення розбиваються на обчислення набору функцій «Map» і «Reduce», розподілених за сукупністю обчислювальних вузлів. Пропонується стратегія віртуального перетасовування [15], що дозволяє забезпечити ефективне переміщення даних та скорочення кількості операцій введення-виводу для перетасовування MapReduce, тим самим знижуючи енергоспоживання та зберігаючи енергію. Віртуальне перетасовування реалізується за допомогою комбінації трьох методів, включаючи тирівневу таблицю сегментів, злиття, близьке до вимоги, а також динамічне та збалансоване злиття піддерев. Економія енергоспоживання програм MapReduce склала 12%. Традиційна система високопродуктивних обчислень (HPC) є прикладом системи з обмеженою пам’яттю. Пропонується нова система MapReduce (Mammoth) [16], метою якої є підвищення продуктивності за рахунок глобального управління пам’яттю, що підвищує швидкодію обробки даних на 40%. Розвиток цифрової індустріалізації [17–21], цифрової економіки великих даних на місцях, досить віддалених від центру, представляє велику телекомунікаційну проблему, яку потрібно вирішувати для того, щоб мати цифрову індустрію інваріантну геопозиції будь-якого населеного пункту. Підкреслюється важливість великих даних у соціальній економіці, розкривається значення цифрової економіки промисловості великих даних, і навіть аналізуються реформи, викликані великими даними та заходи, створені задля їх застосування [21].

Таким чином, розподілена комп’ютерна система складається з множини взаємопов’язаних вузлів. Вони можуть бути фізичними, віртуальними машинами чи контейнерами. Коли група вузлів надає клієнту послуги та програми, якби це була одна машина, її також називають кластером. Hadoop – це платформа для виконання завдань на кластерах комп’ютерів, яка забезпечує гарну якість базового обладнання та програмного забезпечення. Hadoop3 – це проект, написаний на Java з відкритим кодом Apache Software Foundation.

MapReduce – це парадигма програмування, що забезпечує масштабування сотень або тисяч серверів у кластері Hadoop. Файлова система Google використовує методи розподілу, обробки та агрегування великого обсягу даних для пошукової системи Google. Версія MapReduce з відкритим вихідним кодом пізніше була випущена в рамках проекту Apache Hadoop з орієнтацією на функціональне програмування, яке орієнтоване на знання hardware архітектури. Фази виконання завдання MapReduce: поділ даних на кількох комп’ютерних вузлах, застосування функції карти до кожного фрагмента даних. Сортування та перемішування даних, та розподіл за редукторами. Скорочення даних із видачею результату. Чотири етапи: розподіл, зіставлення, сортування та перемішування, скорочення (split, map, sort & shuffle, reduce) (рис. 6, а). Архітектура Hadoop є пакетом файлової системи, механізму MapReduce і HDFS (розподіленої файлової системи Hadoop). Кластер Hadoop складається з одного головного та кількох підлеглих вузлів. Головний вузол включає Job Tracker, Task Tracker, NameNode і DataNode, тоді як підлеглий вузол включає DataNode і TaskTracker. Проблемне місце тут – блок сортування та перемішування даних, який у кожній компанії є секретним ключем, який формує успіх компанії та позитивну економіку обробки великих даних. MapReduce – це макро-виконання механізму глибоких нейронних мереж, де від шару до шару працюють процедури сортування та перемішування даних.

У метриці Hadoop Map-Reduce техніка та архітектура аналізу даних за допомогою ТІ представлена на рис. 6,б. Переваги запропонованого розв’язку: 1) відсутні канали зв’язку між пам’яттю та процесором між процесорами-вузлами кластера; 2) відсутній блок перемішування та переплутування інформації, який замінює розумна структура рядків-адрес таблиці істинності; 3) блок розподілу даних за картками-вузлами замінюється поділом унітарно кодованих даних за адресами, що набагато ефективніше за швидкодією та енерговитратами.

DOI 10.15588/1607-3274-2024-1-15

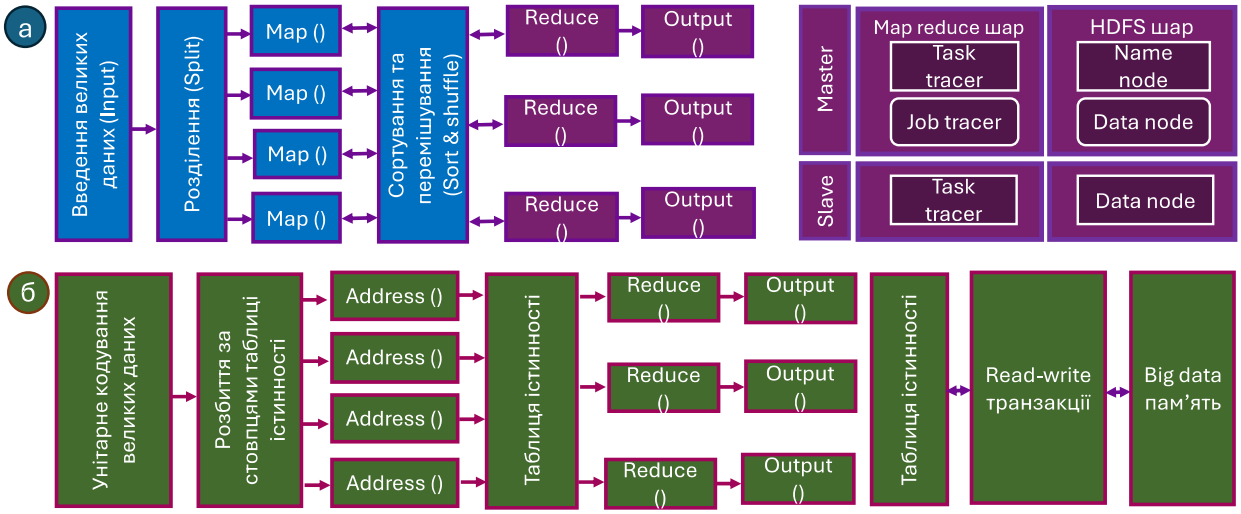


Рисунок 6 – Схема технік та архітектур для обробки великих даних Hadoop Map-Reduce

Оброблені великі дані мають соціальне значення для розвитку суспільства та промисловості. Інтелектуальна обробка великих даних є умовою створення колективного розуму соціальної групи, компанії, держави та планети в цілому. При цьому економіка великих даних (Data Economy) виходить на перше місце в оцінці механізмів обробки, оскільки дуже важливими є два параметри: швидкодія обробки даних та енерговитрати. Тому механізми, орієнтовані на паралельну обробку великих даних усередині центру зберігання даних, будуть завжди потрібні на IT-ринку. Великі дані – кількість інформації, що важко сприймається свідомістю людини. Метрика великих даних VVV: Volume, Velocity, Variety (обсяг, швидкість генерування та різноманітність). 1. Об'єм означає, що Big Data є великі масиви інформації. Сьогодні в кіберпросторі знаходиться 165 зета-байт інформації, з них використовується лише 4% структурованих даних. 2. Швидкість створення даних. Один автомобіль без драйвера генерує сьогодні 4 терабайти даних за день. 3. Різноманітність – це різні джерела інформації та різноманітність форматів (аудіо, відео, фото, тексти, символи) даних. Економіка великих даних – це довести 4 корисні відсотки до 20 з метою отримання штучного розуму або мозку людства з мінімальними витратами за часом та енергією. Далі розглядаються економічні механізми перетворення великих неструктурованих даних, як адрес, в корисні структуровані дані, за рахунок розробки паралельних алгоритмів in-memory комп'ютингу.

3 МАТЕРІАЛИ І МЕТОДИ

Розглянемо online класифікацію великих даних за метрикою подібності-відмінності патернів у бізнес-потоку екранів. Вихідні дані представлені потоком OCR-розпізнаних текстових фрагментів на сотнях зображень екранів комп'ютера. Математична модель представлена у вигляді дедуктивних векторів, які виконують роль фільтра для визначення подібності-відмінності екранів бізнес-потоку даних. При цьому дані на екранах подані двійковими векторами унітарно кодованих примітивів універсуму, які розглядаються як адреси для вилучення інформації з дедуктивного вектора.

Паралельний аналіз двійкової вхідної інформації на матриці трьох векторів (0100 1101 0001) для отримання відмінностей по кожному входу ($D1$, $D2$) і подібності між ними ($S12$) представлений на рис. 7. Адреси формуються комбінацією сигналів з першого та другого вхідного вектора output vectors, які складають стовпець-код для зчитування відповідного стовпця (00 01 10 11) векторної матриці, що формує паралельно стовпці output vectors. Природно, що логічне множення (&) трьох векторів ($D1\&D2\&S12$) = 000000000000 дає нуль вектор, та їх диз'юнкція у даному випадку дорівнює диз'юнкції (+) вхідних векторів: ($D1+D2+S12$)=1110011111110. Векторний аналіз даних перетворюється на матричний комп'ютинг для обробки великих даних. Тому нижня частина рис. 7, б присвячена структурі матричного аналізу великих даних для визначення подібності-відмінності між векторами, яка надходить на входи матричного секвенсора. Кількість входів такого секвенсора – число рядків у матриці D – завжди дорівнює кількості потоків даних, які необхідно обробляти паралельно. Кожен рядок матриці L своїми одиничними значеннями формує деяку функціональність (схожість, відмінність) обробки вхідних даних, як адрес. Сукупність рядків дедуктивної матриці L формує метрику властивостей, кількість яких дорівнює виходам секвенсора (матриці M). Інакше, кількість векторів матриць L і M дорівнює між собою.

Формально можна розглядати запропонований комп'ютинг як xor-відношення трьох матриць, що дорівнює нулю, де T – вхідні дані (Data as Address), F – функціональність (Deductive Matrix L), M – результат аналізу (Result Matrix).

DOI 10.15588/1607-3274-2024-1-15

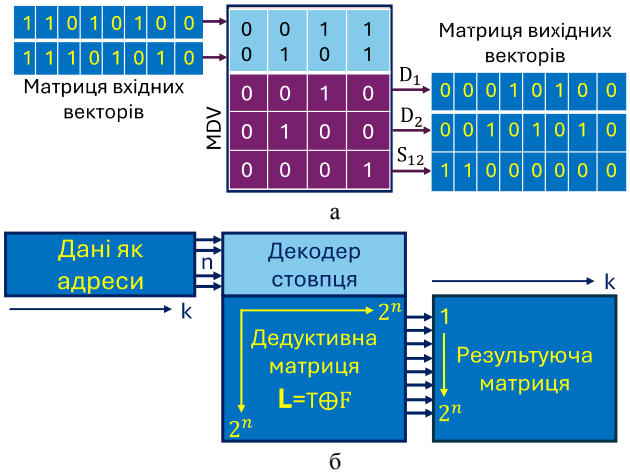


Рисунок 7 – Матричний паралельний аналіз великих даних, як адрес

Швидкодія паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів. Універсум (словник) – впорядкована множина (вектор) неповторних даних на вибраній кількості екранів. Універсум знаходиться шляхом обчислення множини всіх примітивних символів (слів) на основі оператора збору оригінальних текстів всіх аналізованих екранів (рис. 8).

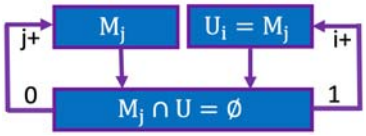


Рисунок 8 – Алгоритм отримання універсуму примітивів

У процесі створення універсуму як метрики бізнес потоку виконується кодування кожного символу десятковим номером (рис. 9). Після цього на отриманій метриці виконується унітарне кодування символів кожного екрана.

1	2	3	4	5	6	7	8	9	10	11	12
ма	he	ye	in	on	no	do	go	di	ha	qu	me
1	1	0	0	0	1	0	0	0	1	0	0
1	1	1	0	0	0	1	1	1	0	1	0

Рисунок 9 – Вектор універсуму примітивів та унітарне кодування символів екрану

Потім кожна множина символів екрану кодується двійковим вектором, де одиницею відзначається позиція, що відповідає символу, що присутня на екрані, що кодується. Отримані вхідні дані, що становлять вхідну матрицю вхідних векторів (MIV), подаються на матрицю векторів дедуктивних MDV для подальшого аналізу, де стовпці вхідної матриці розглядаються як адреси для зчитування відповідних стовпців MDV для формування стовпців MOV. Якщо необхідно проаналізувати n=8 екранів одночасно, тоді синтезується наступна матриця для елемента на вісім входів і кожен дедуктивний вектор буде розмірності 2^n=256 біт.

Архітектура векторної технології обробки великих даних має структуру (рис. 10), яка виконує функції: 1) Preprocessing – підготовка текстів до моделювання шляхом пошуку універсуму текстових примітивів та унітарного кодування екранів двійковими векторами; 2) M-Computing – memory-based computing – моделювання двійкових векторів даних на пам'яті-матриці дедуктивних векторів, що виділяють подібність-відмінність; 3) Postprocessing – виконується перетворення двійкових вихідних векторів у текстові примітиви.

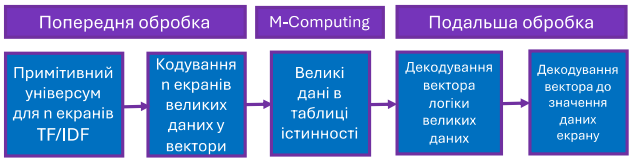


Рисунок 10 – Векторна технологія обробки великих даних

Розглянемо аналіз даних, як адрес ТІ, для ідентифікації екранів та патернів-кейсів бізнес-функціональностей на основі метрики даних, впорядкованих за їх значимістю. Завдання можна зобразити схемою векторно-логічного синтезу моделі бізнес-процесу (рис. 11): 1) використання властивостей ТІ для ідентифікації екранів, еквівалентування ключових даних-слів; 2) знаходження патернів-кейсів на основі подібності логічних векторів, що формують кейси; 3) розкладання бізнес-потоку великих даних щодо суттєвих функціональностей; 4) розпізнавання функціональностей та кейсів у бізнес-потоці даних та їх ідентифікація; 5) побудова online-автомата управління функціональностями бізнес-процесів; 6) щоб відновити функціональність потрібні два компоненти: питання-headers або вхідні умови, а також відповіді-дані або стан виходів, які слід автоматично сконтактенувати: $F=TxorL$ – питання-відповідь. Наявність вихідних даних не гарантує відновлення функціональності, для цього потрібно знати ще й вхідні дані, тобто питання, які задає система для отримання, наприклад, кредиту. Аналіз питань та відповідей в автоматичному режимі дає можливість отримати функціональність бізнес-процесу, представлену header-data ідентифікаторами екранів.

Для вимірювання відношень між процесами та явищами застосовується метрика Similarity-Difference. Найбільш адекватною і простою цифровою моделлю для опису відношень Similarity-Difference є ТІ. Вона поєднує властивості Similarity-Difference у повному комбінаторному виконанні для вирішення завдань класифікації, кластеризації та ідентифікації патернів (екранів). Вирішення завдання ідентифікації екранів пов'язане з виконанням наступних процедур: 1) для потоку екранів необхідно формувати універсум примітивів із ключових слів; 2) виконується кодування потоку екранів двійковими векторами у метриці універ-

DOI 10.15588/1607-3274-2024-1-15

суму ключових слів; 3) потік двійкових даних у форматі-вікні з 8-екранів-змінних стискається в ТІ по стовпцях, які є упорядкованими адресами ТІ. Вона розставляє ключові слова згідно з адресами на вісімці екранів. Кожен стовпець ТІ формує еквівалентну множину ключових слів.

Якість розпізнавання кейсів *Z*, інтегральна оцінка якості розпізнавання кейсів *Qi*; – середня оцінка якості розпізнавання кейсів *Y* визначаються відповідним чином через ідеальний кейс/патерн *Li*; актуальний кейс/патерн *Tj*; кількість входів схеми або функціональності *n* та компоненти патерна *a*1, *a*2, *b*1, *b*2.

Потрібно будувати оцінку покриття еталонних кейсів, або оцінку максимальних покриттів еталонних кейсів відрізками знайдених кейсів.

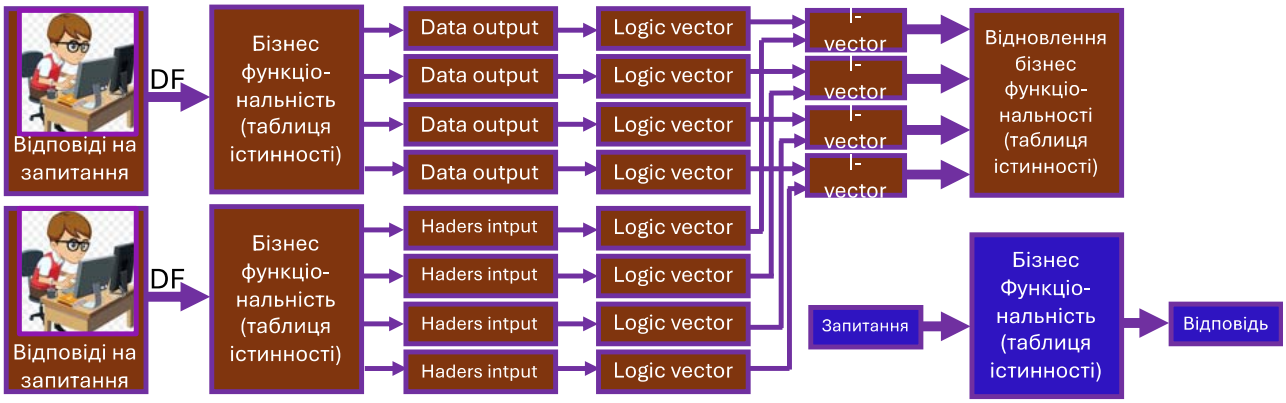


Рисунок 11 – Схема векторно-логічного синтезу моделі бізнес-процесу

4 ЕКСПЕРИМЕНТИ

Взаємодія двигуна з великими даними для отримання результату аналізу показано на рис. 12. Усі три компоненти знаходяться у пам'яті. При цьому двигун Deductive Matrix Engine (автомат з обробки великих даних із зворотними зв'язками) становить computational memory, а вхідні Data as address та вихідні Result Matrix дані – conventional memory. Змінні глобальних зворотних зв'язків враховують передісторію подібності-відмінності вже оброблених патернів визначення метричних властивостей поточних даних. На першому кроці алгоритму псевдозмінні набувають значення *L*=00…00 на метриці знайденого універсуму. Алгоритм закінчує свою роботу після того, як до двигуну будуть завантажені всі дані на шляху його руху зверху вниз. Слід зазначити, що матриця вхідних даних та результат аналізу повністю ідентичні за розмірами. В даному випадку, за розмірами збігається кількість входів *X* і виходів двигуна *Y*.

Продуктивність *Q* даного двигуна з обробки великих даних визначається добутком потужності універсуму та найближчого цілого, що не менше за відношення кількості патернів *P* до кількості активних входів *X* двигуна. Метрика або рівень паралелізму визначається кількістю актуальних входів *X*. Продуктивність обробки великих даних залежить від розмірів матриці вхідних даних, поділеної на кількість актуальних входів *X* Deductive Matrix Engine. Для підвищення швидкодії аналізу даних необхідно збільшувати число вхідних логічних змінних *X* дедуктивної матриці. У цьому слід пам'ятати, що функціональні можливості дедуктивної двійкової матриці обмежуються простором 2^nx2^n, *n* – число вхідних змінних, у якому вирішуються будь-які комбінаторні завдання з класифікації та ідентифікації будь-яких унітарно кодованих процесів і явищ, включаючи біометричні [23, 24, 26].

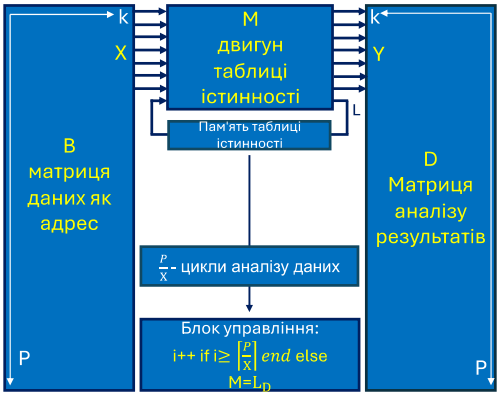


Рисунок 12 – Архітектура in-memory автомата та sliding алгоритм матричної обробки даних, де k – потужність універсуму, P – число патернів або екранів, що підлягають аналізу, X = 7 – кількість активних входів двигуна

Стовпці між собою створюють порядок Similarity-Difference (рис. 13), оцінки яких зменшуються від країв до центру і далі таблиці істинності; 4) тому для ідентифікації екранів необхідно завжди вибирати ключові слова, що відповідають крайнім правим та лівим стовпцям таблиці істинності, що мають максимальні значення difference та similarity. Similarity-стовпець формує клас, Difference-стовпець формують унікальний ідентифікатор екрану. Метрика TF/IDF підтверджує значущість 1-координат стовпців таблиці істинності для ідентифікації екранів за допомогою її комбінаторної структури метрики Similarity-Difference. Стовпці таблиці істинності мають однакову інтегральну оцінку TF/IDF. Це означає, що Difference-координати (*n*=8) у структурі таблиці істинності мають максимальну оцінку значимості будь-якої функції. Для верифікації логічної функціональності ці точки обов'язково мають бути покриті тестами (testbench).

DOI 10.15588/1607-3274-2024-1-15

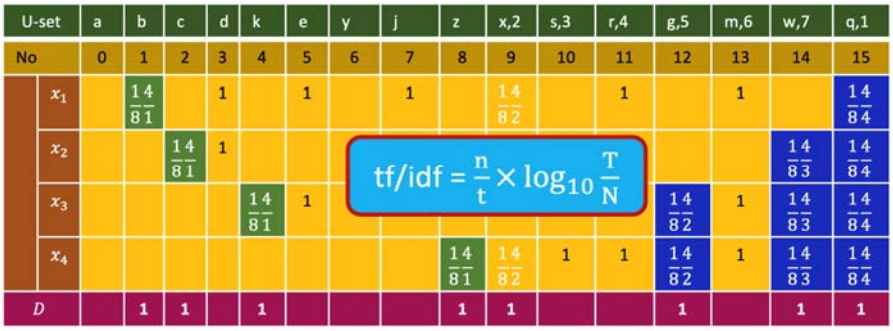


Рисунок 13 – Метрика TF/IDF координат таблиці істинності TF/IDF [28] = [(number of times term t appears in a document) / (total number of terms in the document)] * log10 (Total number of documents / Number of documents with term t in it)

Перевірка несправностей цих точок гарантує перевірку інших несправностей ліній цифрової структури на 90%. Ці точки вказані як критичні для дедуктивної матриці, побудованої для верифікації логічної функціональності. Критичні точки (стану) у будь-якій логічній функціональності кейсів збирають максимальну кількість ключових даних.

5 РЕЗУЛЬТАТИ

Таблиця істинності якісно та метрично визначає еквівалентну множину екранів на координатах стовпців [22, 27]. Логічний вектор визначає сигнатуру функціональності у кадрі. Щоб розрізняти еквівалентні екрани у кадрі, необхідно зробити наступне: 1) виключити екрани, що повторюються; 2) укласти у дужки еквівалентні ключові слова по стовпцях таблиці істинності; 3) similarity зменшується по стовпцях праворуч ліворуч, difference зростає по стовпцях зліва направо. Ідентифікатор екрана складається з крайніх ключових слів.

Локальні та інтегральні оцінки якості розпізнавання кейсів наведено на рис. 14. Потрібно будувати оцінку покриття еталонних кейсів відрізками знайдених кейсів, або оцінку максимальних покриттів еталонних кейсів відрізками знайдених кейсів.

Швидкодія паралельного алгоритму аналізу великих даних на матриці MDV дедуктивних векторів ставиться в лінійну залежність від числа бітів вхідних векторів або потужності універсуму примітивів.

Таким чином, запропоновано паралельні алгоритми in-memory комп'ютингу для економічних механізмів перетворення великих неструктурованих даних, як адрес, у корисні структуровані дані.

Запропоновано архітектуру in-memory computing із глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес. Вона включає структуру матричного аналізу великих даних для визначення подібності між векторами, які надходять на входи матричного секвенсора. Векторний аналіз даних перетворюється на матричний комп'ютинг для обробки великих даних.

Розроблено метод ідентифікації патернів ключовими словами. Він характеризується використанням унітарно-кодованих компонент даних для синтезу таблиці істинності бізнес-процесу. Це дозволяє використовувати read-write транзакції для паралельної обробки великих даних, як адрес.

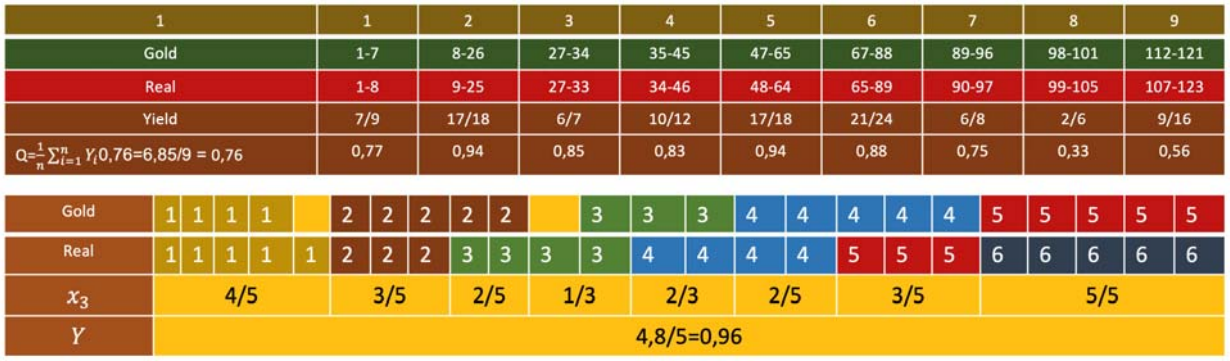


Рисунок 14 – Локальні та інтегральні оцінки якості розпізнавання

6 ОБГОВОРЕННЯ

Завдяки тому, що матриця векторного аналізу визначена в метричному булевому просторі відношень n змінних $2^n x 2^n$, така модель дозволяє вирішувати практично будь-яке комбінаторне завдання з ідентифікації, мінімізації, класифікації або кластеризації великих даних. Паралельність формування всіх рішень для заданих великих даних робить пропонований механізм на пам'яті привабливим для ринку електронних технологій.

Швидкодія векторної технології обробки великих даних, як адрес, за допомогою запропонованого сек-

DOI 10.15588/1607-3274-2024-1-15

венсора визначається потужністю універсуму примітивів k (сумарна розрядність вхідних векторів) або числа екранів P (загальна кількість входів секвенсора), що підлягають обробці: $Q=kP/x$. При цьому апаратурні витрати (кількість восьмивхідних елементів) на реалізацію каскадного секвенсора визначається числом екранів, що підлягають обробці: $N=P/x$, X=6, нижче наступній схемі. Ця оцінка також визначає кількість ітерацій движка, необхідних для паралельної обробки кадрів (8 патернів) вхідних даних.

Оцінка значимості координат таблиці істинності свідчить про те, що найзначніші координати визначаються стовпцях відмінності символів. Незначні координати перебувають у шпальтах подібності символів. Значимість координат стовпців зменшується зліва направо (рис. 15).

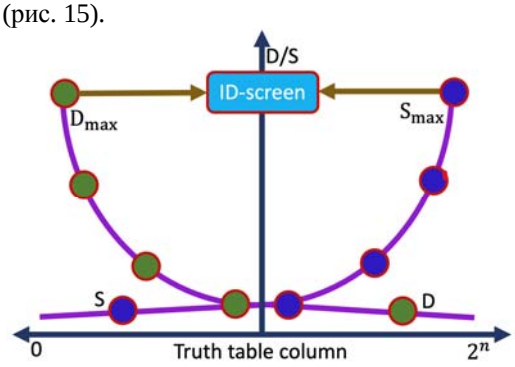


Рисунок 15 – Графіки подібності-відмінності по стовпцях таблиці істинності

У таблицях (див. рис. 14) наведено локальні та інтегральні оцінки автоматичного розпізнавання фрагментів та кейсів у бізнес-потоці даних. Насправді, оцінка покриття більше, ніж Y= 90% вважається прийнятним результатом ідентифікації знайдених у потоці патернів.

Властивості програмного коду, що реалізує in-memory computing на логічних векторах за допомогою read-write транзакцій: точність ідентифікації патернів-екранів – 95%, точність ідентифікації класів еквівалентностей ключових даних – 97%, точність пошуку помилок оператора при виконанні картки функціональності – виконанні карти функціональності. Вхідний потік великих даних містить 10 тисяч екранів.

ВИСНОВКИ

Зроблено крок на шляху створення векторно-логічного in-memory комп'ютингу, який використовує лише read-write транзакції на адресній пам'яті [23, 26, 29, 30]. Показано переваги векторної моделі для компактного опису процесів, явищ, функцій та структур.

Наукова новизна полягає у розробці наступних інноваційних рішень:

1) запропоновано нову векторно-матричну технологію паралельної обробки великих даних, як адрес, яка характеризується використанням read-write транзакцій на матричній пам'яті без використання процесорної логіки;

2) запропоновано архітектуру in-memory computing з глобальним зворотним зв'язком та алгоритм матричної паралельної обробки великих даних, як адрес;

3) запропоновано метод ідентифікації патернів ключовими словами, який характеризується використанням унітарно-кодованих компонентів даних для синтезу таблиці істинності бізнес-процесу, що дає можливість використовувати транзакцію read-write для паралельної обробки великих даних, як адрес.

Практична значимість дослідження полягає в тому, що будь-яку задачу штучного інтелекту (подібність-відмінність, класифікація-кластеризація та розпізнавання, ідентифікація образів) можна технологічно просто та ефективно вирішувати за допомогою таблиці істинності (або її похідних) та унітарно кодованих великих даних.

Перспективи дослідження пов'язані з імплементацією зехнології технології моделювання цифрових пристроїв на ринку EDA.

ЛІТЕРАТУРА

1. Shiqiang Zhu. Intelligent Computing: The Latest Advances, Challenges, and Future [Electronic resource] / [Shiqiang Zhu, Ting Yu, Tao Xu et al.] // Intelligent Computing. – 30 Jan 2023. – Vol 2. – 45 p. – Article ID: 0006 – DOI: 10.34133/icomputing.0006. – Access mode: https://spj.science.org/doi/10.34133/icomputing.0006
2. Hahanova A. Vector-Deductive Faults-As-Address Simulation / A. Hahanova // International Journal of Computing. – 2023. – №22(3). – P. 328–334. – https://doi.org/10.47839/ijc.22.3.3227
3. Faggella Daniel What is Machine Learning? [Electronic resource] / Daniel Faggella. – February 26, 2020. – Access mode: https://emerj.com/ai-glossary-terms/what-is-machine-learning/.
4. Bhardwaj R. Data analyzing using Map-Join-Reduce in cloud storage / R. Bhardwaj, N. Mishra, and R. Kumar // 2014 International Conference on Parallel, Distributed and Grid Computing. – Solan, India. – 2014. – P. 370–373. DOI: 10.1109/PDGC.2014.7030773.
5. Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining / [S. Suresh, T. Rajesh Kumar, M. Nagalakshmi et al] // 2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC). – Dharan, Nepal. – 2022. – P. 420–423. DOI: 10.1109/I-SMAC55078.2022.9986501.
6. Friend Recommendation System Using Map-Reduce and Spark: A Comparison Study / A. M. A. Sai et al // 2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT). – Kottayam, India. – 2023. – P. 1–6. DOI: 10.1109/ICITIIT57246.2023.10068723.
7. Agarwal S. and Sinha R. MR-KClust: An efficient Map Reduce based clustering Technique / S. Agarwal and R. Sinha // 2022 6th International Conference on Electronics, Communication and Aerospace Technology – Coimbatore, India. – 2022. – P. 1460–1464. DOI: 10.1109/ICECA55336.2022.10009369.

DOI 10.15588/1607-3274-2024-1-15

8. Comparative analysis of Gaussian mixture model, logistic regression and random forest for big data classification using map reduce / [V. Singh, R. K. Gupta, R. K. Sevakula and N. K. Verma] // 2016 11th International Conference on Industrial and Information Systems (ICIIS). – Roorkee, India. – 2016. – P. 333–338. DOI: 10.1109/ICIINFS.2016.8262961.
9. Adaptively Accelerating Map-Reduce/Spark with GPUs: A Case Study / [K. R. Jayaram, A. Gandhi, H. Xin, and S. Tao] // 2019 IEEE International Conference on Autonomic Computing (ICAC). – Umea, Sweden. – 2019 – P. 105–114. DOI: 10.1109/ICAC.2019.00022.
10. Besimi N. Overview of data mining classification techniques: Traditional vs. parallel/distributed programming models / N. Besimi, B. Çiço and A. Besimi // 2017 6th Mediterranean Conference on Embedded Computing (MECO). – Bar, Montenegro. – 2017. – P. 1–4. DOI: 10.1109/MECO.2017.7977126.
11. Tang S. DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters / S. Tang, B.-S. Lee and B. He // IEEE Transactions on Cloud Computing. – 1 July-Sept. 2014. – Vol. 2, no. 3. – P. 333–347. DOI: 10.1109/TCC.2014.2329299.
12. Replication-Based Query Management for Resource Allocation Using Hadoop and MapReduce over Big Data / [A. Kumar, N. Varshney, S. Bhatiya and K. U. Singh] // Big Data Mining and Analytics. – December 2023. – Vol. 6, No. 4. – P. 465–477. DOI: 10.26599/BDMA.2022.9020026.
13. Usharani A. V. Secure EMR Classification and Deduplication Using MapReduce / A. V. Usharani and G. Attigeri // IEEE Access. – 2022. – Vol. 10. – P. 34404–34414. DOI: 10.1109/ACCESS.2022.3161439.
14. A Fundamental Tradeoff Between Computation and Communication in Distributed Computing / [S. Li, M. A. Maddah-Ali, Q. Yu, and A. S. Avestimehr] // IEEE Transactions on Information Theory. – Jan. 2018. – Vol. 64, no. 1. – P. 109–128. DOI: 10.1109/TIT.2017.2756959.
15. Virtual Shuffling for Efficient Data Movement in MapReduce / [W. Yu, Y. Wang, X. Que, and C. Xu] // IEEE Transactions on Computers. – Feb. 2015. – Vol. 64, No. 2. – P. 556–568. DOI: 10.1109/TC.2013.216.
16. Shi X. Mammoth: Gearing Hadoop Towards Memory-Intensive MapReduce Applications / X. Shi et al. // IEEE Transactions on Parallel and Distributed Systems. – Aug. 2015. – Vol. 26, No. 8. – P. 2300–2315. DOI: 10.1109/TPDS.2014.2345068.
17. Luo W. Enterprise data economy: A hadoop-driven model and strategy / W. Luo // 2013 IEEE International Conference on Big Data. – Silicon Valley, CA, USA. – 2013. – P. 65–70. DOI: 10.1109/BigData.2013.6691690.
18. Wu J. Empirical Research on the Relationship between Logistics and Regional Economy Based on Big Data / J. Wu, Y. Chen, and Y. Chen // 2022 International Conference on Big Data, Information and Computer Network (BDICN). – Sanya, China. – 2022. – P. 12–15, DOI: 10.1109/BDICN55575.2022.00010.
19. Dai G. Research on Digital Economy Information System through Cloud Computing and Big Data Technology / G. Dai // 2021 IEEE International Conference on Data Science and Computer Application (ICDSCA). – Dalian, China. – 2021. – P. 734–737. DOI: 10.1109/ICDSCA53499.2021.9650173.
20. Liu R. Placement of High Availability Geo-Distributed Data Centers in Emerging Economies / R. Liu, W. Sun, and W. Hu // IEEE Transactions on Cloud Computing. – July-Sept. 2023. – Vol. 11, No. 3 – P. 3274–3288. DOI: 10.1109/TCC.2023.3280983.
21. Approximate Clustering Ensemble Method for Big Data / [M. S. Mahmud, J. Z. Huang, R. Ruby et al], // IEEE Transactions on Big Data. – Aug. 2023. – Vol. 9, No. 4. – P. 1142–1155. DOI: 10.1109/TBDATA.2023.3255003.
22. Davis M. Emil Post's contributions to computer science / M. Davis // Proceedings of the Fourth Annual Symposium on Logic in Computer Science. – 1989. – P. 134–136.
23. Devadze D. Vector-Deductive Memory-Based Transactions for Fault-As-Address Simulation / D. Devadze, Z. Davitadze and A. Hahanova // 2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT). – 2022. – Athens, Greece. – P. 1–6. DOI: 10.1109/DESSERT58054.2022.10018769.
24. Kovalev I. S. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions / [I. S. Kovalev, O. V Drozd, A. Rucinski et al.] // Herald of Advanced Information Technology. Publ. Nauka i Tekhnika – 2020. – Vol. 3, No. 4. – P. 252–262. DOI: 10.15276/hait.04.2020.4.
25. Hahanov V. Cyber Physical Computing for IoT-driven Services / V. Hahanov – New York : Springer, 2018. – 279 p. https://doi.org/10.1007/978-3-319-54825-8
26. Vector-Logic Synthesis Of Deductive Matrices For Fault Simulation / [W. Gharibi, A. Hahanova, V. Hahanov et al.] // Èlektronic modeling. – 2023. – № 45(2). – P. 16–33. doi.org/10.15407/emodel.45.02.016
27. Wroblewski F. J. Undecidability in the completion of truth-function logic / F. J. Wroblewski // Proceedings of the Twenty-First International Symposium on Multiple-Valued Logic. – 1991. – P. 225–229. DOI: 10.1109/ISMVL.1991.130734.
28. Liu H. A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis / H. Liu, X. Chen, and X. Liu // IEEE Access. 2022. – Vol. 10. – P. 32280–32289. DOI: 10.1109/ACCESS.2022.316017.
29. Векторно-логічне моделювання несправностей // [В. І. Хаханов, С. В. Чумаченко, Є. І. Литвинова та ін.] // Радіоелектроніка, інформатика, управління. – 2023. – №2. – P. 37–51. DOI: 10.15588/1607-3274-2023-2-5.
30. Векторні моделі логіки і структури для тестування та моделювання цифрових схем / [Г. В. Хаханова, В. І. Хаханов, С. В. Чумаченко та ін.] // Радіоелектроніка, інформатика, управління. – 2021. – №3. – P. 69–85. Doi: 10.15588/1607-3274-2021-3-7.

Received 04.01.2024.
Accepted 28.02.2024.

UDC 681.326

IN-MEMORY INTELLIGENT COMPUTING

Hahanov V. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Abdullayev V. H. – PhD, Associate Professor of the Computer Engineering Department, Azerbaijan State University of Oil and Industry, Baku, Azerbaijan.

DOI 10.15588/1607-3274-2024-1-15

Chumachenko S. V. – Dr. Sc., Professor, Head of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Lytvynova E. I. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Hahanova I. V. – Dr. Sc., Professor of the Design Automation Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. Processed big data has social significance for the development of society and industry. Intelligent processing of big data is a condition for creating a collective mind of a social group, company, state and the planet as a whole. At the same time, the economy of big data (Data Economy) takes first place in the evaluation of processing mechanisms, since two parameters are very important: speed of data processing and energy consumption. Therefore, mechanisms focused on parallel processing of large data within the data storage center will always be in demand on the IT market.

Objective. The goal of the investigation is to increase the economy of big data (Data Economy) thanks to the analysis of data as truth table addresses for the identification of patterns of production functionalities based on the similarity-difference metric.

Method. Intelligent computing architectures are proposed for managing cyber-social processes based on monitoring and analysis of big data. It is proposed to process big data as truth table addresses to solve the problems of identification, clustering, and classification of patterns of social and production processes. A family of automata is offered for the analysis of big data, such as addresses. The truth table is considered as a reasonable form of explicit data structures that have a useful constant – a standard address routing order. The goal of processing big data is to make it structured using a truth table for further identification before making actuator decisions. The truth table is considered as a mechanism for parallel structuring and packing of large data in its column to determine their similarity-difference and to equate data at the same addresses. Representation of data as addresses is associated with unitary encoding of patterns by binary vectors on the found universe of primitive data. The mechanism is focused on processorless data processing based on read-write transactions using in-memory computing technology with significant time and energy savings. The metric of truth table big data processing is parallelism, technological simplicity, and linear computational complexity. The price for such advantages is the exponential memory costs of storing explicit structured data.

Results. Parallel algorithms of in-memory computing are proposed for economic mechanisms of transformation of large unstructured data, such as addresses, into useful structured data. An in-memory computing architecture with global feedback and an algorithm for matrix parallel processing of large data such as addresses are proposed. It includes a framework for matrix analysis of big data to determine the similarity between vectors that are input to the matrix sequencer. Vector data analysis is transformed into matrix computing for big data processing. The speed of the parallel algorithm for the analysis of big data on the MDV matrix of deductive vectors is linearly dependent on the number of bits of the input vectors or the power of the universe of primitives. A method of identifying patterns using key words has been developed. It is characterized by the use of unitary coded data components for the synthesis of the truth table of the business process. This allows you to use read-write transactions for parallel processing of large data such as addresses.

Conclusions. The scientific novelty consists in the development of the following innovative solutions: 1) a new vector-matrix technology for parallel processing of large data, such as addresses, is proposed, characterized by the use of read-write transactions on matrix memory without the use of processor logic; 2) an in-memory computing architecture with global feedback and an algorithm for matrix parallel processing of large data such as addresses are proposed; 3) a method of identifying patterns using keywords is proposed, which is characterized by the use of unitary coded data components for the synthesis of the truth table of the business process, which makes it possible to use the read-write transaction for parallel processing of large data such as addresses. The practical significance of the study is that any task of artificial intelligence (similarity-difference, classification-clustering and recognition, pattern identification) can be solved technologically simply and efficiently with the help of a truth table (or its derivatives) and unitarily coded big data . Research prospects are related to the implementation of this digital modeling technology devices on the EDA market.

KEYWORDS: Intelligent Computing, Cloud, fog, and edge computing, Big data computing, In-memory computing, Cyber social competing, Hadoop Map-Reduce technique, big data as addresses, truth table, logical vector, similarities-differences, equivalence data, universe of primitives, patterns as a binary vector.

REFERENCES

1. Shiqiang Zhu, Ting Yu, Tao Xu, et al Intelligent Computing: The Latest Advances, Challenges, and Future [Electronic resource], *Intelligent Computing*, 30 Jan 2023, Vol. 2, 45 p. Article ID: 0006 DOI: 10.34133/icomputing.0006. Access mode: https://spj.science.org/doi/10.34133/icomputing.0006
2. Hahanova A. Vector-Deductive Faults-As-Address Simulation, *International Journal of Computing*, 2023, №22(3), pp. 328–334. https://doi.org/10.47839/ijc.22.3.3227
3. Daniel Faggella. What is Machine Learning? [Electronic resource], February 26, 2020. Access mode: https://emerj.com/ai-glossary-terms/what-is-machine-learning/.
4. Bhardwaj R., Mishra N., and Kumar R. Data analyzing using Map-Join-Reduce in cloud storage, *2014 International Conference on Parallel, Distributed and Grid Computing*. Solan, India, 2014, pp. 370–373. DOI: 10.1109/PDGC.2014.7030773.
5. Suresh S., Rajesh Kumar T., Nagalakshmi M., Bennilo Fernandes J. and Kavitha S. Hadoop Map Reduce Techniques: Simplified Data Processing on Large Clusters with Data Mining, *2022 Sixth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. Dharan, Nepal, 2022, pp. 420–423. DOI: 10.1109/I-SMAC55078.2022.9986501.
6. Sai A. M. A. et al. Friend Recommendation System Using Map-Reduce and Spark: A Comparison Study, *2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT)*. Kottayam, India, 2023, pp. 1–6. DOI: 10.1109/ICITIIT57246.2023.10068723.
7. Agarwal S. and Sinha R. MR-KClust: An efficient Map Reduce based clustering Technique, *2022 6th International*

DOI 10.15588/1607-3274-2024-1-15

Conference on Electronics, Communication and Aerospace Technology. Coimbatore, India, 2022, pp. 1460–1464. DOI: 10.1109/ICECA55336.2022.10009369.
8. Singh V., Gupta R. K., Sevakula R. K. and Verma N. K. Comparative analysis of Gaussian mixture model, logistic regression and random forest for big data classification using map reduce, *2016 11th International Conference on Industrial and Information Systems (ICIIS*). Roorkee, India, 2016, pp. 333–338. DOI: 10.1109/ICIINFS.2016.8262961.
9. Jayaram K. R., Gandhi A., Xin H., and Tao S. Adaptively Accelerating Map-Reduce/Spark with GPUs: A Case Study, *2019 IEEE International Conference on Autonomic Computing (ICAC)*. Umea, Sweden, 2019, pp. 105–114. DOI: 10.1109/ICAC.2019.00022.
10. Besimi N., Çiço B. and Besimi A. Overview of data mining classification techniques: Traditional vs. parallel/distributed programming models, *2017 6th Mediterranean Conference on Embedded Computing (MECO)*. Bar, Montenegro, 2017, pp. 1–4. DOI: 10.1109/MECO.2017.7977126.
11. Tang S., Lee B.-S. and He B. DynamicMR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters, *IEEE Transactions on Cloud Computing*, 1 July-Sept. 2014, Vol. 2, No. 3, pp. 333–347. DOI: 10.1109/TCC.2014.2329299.
12. Kumar A., Varshney N., Bhatiya S. and Singh K. U. Replication-Based Query Management for Resource Allocation Using Hadoop and MapReduce over Big Data, *Big Data Mining and Analytics*, December 2023, Vol. 6, No. 4, pp. 465–477. DOI: 10.26599/BDMA.2022.9020026.
13. Usharani A. V. and Attigeri G. Secure EMR Classification and Deduplication Using MapReduce, *IEEE Access*, 2022, Vol. 10, pp. 34404–34414. DOI: 10.1109/ACCESS.2022.3161439.
14. Li S., Maddah-Ali M. A., Yu Q., and Avestimehr A. S. A Fundamental Tradeoff Between Computation and Communication in Distributed Computing, *IEEE Transactions on Information Theory*, Jan. 2018, Vol. 64, No. 1, pp. 109–128. DOI: 10.1109/TIT.2017.2756959.
15. Yu W., Wang Y., Que X., and Xu C. Virtual Shuffling for Efficient Data Movement in MapReduce, *IEEE Transactions on Computers*, Feb. 2015, Vol. 64, No. 2, pp. 556–568. DOI: 10.1109/TC.2013.216.
16. Shi X. et al. Mammoth: Gearing Hadoop Towards Memory-Intensive MapReduce Applications, *IEEE Transactions on Parallel and Distributed Systems*, Aug. 2015, Vol. 26, No. 8, pp. 2300–2315. DOI: 10.1109/TPDS.2014.2345068.
17. Luo W. Enterprise data economy: A hadoop-driven model and strategy, *2013 IEEE International Conference on Big Data*. Silicon Valley, CA, USA, 2013, pp. 65–70. DOI: 10.1109/BigData.2013.6691690.
18. Wu J., Chen Y., and Chen Y. Empirical Research on the Relationship between Logistics and Regional Economy Based on Big Data, *2022 International Conference on Big Data, Information and Computer Network (BDICN*). Sanya, China, 2022, pp. 12–15, DOI: 10.1109/BDICN55575.2022.00010.
19. Dai G. Research on Digital Economy Information System through Cloud Computing and Big Data Technology, *2021 IEEE International Conference on Data Science and Computer Application (ICDSCA)*. Dalian, China, 2021, pp. 734–737. DOI: 10.1109/ICDSCA53499.2021.9650173.
20. Liu R., Sun W., and Hu W. Placement of High Availability Geo-Distributed Data Centers in Emerging Economies, *IEEE Transactions on Cloud Computing*, July-Sept. 2023, Vol. 11, No. 3, pp. 3274–3288. DOI: 10.1109/TCC.2023.3280983.
21. Mahmud M. S., Huang J. Z., Ruby R., Ngueilbaye A. and Wu K. Approximate Clustering Ensemble Method for Big Data, *IEEE Transactions on Big Data*, Aug. 2023, Vol. 9, No. 4, pp. 1142–1155. DOI: 10.1109/TBDATA.2023.3255003.
22. Davis M. Emil Post's contributions to computer science, *Proceedings of the Fourth Annual Symposium on Logic in Computer Science*, 1989, pp. 134–136.
23. Devadze D., Davitadze Z. and Hahanova A. Vector-Deductive Memory-Based Transactions for Fault-As-Address Simulation, *2022 12th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, 2022. Athens, Greece, pp. 1–6. DOI: 10.1109/DESSERT58054.2022.10018769.
24. Kovalev I. S., Drozd O. V., Rucinski A., Drozd M. O., Antoniuk V. V., Sulima Y. Y. Development of Computer System Components in Critical Applications: Problems, Their Origins and Solutions, *Herald of Advanced Information Technology. Publ. Nauka i Tekhnika*, 2020, Vol. 3, No.4, pp. 252–262. DOI: 10.15276/hait.04.2020.4.
25. Hahanov V. Cyber Physical Computing for IoT-driven Services. New York, Springer, 2018, 279 p. https://doi.org/10.1007/978-3-319-54825-8
26. Gharibi W., Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Hahanov I. Vector-Logic Synthesis Of Deductive Matrices For Fault Simulation, *Èlektronic modeling*, 2023, № 45(2), pp. 16–33. doi.org/10.15407/emodel.45.02.016
27. Wroblewski F. J. Undecidability in the completion of truth-function logic, *Proceedings of the Twenty-First International Symposium on Multiple-Valued Logic*, 1991, pp. 225–229. DOI: 10.1109/ISMVL.1991.130734.
28. Liu H., Chen X., and Liu X. A Study of the Application of Weight Distributing Method Combining Sentiment Dictionary and TF-IDF for Text Sentiment Analysis, *IEEE Access*. 2022, Vol. 10, pp. 32280–32289. DOI: 10.1109/ACCESS.2022.316017.
29. Hahanov V., Chumachenko S., Litvinova Y., Hahanova I., Khakhanova A., Shkil A., Rakhlis D., Hahanov I., Shevchenko O. Vector-Logical Fault Simulation, *Radio Electronics, Computer Science, Control*, 2023, №2, pp. 37–51. https://doi.org/10.15588/1607-3274-2023-2-5
30. Hahanova A., Hahanov V., Chumachenko S., Litvinova E., Rakhlis D. Vector-Driven Logic And Structure For Testing And Deductive Fault Simulation, *Radio Electronics, Computer Science, Control*, 2021, No. 3, pp. 69–85. https://doi.org/10.15588/1607-3274-2021-3-7.

DOI 10.15588/1607-3274-2024-1-15

p-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2024. № 1
e-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2024. № 1

ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

PROGRESSIVE INFORMATION TECHNOLOGIES

UDC 004.9

REALIZATION OF THE DECISION-MAKING SUPPORT SYSTEM FOR TWITTER USERS' PUBLICATIONS ANALYSIS

Batiuk T. – Postgraduate student of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

Dosyn D. – Dr. Sc., Professor of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. The paper emphasizes the need for a decision-making system that can analyze users' messages and determine the sentiment to understand how news and events impact people's emotions. Such a system would employ advanced techniques to analyze users' messages, delving into the sentiment expressed within the text. The primary goal is to gain insights into how news and various events reverberate through people's emotions.

Objective. The objective is to create a decision-making system that can analyze and determine the sentiment of user messages, understand the emotional response to news and events, and distribute the data into clusters to gain a broader understanding of users' opinions. This multifaceted objective involves the integration of advanced techniques in natural language processing and machine learning to build a robust decision-making system. The primary goals are sentiment analysis, comprehension of emotional responses to news and events, and data clustering for a holistic view of user opinions.

Method. The use of long-short-term memory neural networks for sentiment analysis and the *k*-means algorithm for data clustering is proposed for processing large volumes of user data. This strategic combination aims to tackle the challenges posed by processing large volumes of user-generated data in a more nuanced and insightful manner.

Results. The study and conceptual design of the decision-making system have been completed and the decision-making system was created. The system incorporates sentiment analysis and data clustering to understand users' opinions and the sentiment value of such opinions dividing them into clusters and visualizing the findings.

Conclusions. The conclusion is that the development of a decision-making system capable of analyzing user sentiment and clustering data can provide valuable insights into users' reactions to news and events in social networks. The proposed use of long-short-term memory neural networks and the *k*-means algorithm is considered suitable for sentiment analysis and data clustering tasks. The importance of studying existing works and systems to understand available algorithms and their applications is emphasized. The article also describes created and implemented a decision-making system and demonstrated the functionality of the system using a sample dataset.

KEYWORDS: natural language processing, convolutional neural network, recurrent neural network, LSTM, *k*-means clustering.

ABBREVIATIONS

NLP is a natural language processing;
CNN is a convolutional neural network;
RNN is a recurrent neural network;
LSTM is a long-short-term memory;
HC is a hierarchical clustering;
KMC is a *k*-means clustering;
DB is a database;
DMS is a decision-making system;
SDS is a sample data set.

NOMENCLATURE

S is a decision-making analysis system;
I is a set of inputs;
O is a set of outputs;
R are the basic rules for processing the flow of input data to the sentiment analysis system;
F is an input data processing parameters;
N is a recurrent neural network;
α is an input validation operator;
β is an input data processing operator;
γ is a the search operator for relevant users after clustering;
P is an improved recurrent model of searching for validated users;
μ is an user authentication operator;
χ is an input data set formation operator;
ω is an operator of list formation and sentiment analysis data;
λ is a validation request resolution operator;
i_1 is a set of authentication data (login, password, set of distributed data);
i_2 is a data storage of publications of a specific social network;
i_3 is a different sets of user data;
i_4 is a specific user request;
o_1 is a clustering requests using the elbow method;
o_2 is a set of updates for the user's profile in the selected social network;

DOI 10.15588/1607-3274-2024-1-16

o_3 is a saving at the request of the DMS user;
r_1 is a set of rules of the data saving algorithm;
r_2 is an operating rules of recurrent neural network;
r_3 is a set of rules of operations of a convolutional neural network;
r_4 is a set of hierarchical clustering rules;
r_5 is a set of *k*-means clustering rules;
u_1 is a set of levels of data processing;
u_2 is a set of data processing requirements;
u_3 is a set of text validation requirements;
u_4 is a set of multiple levels of creating a linked list of relevant users;
u_5 is a set of LSTM data analysis requirements;
X_{UF} is a the result of an authenticated user.

INTRODUCTION

Creating and implementing an intelligent system for sentiment analysis and clustering publications is a relevant and promising task these days, most communication between people occurs in social networks according to certain situations or circumstances. Each message of a social network user has particular semantics, reflects certain thoughts and analysis of the relevant situation, or is a reaction to a specific event. Modern algorithms and approaches to data analysis make it possible to effectively and relatively analyze large volumes of text data, thanks to which it is possible to determine the average tone of users' reactions to certain events and, as a result, conclude the analyzed content. In this way, it is possible to understand the attitude of different groups of users to a certain type of content, offers, goods, and other market offers, and large brands and corporations actively use these approaches to the analysis of information by determining the tone of the user text and further dividing this messages and, accordingly, the users themselves into clusters for further work with the resulting clusters.

To implement this kind of algorithms, first of all, it is necessary to determine precisely which group of users we want to research, that is, it is essential to have certain keywords that will be used for searching, geolocation, setting tags, etc. This is the starting point of the current research, next it is necessary to load the desired dataset, that is, a pool of user messages stored according to a given predicate. After receiving the file with the saved messages, it is necessary to perform a general check of the file for the correctness of the data and, after that, to digitally structure the data in such a way that it is possible to operate with the data as accurately as possible, that is, they need to be formatted and brought to a single structure. After that, it is necessary to analyze user messages. The general algorithm can be divided into two main sub-algorithms: it is an analysis of the sentiment of user messages, according to a confident approach, where each message has its own sentiment level and rating, and also clustering of messages to divide users and their messages into a number determined during the operation of the intelligent system clusters. It is most convenient to show the obtained results in the form of graphs and charts for visualization and understanding of the general situation in the results of the analysis of user publications. These algorithms are not new and contain vast possibilities for modification and optimization, which will be carried out in further work.

1 PROBLEM STATEMENT

The system of sentiment analysis of user publications S is represented by a tuple simulation model:

$$S = < R, I, F, O, N, \alpha, \beta, \gamma >,$$

where $R = \{r_1, r_2, r_3, r_4\}$, $F = \{u_1, u_2, u_3, u_4, u_5\}$, $I = \{i_1, i_2, i_3, i_4\}$, $O = \{o_1, o_2, o_3\}$.

The main processes of DMS users' publication analysis are "User authentication", "Lemmatization of user data", "Performing sentiment analysis" and "Performing *k*-means clustering".

The DMS users' publication analysis user authentication process will be described by superposition:

$$X_{UF} = \alpha \circ \beta \circ \mu,$$
$$X_{UF} = \alpha\,(\beta(\mu\,(i_1, i_3, i_4), r_5, u_5), u_4).$$

The process of forming lemmatized user data of DMS users' publication analysis will be described by superposition: $C_{CU} = \alpha \circ \beta \circ \chi$, so

$$X_{UC} = \alpha\,(\beta(\chi(X_{UF}, i_2, i_3, i_4), r_1, u_3), r_2).$$

The process of performing sentiment analysis will be described by superposition:

$$X_{UK} = \omega \circ \gamma \circ \beta \circ \alpha,$$
$$X_{UK} = \beta\,(\gamma(\omega(\alpha(X_{UC}, i_4), i_2), u_5), r_3).$$

The process of performing *k*-means clustering will be described by superposition:

$$X_{UM} = \lambda \circ \gamma \circ \beta \circ \alpha,$$
$$X_{UM} = \lambda(\gamma(\beta(\alpha(X_{UK}, i_4), i_3), u_3), r_5).$$

2 REVIEW OF THE LITERATURE

Neural networks have become an indispensable part of the work of various companies and corporations. In the article [1], the authors investigate the exact role of deep learning in e-commerce and the principles of working with users using the example of online store networks. The authors investigated the concept of user reviews of certain products of different quality and the impact of current reviews on the sale of products by other users of the system in the future, investigated when precisely and under what conditions users pay the most or least attention to product reviews and, accordingly, how exactly positive feedback is composed whether negative feedback can affect the purchase attractiveness of the product. In the article [2], the authors investigated user comments under videos on the YouTube social network, a dataset with videos about the COVID-19 virus and user comments was selected, as the sentiment of written messages, their frequency, and the activity of writing by users were investigated, accordingly, it was assumed that a part of users with negative comments is bots, due to

DOI 10.15588/1607-3274-2024-1-16

very similar message patterns and almost the same level of negative tone. On the contrary, the authors in the article [3] examined the comments of famous personalities on social networks. They found many comments written at certain selected moments in time and had almost the same positive tone in the range from 0 to 1, which also indicated the unnatural state of these messages.

Also, the authors in the article [4] considered several types of user personification models in the Twitter social network using LSTM-neural networks, several critical parameters for each user were taken from the open API and considered as a separate dataset with the subsequent division of users into groups according to their location, description, and profile avatar, and a corresponding neural network was trained to assign users to certain groups. A system was also created [5] that processes author citations in articles and provides an opportunity to analyze the correctness and correctness of the current text and correct it using a recurrent neural network, the main task of which is to process text data and analyze subsequent data based on training with the teacher. Also, in the article [6], the authors considered machine learning based on customer feedback on hotels to understand the situation in this market and further develop a hotel development plan based on positive and negative customer feedback.

3 MATERIALS AND METHODS

The purpose of this work is the implementation of an intelligent system of sentiment analysis and clustering of publications in the Twitter social network. The idea of analyzing the sentiment of user messages or publications in social networks is familiar because several practically implemented systems perform a similar task. At the moment, an important task is the optimization and the most effective use of already existing technologies and the correct selection of models and algorithms to perform a particular task, which may depend both on the size of the input dataset and on the size of individual text tokens [7] within the dataset, or even parameters search of text information for users using a particular system, or only its partial functionality for text research.

The process of analyzing text publications or user comments can be divided into two essential parts: the analysis of the text's sentiment and the clustering of text data. This task is non-trivial and quite complex since many parameters must be considered before creating such an intelligent system for analyzing textual data: the size of the sample, the textual data, and the context to which these data belong. These users also write posts or messages to react to a particular event or set of events happening at a certain time. This all means that it is impossible to create a unique intelligent system that can cover all cases and analyze text data with approximately the same efficiency. One way or another, there are tasks that each specific system can perform efficiently and with maximum accuracy. Still, there are also tasks for which the same set of algorithms cannot function efficiently or accurately. Thus, convolutional neural networks are often used to analyze the text's sentiment, and hierarchical clustering is used for cluster analysis. These algorithms are efficient and time-tested but can only perform analysis on small to medium-sized datasets or data samples. These algorithms can be used without problems for large volumes of data. Still, such research will be inefficient and less accurate, especially if the number of text data units is large and the tokenized object is small [8]. A clear example is the social network Twitter, where one user message can have a maximum of 280 characters to create a post. In order to analyze the sentiment of messages and their clustering as effectively as possible, we will use the LSTM neural network and the *k*-means clustering algorithm, thanks to which we can achieve 10...15% greater efficiency compared to convolutional neural networks and hierarchical clustering. Before describing the functionality of the system and the main algorithms of the long-short-term memory neural network and the *k*-means clustering algorithm, it is worth paying attention to unsolved problems, namely, why convolutional neural networks specifically in the context of user posts and comments on the Twitter social network and conventional hierarchical clustering are inefficient and may show unexpected results in the results of text sentiment analysis and subsequent clustering [9], namely due to nuances in the implementation of these algorithms for working with text and due to the properties of the weights provided by the convolutional neural network, which are important to consider, since the generated model may be invalid if the weights are miscalculated.

A convolutional neural network, or CNN, can be thought of as a set of matrices that make up one large matrix, in which both horizontal and vertical sets of elements, formed in the process of learning a neural network to build a model, can be chosen as vectors for training. In the course of the work, a collection of words is superimposed on each other with the help of vectors since each word is represented by a separate vector of letters that together form a particular image. With the help of the created system predicates, the generated image of vectors representing a set of text data or words is filtered, since the predicate has the same width and length as the value check, it is possible to analyze only part of the vector, therefore, for the efficiency of work, vectors are most often put into temporary matrices for correct and more effective filtering using a given predicate[10]. Since this is a natural language processing task, the filters we employ in the work process have the same width as the length of the investigated text element or a single word. The height, on the other hand, is more static and can usually change its size from 0 to 5 since it is necessary to form certain sequences of values from the received text information, which will later be written into lists of values, accordingly, it is required to limit the number of lists to 5 pieces to be able to carry out effective parallel processing of text information and model training with given limitations of the current neural network [11]. Due to these algorithmic limitations, the main problem of the impossibility of accurate learning and subsequent text analysis using convolutional neural networks arises. The

DOI 10.15588/1607-3274-2024-1-16

text with restrictions will be analyzed, and the result of the analysis of the sentiment of the text will be correct. Still, due to the height restriction of the vector, the neural network can only be trained on posts and messages with polar sentiment values, i.e., –1, 0, or 1 [12], respectively clearly negative, neutral or positive comments without the possibility of their distribution over a certain range, which, on the one hand, is not critical and can be helpful in a general understanding of the sentiment of messages. Still, on the other hand, with the help of such a convolutional neural network, it is impossible to carry out an accurate sentiment analysis. There are also reservations about the speed of operation because such a neural network has low speed and efficiency since the memory algorithm of repeated data is used to save the polar values of the vectors.

If we talk about hierarchical clustering, then it also has certain disadvantages in its use. The biggest drawback is similar to what was described earlier about the convolutional neural network, namely, the inability to perform complex tasks, as well as the problem of low speed and efficiency of such a clustering algorithm. To begin with, it is worth noting that hierarchical clustering is formed based on a tree graph, also known as a dendrogram. Accordingly, when constructing this tree graph, we use an agglomerative approach to work with input data [13]. Having a dendrogram and applying an agglomerative approach, we observe relatively monotonous clusters and simultaneously look for possible connections between them, in the second step, we successively combine clusters into a separate "connection" based on the predicates we need. The main advantage is simplicity and clarity of use, that is, the presence of a convenient tree-like structure, in which clusters can often be seen "by eye" with the correct generation of the model [14]. Unfortunately, the downside of this simplicity is that it can usually only be used for small amounts of data and small datasets. One of the main disadvantages is the complexity of the tree hierarchy, which is clustered in time with an algorithmic complexity of $O(n^2 \log n)$, where n is the number of total data points.

If, on the contrary, we take the *k*-means algorithm, in it, we will use the optimization of a specific objective function, for example, within a certain range of values from k to l, that is, we do not actually have an objective function, and therefore we will have a much more efficient implementation due to the complexity of the algorithm $O(nKm)$, where K is the number of clusters, and m is the number of average values [15]. Also, the problem of hierarchical clustering is a certain staticity, namely the impossibility of undoing the previous steps of the algorithm, that is if we cluster n–1 points. Then it turns out that the connection between the clusters was incorrect, or there were problems creating a tree graph on one of the hierarchy levels. We cannot cancel this step at the program level during execution because either the final dendrogram with distorted values will be obtained, or it will be necessary to stop the program and start the system from the beginning, which is also a big minus compared to the work the *k*-means algorithm, where it is possible to check the correctness of average values using a specified execution condition [16]. Based on the above information, it can be concluded that convolutional neural networks and hierarchical clustering are suboptimal for analyzing the sentiment of user posts and messages in the Twitter social network. Instead, to optimize and speed up the operation of the intelligent system, it is more appropriate choice to use a neural network based on LSTM architecture and the modern *k*-means clustering algorithm.

4 EXPERIMENTS

Speaking about the functionality of the created intelligent system for the analysis of sentiment and clustering of posts in the Twitter social network, first of all, it is worth paying attention to two main algorithms, which will perform the main volume of work in the middle of the system, this is an LSTM neural network, with the help of which an effective and fast analysis of the sentiment of posts and comments by Twitter users and the *k*-means clustering algorithm, which will be used to select the main clusters and distribute the user text according to these clusters. To begin with, it is worth revealing the essence of the neural network that was built.

One of the most popular and effective neural network models focused on processing time series is the Long Short-Term Memory (LTSM) model [17]. It is efficient to use and much more accurate than the previously mentioned convolutional neural network, but at the same time, it is challenging to implement because it is essentially a recurrent neural network whose primary function is to predict a sequence of data from an input dataset and the corresponding problem that must be resolved. For a better understanding of the context of a neural network application, it is worth formulating what is meant by time series analysis for which we are building a neural network. The point is that the input data represent particular points of information that are analyzed in certain time intervals. Thanks to this, it is possible to create and analyze patterns of processes that occur in certain time intervals. Since we have certain related data points [18], we use a Recurrent Neural Network (RNN).

The Long Short-Term Memory neural network, or LSTM, is a separate and special case of recurrent neural networks because this model is even more efficient. After all, it can store a collection of information for extended periods. In the LSTM neural network, two main recurrent problems are overcome: gradients that disappear and gradients that are deformed during the operation of the intelligent system. An LSTM model consists of a memory cell state and three main passes. The memory cell's state stores the cell's value for a certain period of time and is a kind of tape that moves and linearly transmits data further along the conveyor with practically no additional deformations. In the LSTM network model, it is possible to add, modify and delete information using the three passes mentioned earlier. Passages help regulate information and are key in the LSTM architecture of a neural network, as thanks to the state of the memory cell,

DOI 10.15588/1607-3274-2024-1-16

the data stream is formed into a particular linear structure and allows for uniform distribution of memory throughout the neural network's operation time in the intelligent system. A neural network is in 3 main states [19]: either data is input or output, or it is forgotten due to distortion or unnecessary information.

Accordingly, the work of the neural network is implemented using the previously mentioned three passes. The first is the "forgetting" pass, which is responsible for removing information that, due to distortion or use, is no longer needed to analyze the text's sentiment so that it can be removed and make room for the following information at the input. Thanks to this, the model becomes more efficient at each neural network step. This pass has two main inputs: the hidden state of the previous memory cell and the current input at this step. These data are multiplied with previously created weight matrices, after which a certain displacement factor is added. Next, a sigmoid function is applied, which produces a result from 0 to 1, thanks to which the neural network "knows" which information can be "forgotten" and which data can be passed on. If we have a value of 0, the information about this state of the memory cell can be deleted, if the value is close to 1, then all the information about the cell must be saved and passed on. The vector output of the sigmoid function, obtained as a result of working out part of the neural network model, must be multiplied by the state of the memory cells that were not deleted during the work, and the result passed on [20], as a result of which the first pass is formed, which performs the vital function of removing all unnecessary for analysis of the sentiment of the text of the memory cells. Figure 1 depicts the main aspects of the LSTM neural network architecture.

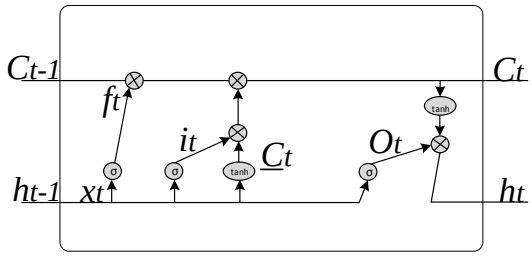


Figure 1 – Architecture of Long Short-Term Memory neural network

The second is the "entrance" passage. This pass is used to add information to the state of the memory cell. Initially, the allocated values to be added to the cell are adjusted using a sigmoid function, the inputs are still the hidden state of the previous memory cell and the current input of the algorithm step. During operation, a vector is created that contains all or almost all possible values that must be added to the state of the memory cell. This happens using the tanh function, so the tangent outputs values from –1 to 1. Next, you need to determine [21] the value of the sigmoid function, a regulatory function. The available values of the regulatory function must be multiplied by the previously created vector of values; all vital information for the intelligent system to analyze the sentiment of a certain text is added to the state of the memory cell using the data addition operation, thanks to which you can be sure that only valid and necessary information, which was previously filtered and checked, was added through the "input" pass during network operation.

Last is the "output" pass, which is used to use the current information that is available at a given point in time and display the most relevant results. First, a separate vector is created after applying the tanh function to the state of the memory cell, where the output value ranges from –1 to 1. The same sigmoid function again acts as a regulatory function, that is, it is used to regulate precisely those values that need to be output from the vector by using the two previously mentioned inputs, namely the hidden state of the previous memory cell and the current input of information at this step. The value of the sigmoid function must be multiplied by a vector, and the obtained result of the operation is used as the output value. Also, the neural network sends the result to the hidden state of the following memory cell, which, accordingly, is a modern solution, thanks to which the LSTM neural network [22] is the most efficient and convenient in predicting sequences and performs exceedingly well in the task of analyzing the sentiment of posts on the Twitter social network.

It is also worth clarifying the importance of the sigmoid activation function during the "forgetting" pass operation because instead of distributing the values between –1 and 1, the input values are distributed between 0 and 1. This helps to update the changed data in time or "forget" the distorted ones. that is, those that will no longer be of any use in analyzing the text's sentiment. The division is made precisely between 0 and 1 through mathematical multiplication since any number we multiply by 0 will result in 0, and vice versa, any number multiplied by one will remain unchanged. Thanks to this, the sigmoid function helps to effectively determine which data should be "forgotten" or deleted, which should be updated to the current pitch value, and which should be kept and passed on to the next "input" pass. So, Figure 2 shows the step-by-step algorithm of the Long Short-Term Memory neural network.

In addition to implementing the LSTM neural network, an important part of the intelligent system for sentiment analysis and clustering of posts in the Twitter social network is the correct implementation of clustering using the *k*-means algorithm. As already described earlier, the usual hierarchical clustering, although it is pretty popular and has certain clearly expressed advantages, is still not suitable for our task, namely for processing the exact values of the text's sentiment and, accordingly, the selection of the necessary clusters. To begin with, it is worth clarifying that clustering is the process of dividing a volume of data of a certain size into several clearly defined groups similar in structure in such a way that the values of data points in one group are more similar to the importance of other data points in the same group than other points data that were assigned to other groups. It is

DOI 10.15588/1607-3274-2024-1-16

also worth noting that clustering is a learning algorithm without a teacher due to its work features.

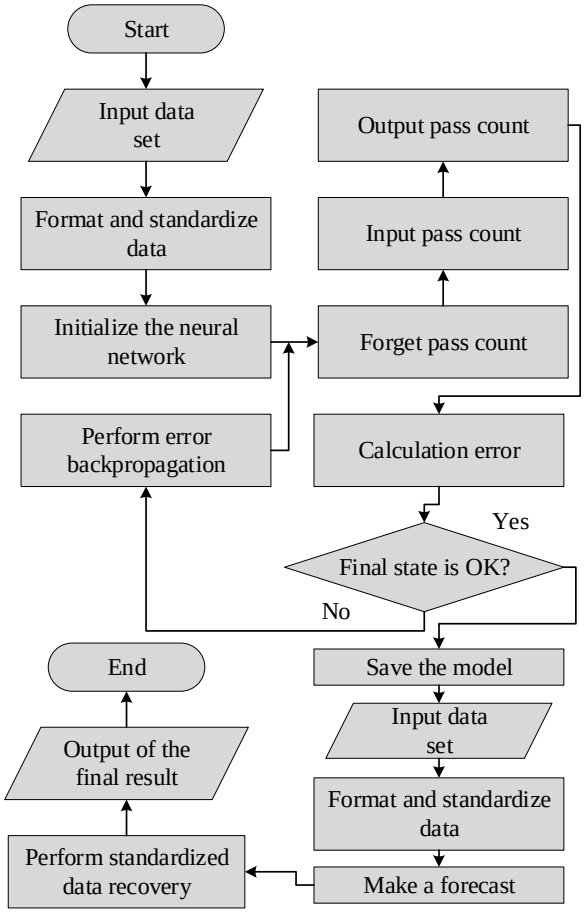


Figure 2 – LSTM neural network algorithm

k-means clustering is an algorithm that does not need to mark the input data in the learning process, unlike learning algorithms with a teacher. *k*-means divides objects into clusters in such a way that all objects within a cluster are similar to each other and dissimilar to objects in different clusters. The Latin letter k stands for a number that represents the number of clusters to be created. Also, the essence of the algorithm is to find the best or optimal value of the clusters to be used for better work with sentiment analysis. The *k*-means clustering procedure itself is a relatively simple and straightforward mathematical problem. To begin with, a general notation should be defined. For example, we have from C_1 to C_k sets that have indices of observations in each cluster, these sets satisfy two main properties: firstly, each statement belongs to at least one of the k existing clusters, and secondly, the clusters cannot overlap each other, that is, one set of observations can belong to only one unique cluster. The basis of *k*-means clustering is the idea that the best clustering is the one where the variation of values within the cluster is minimal, that is, the variation within a particular cluster C_k is a measure of the magnitude of the cluster observation, where one cluster differs from another, this is the problem that is solved *k*-means clustering within an intelligent system.

It is worth noting the importance of the parameter k, which determines the number of clusters, a certain approximate value can often determine it simply by estimating the size of the dataset and the data it contains. For our task of analyzing the text's sentiment, the "elbow" method was chosen, allowing you to more accurately determine the number of clusters required for work by running the *k*-means algorithm with a different set of clusters to determine the optimal value empirically. The "elbow" method involves finding a certain metric to estimate how good the clustering result is for different k values by finding the "elbow" point to separate all unnecessary further values and choose the optimal one up to the specified point. A sharp drop in values in the corresponding graph means that the clustering value is being optimized, but there is a point where the sharp drop in values stops falling and stabilizes. This is the same "elbow" point. That is, all values after the "elbow" point should be discarded, and only those where there is a decline in the average sum of squares of the observation values within each cluster, where the fall is the largest – this point will most likely represent the optimal number of clusters.

The clustering algorithm is essentially a method of dividing the observation indices in each cluster so that the goal of the last equation of selecting *k* clusters is minimized. The problem is quite tricky because there are k^n ways to partition n observations into clusters, but there is an algorithm that can be used to find a local optimum for the *k*-means optimization problem. The *k*-means clustering algorithm itself consists of two global steps. To begin with, it is necessary to choose a random value from 1 to k for each of the data observations that were separately selected from the dataset, they represent certain initial values for the clusters. This procedure should be repeated until the values of the clusters stop changing. For each of the k clusters, it is necessary to calculate the cluster's centroid. The resulting centroid of the *k*-th cluster is a separate vector of the average values of the observation indices in the *k*-th cluster. Each cluster needs to be assigned an observation identifier whose centroid is the closest, it is necessary to make sure that all clusters are stable and do not contain uncertainties, as this can prevent the correct distribution of observation indices relative to clusters. The *k*-means clustering algorithm is shown in Figure 3.

Having outlined the logic and functionality of the LSTM neural network and the *k*-means clustering algorithm, it is clear how sentiment analysis and clustering of posts and messages of users of the Twitter social network will take place within the system. In addition, it is necessary to understand the logic of the intelligent system as a whole. Neural network and clustering algorithms perform only a certain function in the system. The system itself consists of many processes, which is not surprising since we interact with a genuine

DOI 10.15588/1607-3274-2024-1-16

user and push back from the parameters set by the user at a certain time when this user works with the system.

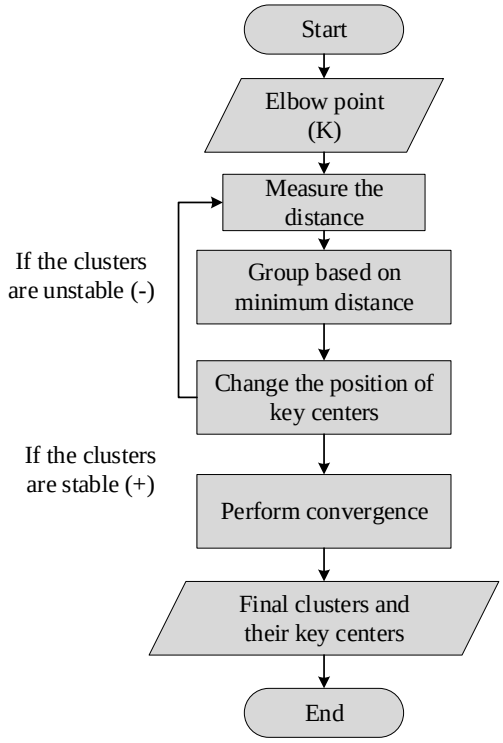


Figure 3 – The *k*-means clustering algorithm

The use case diagram is the most convenient diagram for displaying the general concept of user interaction, the system, and other elements of the overall structure because it mainly consists of actors and their use cases. In this intelligent system of sentiment analysis and clustering of textual data, there is a single actor, the “User”, who is a separate entity in relation to the system and is depicted separately from other usage options, representing the main aspects of the system’s operation. The diagram of use cases itself is shown in Figure 4.

In the diagram, the intelligent system is depicted inside a rectangle, it consists of several prominent use cases responsible for saving tweets, that is, posts and comments of users of the social network Twitter, and there are also use cases that are responsible for formatting text messages, data to the desired form, performing LSTM analysis of the sentiment of text data, performing cluster analysis, outputting all the necessary received results in the format that is most suitable according to the received data and ending the system. In addition to the usual use cases that perform the main functions of the intelligent system, the diagram shows inclusion and extension options that help to explain in more detail the essence of the operation of the prominent use cases. For example, the use case that describes saving tweets contains inclusion options that describe setting parameters in a particular format and saving the formed dataset in a .csv file. There is also an extension option that illustrates the process of selecting the main search keywords. The use case that describes the formatting of text data only has extension options that describe removing redundant characters from text, standardizing existing data, and removing text processors that were loaded by default. Also, another use case that is responsible for LSTM sentiment analysis of text data only has options for expanding the possibilities, such as performing the calculation of all current network layers and memory cell state and three main passes, performing sentiment prediction of text information, i.e., posts and comments of a Twitter user and final saving of the created model. Then there is the option of using cluster analysis, including options for expansion and inclusion, such as finding the “elbow” point, determining the final clusters and their key centers, and implementing convergence. It is also worth noting the last two options of use, namely the results’ output and the intelligent system’s termination. The output of the results has only options to include, namely

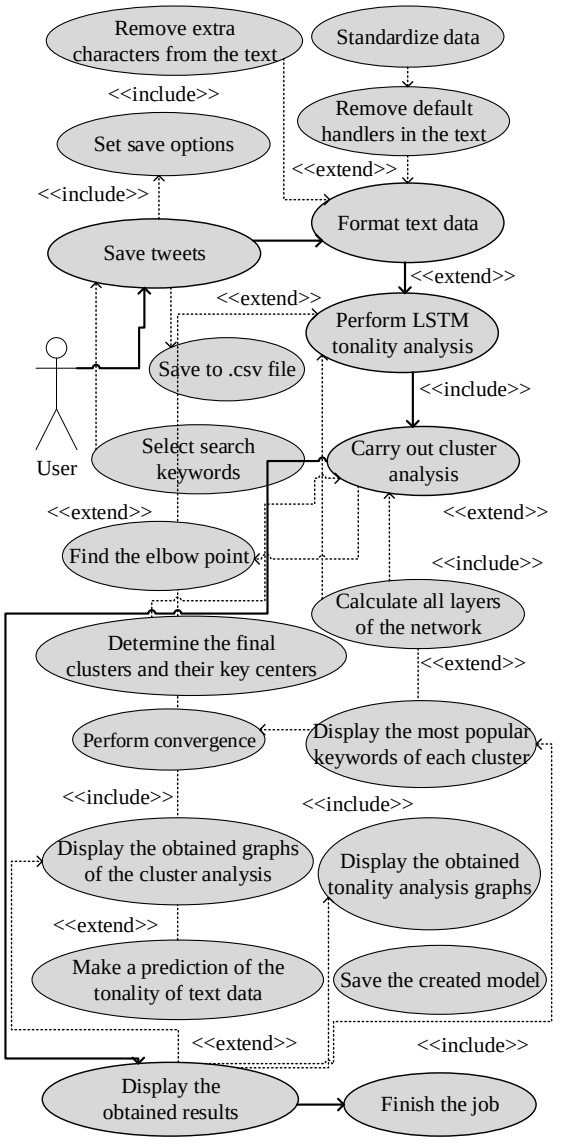


Figure 4 – Diagram of options for using the intelligent system of sentiment analysis and clustering of publications

DOI 10.15588/1607-3274-2024-1-16

the output of the tone of user posts and comments and the output of the cluster analysis in the form of textual information and using charts for a more detailed explanation of the obtained data according to the search keywords.

Having described the general structure of the intelligent system with the help of a diagram of use cases, it is necessary to more precisely describe the design of the created system and its functionality, for this the activity diagram is ideal, with its help, you can highlight the entities of the system with the help of tracks, the conditions for performing calculations and calculations, clearly show the state of operation and its interaction with other states within the initial and final state of the intelligent system of sentiment analysis and clustering of posts in the Twitter social network. It is also convenient to display all existing branching of flows during work and their results. To begin with, it is worth highlighting two main entities, namely “User” and “Server”, on the activity diagram they are presented in the form of separate tracks, accordingly, it is possible to describe their context not only in space but also in time, that is, to understand what processes in the intelligent system will occur at a certain point in time and how they will interact with each other. The first is the initial state, which describes the functional start of the system, then the states of actions belonging to the user of the system interact sequentially with each other, i.e. entering save parameters, setting keywords and other actions, or states of actions that are completely encapsulated within the logic of a separate instance class of task execution. The constructed activity diagram is shown in Figure 5.

Each state of action can be considered independently of others. Also, the “Server” entity in the work process checks the condition for the admissibility of saving the created model or the need to implement the process of reverse propagation of the error. In addition to conditions, the “Server” entity contains parallelization of action states, where at the same time, during the execution of two different predicates, the change of crucial centers can be divided into both the measurement of distances between clusters and the implementation of convergence with already existing clusters as a result of the calculation of the centroid of the cluster containing divided indices observation in an intelligent system. So, after describing the functionality of the system using object-oriented diagrams, for a better understanding of the context of the system’s work, it is necessary to describe its main elements and relationships using a functional diagram, the data flow diagram is the most convenient for performing this task, as it additionally shows the system, as a set of processes that interact with each other during the entire life cycle of the system. The data flow diagram of the intelligent system is shown in Figure 6.

The diagram consists of 7 consecutive data flows, where each flows into the next and transmits certain information within the request. Each thread fulfills all the requirements for transactions, that is, the intelligent system transmits data without the risk of its loss: if the transaction does not take place or information is lost or distorted in the process, the transaction will simply stop the current thread and return control to the previous thread, which generally ended without distorting existing user information. Also, under the number 1 and the name R on the diagram are data stores, in our case, it is a text file with the extension .csv for saving the dataset and a server for dynamically saving data between requests. Based on the completed description of the main essence of the system being created, its functionality, the LSTM neural network implementation, and the *k*-means clustering algorithm, an experiment was conducted to create a complete dataset of publications and comments on the social network Twitter, according to the keywords specified by the user, and an analysis of sentiment was carried out individual messages and corresponding division into final clusters.

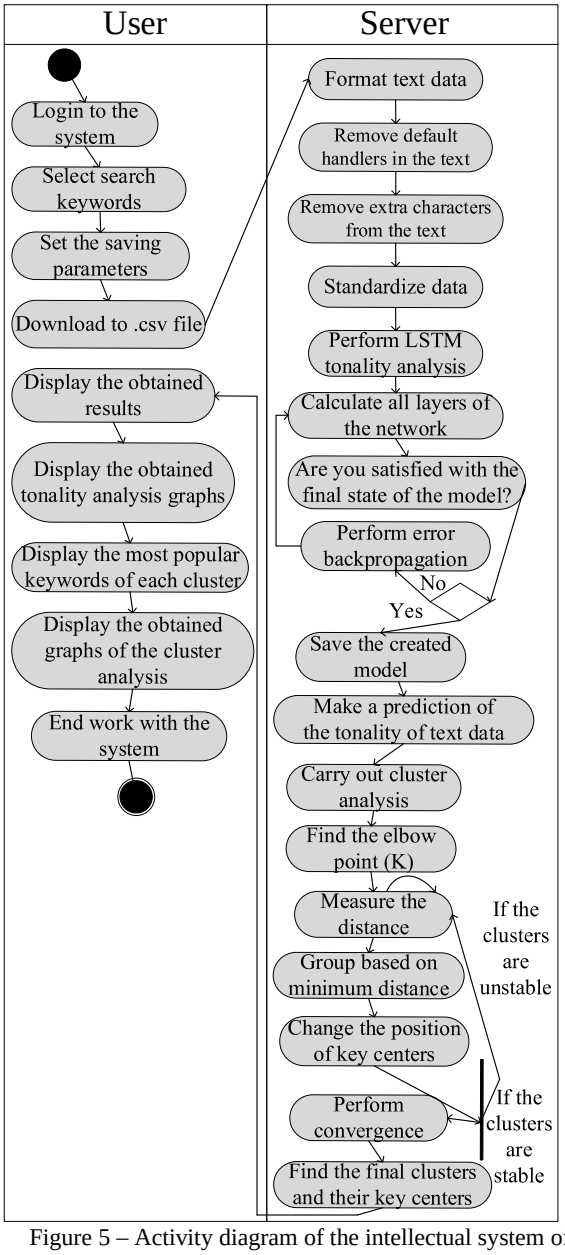


Figure 5 – Activity diagram of the intellectual system of sentiment analysis and publication clustering

DOI 10.15588/1607-3274-2024-1-16

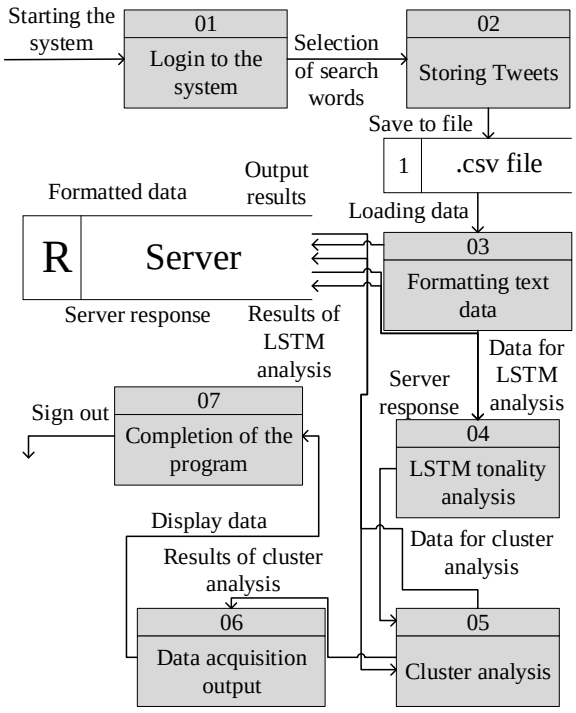


Figure 6 – Diagram of data flows of the intelligent system of sentiment analysis and clustering of publications

5 RESULTS

An experiment was conducted on applying the implemented intelligent system of sentiment analysis and clustering of posts in the social network Twitter, in which the work of the created LSTM neural network and the *k*-means clustering algorithm was tested. First, you need to find data to analyze. For maximum relevance, it was decided to make a separate dataset and download all user comments and posts on the Twitter social network over the past few months. Before downloading data, it was necessary to register a Twitter developer account and obtain four keys: consumer key, consumer secret key, access token, and access token secret. All of them are required to use the official Twitter API to save posts.

Since Russia attacked Ukraine in February 2022, it became an event that caused a corresponding reaction, both in Ukraine and abroad. People on social media, including Twitter, have been actively discussing the war and its related topics since February. Therefore, it was decided to analyze the tone of publications and comments of Ukrainian users regarding the war. The search was carried out using the geo-tag “Ukraine” and tweets were searched for the past six months, starting from July 2022, for posts and comments containing the mandatory keywords “Ukraine” and “war”, as well as the optional words “missile”, “offensive” and “invasion”. Given the API limit for downloading post instances, it was decided to download 2,000 tweets matching the specified keywords. To form the dataset, only specific parameters of downloaded tweets are needed, accordingly, those that are not used must be discarded by looping through all downloaded tweets and saving the user’s nickname, profile description, number of tweets, number of subscriptions, and followers, and most importantly, the text of the publication or comment All data is initially stored in the data dictionary and converted to the required format using the DataFrame function for data processing.

Considering the peculiarity of the created intelligent system, it was decided that the most convenient format for working with text data of this type using a neural network and a clustering algorithm will be the .csv format since it is universal and almost every programming language or data processing tool has functionality for working with files of this type. So, a dataset was created, which was processed during the training of the LSTM model of the neural network, and data was distributed into clusters. Before that, it became necessary to process all text data, that is, to bring them to one form for the most efficient processing.

Since we have a ready-made dataset, we can perform sentiment analysis, but before that, we need to format and standardize all the text so it can be conveniently analyzed. All hashtags were removed from the text, as they distorted the post’s content or comment. All words have been reduced to lowercase, all URL links have been removed, and all special characters not needed to determine the text’s tone have been drawn. In the end, all unnecessary spaces and single characters in the text were removed, and all system characters that were added utilizing the Twitter social network were removed, after which an LSTM analysis of the sentiment of the messages was performed, an example of the results of which is shown in Figure 7.

	text	Sentiment Score	\ Overall Sentiment
0	rt mtracey this woman literally works forus go...	0.000000	Neutral
1	rt smelyansky_igor 6000 branches of ukrposhta ...	0.000000	Neutral
2	snekotron fine with me we all knew they were c...	0.208333	Positive
3	rt euromaidanpress vilnius lithuania protest i...	0.000000	Neutral
4	rt andrew__roth putin is losing the war facing...	0.000000	Neutral

Figure 7 – An example of the results of determining the sentiment of the text by means of LSTM analysis

In the course of the work, the value of not only the general sentiment of words but also the polarity of the emotion was determined quite precisely, that is, the user can express a certain emotion with different strength and intensity according to the situation and the written publication or comment, accordingly, we determined the sentiment of the text instance of the data, as well as the emotional range, is from –1.0 – the most negative text, to 1.0 – the most positive, everything in between – the text with a certain sentiment and scope, there is also a value of 0, which means an average neutral text. The analysis of the text was carried out according to the previously described algorithm of memorizing certain moments in time from the test sample on which the neural network was trained. To begin with, we determine the language in which the text message was written, for convenience, only tweets in English were selected, after which two functions, spellcheck and correct, are performed, where we check the correctness of the written words using basic algorithms for working with text. Next, according to the algorithm, we perform word systematization, that is, using the definitions method, we obtain a list of possible meanings for the word according to the trained model and

DOI 10.15588/1607-3274-2024-1-16

choose the closest value. A neural network containing two levels of 100 elements, i.e., 50 elements per level, was generated. We set the value of spatial descent, take 500 epochs for the correct processing of 2000 messages, 0.5 is the value of spatial descent, and, accordingly, we set a similar value of 0.5 for gradient descent. Added a density value of 1 and a sigmoid activation function for the LSTM algorithm and correct handling of memory cell states. As a result, our recurrent model consists of an embedded layer, an LSTM layer, and a density layer, in which the sigmoid function is responsible for the native activation process. The training is carried out with a training batch size of 20 elements and a distribution factor of 0.2. The next step was to extract the level of emotionality of the sentiment of the text for a more comprehensive analysis of the text sample, but first, a general experiment was carried out on a test sample of 2000 words.

The model continues to run within 500 epochs, but instead of one block of memory, we have three different ones, called functional cells. Each cell has a certain state that can change according to the learning process and transfer the model's state. Accordingly, there is a hidden state that is unique to each cell and is not accessible from the others, and a distributed layer that is common to all three cells and, accordingly, can be changed by choosing the optimal value, which is carried out by selecting the maximum value at each of the steps of learning the subjective models. We will carry out not only a general analysis of sentiment but also an analysis of emotional, subjective sentiment in the range from –1 to 1. In this way, it is possible to understand how strongly the Twitter social network users expressed positive or negative emotions within the limits of the publication or comment, which is a reaction to military actions. Such an uneven distribution can be explained by the fact that people react to good news during the war much more emotionally and impulsively, rejoicing in something positive, as can be seen on the graph, some of the comments and publications reach 0.8 emotionality value and, on the contrary, negative comments written by people after several months of war much less emotional. A graph of the subjectivity of sentiment is shown in Figure 8. From this graph we can see the highest density of positive and negative emotions in the text instances has the same value of 0.5.

Figure 8 – Subjectivity of sentiment

So, after analyzing the sentiment of user posts and comments, we saw how many positive, negative, and neutral tweets were written during a specific period using certain thematic keywords and the distribution between their objectivity. Next, clustering was carried out by the *k*-means method to see the distribution of text messages of different tonalities by clusters. The clustering process consists of 3 main steps: initialization of the model, adjustment of model fit parameters, and forecast for further processing and division into clusters. To begin with, n was set – the number of clusters, which can be from 3 to 5, according to the number of text elements, and random state – the state of the clustering process, which is a random number necessary for the initialization of the data model.

After determining the number of clusters, it is time to distribute the text instances by these clusters, since there are only 3 of them, it is convenient to divide them into positive, neutral, and negative. To do this, the groupby.mean function was used, the essence of which is to determine the average value for each cluster. Next, standardization is carried out, for which it is necessary to subtract the cluster value from the sample mean and divide it by the mean deviation, after which the mean value is removed, and the unit variance is scaled. The distribution of user text data by clusters is depicted in Figures 9 and 10 using different graphs. Therefore, the PCA plot reflects the general structure of the data, and the TSNE plot best demonstrates the relationship between neighbors. For a more convenient display of data, 40 text instances were displayed on the PCA graph instead of all possible objects, and 400 text instances were displayed on the TSNE.

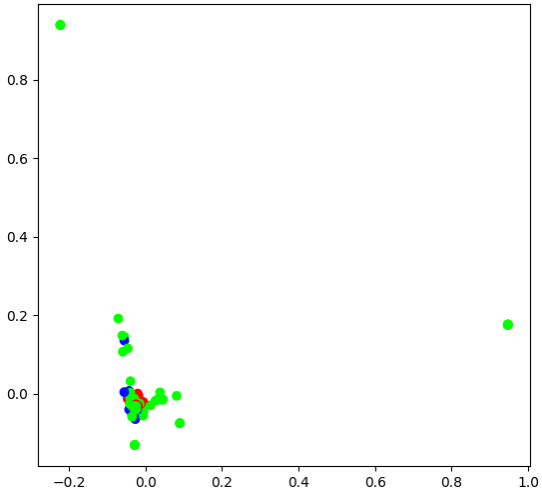


Figure 9 – Distribution graph of PCA type clusters

In the end, the most popular keywords for each cluster were extracted, from which certain conclusions can be drawn or trends can be observed, for this, the average value was calculated for all dimensions of the created model, which were grouped in each cluster. The next step was sorting the arrays of average values of each cluster in descending order and selecting the first ten elements,

DOI 10.15588/1607-3274-2024-1-16

shown in Figure 11, where zero cluster contain negative user posts and comments, the first cluster – positive, and the second – neutral.

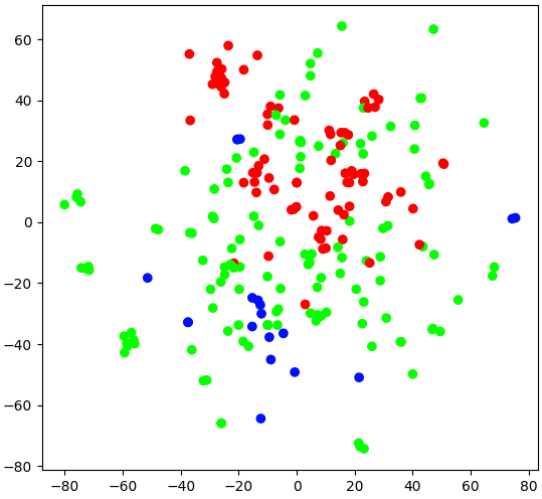


Figure 10 – Distribution graph of TSNE-type clusters

```
Cluster 0
one,country,nato,stop,people,putin,rt,russia,war,ukraine

Cluster 1
apmassaro3,attack,supporter,russia,civilian,today,russian,military,ukraine,rt

Cluster 2
monday,ukraine,oct,10,defense,pic,russian,air,rt,missile
```

Figure 11 – The most popular keywords

6 DISCUSSION

After experimenting, it can be concluded that the used combination of algorithms, namely the LSTM sentiment analysis neural network and the *k*-means clustering algorithm, works effectively and allow more accurate analysis of datasets than their counterparts, especially when the datasets reach large sizes and it is necessary to accurately analyze the sentiment of the text together with the level of emotionality and effectively distribute the number of clusters required for further research. Figure 12 shows a comparison of the work of combinations of the CNN neural network and the hierarchical clustering algorithm, which have the complexity of the $O(n^2)$ algorithm, and the LSTM neural network and the *k*-means clustering algorithm created in the intelligent system, which have the complexity of the $O(n\log n)$ algorithm, respectively, the implemented algorithms in an intelligent system, they work at least 10...15% faster and more efficiently than the previously described convolutional neural network and hierarchical clustering, and with large volumes of data they can work up to 20% better.

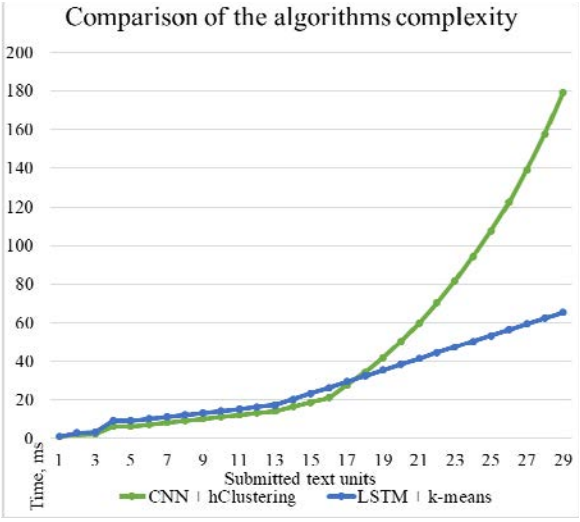


Figure 12 – Comparison of the complexity of algorithms

CONCLUSIONS

During the work, an intelligent system of sentiment analysis and clustering of posts in the Twitter social network was implemented, with the help of which the user can enter certain parameters and keywords that will be used to download posts and comments from the Twitter social network for analyzing the sentiment of messages, their emotional evaluation and division into clusters. To begin with, the importance of creating such an intelligent system was described, and the analysis of recent research and literature sources to understand what has already been implemented and what still needs to be implemented and, accordingly, repeat the advantages and avoid the disadvantages. The application of neural networks of various directions and clustering algorithms was mainly considered. The purpose of the work was described, and it was explained why implementing a modern LSTM neural network, and the *k*-means clustering algorithm is an urgent task to increase the system's efficiency and improve the accuracy of the final results. The functionality of the system was also described, namely the essence of the work of its main algorithms and the general implementation of the main functional components, and presented with the help of diagrams that depict both the block diagrams of the algorithms and the options for using and interacting with all the created data flows of the intelligent system as a whole.

In the course of the work, an actual experiment was carried out using the created intelligent system, during which keywords and search parameters for publications and comments in the Twitter social network were set, and a set of tweets was downloaded, which was saved and formed into a dataset. After that, formatting, lemmatization, and standardization of text instances were carried out for the correct operation of the algorithms. Concerning this text, a Long Short-Term Memory neural network was trained and used, with the help of which both a general sentiment analysis was carried out with the division of the text into negative, neutral, and positive, and an analysis of the emotional value of the text, i.e., the distribution from –1.0, i.e., the maximum possible negative to 1.0 – the maximum possible positive, and the value and subjectivity of sentiment are reflected. Similarly, a cluster analysis of the text was carried out using the *k*-means clustering algorithm, a model was created and trained, and the optimal number of clusters for the downloaded dataset was selected using the

DOI 10.15588/1607-3274-2024-1-16

"elbow" method. After dividing the text data into clusters, the formed clusters were graphically displayed using PCA and TSNE graphs. The average value was calculated for the current dimensions of the created model, and the ten most popular words in each cluster were displayed, the image of the text elements of each cluster was demonstrated using a three-dimensional graph, and the difference in efficiency and complexity between the CNN neural network and hierarchical clustering and the LSTM neural network was shown and *k*-means clustering algorithm implemented in the intelligent system.

ACKNOWLEDGEMENTS

The study was conducted within the framework of joint scientific research of the Department of Information Systems and Networks of the "Lviv Polytechnic" National University on the topic "Research, development and implementation of intelligent distributed information technologies and systems based on the resources of databases, data warehouses, data spaces and knowledge in order to accelerate the processes of formation modern information space using neural networks and data clustering algorithms". Scientific research was also carried out as part of the initiative research topic of the Department of Information Systems and Networks of the "Lviv Polytechnic" National University on the topic "Development of intelligent distributed systems based on the ontological dynamic system analysis approach for the purpose of integrating information resources as a data distribution services".

REFERENCES

1. Alhakiem H. R. Setiawan E. B. Aspect-Bas1ed Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion, *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 2022,Vol. 6, No. 5, pp. 840–846. DOI: 10.29207/resti.v6i5.4429
2. Huang Xiaoran et al. Machine Learning Modeling of Vitality Characteristics in Historical Preservation Zones with Multi-Source Data, *Buildings*, 2022, Vol. 12, No. 11, P. 1978. DOI: 10.3390/buildings12111978
3. Lampropoulos G., Keramopoulos E. Virtual Reality in Education: A Comparative Social Media Data and Sentiment Analysis Study, *International Journal of Recent Contributions from Engineering, Science & IT (iJES)*, 2022, Vol. 10, No. 03, p. 19–32. DOI: 10.3991/ijes.v10i03.34057
4. Wang Y., Chen Zh., Fu Ch. Synergy Masks of Domain Attribute Model DaBERT: Emotional Tracking on Time-Varying Virtual Space Communication, *Sensors*, 2022, Vol. 22, No. 21, P. 8450. DOI: 10.3390/s22218450
5. Karyukin Vl. et al. On the development of an information system for monitoring user opinion and its role for the public, *Journal of Big Data*, 2022, Vol. 9, No. 1. DOI: 10.1186/s40537-022-00660-w
6. Albahli Saleh et al. A Machine Learning Method for Prediction of Stock Market Using Real-Time Twitter Data, *Electronics*, 2022, Vol. 11, No. 20, P. 3414. DOI: 10.3390/electronics11203414
7. Li Qizhi et al. A New Sentiment-Enhanced Word Embedding Method for Sentiment Analysis, *Applied Sciences*, 2022, Vol. 12, No. 20, P. 10236. DOI: 10.3390/app122010236
8. Abulhaija Sabreen et al. Mobile Applications Rating Performance: A Survey, *International Journal of Interactive Mobile Technologies (iJIM)*, 2022, Vol. 16, No. 19, pp. 133–146. DOI: 10.3991/ijim.v16i19.32051
9. Kamath A. N., Shenoy S. S., Kumar N. S. An overview of investor sentiment: Identifying themes, trends, and future direction through bibliometric analysis, *Investment Management and Financial Innovations*, 2022, Vol. 19, No. 3, pp. 229–242. DOI: 10.21511/imfi.19(3).2022.19
10. Zhang M. et al. Intelligent Vehicle Sales Prediction Based on Online Public Opinion and Online Search, *Sustainability*, 2022, Vol. 14, No. 16, P. 10344. DOI: 10.3390/su141610344
11. Pawełoszek I. Towards a Smart City – The Study of Car-Sharing Services in Poland, *Energies*, 2022, Vol. 15, No. 22, P. 8459. DOI: 10.3390/en15228459
12. Li Chih-yuan et al. Public Health Policy Monitoring through Public Perceptions: A Case of COVID-19 Tweet Analysis, *Information*, 2022, Vol. 13, No. 11, P. 543. DOI: 10.3390/info13110543
13. Corti L. et al. Social media analysis of Twitter tweets related to ASD in 2019–2020, with particular attention to COVID-19: topic modelling and sentiment analysis, *Journal of Big Data*, 2022, Vol. 9, No. 1. DOI: 10.1186/s40537-022-00666-4
14. Liu H. Online review analysis on various networks' consumer feedback using deep learning, *IET Networks*, 2022, Vol. 11, pp. 234–244. DOI: 10.1049/ntw2.12045
15. Wang Yili et al. Sentiment Analysis of Twitter Data, *Applied Sciences*, 2022, Vol. 12, No. 22, P. 11775. DOI: 10.3390/app122211775
16. Kim D., Kim Y.-J., Jeong Y.-S. Graph Convolutional Networks with POS Gate for Aspect-Based Sentiment Analysis, *Applied Sciences*, 2022, Vol. 12, No. 19, P. 10134. DOI: 10.3390/app121910134
17. Opiła J. On Employing of Extended Characteristic Surface Model for Forecasting of Demand in Tourism, *Interdisciplinary Description of Complex Systems*, 2022, Vol. 20, No. 5, pp. 621–639. DOI: 10.7906/indecs.20.5.8
18. Bagate R. A., Suguna R. Sarcasm detection of tweets without #sarcasm: data science approach, *Indonesian Journal of Electrical Engineering and Computer Science*, 2021, Vol. 23, No. 2, P. 993. DOI: 10.11591/ijeecs.v23.i2.pp993-1001
19. Asgari T. et al. Identifying key success factors for startups With sentiment analysis using text data mining, *International Journal of Engineering Business Management*, 2022, Vol. 14, pp. 435–453. DOI: 10.1177/18479790221131612
20. Kowalska-Styczeń A. et al. Game problem of assigning staff to project implementation, *Decision Making: Applications in Management and Engineering*, 2023, Vol. 6, No. 2, pp. 691–721. DOI: 10.31181/dmame622023713
21. Lytvyn V. et al. Identification and Correction of Grammatical Errors in Ukrainian Texts Based on Machine Learning Technology, *Mathematics*, 2023,Vol. 11, No. 4, P. 904. DOI: 10.3390/math11040904
22. Prokipchuk O. et al. Intelligent Analysis of Ukrainian-language Tweets for Public Opinion Research based on NLP Methods and Machine Learning Technology, *International Journal of Modern Education and Computer Science*, 2023, Vol. 15, No. 3, pp. 70–93. DOI: 10.5815/ijmecs.2023.03.06

Accepted 04.01.2024.
Received 22.02.2024.

УДК 004.9

РЕАЛІЗАЦІЯ СИСТЕМИ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ АНАЛІЗУ ПУБЛІКАЦІЙ КОРИСТУВАЧІВ TWITTER

Батюк Т. М. – аспірант кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

Досин Д. Г. – д-р техн. наук, старший науковий співробітник, професор кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

DOI 10.15588/1607-3274-2024-1-16

АНОТАЦІЯ

Актуальність. У статті наголошується на необхідності створення системи прийняття рішень, яка може аналізувати повідомлення користувачів і визначати настрої, щоб зрозуміти, як новини та події впливають на емоції людей. Така система використовуватиме передові методи для аналізу повідомлень користувачів, заглиблюючись у почуття, виражені в тексті. Основна мета – отримати уявлення про те, як новини та різноманітні події відбиваються на емоціях людей.

Метою дослідження є створення системи прийняття рішень, яка зможе аналізувати та визначати настрої повідомлень користувачів, розуміти емоційну реакцію на новини та події та розподіляти дані в кластери, щоб отримати ширше розуміння думок користувачів. Ця багатогранна мета передбачає інтеграцію передових методів обробки природної мови та машинного навчання для створення надійної системи прийняття рішень. Основними цілями є аналіз настроїв, розуміння емоційних реакцій на новини та події та кластеризація даних для цілісного уявлення про думки користувачів.

Метод. Для обробки великих обсягів даних користувача пропонується використання нейронних мереж довгострокової пам'яті для аналізу настрою та алгоритму *k*-середніх для кластеризації даних. Ця стратегічна комбінація спрямована на вирішення проблем, пов'язаних із обробкою великих обсягів даних, створених користувачами, у більш глибокий та цілеспрямований спосіб.

Результати. Виконано дослідження та концептуальне проектування системи прийняття рішень та створено систему прийняття рішень. Система включає аналіз настроїв і кластеризацію даних для розуміння думок користувачів і цінності настроїв таких думок, розділяючи їх на кластери та візуалізуючи результати.

Висновки. Розробка системи прийняття рішень, здатної аналізувати настрої користувачів і кластеризувати дані, може надати цінну інформацію про реакцію користувачів на новини та події в соціальних мережах. Запропоноване використання нейронних мереж довгострокової пам'яті та алгоритму *k*-середніх вважається придатним для аналізу настроїв і завдань кластеризації даних. Підкреслюється важливість вивчення існуючих робіт і систем для розуміння доступних алгоритмів і їх застосування. У статті також описано створену та впроваджену систему прийняття рішень та продемонстровано функціональність системи на прикладі набору даних.

КЛЮЧОВІ СЛОВА: обробка природної мови, згорткова нейронна мережа, рекурентна нейронна мережа, LSTM, кластеризація *k*-середніх.

ЛІТЕРАТУРА

1. Alhakiem H. R. Aspect-Bas1ed Sentiment Analysis on Twitter Using Logistic Regression with FastText Feature Expansion / Hanif Reangga Alhakiem, Erwin Budi Setiawan // Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi). – 2022. – Vol. 6, No. 5. – P. 840–846. DOI: 10.29207/resti.v6i5.4429
2. Machine Learning Modeling of Vitality Characteristics in Historical Preservation Zones with Multi-Source Data / [Xiaoran Huang et al.] // Buildings. – 2022. – Vol. 12, No. 11. – P. 1978. DOI: 10.3390/buildings12111978
3. Lampropoulos G. Virtual Reality in Education: A Comparative Social Media Data and Sentiment Analysis Study / Georgios Lampropoulos, Euclid Keramopoulos // International Journal of Recent Contributions from Engineering, Science & IT (iJES). – 2022. – Vol. 10, No. 03. – P. 19–32. DOI: 10.3991/ijes.v10i03.34057
4. Wang Y. Synergy Masks of Domain Attribute Model DaBERT: Emotional Tracking on Time-Varying Virtual Space Communication / Ye Wang, Zhenghan Chen, Changzeng Fu // Sensors. – 2022. – Vol. 22, No. 21. – P. 8450. DOI: 10.3390/s22218450
5. On the development of an information system for monitoring user opinion and its role for the public / [Vladislav Karyukin et al.] // Journal of Big Data. – 2022. – Vol. 9, No. 1. DOI: 10.1186/s40537-022-00660-w
6. A Machine Learning Method for Prediction of Stock Market Using Real-Time Twitter Data / [Saleh Albahli et al.] // Electronics. – 2022. – Vol. 11, No. 20. – P. 3414. DOI: 10.3390/electronics11203414
7. A New Sentiment-Enhanced Word Embedding Method for Sentiment Analysis / [Qizhi Li et al.] // Applied Sciences. – 2022. – Vol. 12, No. 20. – P. 10236. DOI: 10.3390/app122010236
8. Mobile Applications Rating Performance: A Survey / [Sabreen Abulhaija et al.] // International Journal of Interactive Mobile Technologies (iJIM). – 2022. – Vol. 16, No. 19. – P. 133–146. DOI: 10.3991/ijim.v16i19.32051
9. N Kamath A. An overview of investor sentiment: Identifying themes, trends, and future direction through bibliometric analysis / Aditi N Kamath, Sandeep S. Shenoy, Subrahmanya Kumar N. // Investment Management and Financial Innovations. – 2022. – Vol. 19, No. 3. – P. 229–242. DOI: 10.21511/imfi.19(3).2022.19
10. Intelligent Vehicle Sales Prediction Based on Online Public Opinion and Online Search / [Mingyang Zhang et al.] // Sustainability. – 2022. – Vol. 14, No. 16. – P. 10344. DOI: 10.3390/su141610344
11. Pawełoszek I. Towards a Smart City–The Study of Car-Sharing Services in Poland / Ilona Pawełoszek // Energies. – 2022. – Vol. 15, No. 22. – P. 8459. DOI: 10.3390/en15228459
12. Public Health Policy Monitoring through Public Perceptions: A Case of COVID-19 Tweet Analysis / [Chih-yuan Li et al.] // Information. – 2022. – Vol. 13, No. 11. – P. 543. DOI: 10.3390/info13110543
13. Social media analysis of Twitter tweets related to ASD in 2019–2020, with particular attention to COVID-19: topic modelling and sentiment analysis / [Luca Corti et al.] // Journal of Big Data. – 2022. – Vol. 9, No. 1. DOI: 10.1186/s40537-022-00666-4
14. Liu H. Online review analysis on various networks' consumer feedback using deep learning / Huajin Liu // IET Networks. – 2022. – Vol. 11. – P. 234–244. DOI: 10.1049/ntw2.12045
15. Sentiment Analysis of Twitter Data / [Yili Wang et al.] // Applied Sciences. – 2022. – Vol. 12, No. 22. – P. 11775. DOI: 10.3390/app122211775
16. Kim D. Graph Convolutional Networks with POS Gate for Aspect-Based Sentiment Analysis / Dahye Kim, YoungJin Kim, Young-Seob Jeong // Applied Sciences. – 2022. – Vol. 12, No. 19. – P. 10134. DOI: 10.3390/app121910134
17. Opiła J. On Employing of Extended Characteristic Surface Model for Forecasting of Demand in Tourism / Janusz Opiła // Interdisciplinary Description of Complex Systems. – 2022. – Vol. 20, No. 5. – P. 621–639. DOI: 10.7906/indecs.20.5.8
18. Bagate R. A. Sarcasm detection of tweets without #sarcasm: data science approach / Rupali Amit Bagate, R. Suguna // Indonesian Journal of Electrical Engineering and Computer Science. – 2021. – Vol. 23, No. 2. – P. 993. DOI: 10.11591/ijeecs.v23.i2.pp993-1001
19. Identifying key success factors for startups With sentiment analysis using text data mining / [Tina Asgari et al.] // International Journal of Engineering Business Management. – 2022. – Vol. 14. – P. 435–453. DOI: 10.1177/18479790221131612
20. Game problem of assigning staff to project implementation / [Agnieszka Kowalska-Styczeń et al.] // Decision Making: Applications in Management and Engineering. – 2023. – Vol. 6, No. 2. – P. 691–721. DOI: 10.31181/dmame622023713
21. Identification and Correction of Grammatical Errors in Ukrainian Texts Based on Machine Learning Technology / Vasyl Lytvyn [et al.] // Mathematics. – 2023. – Vol. 11, No. 4. – P. 904. DOI: 10.3390/math11040904
22. Intelligent Analysis of Ukrainian-language Tweets for Public Opinion Research based on NLP Methods and Machine Learning Technology / [Oleh Prokipchuk et al.] // International Journal of Modern Education and Computer Science. – 2023. – Vol. 15, No. 3. – P. 70–93. DOI: 10.5815/ijmecs.2023.03.06

DOI 10.15588/1607-3274-2024-1-16

p-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2024. № 1
e-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2024. № 1

UDC 519.172.1: 62-503.57

METHOD OF CREATING A MINIMAL SPANNING TREE ON AN ARBITRARY SUBSET OF VERTICES OF A WEIGHTED UNDIRECTED GRAPH

Batsamut V. M. – Dr. Sc., Professor, The Deputy Head of the Scientific Research Center of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

Hodlevsky S. O. – The Researcher of the Scientific Research Center of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

Babkov Yu. P. – PhD, Associated Professor, Professor of the State Security Department of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

Morkvin D. A. – PhD, Head of the Scientific Research Laboratory of the Scientific Research Center of the National Academy of the National Guard of Ukraine, Kharkiv, Ukraine.

ABSTRACT

Context. The relevance of the article is determined by the need for further development of models for optimal restoration of the connectivity of network objects that have undergone fragmentation due to emergency situations of various origins. The method proposed in this article solves the problematic situation of minimizing the amount of restoration work (total financial costs) when promptly restoring the connectivity of a selected subset of elements of a network object after its fragmentation.

The purpose of the study is to develop a method for creating a minimal spanning tree on an arbitrary subset of vertices of a weighted undirected graph to minimize the amount of restoration work and/or total financial costs when promptly restoring the connectivity of elements that have a higher level of importance in the structure of a fragmented network object.

Method. The developed method is based on the idea of searching for local minima in the structure of a model undirected graph using graph vertices that are not included in the list of base vertices to be united by a minimal spanning tree. When searching for local minima, the concept of an equilateral triangle and a radial structure in such a triangle is used. In this case, there are four types of substructures that provide local minima: first, those with one common base vertex; second, those with two common base vertices; third, those with three common base vertices; fourth, those without common base vertices, located in different parts of the model graph. Those vertices that are not included in the list of basic ones, but through which local minima are ensured, are added to the basic ones. Other vertices (non-basic) along with their incident edges are removed from the structure of the model graph. Then, using one of the well-known methods of forming spanning trees, a minimal spanning tree is formed on the structure obtained in this way, which combines the set of base vertices.

Results. 1) A method for creating a minimal spanning tree on an arbitrary subset of vertices of a weighted undirected graph has been developed. 2) A set of criteria for determining local minima in the structure of the model graph is proposed. 3) The method has been verified on test problems.

Conclusions. The theoretical studies and several experiments confirm the efficiency of the developed method. The solutions developed using the developed method are accurate, which makes it possible to recommend it for practical use in determining strategies for restoring the connectivity of fragmented network objects.

KEYWORDS: network object, weighted undirected graph, connectivity, transitive closure, minimum spanning tree, local optimum, optimization criterion, method.

ABBREVIATIONS

MST is a minimal spanning tree;

TC is a transitive closure.

NOMENCLATURE

G is an undirected weighted graph modeling a network object;

V is a set of vertices of the model graph G;

E is a set of edges of a model graph G;

S_G is a set of edges of a model graph G;

R_G is a matrix of shortest paths of the model graph G;

(u,v) is the graph edge G;

$w(u,v)$ is a weighting coefficient of some edge (u,v);

K is an arbitrarily selected subset of vertices of the model graph G;

$G'_{[K]}$ is an MST, which is created on an arbitrarily selected subset of vertices of the model graph G;

E' is a set of edges that make up the required graph $G'_{[K]}$;

W is a total weight of the constructed tree;

w_{ij} is a weight of the TC between the corresponding vertices of the model graph G;

v_i is a vertex of the model graph G;

n is a number of graph vertices.

INTRODUCTION

Objects with a distributed structure, so-called network objects, have long ago and forever entered the life of mankind. Such facilities include transport networks (road, rail, water transport, air transport); data transmission networks; power grids, water supply networks, gas supply networks and others. A distinctive feature of such objects is the presence in their composition of nodal elements (passenger stations, communication nodes, distribution

DOI 10.15588/1607-3274-2024-1-17

and pumping stations, producers of services and their consumers, etc.) and communication lines between these nodal elements (transport routes, cable facilities, power lines, etc.).

The efficiency of such complex objects depends on the performance of their individual elements, but this dependence is more pronounced on communication lines, since the latter have larger linear dimensions and, therefore, are more often and more exposed to external undesirable influences [1, 2, 3, 4]. The causes of external influences include man-made and natural emergencies. Network facilities undergo particularly significant fragmentation (destruction) because of military (combat) operations in the territories where such facilities are located [5].

As a rule, after the situation in the crisis area is normalized, and in many cases during the emergency response, the issue of conducting restoration work on the destroyed network facility arises to bring the structure of the facility and all its functioning parameters to the design parameters.

In conditions of limited funding or limited time for restoration work, the network facility is usually restored to the spanning structure first [6] as the first stage of restoration work. In such a structure, each nodal element is connected to any other nodal element, although not always by optimal routes. The facility continues to operate, although with some loss of quality of service to end users (subscribers). At the second stage, based on the spanning structure, the network object is restored to its original structure, and even improved, to make it more efficient.

If the nodal elements differ in their degree of importance, the first stage of restoration work involves restoring the connectivity of not all elements of the network object, but the more significant ones selected according to a certain rule (criterion). It should be noted here that when determining the set of such elements, the current situation at the facility due to its fragmentation and the functional purpose of the facility itself are also considered.

Using well-known methods for constructing minimal spanning trees (MSTs) on graphs, such as Prima [7], Kruskala [8], Boravki-Solina [9], and others [10, 11, 12], it is generally possible to construct a spanning tree on an arbitrary subset of the vertices of the initial undirected graph, but in most cases such a tree will not be optimal in terms of the minimum total weight of the weighted edges it is composed of. On network objects with a significant number of node elements and a significant density of communication lines, such an error can be significant and decisive in matters of choosing a strategy for restoring the connectivity of a fragmented network object.

Thus, the article is aimed at minimizing the amount of restoration work (financial costs for such work) aimed at promptly restoring the connectivity of a certain (prescribed) set of node elements of a fragmented network object.

The object of the study is the process of restoring connectivity between an arbitrary subset of node elements of a fragmented network object.

The subject of the study is the method for creating a minimal spanning tree on an arbitrary subset of vertices of a weighted undirected graph.

The purpose of the study is to develop a method for creating a minimal spanning tree on an arbitrary subset of vertices of a weighted undirected graph to minimize the amount of restoration work and/or total financial costs when promptly restoring the connectivity of elements that have a higher level of importance in the structure of a fragmented network object.

1 PROBLEM STATEMENT

The tasks of determining the optimal structures of network objects are, for the most part, formalized and solved using graph theory models and methods [13]. That is why we will model the structure of the network object with some weighted undirected graph $G=(V,E)$, where V is a set containing the vertices of a graph that model the node elements of a network object; E is the set containing the edges of a graph that model the communication lines of a network object.

For each edge $(u,v)\in E$, its weight is known $w(u,v)$. In the plural V the vertices of the initial graph are an arbitrarily selected subset of vertices K, so that $K\subseteq V$, $|K|<|V|$. Vertices that make up a subset K, will be called basic. The task is to create MST $G'_{[K]}=\left(K,E'\subseteq E\right)$, connecting a selected subset of base vertices K, namely:

$$w(E')=\sum_{(u,v)\in E'} w(u,v)\to \min, \qquad (1)$$

under the conditions:

$$\forall < x,y > \in K\ \exists\ x \xrightarrow{\text{TC}} y, \qquad (2)$$

where $< x,y >$ – an arbitrary pair of vertices from the set K; $x \xrightarrow{\text{TC}} y$ – transitive closure between an arbitrary pair of vertices $< x,y >$.

2 REVIEW OF THE LITERATURE

Currently, the theoretical basis for restoring the connectivity of fragmented (broken) network objects is the graph theory.

A well-known and studied problem of graph theory with numerical practical applications is the problem of creating an initial undirected MST graph on the structure, that is, an acyclic subgraph in which all vertices of the initial graph are transitively closed (there is a path connecting any pair of vertices), and the total weight of the edges of this acyclic subgraph is minimal.

DOI 10.15588/1607-3274-2024-1-17

Currently under creating MST $G'_{[+]}$, where "+" – the entire nodal basis of the graph G, as it was mentioned above, the well-known methods of Prim, Kruskal, Boruvka-Solin are used. These methods can also be used to search for minimal covering trees $G'_{[K]}$, by checking at each step whether the tree being built has connectivity between all vertices $v_i \in K$.

Thus, a connected undirected graph is applied to the input of Prim's algorithm [7]. For each edge, its cost is set. First, an arbitrary vertex is taken and the edge incident to this vertices, which has the lowest cost, is found. The found edge and two vertices connected by it form a tree. Then the edges of the graph are considered, one end of which is a vertex that already belongs to the tree, and the other is not; from these edges, the edge of the lowest cost is selected. The edge selected at each step is joined to the tree. The tree grows until all the vertices of the initial graph are explored. The result of the algorithm is the MST.

If the initial graph is given by the adjacency matrix, the computational complexity of this algorithm is estimated $O(n^2)$.

In Kruskal's algorithm [8], the current set of edges is initially set to be empty. All the edges of the graph are ordered as the weight increases and are presented in a separate list. An edge of minimum weight is selected from the list and added to the already existing set (the tree being created). A cycle check is performed immediately. If there is no cycle, then the next edge is taken and added to the set. If there is a cycle, the edge that created it is discarded. The process is iteratively repeated until all vertices of the initial graph are included into the required tree. The tree found in this way is the minimum spanning tree of the initial graph.

The computational complexity of this algorithm will be evaluated $O(E\log(E))$, and is mainly determined by the complexity of the process of sorting the edges of the graph.

The Boruvka-Solin algorithm [9] is practically no different from Kruskal's algorithm.

The conducted analysis of the literature shows that the problem in the formal statement (1)-(2) has not been posed or solved by anyone. Our article is dedicated to solving this problem.

3 MATERIALS AND METHODS

The analysis of the Prim, Kruskal, Boruvka-Solin methods on various structures proved that their use for creating trees $G'_{[K]}$, may give some error in the final result, because in the structure of the initial graph G spanning trees may exist $G'_{[K]}$ with less weight. The fact is that at each step of these methods, vertices are needed for transitive linking $v_i \in K$, an edge of minimum weight is added to the structure of the required tree, followed by a check for the presence of a cycle. The total weight of the added edges may exceed the weight of some edge, the weight of which is greater than each of the added ones, but through which the optimal (by the minimum weight criterion) transitive closure of the vertices is carried out $v_i \in K$.

For example, there is some communication network modeled by an undirected weighted graph G, Fig. 1, a. Minimal spanning tree $G'_{[1,\,4,\,5,\,6]}$, built according to the Kruskal's method, provided by Fig. 1, b bolder lines.

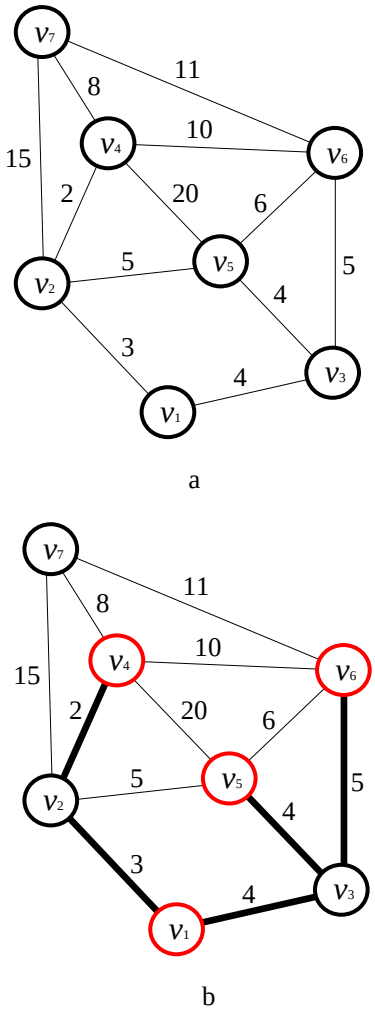


Figure 1 – Building a minimal spanning tree on an arbitrarily subset of vertices of the initial graph G:
a – the initial undirected graph G;
b – a minimal spanning tree $G'_{[1,4,5,6]}$ of the initial undirected graph G, built according to Kruskal's method

At the same time, the total weight of five edges (v_1,v_2), (v_2,v_4), (v_1,v_3), (v_3,v_5), (v_3,v_6), which are part of the spanning tree $G'_{[1,4,5,6]}$, is equal to $W = \sum_{(i,j)\in G'_{[1,4,5,6]}} w_{ij} = 18$.

But it can be seen that in fact MST $G'_{[1,4,5,6]}$ consists of four edges (v_1,v_2), (v_2,v_4), (v_2,v_5), (v_5,v_6).

DOI 10.15588/1607-3274-2024-1-17

Herewith $W = \sum_{(i,j)\in G'_{[1,4,5,6]}} w_{ij} = 16$, see Fig. 2. The absolute difference in the weights of these two trees is 2 units.

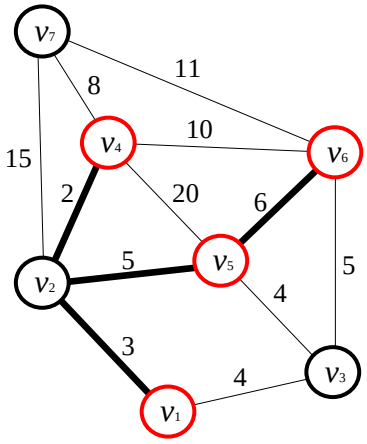


Figure 2 – A minimal spanning tree $G'_{[1,4,5,6]}$ of the initial undirected graph G

Considering the above, we will formulate and prove the following theorem.

Theorem. Let $G=(V,E)$ be an arbitrary weighted undirected graph. Minimal spanning tree $G'_{[K]}=(K,E')$ on a subset of selected vertices $v_i \in K$ of the graph G, where $K \subseteq V$, can be created by adding a subset to the composition K of some vertex $v_i \notin K$, if the optimal (of minimal weight) transitive closure (TC) of some vertices is carried out through it $v_i \in K$.

Proof: It is obvious that is a minimal spanning tree on a subset of vertices K, in case $|K|=2$ is a shortest path connecting these two vertices. If $|K|>2$, there can be several such paths. Thus, to obtain the connectivity of some vertices s, d, t to the structure of the required $G'_{[s,d,t]}$ can be added $(s,d_1),(d_1,d_2),...,(d_{n-1},d_n),(d_n,d)$ and $(s,t_1),(t_1,t_2),...,(t_{n-1},t_n),(t_n,t)$ edge. Suppose that in the structure of the initial graph G some vertices is present $t_n \notin K$ and edge (t_n,d) for which the following condition is true: $w_{t_n,d} > w_{s,d_1}$, $w_{t_n,d} > w_{d_1,d_2}$,..., $w_{t_n,d} > w_{d_n,d}$ and $w_{t_n,d} < (w_{s,d_1} + ... + w_{d_{n-1},d_n} + w_{d_n,d})$. So, considering the edge (t_n,d) it is possible to reduce the total weight of the required spanning tree $G'_{[s,d,t]}$.

Therefore, the adjacent edges whose weight coefficients are in parentheses can increase the total weight of $G'_{[s,d,t]}$. Thus, the problem should be solved taking into account the possible addition to the structure of the required tree $G'_{[K]}$ of additional vertices $v_i \notin K$, the total weight of the transitive closure through which will ensure the minimization of the total weight of the required spanning tree. The theorem is proved.

Let us note an important consequence of the theorem.

Consequence. The weight of transitive closure of vertices $v_i \in K$, can be reduced through some vertices $v_i \notin K$, starting from $|K|=3$.

Let's explain the mentioned consequence graphically. For example, there are two connected networks with lengths L_1 and L_2, see Fig. 3.

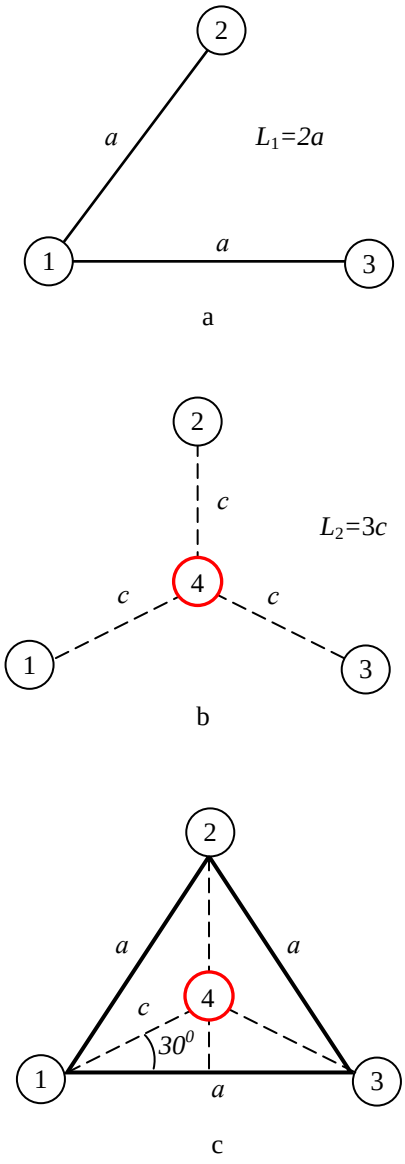


Figure 3 – Geometric comparison of the total weight of TC in networks with $|K|=3$ and different organization of the structure:
a – without using an additional vertices – a linear substructure.
b – using an additional vertices (v_4) – a radial substructure.
c – geometric interpretation of TC weight based on an equilateral triangle

DOI 10.15588/1607-3274-2024-1-17

The size of an edge c in an equilateral triangle, see Fig. 3, c is determined as follows: $\text{Cos } 30^0 = \frac{a/2}{c} \rightarrow c = \frac{a}{\sqrt{3}}$. Thus, in an equilateral triangle (or close to it), the inequality $L_1 > L_2$ will always be valid.

Based on the above, the main idea of the method is to check the structure of the initial graph G on the possibility of reducing the weight of TC between threes $v_i \in K$ in their various combinations (sets) due to the addition of some vertices $v_i \notin K$ (see Fig. 3, b). If such a possibility exists, we will speak of the existence of a local minimum, which is ensured by this $v_i \notin K$. Vertices $v_i \notin K$, which do not provide local minima will be removed from the structure of the initial graph G together with the edges, incidental to them, and they will not be considered in the further creating of the MST.

If several radial substructures that provide local minima are found in the graph G structure, they should be analyzed for the extent to which the base vertices $v_i \in K$ are used together. The following options are possible here, see Fig. 4:

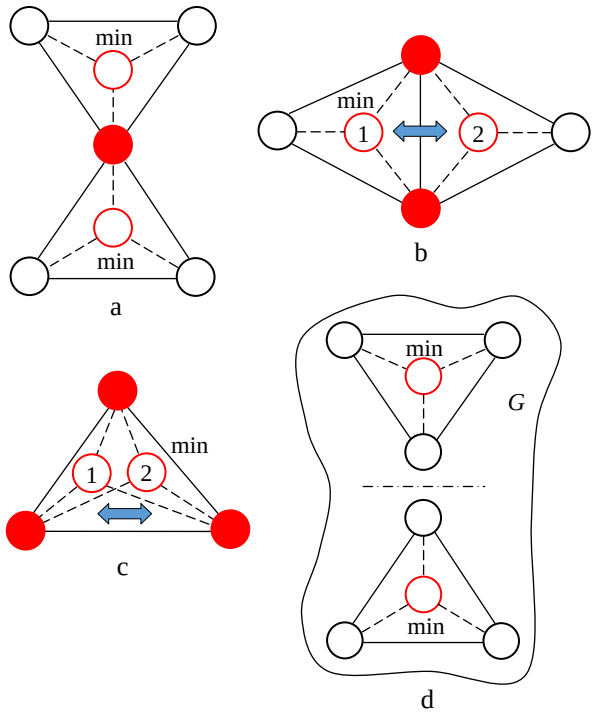


Figure 4 – Degree of compatible use by radial substructures of base vertices $v_i \in K$ (such vertices are marked in solid red):
a – first; b – second; c – third; d – null (no compatible use of base vertices)

Thus, when detected in the structure of the model graph G several radial substructures with the first degree of their joint use of basic vertices (see Fig. 4, a), all vertices $v_i \notin K$, through which such substructures are formed, remain in the structure of the graph G, providing corresponding local minima in it. The same situation occurs with radial substructures with zero degree of use of base vertices (see Fig. 4, d), because substructures are at some distance from each other. In this case, all vertices $v_i \notin K$, due to which such substructures are formed, remain in the structure of the graph G, providing corresponding local minima in it.

The situation will be different in the presence of the second and third degrees of base vertex usage. In such a situation (see Fig. 4, b and Fig. 4, c) you need to find out which of the vertices $v_i \notin K$ (in this case v_1 or v_2) will ensure a lower weight of the TC of the base vertices $v_i \in K$. In the case of the third degree of use of base vertices, competition between vertices $v_i \notin K$ occurs on several parallel radial substructures that connect some triplet of base vertices $v_i \in K$, see Fig. 4, c. The result of such competition is the selection of a single vertex $v_i \notin K$, which ensures the smallest weight of the TC of this trio of vertices. At the same time, the local minimum remains within the three analyzed vertices. In the case of the second degree of using base vertices (see Fig. 4, b), vertices $v_i \notin K$ are the roots of adjacent radial substructures that connect different sets of basic vertices triples $v_i \in K$. In this case, the result of competition between the corresponding vertices $v_i \notin K$ is the choice of the radial substructure that provides the smallest weight of the TC of the corresponding triple of vertices (within the example shown in Fig. 4, b, the local minimum is provided on the triple of vertices, the root of which is the vertex v_1). Both in the first and in the second cases, the vertices $v_i \notin K$, that lost the competition are removed from the structure of the modeling graph G along with the edges, incident to them.

Having considered the general theoretical provisions, we will present the developed method in the form of the following six steps:

Step 1. Based on the modeling weighted undirected graph G, creating its adjacency matrix S_G. According to the matrix S_G creating a matrix of the shortest paths R_G between all pairs of the vertices of graph G. For this purpose, we can use the Warshall-Floyd algorithm [14, 15] or Shimbel [16] and some others.

Step 2. For each $v_i \notin K$ according to the R_G to find the weight of the TC with three base vertices $v_i \in K$, for which the condition $\sum_{j=1}^{n} w_{ij} \rightarrow \min$ is valid.

The result of the operation: the column indices (vertices $v_i \in K$) are *idx*1, *idx*2, *idx*3; the total weight of the TC connecting the given trio of base vertices $v_i \in K$

DOI 10.15588/1607-3274-2024-1-17

with $v_i \notin K$, that is analyzed is $\sum w_{v_i \notin K}^{[v_{idx1}, v_{idx2}, v_{idx3}] \in K}$. To save the results of the operation.

Step 3. For each $v_i \in K$ by the column index sets defined in step 2 (*idx*1, *idx*2, *idx*3) according to the matrix R_G to find the weight of the corresponding vertices. The result of the operation: the weight of the TC connecting the base vertices $v_i \in K$ with three base vertices $v_i \in K$ with numbers *idx*1, *idx*2, *idx*3 is $\sum w_{v_i \in K}^{[v_{idx1}, v_{idx2}, v_{idx3}] \in K}$. To save the results of the operation.

Step 4. To remove vertices $v_i \notin K$ for which the condition is valid:

$$\sum w_{v_i \notin K}^{[v_{idx1}, v_{idx2}, v_{idx3}] \in K} \geq \forall \sum w_{v_i \in K}^{[v_{idx1}, v_{idx2}, v_{idx3}] \in K}, i = \overline{1, n}, \quad (3)$$

from the structure of the model graph G together with the edges, incident to it. Appropriate changes to the matrix S_G should be made.

Step 5. To carry out a pairwise check of the vertices $v_i \notin K$ remaining after the previous steps for the degree of compatible use by the radial substructures of the base vertices $v_i \in K$:

a) if because of such a check zero or first degree was found (without a match by indices or a match by one index), then such vertices should be left in the structure of the model graph G;

b) in the case of detection of the second or third degree (a match according to two or three indices, respectively), determine the vertices $v_i \notin K$ through which the minimum TC of the corresponding trio of base vertices is ensured $v_i \in K$. To remove the vertices that lost the competition from the structure of the model graph G together with the edges, incident to it. An appropriate changes to the matrix S_G should be made.

Step 6. On the modified in this way graph G, by one of the well-known algorithms for creating the MST the required minimum tree $G'_{[K]} = (K, E' \subseteq E)$ is created.

4 EXPERIMENTS

Let us illustrate the application of the method on the example of the graph provided by Fig. 1, *a*. As before, the MST $G'_{[1, 4, 5, 6]}$ is to be found.

Step 1. A calculated matrix of shortest paths R_G between all pairs of graph vertices has the following form:

$$R_G = \begin{array}{c|ccccccc} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 \\ \hline v_1 & 0 & 3 & 4 & 5 & 8 & 9 & 13 \\ v_2 & 3 & 0 & 7 & 2 & 5 & 11 & 10 \\ v_3 & 4 & 7 & 0 & 9 & 4 & 5 & 16 \\ v_4 & 5 & 2 & 9 & 0 & 7 & 10 & 8 \\ v_5 & 8 & 5 & 4 & 7 & 0 & 6 & 15 \\ v_6 & 9 & 11 & 5 & 10 & 6 & 0 & 11 \\ v_7 & 13 & 10 & 16 & 8 & 15 & 11 & 0 \end{array} \quad (4)$$

In expression (4), vertices $v_i \in K$ are marked in red.

Step 2. For each $v_i \notin K$ we define $\sum w_{v_i \notin K}^{[v_{idx1}, v_{idx2}, v_{idx3}] \in K} \to \min$. For the vertex $v_2 \notin K$ it is $\sum w_{v_2}^{[v_1, v_4, v_5]} = 10$. For the vertex $v_3 \notin K$ it is $\sum w_{v_3}^{[v_1, v_5, v_6]} = 13$. For the vertex $v_7 \notin K$ it is $\sum w_{v_7}^{[v_1, v_4, v_6]} = 32$.

Step 3. The results of calculations for this step are presented in Table 1.

Table 1 – The weight of TC of the basic vertices $v_i \in K$ with triples of base vertices $v_i \in K$, having the indices defined in step 2 of the method

$v_i \in K$	$\sum w_{v_i \in K}^{[v_1, v_4, v_5]}$	$\sum w_{v_i \in K}^{[v_1, v_5, v_6]}$	$\sum w_{v_i \in K}^{[v_1, v_4, v_6]}$
1	2	3	4
v_1	13	17	14
v_4	12	22	15
v_5	15	14	21
v_6	25	15	25

Step 4. According to inequality (3), we compare the received sums of weights $\sum w_{v_2}^{[v_1, v_4, v_5]} = 10$, $\sum w_{v_3}^{[v_1, v_5, v_6]} = 13$, $\sum w_{v_7}^{[v_1, v_4, v_6]} = 32$ with sums of weights on the corresponding indices for $v_i \in K$, which are presented in Table 1. The inequality is valid only for the vertex v_7, since $\sum w_{v_7}^{[v_1, v_4, v_6]} = 32$ is greater than any value in column 4, see Table 1. So, the vertex v_7 is removed from the structure of the model graph G with all the edges incident to it. Corresponding changes are also to be made to the matrix S_G.

Step 5. Let us perform a pairwise check of the vertices $v_i \notin K$ remaining after the previous steps for the degree of compatible use by the corresponding radial substructures of the base vertices $v_i \in K$. Such vertices are v_2 and v_3 for which respectively $\sum w_{v_2}^{[v_1, v_4, v_5]} = 10$

DOI 10.15588/1607-3274-2024-1-17

and $\sum w_{v_3}^{[v_1,v_5,v_6]}=13$. As we can see, the radial substructures, the roots of which are these vertices, jointly use the base vertices v_1 and v_5. Therefore, we got the second degree of joint use of basic vertices by radial substructures $v_i \in K$ (match by two indices). Since the weight of the transitive closure over the vertex v_3 is greater than through the vertex v_2 (13>10), the vertex v_3 is also removed with all its incident edges from the structure of the model graph G. Corresponding changes are also to be made to the matrix S_G.

Step 6. On the modeling graph G modified in this way (Fig. 5) using the Kruskal method, we will create the MST $G'_{[1,4,5,6]}$. It will be identical to the MST presented at Fig. 2.

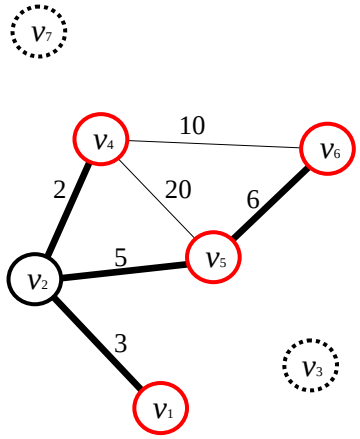


Figure 5 – Modified model graph G and the MST on the selected subset of vertices

5 RESULTS

When applying the developed method to the initial undirected graph G (see Fig. 1), three radial substructures were successively considered, the roots of which were vertices v_2, v_3, v_7, not included in the set K. During the verification, it was found that the vertex of v_7 does not provide a minimum TC between the specified three base vertices $v_i \in K$. This fact made it possible to modify the initial graph G by removing this vertices and all edges incident to it from its structure. The vertices v_2 and v_3 have provided the minimum TC. At the same time, the radial structures (triplets of vertices), of whose roots they are, intersect along two vertices and have the second degree of joint use of the base vertices $v_i \in K$. This fact led to the need to compare the vertices v_2, v_3, and choose the one that provides the local minimum of TC. This vertex appeared to be the vertex v_2. Consequently, vertex v_3 was also removed from the original graph G structure.

Therefore, the internal tools of the proposed method allow testing the structure for the presence of local minima in the TC of the base vertices $v_i \in K$ through the vertices $v_i \notin K$, and modifying the structure of the initial graph G to further find MST $G'_{[K]}$ in this structure.

6 DISCUSSION

The combination of the approaches proposed in the article allowed us to develop a method by which it is possible to build a MST on an arbitrary subset of vertices of the initial undirected graph. This became possible due to the analysis of radial substructures whose roots are vertices $v_i \notin K$, in terms of the weight of the TC of these substructures, and the search for local minima among them. At the same time, this became possible due to the use of the shortest paths matrix (R_G) between all pairs of vertices of the model graph G. Due to the fact that such a matrix contains information not only about the presence of TC between any pair of vertices, but also quantitatively characterizes this relationship, it became possible to analyze different sets of three basic vertices $v_i \in K$, from different locations of the model graph relative to the root of the current radial substructure. The above allows us to launch a mechanism for revealing local minima in the structure of the model graph G and selecting vertices $v_i \notin K$ that provide this minimum. On the other hand, those vertices $v_i \notin K$, which do not provide local minima are removed from the structure of the model graph G, thereby not increasing the weight of the required MST $G'_{[K]}$.

Several dozen full-scale experiments on various network objects of low density have shown the efficiency of the developed method, and the solutions obtained were optimal. At the same time, the behavior of the method on dense network objects of high dimensionality remains a challenge ($n>30$). Thus, the method could be considered quasi-optimal at the moment.

The computational complexity of the combinatorial algorithm that implements the developed method will be determined by the computational complexity of its "basic elements" – the algorithm for finding the shortest paths between all pairs of vertices of the model graph and the algorithm for creating the MST. If the Warshall-Floyd algorithm and the Kruskal algorithm are taken as the basic algorithms, respectively, the overall computational complexity of the combinatorial algorithm will be estimated $O(n^3 + E\log(E))$.

The obtained polynomial estimate of the computational complexity is suitable for using such an algorithm in solving relevant management problems in real life.

CONCLUSIONS

The article solves the actual scientific and applied problem of creating the MST $G'_{[K]}$ on an arbitrarily chosen subset of vertices of the initial undirected weighted graph, where K is an arbitrarily chosen subset of vertices of the initial graph G.

DOI 10.15588/1607-3274-2024-1-17

The scientific novelty of the developed method is as follows:

1) in the formulation of the consequence that to reduce the weight of the transitive closure of the base vertices $v_i \in K$, through some vertices $v_i \notin K$, starting from $|K| = 3$;

2) in the proposed approach to vertices selection $v_i \notin K$. The essence of the approach is to compare the weights of transitive closures of different radial substructures whose roots are vertices $v_i \notin K$, combining different sets of three basic vertices $v_i \in K$;

3) in the proposed approach to determining the local minimum of the weight of radial substructures, among substructures that are in competition. The core of the approach is to analyze the degree of joint use of base vertices $v_i \in K$ by different radial substructures.

The practical value of the method is when it is applied to large and dense network objects that have undergone fragmentation (destruction due to external influences), it is possible to significantly reduce the amount of restoration work and/or total financial costs while quickly restoring the connectivity of elements that are of higher importance in the structure of such an object.

A promising direction for further research is the final verification of the developed method to determine its optimality class.

ACKNOWLEDGEMENTS

This article highlights one of the results obtained by the authors in 2021-2022 while implementing the research project (state registration number 0120U002173) at the Research Center of the National Academy of the National Guard of Ukraine. The authors are grateful to their colleagues for their support during the research and active participation in the discussion of the results. All authors declare that they have neither financial support nor obligations.

REFERENCES

1. Kerigan-Kyrou D. Critical Energy Infrastructure: Operators, NATO, and Facing Future Challenges, *Connections: The Quarterly Journal 12*, 2013, Vol. 3, pp. 109–117. DOI: 10.11610/Connections.12.3.06.
2. Tichý L., Eichler J. Terrorist Attacks on the Energy Sector: The Case of Al Qaeda and the Islamic State, *Studies in Conflict & Terrorism*, 2018, Vol. 41, No. 6, pp. 450–473. DOI: 10.1080/1057610X.2017.1323469.
3. Dunn S., Wilkinson S. Hazard Tolerance of Spatially Distributed Complex Networks, Re*liability Engineering & System Safety 157*, 2017, pp. 1–12. DOI:10.1016/j.ress.2016.08.010.
4. Varianou Mikellidou C., Shakou L., Boustras G. et al. Energy Critical Infrastructures at Risk from Climate Change: A State-of-the-Art Review, *Safety Science*, 2018, Vol. 110, Part C, pp. 110–120. DOI: 10.1016/j.ssci.2017.12.022.
5. Danyk Yu., Maliarchuk T., Briggs Ch. Hybrid War: High-tech, Information and Cyber Conflicts. Connections, *The Quarterly Journal 16*, 2017, Vol. 2, pp. 5–24. DOI: 10.11610/Connections.16.2.01.
6. Sulema O. K., Lande D. V. Finding the optimal hierarchy in a quasi-hierarchical graph by centrality criteria, *Registration, storage and processing of data*, 2015, Vol. 17, No 4, pp. 3–10. ISSN 1560-9189.
7. Prim R. Shortest connection networks and some generalizations, *Bell System Technical Journal*, 1957, Vol. 36, No. 6, pp. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x.
8. Kruskal J. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem, *Proc. AMS*, 1956, Vol 7, No. 1, pp. 48–50. DOI: 10.1090/S0002-9939-1956-0078686-7.
9. Pawan Harish, Narayanan P., Vineet Vibhav, Suryakant Patidar Chapter 7 – Fast Minimum Spanning Tree Computation. GPU Computing Gems Jade Edition, Applications of GPU Computing Series. San Francisco, Morgan Kaufmann, 2012, pp. 77–88. ISBN 9780123859631, DOI: 10.1016/B978-0-12-385963-1.00007-1.
10. Cheriton D., Endre T. Finding minimum spanning trees, *SIAM Journal on Computing*, 1976, Vol. 5 (4), pp. 724–742. DOI:10.1137/0205051.
11. Seth P., Vijaya R. An optimal minimum spanning tree algorithm, *Journal of the ACM*, 2002, Vol. 49 (1), pp. 16–34. DOI:10.1145/505241.505243.
12. Johnson D. Priority queues with update and finding minimum spanning trees, *Information Processing Letters*, 1975, 4 (3), pp. 53–57. DOI:10.1016/0020-0190(75)90001-0.
13. Christofides N. Theory of graphs. Algorithmic approach. Moscow, Mir, 1978, 432 p.
14. Floyd R. Algorithm 97: Shortest Path, *Communications of the ACM*, 1961, Vol. 5 (6), 345 p. DOI:10.1145/367766.368168.
15. Warshall S. Algorithm on Boolean matrices, *Journal of the ACM*, 1962, Vol. 9 (1), pp. 11–12. DOI:10.1145/321105.321107.
16. Shimbel A. Structural parameters of communication networks, *Bulletin of Mathematical Biophysics*, 1953, Vol. 15 (4), pp. 501–507. DOI:10.1007/BF02476438.

Received 11.12.2023.
Accepted 25.02.2024.

УДК 519.172.1: 62-503.57

МЕТОД ПОБУДОВИ МІНІМАЛЬНОГО КІСТЯКОВОГО ДЕРЕВА НА ДОВІЛЬНІЙ ПІДМНОЖИНІ ВЕРШИН ЗВАЖЕНОГО НЕОРІЄНТОВАНОГО ГРАФА

Бацамут В. М. – д-р військ. наук, професор, заступник начальника науково-дослідного центру службово-бойової діяльності Національної гвардії України Національної академії Національної гвардії України, Харків, Україна.

Годлевський С. О. – науковий співробітник науково-дослідного центру службово-бойової діяльності Національної гвардії України Національної академії Національної гвардії України, Харків, Україна.

Бабков Ю. П. – канд. техн. наук, доцент, професор кафедри державної безпеки Національної академії Національної гвардії України, Харків, Україна.

DOI 10.15588/1607-3274-2024-1-17

Морквін Д. А. – д-р філософ., начальник науково-дослідної лабораторії науково-дослідного центру службово-бойової діяльності Національної гвардії України Національної академії Національної гвардії України, Харків, Україна.

АНОТАЦІЯ

Актуальність. Актуальність статті обумовлюється потребою у подальшому розвитку моделей оптимального відновлення зв'язності мережних об'єктів, що зазнали фрагментації внаслідок надзвичайних ситуацій різного характеру походження. Запропонований у статті метод усуває проблемну ситуацію, що полягає у необхідності мінімізації обсягу відновлювальних робіт (загальних фінансових витрат) при оперативному відновленні зв'язності обраної підмножини елементів мережевого об'єкту після його фрагментації.

Мета роботи полягає у розробленні методу побудови мінімального кістякового дерева на довільній підмножині вершин зваженого неорієнтованого графу для мінімізації обсягу відновлювальних робіт і/або загальних фінансових витрат при оперативному відновленні зв'язності елементів, які мають вищий рівень важливості в структурі фрагментованого мережного об'єкту.

Метод. Розроблений метод заснований на ідеї пошуку в структурі модельного неорієнтованого графа локальних мінімумів з використанням вершин графу, що не входять до переліку базових вершин, які потрібно об'єднати мінімальним кістяковим деревом. Під час пошуку локальних мінімумів використовується поняття рівностороннього трикутника та радіальної структури в такому трикутнику. При цьому розрізняються чотири типи підструктур, які забезпечують локальні мінімуми: перші, ті що мають одну спільну базову вершину; другі, ті що мають дві спільні базові вершини; треті, ті що мають три спільні базові вершини; четверті, ті що не мають спільних базових вершин – знаходяться в різних частинах модельного графа. Ті вершини, що не входять до переліку базових, але через які забезпечуються локальні мінімуми, додаються до складу базових. Інші вершини (небазові) разом з інцидентними їм ребрами видаляються з структури модельного графа. Далі, на отриманій таким чином структурі, одним із відомих методів побудови кістякових дерев, будується мінімальне кістякове дерево, яке поєднує набір базових вершин.

Результати. 1) Розроблено метод побудови мінімального кістякового дерева на довільній підмножині вершин зваженого неорієнтованого графа. 2) Запропонована сукупність критеріїв для визначення локальних мінімумів в структурі модельного графа. 3) Виконано верифікацію методу на тестових задачах.

Висновки. Проведені теоретичні дослідження та низка експериментів підтверджують працездатність розробленого методу. Рішення, що виробляються із використанням розробленого методу, є точними, що дозволяє рекомендувати його до практичного використання при визначенні стратегій відновлення зв'язності фрагментованих мережевих об'єктів.

КЛЮЧОВІ СЛОВА: мережевий об'єкт, зважений неорієнтований граф, зв'язність, транзитивне замкнення, мінімальне кістякове дерево, локальний оптимум, критерій оптимізації, метод.

ЛІТЕРАТУРА

1. Kerigan-Kyrou D. Critical Energy Infrastructure: Operators, NATO, and Facing Future Challenges / D. Kerigan-Kyrou // Connections: The Quarterly Journal 12. – 2013. – Vol. 3. – P. 109–117. DOI: 10.11610/Connections.12.3.06.
2. Tichý L. Terrorist Attacks on the Energy Sector: The Case of Al Qaeda and the Islamic State / L. Tichý, J. Eichler // Studies in Conflict & Terrorism. – 2018. Vol. 41, No 6. – P. 450–473. DOI: 10.1080/1057610X.2017.1323469.
3. Dunn S. Hazard Tolerance of Spatially Distributed Complex Networks. / S. Dunn, S. Wilkinson // Reliability Engineering & System Safety 157. – 2017. – P. 1–12. DOI:10.1016/j.ress.2016.08.010.
4. Energy Critical Infrastructures at Risk from Climate Change: A State-of-the-Art Review / [C. Varianou Mikellidou, L. Shakou, G. Boustras et al.] // Safety Science. – 2018. – Vol. 110, Part C. – P. 110–120. DOI: 10.1016/j.ssci.2017.12.022.
5. Danyk Yu. Hybrid War: High-tech, Information and Cyber Conflicts. Connections: / Yu. Danyk, T. Maliarchuk, Ch. Briggs // The Quarterly Journal 16. – 2017. – Vol. 2. – P. 5–24. DOI: 10.11610/Connections.16.2.01.
6. Сулема О. К. Знаходження оптимальної ієрархії у квазіієрархічному графі за критеріями центральності. / О. К. Сулема, Д. В. Ланде // Реєстрація, зберігання і обробка даних. – 2015. – Т. 17. – № 4. – С. 3–10. ISSN 1560-9189.
7. Prim R. Shortest connection networks and some generalizations / R. Prim // Bell System Technical Journal. – 1957. – Vol. 36, No. 6. – P. 1389–1401. DOI: 10.1002/j.1538-7305.1957.tb01515.x.
8. Kruskal J. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem / J. Kruskal // Proc. AMS. – 1956. – Vol 7, No. 1. – C. 48–50. DOI: 10.1090/S0002-9939-1956-0078686-7.
9. Pawan Harish. Chapter 7 – Fast Minimum Spanning Tree Computation. GPU Computing Gems Jade Edition, Applications of GPU Computing Series / Pawan Harish, P. Narayanan, Vibhav Vineet, Suryakant Patidar. – San Francisco: Morgan Kaufmann, 2012. – P. 77–88. ISBN 9780123859631, DOI: 10.1016/B978-0-12-385963-1.00007-1.
10. Cheriton D. Finding minimum spanning trees / D. Cheriton, T. Endre // SIAM Journal on Computing. – 1976. Vol. 5 (4). – P. 724–742. DOI:10.1137/0205051.
11. Seth P. An optimal minimum spanning tree algorithm / P. Seth, R. Vijaya // Journal of the ACM. – 2002. – Vol. 49 (1). – P. 16–34. DOI:10.1145/505241.505243.
12. Johnson D. Priority queues with update and finding minimum spanning trees / D. Johnson. // Information Processing Letters. – 1975. – 4 (3). – P. 53–57. DOI:10.1016/0020-0190(75)90001-0.
13. Кристофидес Н. Теория графов. Алгоритмический подход / Н. Кристофидес. – Москва : Мир, 1978. – 432 с.
14. Floyd R. Algorithm 97: Shortest Path / R. Floyd // Communications of the ACM. – 1961. – Vol. 5 (6). – 345 p. DOI:10.1145/367766.368168.
15. Warshall S. Algorithm on Boolean matrices / S. Warshall // Journal of the ACM. – 1962. – Vol. 9 (1). – P. 11–12. DOI:10.1145/321105.321107.
16. Shimbel A. Structural parameters of communication networks / A. Shimbel // Bulletin of Mathematical Biophysics. – 1953. – Vol. 15 (4). – P. 501–507. DOI: 10.1007/BF02476438.

DOI 10.15588/1607-3274-2024-1-17

УДК 004.318

ПРОЕКТУВАННЯ КОНВЕЄРНОГО ПРОЦЕСОРА RISC-V АРХІТЕКТУРИ З АПАРАТНИМ СПІВПРОЦЕСОРОМ ЦИФРОВОЇ ОБРОБКИ СИГНАЛІВ

Ваврук Є. Я. – канд. техн. наук, доцент, доцент кафедри електронних обчислювальних машин Національного Університету «Львівська політехніка», Львів, Україна.

Махров В. В. – студент кафедри комп'ютерних систем та мереж ДВНЗ «Ужгородський національний університет», Ужгород, Україна.

Гедеон Г. О. – асистент кафедри комп'ютерних систем та мереж ДВНЗ «Ужгородський національний університет», Ужгород, Україна.

АНОТАЦІЯ

Актуальність. Цифрова обробка сигналів використовується в багатьох сферах науки, техніки та діяльності людини. Одним із шляхів реалізації алгоритмів цифрової обробки сигналів є розробка співпроцесорів, як складової частини відомих архітектур. У випадку розробки конвеєрного пристрою такий підхід дозволить використовувати програмні та апаратні засоби відповідної архітектури, забезпечити швидше виконання алгоритмів обробки сигналів, скоротити кількість тактів та кількість звернень до пам'яті.

Мета роботи – проектування та дослідження характеристик конвеєрного процесора архітектури RISC-V з співпроцесором цифрової обробки сигналів, що виконує швидке перетворення Фур'є.

Метод. Аналіз технічної літератури та існуючих рішень дозволяє оцінити переваги і недоліки сучасних розробок та на основі них сформувати актуальність обраної теми. Побудова моделей і дані симуляцій дозволяють перевірити працездатність моделі, знайти слабкі ланки компонентів та поліпшити параметри моделі.

Результати. Спроектовано конвеєрний процесор архітектури RISC-V, який виконує базовий набір інструкцій. Проаналізовано час виконання простої асемблерної програми на конвеєрному та однотактному процесорах. Згідно результатів, тестова програма на конвеєрному процесорі виконується за 29 тактів, тоді як на однотактному – за 60 тактів. Розроблено структуру співпроцесора виконання алгоритму швидкого перетворення Фур'є та набір процесорних інструкцій, які дозволяють працювати із співпроцесором. Кількість тактів виконання співпроцесором алгоритму швидкого перетворення Фур'є за основою два для 512 точок складає 2358 тактів, а для 1024 точок – 5180 тактів.

Висновки. Проведені дослідження та розрахунки показали, що використання розробленого апаратного співпроцесора зменшує час виконання алгоритму ШПФ та навантаження на процесор під час обчислень.

КЛЮЧОВІ СЛОВА: RISC-V, процесор, цифрова обробка сигналів, швидке перетворення Фур'є, конвеєр, співпроцесор, FPGA.

АБРЕВІАТУРИ

ЦОС – цифрова обробка сигналів;
ШПФ – швидке перетворення Фур'є;
ISA – instruction set architecture;
RISC – reduced instruction set computer;
ARM – advanced RISC machine;
NOP – no operation;
FPGA – field-programmable gate array;
MIPS – million instruction per second;
DMIPS – Dhrystone MIPS;
CPI – clock per instruction;
SIMD – single instruction multiple data;
SSE – streaming SIMD extensions;
AVX – advanced vector extension;
LUT – look-up table;
MUX – multiplexer;
ALU – arithmetic-logic unit.

НОМЕНКЛАТУРА

CPU_e – час виконання алгоритму;
CPU_p – кількість тактів, необхідних для виконання алгоритму;
CPU_t – тривалість одного такту в секундах;
CPI_a – середня кількість тактів для виконання алгоритму;
IC – кількість інструкцій для виконання алгоритму;
M – мільйон операцій за секунду;
$X(k)$ – вектор частотної складової сигналу;
$x(2n)$ – вектор дискретних значень (парні індекси елементів);
$x(2n+1)$ – вектор дискретних значень (непарні індекси елементів);
W_N – повертаючий множник;
f_n – частота сигналу за номером n;
f_S – частота дискретизації;
N – кількість точок алгоритму ШПФ;
P – кількість тактів для виконання алгоритму ШПФ;
L – кількість стадій конвеєру;
K – кількість точок, яку опрацьовує метелик;
Q – кількість метеликів.

ВСТУП

При побудові сучасних комп'ютерних систем та окремих пристроїв розробники орієнтуються на використання відкритих обчислювальних стандартів архі-

DOI 10.15588/1607-3274-2024-1-18

тектур, наприклад, x86–x64, ARM, SPARC. RISC-V є однією з таких відкритих і доступних стандартів архітектур з простим та модульним набором команд (ISA). ISA описує роботу ядра процесора, кількість регістрів, кожну інструкцію машинного рівня та її байт-код.

Доступна та постійно розширювана архітектура дає можливість інженерам проводити різного роду дослідження, демонструвати неперевершені результати в продуктивності та енергоефективності, а також робити конкуренцію, як комерційним виробникам, так і іншим інженерам.

Алгоритми цифрової обробки використовуються у багатьох сферах: наука, медицина, інженерія, виробництво, телекомунікації тощо. У залежності від вимог, обчислення алгоритмів ЦОС може виконуватися безпосередньо процесором, окремим спеціалізованим вузлом чи на базі апаратного співпроцесора, впровадженого у структуру основного процесора. Останній підхід суттєво зменшує вимоги до параметрів основного процесора.

Одним з найважливіших алгоритмів ЦОС є швидке перетворення Фур'є (ШПФ). ШПФ – це алгоритм прискореного обчислення дискретного перетворення Фур'є, що зменшує об'єм обчислень з $O(n^2)$ до $O(nlog_2(n))$. Складність алгоритму говорить про те, що для виконання прямого або оберненого перетворення необхідно n^2 комплексних операцій додавання та множення для класичного алгоритму, а для ШПФ – $nlog_2(n)$. Під час виконання алгоритму безпосередньо процесором необхідно постійно записувати/читати в/з комірки пам'яті, час доступу до якої в кілька разів більший за час тактового сигналу процесора. Крім того, як до складу універсальних процесорів, так і у процесори ЦОС, не входить повноцінний співпроцесор для виконання алгоритму ШПФ.

Тому, є актуальною розробка структури співпроцесора ШПФ та впровадження його у ядро RISC-V архітектури.

Об'єкт дослідження – конвеєрний процесор архітектури RISC-V з співпроцесором цифрової обробки сигналів.

Предмет дослідження – методи та засоби проектування процесора архітектури RISC-V з модулем ШПФ та дослідження його характеристик.

Мета роботи – проектування та дослідження характеристик конвеєрного процесора архітектури RISC-V з співпроцесором цифрової обробки сигналів, що виконує швидке перетворення Фур'є.

1 ПОСТАНОВКА ЗАДАЧІ

Для забезпечення функціонування процесора необхідно:

– розробити модель одноядерного конвеєрного процесора;

– розробити модель співпроцесора ШПФ;

– розробити набір інструкцій, які дозволяють процесору керувати співпроцесором;

– перевірити роботу розроблених моделей, використовуючи мікросхеми FPGA.

Процесор повинен відповідати таким вимогам: частота роботи – не більше 250 МГц, CPI – не більше 2; MIPS при частоті 250 МГц – не менше 200.

Співпроцесор ШПФ повинен відповідати таким вимогам: частота роботи модуля – не більше 250 МГц, кількість точок ШПФ – 512/1024.

Оскільки дані вимоги пов'язані із часовими параметрами, розглянемо основні з них. Вираз (1) пов'язує найпростіші показники з часом виконання алгоритму (кількість тактів та довжина тактового сигналу):

$$CPU_e = CPU_p \times CPU_t. \quad (1)$$

Вираз (1) не включає жодних посилань на кількість інструкцій, необхідних для алгоритму; показує, що покращити продуктивність можна шляхом зменшення кількості тактових сигналів, необхідних для виконання алгоритму, або скороченням довжини тактового сигналу [1].

Вираз (2) враховує параметр кількості інструкцій, необхідних для виконання алгоритму:

$$CPU_e = IC \times CPI_a \quad (2)$$

Для одноядерного однотактного процесора архітектури RISC-V показник CPI складає 4–5, тоді як для конвеєрного – 1–1.5.

Формула (3) вираховує показник MIPS (M), визначає продуктивність, обернену до часу виконання:

$$M = \frac{IC}{CPU_e \times 10^6}. \quad (3)$$

При розробці інтегральної схеми на FPGA, в даному випадку для RISC-V ядра та співпроцесора ШПФ, використовують термін – рівень логіки. Рівень логіки означає кількість послідовних логічних елементів між початковим (вхідним) вузлом та кінцевим (вихідним). Даний параметр є важливим при проектуванні, оскільки від нього залежить частота роботи модуля та енергетичні витрати при його роботі.

2 ОГЛЯД ЛІТЕРАТУРИ

На базі RISC-V архітектури розроблений ряд процесорів і мікропроцесорів, IP-ядер. Найближчі за задачами та функціями до розроблюваного процесора наведені нижче.

Проект NEORV32 розрахований для мікросхем програмованої логіки. Містить дві стадії конвеєру, кеш пам'ять для інструкцій та даних, два привілейовані режими ядра, 32 входи переривань, периферія: UART, SPI, DMA, CRC і т.д. Мова програмування – VHDL [2]. Максимальна частота процесора – не більше 150 МГц. Це, по-перше, зв'язано з тим, що процесор містить довгі ланцюги логічних елементів на ста-

DOI 10.15588/1607-3274-2024-1-18

дії виконання, які приводять до відчутних часових затримок. По-друге, цифрові компоненти процесора, як: АЛП, модуль керування, модуль генерації констант, модуль вирішення конфліктних ситуацій, використовують вбудовані функції мови VHDL. З однієї сторони компілятор намагається побудувати операції множення, ділення, зсуви та ін. на базових комбінаційних елементах LUT (які мають значні транспортні затримки). З іншої сторони – алгоритм, що написаний операторами, умовними та циклічними блоками мови VHDL, не є оптимізований при оптимально налаштованих параметрах.

The Potato – це проект процесора RISC-V архітектури, який підтримує базовий набір інструкцій та розроблений на мові VHDL. Процесор підтримує один привілейований рівень, апаратний таймер, вісім незалежних сигналів переривань, Wishbone шину та кеш пам'ять інструкцій та даних. Ядро містить п'ять стадій конвеєру. Проект можна застосовувати на різних типах мікросхем програмованої логіки, але робоча частота такого процесора буде не більшою за 150 МГц [3].

При обчисленні алгоритму ШПФ, програми використовують принцип SIMD, а саме в архітектурі Intel це розширення SSE та AVX, в ARM – SVE. Дані розширення застосовуються при обчисленні арифметичних операцій над чотирма-вісьмома числами рухомої коми одинарної точності. Проте програмні забезпечення з даними розширеннями мають обмеження, оскільки обчислення ШПФ проводиться над комплексними числами, а кількість комплексних чисел, яку можна обробити однією інструкцією, становить два-чотири відповідно до обраного розширення.

Під час виконання алгоритму ШПФ, процесор звертається до пам'яті, в якій містяться дані та таблиця повертаючих множників. Якщо процесор не містить кеш пам'яті, час читання даних із зовнішньої пам'яті складає 7–20 нс [4]. При роботі процесора з кеш пам'яттю час читання даних складає від 1 до 4 нс [5].

У портативних системах, де переважно використовуються дешеві ARM Cortex-M ядра порівняно з Cortex-A, використовують ШПФ для стиснення та декодування аудіо сигналу формату MP3, спектрального аналізу сигналу в осцилографах та ін. Такі ядра не мають SIMD розширення для чисел з рухомою комою та виконують алгоритм ШПФ безпосередньо ядром, а час виконання залежить від робочої частоти ядра, обраного формату даних та кількості даних. В дослідженнях [6–7] наведено швидкодію програмної реалізації алгоритму на ядрах Cortex-M4 та Cortex-M7. Час виконання алгоритму ШПФ для кількості точок 512/1024 з числами з рухомою комою одинарної точності для ядра Cortex-M4 при частоті 180 МГц становить 547 мкс, кількість тактів – 98624, тоді як для Cortex-M7 при частоті 216 МГц час виконання – 339 мкс, кількість тактів – 73292.

У наведених матеріалах [8–10] будова модуля ШПФ подібна за структурою: пам'ять, де зберігається ціла та уявна частина комплексних чисел; таблиця повертаючих множників; схема метелика за основою два та контролюючий модуль. Робоча частота модуля – не більше 100 МГц. Модуль [9] максимально опрацьовує 256 точок на одному метелику за основою два, що не є достатнім для більшості прикладних задач.

Проект модуля Radix-2 FFT [11] представляє реалізацію 16-ти точкового ШПФ, написаний на мові VHDL. На вхід даного модуля подаються 32 сигнали у форматі IEEE-754, які формують 16 комплексних значень. Модуль виконує обчислення за один такт. Складність такої схеми полягає у великих обчислювальних ресурсах: для модуля необхідно 64 комплексних помножувачів, 128 комплексних суматорів.

TMS320VC5505/TMS320C5505/TMS320C5515 – це процесори ЦОС, які містять вбудований співпроцесор для ШПФ (HWAFFT). Дані процесори побудовані на ядрі C55x, RISC архітектури, із розрядністю даних 16 бітів. Показник MIPS складає 800 мільйонів операцій за секунду при частоті 400 МГц [12].

Співпроцесор HWAFFT складається з одного метелика за основою два, який виконує алгоритм прорідження за часом. Апаратний модуль підтримує двоетапний режим, в якому два яруси ШПФ обчислюються за один прохід. У даному режимі, керуючий модуль направляє результати обчислень з першого етапу на вхід метелика для обчислення другого етапу. Це забезпечує прискорення обчислень для великої кількості точок. На вхід модуля надходять 16-ти розрядні числа, формат числа – фіксована крапка. Апаратна частина співпроцесора HWAFFT містить дві стадії конвеєру. Комплексне множення з повертаючим множником виконується на першій стадії конвеєра, а комплексне додавання та віднімання – на другій. Результати отримуються через кілька тактів із моменту надходження вхідних даних: п'ять тактів затримки для одноетапного режиму, дев'ять тактів для двоетапного режиму.

Інтерфейс між процесором та співпроцесором реалізований через набір інструкцій, які дозволяють виконувати операції ініціалізації, завантаження/ читання результату та виконання самого алгоритму. Вхідні дані повинні надходити у біт-реверсивному порядку.

Процесор виконує перетворення 512 точок за 3740 тактів, а 1024 – за 7315 тактів [13].

3 МАТЕРІАЛИ ТА МЕТОДИ

Для покращення продуктивності процесора необхідно зменшити кількість тактів на виконання програми або скоротити довжину тактового сигналу. Конвеєрний підхід проектування процесора дозволяє зменшити кількість тактів на виконання алгоритму.

Розробники RISC-V архітектури пропонують розроблювати конвеєрний процесор, який складається з 5 стадій: Fetch (читання інструкції з пам'яті програм), Decode (декодування зчитаної інструкції), Execute (виконання декодованої інструкції), Memory (читання/запис даних з пам'яті даних), WriteBack (запис результату в регістр) [1].

DOI 10.15588/1607-3274-2024-1-18

При роботі процесорів із конвеєрною архітектурою виникають конфліктні ситуації, які призводять до неможливості виконання чергових інструкцій. Налічують три класи конфліктів: конфлікт між даними, конфлікт керування, структурний конфлікт.

Конфлікт між даними виникає, коли для виконання інструкції, необхідні дані від попередньо отриманого результату. Існує два методи усунення конфлікту:

– вставлення NOP операції в конвеєр. При цьому збільшується час виконання інструкції;

– використання методів пересилання. Пересилання відбувається зі стадій Memory та WriteBack до стадії Execute.

Конфлікт керування виникає, коли процесор виконує команду переходу, а інструкція, яка йде після неї, – залишається в конвеєрі. Щоб вирішити даний конфлікт, необхідно передати NOP операцію зі стадії Fetch до стадії Decode коли виконується перехід.

Структурний конфлікт виникає у суперскалярних та багатотактових процесорах, коли інструкції використовують один і той же апаратний ресурс (пам'ять) в однаковий момент часу. Єдиний спосіб уникнути такого конфлікту – виконати NOP операцію та дати можливість конвеєру, з яким виникає конфлікт, завершити свою роботу.

Згідно рекомендацій [1] розробників RISC-V архітектури розроблено функціональну схему конвеєрного процесора (рис. 1). Згідно схеми, конвеєр складається з п'яти стадій та містить наступні компоненти: програмний лічильник (PC), пам'ять інструкцій (SRAM), модуль керування (Control Unit), регістрова пам'ять для цілих чисел (GPRs), регістрова пам'ять для дійсних чисел (FGPRs), модуль генерації констант (IMM.GEN), модуль усунення конфліктів при роботі з пам'яттю (Hazard Detection Unit), АЛП виконання базових команд (ALU), АЛП для команд множення/ділення (M Extension), АЛП дійсних чисел (FPU), модуль керування АЛП (ALU Control Unit), блок виконання порівняння (Branch Unit), модуль пересилання даних (Forwarding Unit ALU), пам'ять даних (DRAM). Процесор виконує 37 базових інструкцій та вісім інструкцій множення/ділення.

При виконанні алгоритмів ШПФ використовують алгоритми прорідження за частотою та прорідження за часом. Для розкладу дискретних даних в спектральний ряд за алгоритмом ШПФ за основою два, використовується наступний вираз [14–15]:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x(2n) \times W_N^{2nk} + \sum_{n=0}^{\frac{N}{2}-1} x(2n+1) \times W_N^{k(2n+1)}. \qquad (4)$$

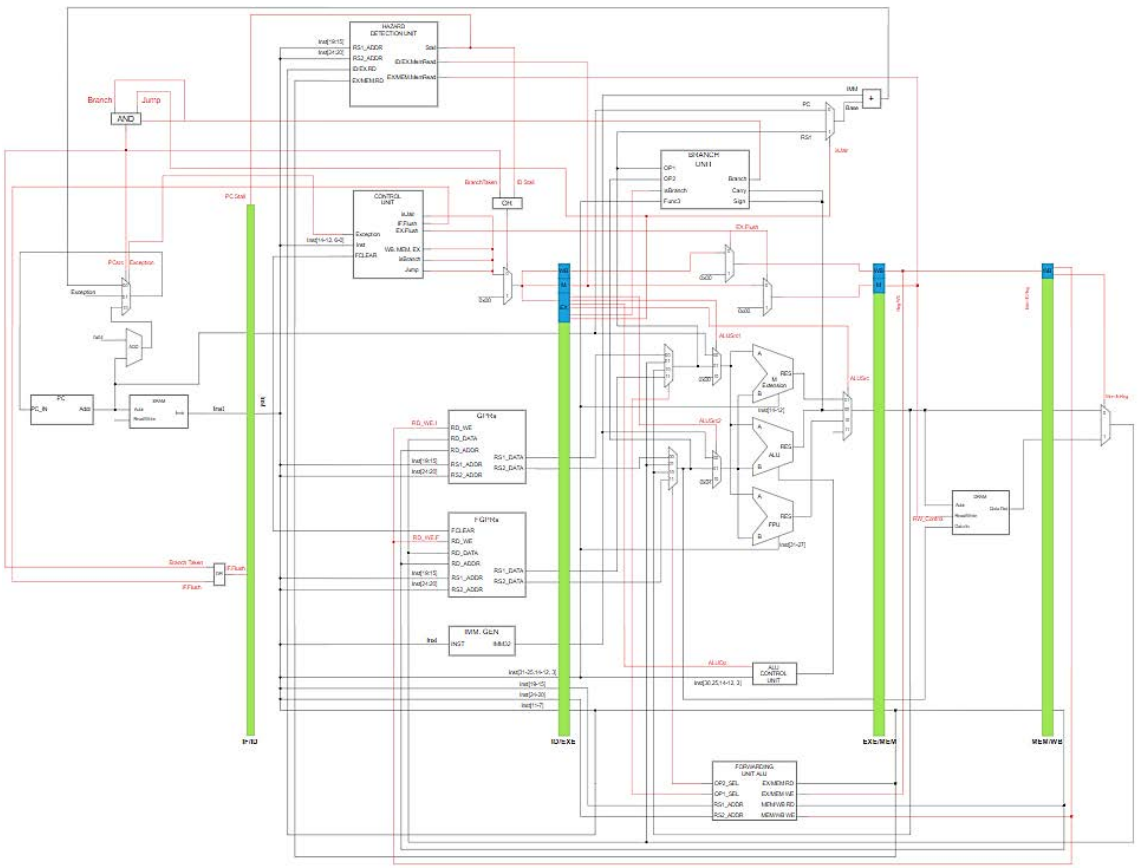


Рисунок 1 – Функціональна схема конвеєрного RISC-V ядра

DOI 10.15588/1607-3274-2024-1-18

Складність обох алгоритмів однакова, але є різниця у порядку вхідних та вихідних даних. Для алгоритму прорідження за часом вхідні дані розміщені в бітреверсивному порядку, а вихідні – в прямому. Зворотній підхід використаний і для прорідження за частотою.

Основним елементом ШПФ є метелик (рис. 2). На рисунку 2 наведено схеми метелика за основою два, які містять дві операції додавання та одну операцію множення комплексних змінних. Стрілка на рисунку 2а означає множення повертаючого множника на різницю двох комплексних величин, а на рисунку 2б – множення повертаючого множника на комплексну величину *b*.

Результатом виконання ШПФ є вектор обчислених даних, який містить номери частот з яких утворена функція. Щоб знайти частоту, маючи номер частоти, використовують наступний вираз:

$$f_n = \frac{f_S \times n}{N}. \quad (5)$$

а б

Рисунок 2 – Схема метелика ШПФ:
а – з прорідженням за частотою, б – з прорідженням за часом

У рамках дослідження схеми модуля [10], авторами даної статті проведено аналіз, зроблено висновки стосовно його працездатності та визначено компоненти, які можна покращити, що в результаті дозволить пришвидшити процес обчислень.

Модуль [10] складається з пристрою генерації адрес AGU (Address Generation Unit); таблиці повертаючих множників Twiddle factor ROM; одного метелика за основою два; двох модулів пам'яті для зберігання дійсної та уявної частини комплексних змінних; з'єднаннями, які з'єднують дані компоненти та утворюють працюючий комплекс.

Пам'ять для комплексних значень та пам'ять для констант повертаючих множників має затримку читання один такт. Модуль AGU контролює генерацію адрес для читання/запису вмісту пам'яті в/з метелика.

Кількість тактів, яка необхідна модулю, щоб виконати операцію на одному метелику і записати результат в пам'ять, рівна десяти [10]. На основі вище описаних параметрів наведено загальну кількість тактів для виконання алгоритму (табл. 1).

Таблиця 1 – Кількість тактів виконання алгоритму ШПФ на однотактному метелику за основою два

Кількість точок	Кількість тактів
32	160
512	23040
1024	51200
2048	92160

Схема метелика виконує алгоритм з частотним прорідженням, що не є повним функціоналом, оскільки можуть виникнути випадки, в яких необхідно саме алгоритм прорідження за часом.

Для перевірки працездатності модуля за описаним принципом [8–10] побудовано математичну модель у системі MathCAD, яка виконує пряме ШПФ над 16 точками. Дана модель складається з матриці повертаючих множників, матриці перестановки повертаючих множників, матриці перестановки точок, функції метелика за основою два та основної функції ШПФ. Функція метелика має можливість виконувати алгоритм, як за прорідженням за часом, так і за частотою.

На рисунку 3 наведено схему виконання алгоритму прямого та оберненого ШПФ (функція FFT16), де позначено: *X* – вхідні дискретні дані; *W* – матриця повертаючих множників; *PermXI* – матриця перестановки точок; *PermW* – матриця перестановки повертаючих множників; *InPerm* – ознака порядку вхідних даних (біт-реверсивний або прямий); *OutPerm* – ознака порядку вихідних даних (біт-реверсивний або прямий); *Inverse* – ознака виконання алгоритму (прорідження за часом або частотою).

Перший крок функції FFT16 – визначення параметрів *a* (збільшення або зменшення ітерацій) та *s* (ітераційний лічильник) за допомогою вхідного параметру *InPerm*. Основний алгоритм виконується в двох операторах циклу. Перший цикл описує ітерацію по чотирьом ярусам, другий – обчислення даних на одному метелику. Після завершення обчислень функція повертає вектор з елементами, розміщеними у бітреверсивному порядку при *OutPerm* рівне одиниці, в інакшому випадку – повертає обчислений вектор.

Результати функції FFT16 порівняно із вбудованою функцією MathCAD – fft, яка приймає один параметр – вектор дискретних даних. Для перевірки використано функцію синуса з такими параметрами: значення амплітуди – 16, частота – 5 Гц, початкова фаза – 0 радіан.

Побудовано амплітудно-частотний графік функції синуса (рис. 4). Згідно графіку, номер частоти сигналу – один, сама частота сигналу за п'ятим виразом – 5 Гц. Отже, обчислена частота рівна частоті дискретного сигналу, що підтверджує правильність обчислень.

DOI 10.15588/1607-3274-2024-1-18

Процес можна прискорити, використовуючи конвеєрний підхід. Модуль ШПФ [10] виконує обчислення двох комплексних значень на одному метелику за десять тактів. Для покращення характеристик авторами даної статті пропонується побудувати шести стадійний конвеєр. Тоді загальна кількість тактів для виконання алгоритму за даним принципом рівна:

$$P = \left(\frac{N}{Q \times K} + L \right) \times \log_K (N). \quad (5)$$

З виразу 6 обчислено загальну кількість тактів, необхідну для виконання алгоритму, використовуючи один конвеєрний метелик (Q = 1) за основою два (K = 2) (табл. 2).

Таблиця 2 – Кількість тактів виконання алгоритму на конвеєрному метелику

Кількість точок	Кількість тактів
32	110
512	2358
1024	5180
2048	11330

На рисунку 5 наведено функціональну схему конвеєрного шести стадійного метелика. В метелику виконується обчислення, як за алгоритмом з прорідження за часом, так і з прорідження за частотою. Згідно рисунку 5, X та Y – вхідні дані, W – повертаючий множник, Ctrl – ознака нормалізації вихідних даних, Mode – ознака виконання алгоритму (прорідження за часом або частотою) CLK – тактовий сигнал для D-тригеру.

Функціональна схема співпроцесора ШПФ наведена на рисунку 6. Даний співпроцесор складається з модуля керування (Control Unit), модуля генерації адреси (Address Generation Unit), пам'яті даних (Data Memory), пам'яті повертаючих множників (Exponent Memory), метелика ШПФ за основою два (Radix-2). Обмін інформацією між процесором та співпроцесором відбувається за допомогою вхідних та вихідних сигналів модуля керування.

```
FFT16(X, W, PermXI, PermW, InPerm, OutPerm, Inverse) :=
    a ← −1  if  InPerm = 1
    a ← 1  otherwise
    s ← 0  if  InPerm = 0
    s ← 3  otherwise
    for  i ∈ 0.. 3
        for  j ∈ 0.. 7
            (A  B) ← Radix_2[X(PermXI2·j,s), X(PermXI2·j+1,s), W(PermWj,s), InPerm, Inverse]
            X(PermXI2·j,s) ← A
            X(PermXI2·j+1,s) ← B
        s ← s + a
    for  k ∈ 0.. 15
        rk ← PermXIk,0
    X ← Permutation(X,r)  if  OutPerm = 1
    X  otherwise
```

Рисунок 3 – Схема алгоритму ШПФ

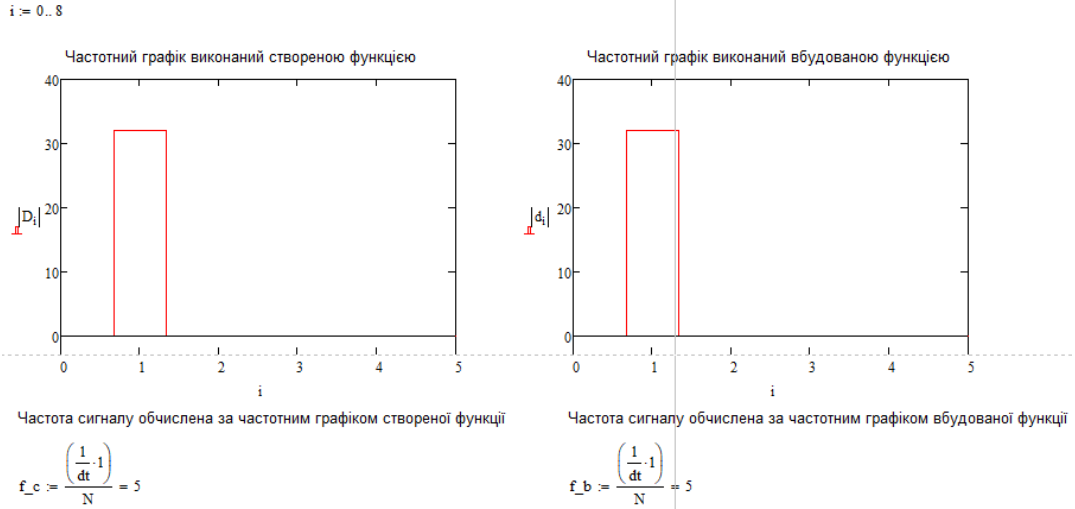


Рисунок 4 – Амплітудно-частотний графік функції синуса

DOI 10.15588/1607-3274-2024-1-18

Пам’ять повертаючих множників має об’єм $\frac{N}{2} \times 4$ байти.

Для керування модулем ШПФ використовуються наступні сигнали: Command – шина команд керування модулем; User Data WR – шина вхідних даних; User Data RD – шина вихідних даних; User Address – шина адреси читання/запису, Enable – сигнал активації модуля.

Розроблений співпроцесор з одним метеликом за основою два та об’ємом пам’яті 4 кБ є невитратним рішенням для покращення процесора. Збільшення пам’яті співпроцесора або основи метелика збільшить складність апаратної розробки та апаратних модулів.

Модуль можна модифікувати збільшенням розміру пам’яті (Data Memory), що дозволить обробити більше точок. При збільшені розміру пам’яті модифікується модуль генерації адрес (Address Generator Unit) та керуючий модуль (Control Unit). Для більш оптимізованого рішення варто використовувати пам’ять розмірністю 2^n, $n \in N$. Кількість точок, яку може обробити співпроцесор, обмежена розмірністю пам’яті.

Введення у схему додаткового метелика за основою два дозволить зменшити час виконання алгоритму. При впровадженні додаткового метелика додається один порт до пам’яті, модифікується модуль генерації адрес (Address Generator Unit) та керуючий модуль (Control Unit). Кількість метеликів, яку можливо встановити у автомат, залежить від кількості портів входу/виходу, яку підтримує пам’ять.

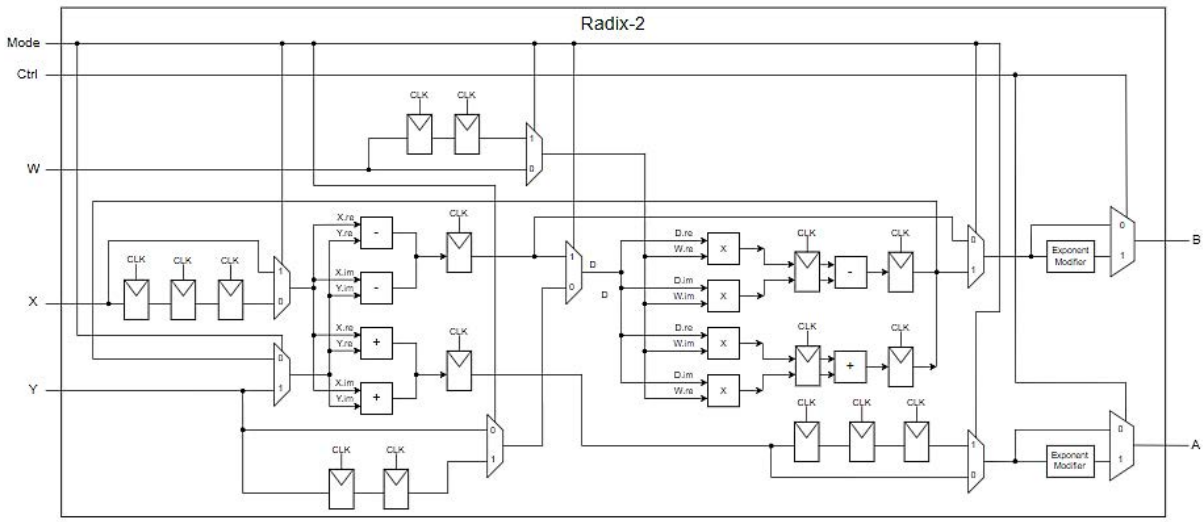


Рисунок 5 – Функціональна схема конвеєрного метелика

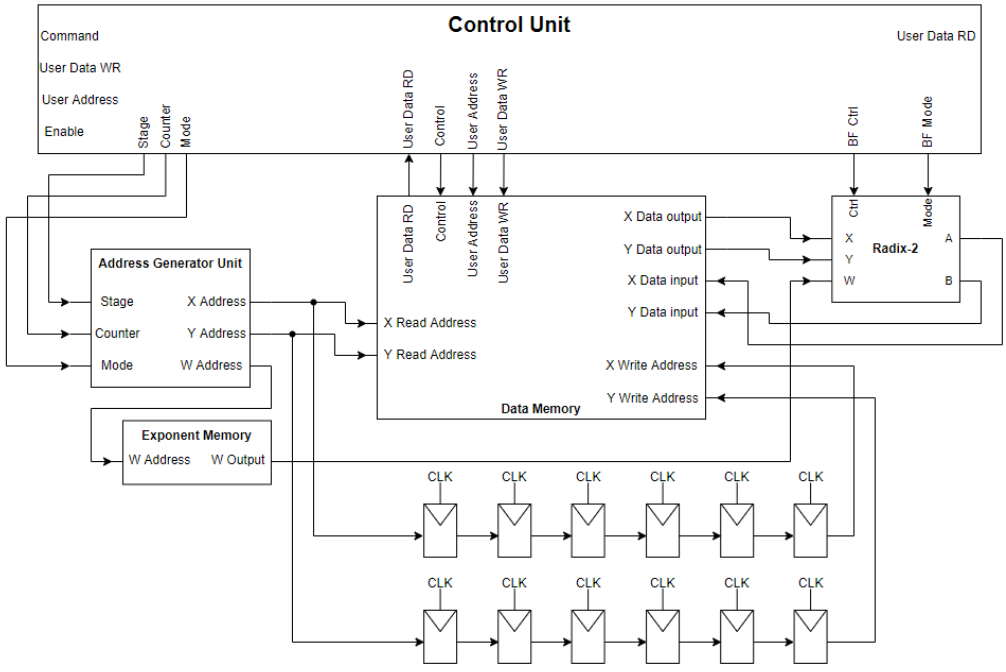


Рисунок 6 – Функціональна схема автомату ШПФ

DOI 10.15588/1607-3274-2024-1-18

На рисунку 7 наведено функціональну схему процесора RISC-V архітектури з впровадженим співпроцесором ШПФ. Співпроцесор знаходиться у стадії виконання (Execute). На вхід мультиплексора під'єднані шина User Data RD разом з іншими АЛП. Інформація для шин Command, User Data RW, User Address, Enable генерується модулем керування ядра процесора (Control Unit) та посилається до входу станового регістра EX/MEM (рис. 1).

Формат інструкцій процесора для роботи з модулем ШПФ наведено на рисунку 8. Команди схожі до формату R-типу, оскільки працюють із двома регістрами і мають певний ідентифікатор інструкції у полі func3 та func7. Створення нового формату інструкцій призвело б до часткової модифікації архітектури та її ускладнення.

Введено 11 інструкцій, чотири з яких дозволяють зчитувати дані з пам'яті дійсної або уявної частини в регістри дійсних або цілих чисел. Наступні чотири інструкції дозволяють записувати дані у пам'ять дійсної або уявної частини з регістрів дійсних або цілих чисел. Останні три команди дозволяють керувати модулем, а саме: старт виконання алгоритму (fftstart), скидання автомату в початкове значення (fftreset), отримання інформації про співпроцесор (fftstatus).

Читання даних з пам'яті та запис їх процесором у співпроцесор ШПФ відбувається один раз для одного виконання перетворення Фур'є. Індексація запису/читання даних в/з пам'яті співпроцесора є прямою або біт-реверсивною при вказанні параметра інструкцією співпроцесора завантаження/читання.

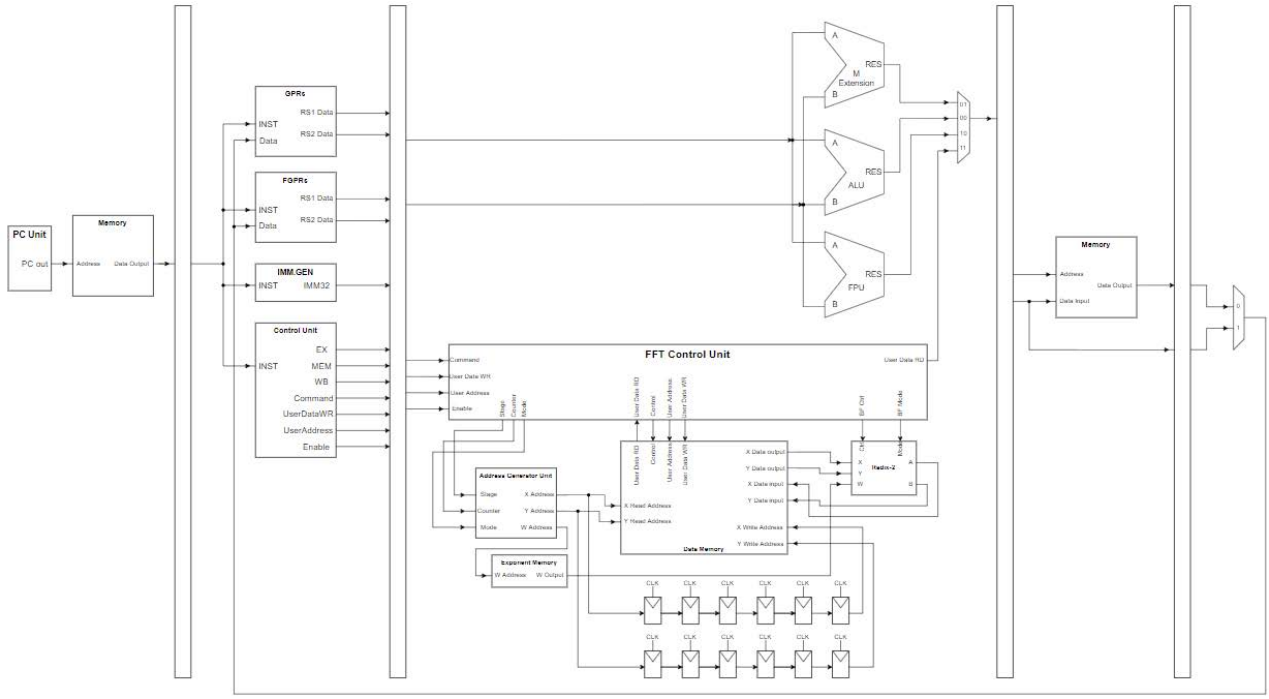
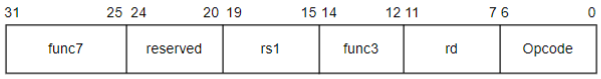


Рисунок 7 – Функціональна схема конвеєрного процесора RISC-V архітектури з впровадженим модулем ШПФ

FFT INSTRUCTION FORMAT



Inst	Name	Opcode	func3	func7
fftstoreRe.f	Store data to FFT from FP register	1011011	000	0x0
fftstoreIm.f	Store Imaginary part to FFT from FP register	1011011	000	0x1
fftstoreRe.i	Store data to FFT from Integer register	1011011	001	0x0
fftstoreIm.i	Store Imaginary part to FFT from Integer register	1011011	001	0x1
fftloadRe.f	Load Real part from FFT to FP register	1011011	010	0x0
fftloadIm.f	Load Imaginary part from FFT to FP register	1011011	010	0x1
fftloadRe.i	Load Real part from FFT to Integer register	1011011	011	0x0
fftloadIm.i	Load Imaginary part from FFT to Integer register	1011011	011	0x1
fftstart	Start execution	1011011	100	
fftreset	Reset module	1011011	101	
fftstatus	Get module status	1011011	110	

Рисунок 8 – Формат інструкцій процесора для роботи з модулем ШПФ

DOI 10.15588/1607-3274-2024-1-18

4 ЕКСПЕРИМЕНТИ

Модель запропонованого процесора розроблена на мікросхемі програмованої логіки Xilinx SPARTAN 6. Дана мікросхема має достатній набір програмованих логічних елементів (LUT), однопортових та багатопортових елементів пам'яті (RAMB16BWER), тригерів, модулів ЦОС (DSP48A1) та інших вбудованих компонентів [16].

Для розробки апаратного забезпечення на мікросхемі Xilinx SPARTAN 6 використовується програмний додаток Xilinx ISE Suite, який дозволяє розробляти логічні схеми різної складності, використовуючи дві мови опису апаратури: VHDL та Verilog. Даний програмний додаток дає можливість оцінити складність моделі, визначити її максимальну частоту, переглянути цифрову схему розробленої моделі та на цій основі провести удосконалення моделі. Обрано мову опису апаратури – VHDL.

Для знаходження параметру CPI, ядра RISC-V архітектури, розроблено програму мовою асемблера для архітектури RISC-V, яка виконує чотири арифметичні операції та одну операцію переходу. Виконуючи програму, необхідно знайти кількість інструкцій та загальну кількість тактів на її виконання.

Для знаходження параметру MIPS, ядра RISC-V архітектури, розроблено програму мовою асемблера множення двох цілочисельних матриць із розрядністю елемента 32 біти. По завершенню виконання програми протягом однієї секунди, необхідно визначити кількість інструкцій, які виконалися за даний час.

Для визначення швидкодії виконання алгоритму ШПФ, написано асемблерну програму, яка передає дискретні дані синуса амплітудою в 3 одиниці, частотою 44100 Гц, початковою фазою – 0 радіан співпроцесору ШПФ та виконує прямий розклад 512/1024 точок у частотний спектр. По завершенню виконання даної програми необхідно визначити кількість тактів для виконання модулем алгоритму та виконати порівняння із готовими результатами досліджень.

5 РЕЗУЛЬТАТИ

Дослідження проведені у симуляторі ISim, який є частиною додатку Xilinx ISE. В симуляторі налаштовано тактову частоту процесора на 250 МГц. Знайдено параметр CPI ядра RISC-V та порівняно його з однотактним процесором. Програма на конвеєрному процесорі виконалась за 29 тактів, а на однотактному – 60. Відповідно CPI конвеєрного процесора – 1.6, а однотактного – 4.6.

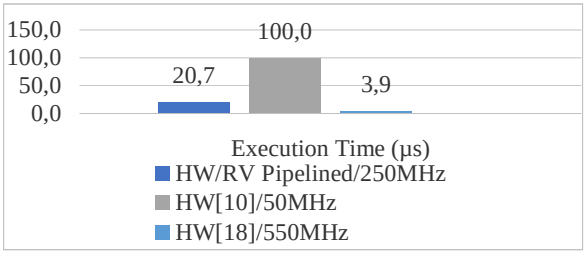
Алгоритм множення двох матриць розмірами 10 на 10 виконано за час 14.224 мкс з виконаною кількістю інструкцій – 3456. Знайдено параметр MIPS конвеєрного процесора – 242, тобто 242 мільйонів інструкцій за секунду.

Схема метелика використовує вбудований апаратний модуль DSP на якому створено 64-х бітний помножувач [17].

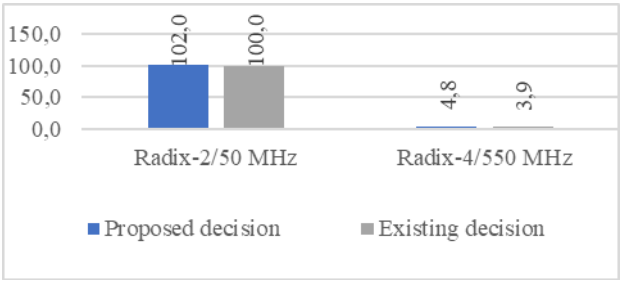
Знайдено кількість тактів, необхідну для виконання алгоритму ШПФ для 512 та 1024 точок (діаграма 3). Для 512 точок модулю необхідно 2358 тактів, а час виконання при частоті 250 МГц – 9.4 мкс. Для 1024 точок кількість тактів складає 5180, а час виконання – 20.7 мкс.

На діаграмі 1 зображено порівняльну характеристику часу виконання ШПФ для 1024 точок для розробленого співпроцесора ШПФ та існуючих рішень. Розроблений співпроцесор виконує алгоритм ШПФ за основою два при тактовій частоті роботи 250 МГц у п'ять разів повільніше у порівнянні з модулем [18] при частоті 550 МГц; модуль [10] при частоті 50 МГц виконує ШПФ за 100 мкс, тоді як розроблений співпроцесор – за 20.7 мкс при тактовій частоті роботи 250 МГц.

На діаграмі 2 наведено порівняльну характеристику часу виконання алгоритму ШПФ спроектованим співпроцесором (Proposed decision) та іншими рішеннями (Existing decision) при частотах 50 МГц і 550 МГц. Кількість точок для алгоритму ШПФ – 1024. Згідно діаграми, співпроцесор виконує алгоритм на 2 мкс повільніше у порівнянні з модулем [18] при тактовій частоті роботи 50 МГц та на 0.9 мкс повільніше за модуль [18] при частоті 550 МГц.



Діаграма 1 – Порівняльна характеристика часу виконання (мкс) ШПФ для 1024 точок



Діаграма 2 – Порівняльна характеристика часу виконання (мкс) алгоритму ШПФ співпроцесором

У таблиці 3 наведено дані використаних ресурсів мікросхеми Xilinx SPARTAN 6.

Таблиця 3 – Використані ресурси системи

LUT	DSP48A1	RAMB16BWER	DFF
7232 з 9112	24 з 32	6 з 32	9872 з 18224

6 ОБГОВОРЕННЯ

Аналізуючи результати досліджень, що проводилися для мікросхеми програмованої логіки, запропонований метод з апаратним співпроцесором ЦОС демонструє прискорення в обчисленнях алгоритму ШПФ у порівнянні з програмними реалізаціями для ядер ARM Cortex-M4 та ARM Cortex-M7 [6–7].

DOI 10.15588/1607-3274-2024-1-18

Час виконання запропонованого методу дещо поступається існуючим апаратним рішенням [8–10] із-за складності апаратних модулів, які складаються з довгих ланцюгів LUT елементів, апаратних вбудований модулів DSP, що збільшують час виконання інструкцій розширення.

Основною перевагою запропонованого методу є його модульність, що дозволяє збільшувати кількість метеликів, змінювати основу метелика, збільшувати кількість точок для опрацювання для зменшення часу виконання без модифікації структури. Варто зазначити, що даний співпроцесор працює з числами з рухомою комою одинарної точності, які забезпечують точність обчислень.

Таким чином, удосконалено процесор RISC-V архітектури, впровадженням співпроцесора ШПФ, без модифікації самої архітектури.

ВИСНОВКИ

Розглянуто відкриту архітектуру набору команд RISC-V, що дозволяє освітнім, науково-дослідним та промисловим організаціям працювати над розробками на основі даних процесорів.

На основі програмованої логіки, використовуючи розроблену модель, розроблена схема конвеєрного процесора RISC-V архітектури. Використовуючи симулятор Isim, проведено дослідження працездатності та продуктивності процесора. Співпроцесор виконує алгоритм ШПФ розмірністю 512/1024 точки. Час виконання співпроцесором алгоритму на одному метелику за основою два для 512 точок склав 9.4 мкс, а для 1024 точок – 20.7 мкс при тактовій частоті роботи 250 МГц.

Проведено дослідження часу виконання алгоритму ШПФ розробленим співпроцесором на частотах 50 МГц і 550 МГц. Згідно результатів, час виконання алгоритму співпроцесором з метеликом за основою два є близьким до значення, наведеного у джерелі [18], тоді як з метеликом за основою чотири, – розроблений співпроцесор виконує ШПФ на 0.9 мкс повільніше.

Подальші дослідження розробленого процесора спрямовані на визначення продуктивності процесора з використанням тестової програми Dhrystone. Це пов’язано з тим, що параметр MIPS не дозволяє об’єктивно оцінити продуктивність процесорів різних архітектур, тоді як DMIPS є стандартизованим та використовується виробниками обчислювальних машин. Для правильності розрахунків продуктивності необхідно при тестуванні використовувати ідентичні версії синтетичного тесту.

Науковою новизною роботи є покращення архітектури RISC-V, розширенням системи команд та впровадженням апаратного співпроцесора ШПФ.

Практичне значення розробленого процесора з апаратним модулем ШПФ полягає у зменшенні часу виконання алгоритму ШПФ, покращенням архітектури RISC-V та можливістю використання апаратних модулів виконання інших алгоритмів ЦОС.

DOI 10.15588/1607-3274-2024-1-18

ПОДЯКИ

Роботу виконано в рамках науково-дослідної теми кафедри комп’ютерних систем та мереж Ужгородського національного університету «Методи і засоби апаратної та програмної реалізації високопродуктивних комп’ютерних систем та мереж» (номер державної реєстрації 0121U110031). Автори вдячні Горвату Петру Петровичу, доценту кафедри комп’ютерних систем та мереж ДВНЗ «УжНУ» за цінні та конструктивні поради під час написання роботи.

ЛІТЕРАТУРА

1. Patterson D. A. Computer Organization and Design. The Hardware/Software Interface: RISC-V Edition / D. A. Patterson, J. L. Hennessy. – California : Morgan Kaufmann Publishers, 2018. – 1665 p.
2. The NEORV32 RISC-V Processor [Electronic resource]. – Access mode: https://github.com/stnolting/neorv32
3. The Potato Project [Electronic resource]. – Access mode: https://github.com/skordal/potato
4. Keeth B. DRAM Circuit Design : Fundamental and High-Speed Topics / B. Keeth, R. J. Baker, B. Johnson, F. Lin. – Hoboken : John Wiley & Sons, 2007. – 440 p.
5. Memory Performance in a Nutshell [Electronic resource]. – Access mode: https://www.intel.com/content/www/us/en/developer/articles/technical/memory-performance-in-a-nutshell.html
6. The DSP capabilities of ARM Cortex-M4 and Cortex-M7 Processors [Electronic resource] / T. Lorenser. – Access mode: https://community.arm.com/cfs-file/__key/communityserver-blogs-components-weblogfiles/00-00-00-21-42/7563.ARM-white-paper-_2D00_-DSP-capabilities-of-Cortex_2D00_M4-and-Cortex_2D00_M7.pdf
7. Digital signal processing for STM32 microcontrollers using CMSIS // STMicroelectronics. – 2018. – AN4841 Rev 2. – 25 P.
8. Sapiecha K. Modular architecture for high performance implementation of FFT algorithm / K. Sapiecha, R. Jarocki // ACM SIGARCH Computer Architecture New. – 1986. – Vol. 14, № 2. – P. 261–270. DOI: 10.1145/17356.17387
9. A portable hardware design of a FFT algorithm / [C. Gonzalez-Concejero, V. Rodellar, A. Alvarez-Marquina et al.] // Latin American applied research. – 2007. – Vol. 37, № 1. – P. 79–82.
10. The Fast Fourier Transform in Hardware: A Tutorial Based on an FPGA Implementation [Electronic resource] / G. Slade. – Access mode: https://web.mit.edu/6.111/www/f2017/handouts/FFTtutorial121102.pdf
11. Radix-2 FFT – VHDL Implementation [Electronic resource]. – Access mode: https://github.com/bugratufan/radix2-fft-vhdl
12. TMS320C55x Technical Overview [Electronic resource]. – Access mode: https://www.ti.com/lit/ug/spru393/spru393.pdf?ts=1693041737254
13. FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs [Electronic resource] / M. McKeown. – Access mode: https://www.ti.com/lit/an/sprabb6b/sprabb6b.pdf?ts=1695025017385
14. Lyons R. G. Understanding Digital Signal Processing, Third Edition / R. G. Lyons. – Boston : Pearson Education, 2011. – 858 p.
15. Ifeachor E. C. Digital Signal Processing: A Practical Approach, Second Edition / E. C. Ifeachor, B. W. Jervis. – Boston : Addison Wesley, 1993. – 779 p.

16. Spartan-6 Family Overview [Electronic resource]. – Access mode: https://docs.xilinx.com/v/u/en-US/ds160
17. Spartan-6 FPGA Data Sheet: DC and Switching Characteristics [Electronic resource]. – Access mode: https://docs.xilinx.com/v/u/en-US/ds162
18. High Throughput and Mixed Radix N-Point Parallel Pipelined FFT VLSI Architectures for Advanced Wireless Communication / [K. Vijayakanthan, K. Hemachandran, M. Anand et al.] // International Journal of Grid and Distributed Computing. – 2020. – Vol. 13, № 1. – P. 400–411.

Стаття надійшла до редакції 11.12.2023.
Після доробки 30.01.2024.

UDC 004.318

THE DESIGN OF THE PIPELINED RISC-V PROCESSOR WITH THE HARDWARE COPROCESSOR OF DIGITAL SIGNAL PROCESSING

Vavruk Y. Y. – PhD, Associate Professor, Associate Professor of Electronic computing department of Lviv Polytechnic National University, Lviv, Ukraine.

Makhrov V. V. – Student of the Department of Computer Systems and Networks of Uzhhorod National University, Uzhhorod, Ukraine.

Hedeon H. O. – Assistant of the Department of Computer Systems and Networks of Uzhhorod National University, Uzhhorod, Ukraine.

ABSTRACT

Context. The digital signal processing is applied in many fields of science, technology and human activity. One of the ways of implementing algorithms of digital signal processing is the development of coprocessors as an integral part of well-known architectures.

In the case of developing a pipelined device, the presented approach will allow to use software and hardware tools of the appropriate architecture, provide the faster execution of signal processing algorithms, reduce the number of cycles and memory accesses.

Objective. Objectives are design and characterization study of a pipelined RISC-V processor and coprocessor of digital signal processing which performs fast Fourier transform.

Method. Analyzing technical literature and existing decisions allow to assess advantages and disadvantages of modern developments and on the basis of which to form the relevance of the selected topic. Model designing and simulation results allow to examine a model efficiency, to determine weak components' parts and to improve model parameters.

Results. The pipelined RISC-V processor has been designed which executes a basic set of instructions. Execution time of assembly program on the single-cycled and the pipelined processors have been analyzed. According to the results, the test program on the pipelined processor is executed in 29 cycles, while on the single-cycle processor it takes 60 cycles. The structure of the coprocessor for the fast Fourier transform algorithm and a set of processor instructions that allow working with the coprocessor have been developed. The number of cycles of the coprocessor based on Radix-2 fast Fourier transform algorithm for 512 points is 2358 cycles, and for 1024 points is 5180 cycles.

Conclusions. Conducted researches and calculations have showed that the application of the developed hardware coprocessor reduces the fast Fourier transform algorithm execution time and the load of the pipelined processor during calculations.

KEYWORDS: RISC-V, processor, digital signal processing, fast Fourier transform, pipelined, coprocessor, FPGA.

REFERENCES

1. Patterson D. A., Hennessy J. L. Computer Organization and Design. The Hardware/Software Interface: RISC-V Edition. California, Morgan Kaufmann Publishers, 2018, 1665 p.
2. The NEORV32 RISC-V Processor. Access mode: https://github.com/stnolting/neorv32
3. The Potato Project. Access mode: https://github.com/skordal/potato
4. Keeth B., Baker R. J., Johnson B., Lin F. DRAM Circuit Design : Fundamental and High-Speed Topics. Hoboken, John Wiley & Sons, 2007, 440 p.
5. Memory Performance in a Nutshell. Access mode: https://www.intel.com/content/www/us/en/developer/articles/technical/memory-performance-in-a-nutshell.html
6. Lorenser T. The DSP capabilities of ARM Cortex-M4 and Cortex-M7 Processors. Access mode: https://community.arm.com/cfs-file/__key/communityserver-blogs-components-weblogfiles/00-00-00-21-42/7563.ARM-white-paper-_2D00_-DSP-capabilities-of-Cortex_2D00_M4-and-Cortex_2D00_M7.pdf
7. Digital signal processing for STM32 microcontrollers using CMSIS, *STMicroelectronics*, 2018, AN4841 Rev 2, 25 p.
8. Sapiecha K., Jarocki R. Modular architecture for high performance implementation of FFT algorithm, *ACM SIGARCH Computer Architecture New*, 1986, Vol. 14(2), pp. 261–270. DOI: 10.1145/17356.17387
9. Gonzalez-Concejero C., Rodellar V., Alvarez-Marquina A., Icaya E. M. de, Gomez-Vilda P. A portable hardware design of a FFT algorithm, *Latin American applied research*, 2007, Vol. 37(1), pp. 79–82.
10. Slade G. The Fast Fourier Transform in Hardware: A Tutorial Based on an FPGA Implementation. Access mode: https://web.mit.edu/6.111/www/f2017/handouts/FFTtutorial121102.pdf
11. Radix-2 FFT – VHDL Implementation. Access mode: https://github.com/bugratufan/radix2-fft-vhdl
12. TMS320C55x Technical Overview. Access mode: https://www.ti.com/lit/ug/spru393/spru393.pdf?ts=1693041737254
13. McKeown M. FFT Implementation on the TMS320VC5505, TMS320C5505, and TMS320C5515 DSPs. Access mode: https://www.ti.com/lit/an/sprabb6b/sprabb6b.pdf?ts=1695025017385
14. Lyons R. G. Understanding Digital Signal Processing, Third Edition. Boston, Pearson Education, 2011, 858 p.
15. Ifeachor E. C., Jervis B. W. Digital Signal Processing: A Practical Approach, Second Edition. Boston, Addison Wesley, 1993, 779 p.
16. Spartan-6 Family Overview. Access mode: https://docs.xilinx.com/v/u/en-US/ds160
17. Spartan-6 FPGA Data Sheet: DC and Switching Characteristics. Access mode: https://docs.xilinx.com/v/u/en-US/ds162
18. Vijayakanthan K., Hemachandran K., Anand M., Hemachandran K. High Throughput and Mixed Radix N-Point Parallel Pipelined FFT VLSI Architectures for Advanced Wireless Communication, *International Journal of Grid and Distributed Computing*, 2020, Vol. 13(1), pp. 400–411.

DOI 10.15588/1607-3274-2024-1-18

UDC 004.93

LAMA-WAVELET: IMAGE IMPAINTING WITH HIGH QUALITY OF FINE DETAILS AND OBJECT EDGES

Kolodochka D. O. – Student of the Institute of Computer Systems, National University "Odessa Polytechnic", Odessa, Ukraine.

Polyakova M. V. – Dr. Sc., Associate Professor, Professor of the Department of Applied Mathematics and Information Technologies, National University "Odessa Polytechnic", Odessa, Ukraine.

ABSTRACT

Context. The problem of the image impainting in computer graphic and computer vision systems is considered. The subject of the research is deep learning convolutional neural networks for image inpainting.

Objective. The objective of the research is to improve the image inpainting performance in computer vision and computer graphics systems by applying wavelet transform in the LaMa-Fourier network architecture.

Method. The basic LaMa-Fourier network decomposes the image into global and local texture. Then it is proposed to improve the network block, processing the global context of the image, namely, the spectral transform block. To improve the block of spectral transform, instead of Fourier Unit Structure the Simple Wavelet Convolution Block elaborated by the authors is used. In this block, 3D wavelet transform of the image on two levels was initially performed using the Daubechies wavelet db4. The obtained coefficients of 3D wavelet transform are splitted so that each subband represents a separate feature of the image. Convolutional layer, batch normalization and ReLU activation function are sequentially applied to the results of splitting of coefficients on each level of wavelet transform. The obtained subbands of wavelet coefficients are concatenated and the inverse wavelet transform is applied to them, the result of which is the output of the block. Note that the wavelet coefficients at different levels were processed separately. This reduces the computational complexity of calculating the network outputs while preserving the influence of the context of each level on image inpainting. The obtained neural network is named LaMa-Wavelet. The FID, PSNR, SSIM indexes and visual analysis were used to estimate the quality of images inpainted with LaMa-Wavelet network.

Results. The proposed LaMa-Wavelet network has been implemented in software and researched for solving the problem of image inpainting. The PSNR of images inpainted using the LaMa-Wavelet exceeds the results obtained using the LaMa-Fourier network for narrow and medium masks in average by 4.5%, for large masks in average by 6%. The LaMa-Wavelet applying can enhance SSIM by 2–4% depending on a mask size. But it takes 3 times longer to inpaint one image with LaMa-Wavelet than with LaMa-Fourier network. Analysis of specific images demonstrates that both networks show similar results of inpainting of a homogeneous background. On complex backgrounds with repeating elements the LaMa-Wavelet is often more effective in restoring textures.

Conclusions. The obtained LaMa-Wavelet network allows to improve the image inpainting with large masks due to applying wavelet transform in the LaMa network architecture. Namely, the quality of reconstruction of image edges and fine details is increased.

KEYWORDS: image inpainting, wavelet transform, LaMa network, Daubechies wavelet, Fréchet inception distance, wavelet convolution.

ABBREVIATIONS

CNN is a convolutional neural network;

MSNPS is Muli-Scale Neural Patch Synthesis;

GLCIC is Globally and Locally Consistent Image Completion;

CoModGAN is co-modulated generative adversarial network;

LaMa-Fourier is Large Mask Inpainting with Fourier Convolutions;

LaMa-Wavelet is Large Mask Inpainting with wavelet transform;

FFC is a fast Fourier convolution;

FFT is a fast Fourier transform;

iFFT is an inverse fast Fourier transform;

LLL, LLH, LHL, HLL, LHH, HLH, HHL, HHH are wavelet coefficient subbands;

DWT is a discrete wavelet transform;

iDWT is an inverse discrete wavelet transform;

BN is a batch normalization layer;

ReLU is a rectified linear unit;

LPIPS is Learned Perceptual Image Patch Similarity;

FID is Fréchet inception distance;

PSNR is a peak signal-to-noise ratio;

MSE is a mean square error;

SSIM is a structural similarity index measure.

NOMENCLATURE

n, m is the number of image rows and columns;

(x,y) are coordinates of the image pixel;

$I(x,y)$ is a vector function representing an image by color channels;

$I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the red, green, blue color channels of an image;

$\circ$ is an element-by-element product of matrixes;

f_θ is an inpainting network;

$struct_{f\theta}$ is the architecture of the f_θ network;

$param_{f\theta}$ is the set of parameters of the f_θ network;

$I_{in}(x,y)$ is an inpainted three-channel color image;

L_2 is a pixel loss;

L_P is a perceptual loss;

L_D is a competitive loss;

k, a, b are the coefficients controling the impact of each of the losses;

m_r and m_g are vectors of mean feature values for real and generated image sets, respectively;

R_r and R_g are the covariance matrices of the features of the real and generated sets of images;

DOI 10.15588/1607-3274-2024-1-19

tr is the trace of the matrix;

L is the number of intensity levels on the image;

$I(v, w)$, $c(v, w)$, $s(v, w)$ are the luminance, contrast, and structure between images v and w ;

m_v, m_w are the local means of images v and w ;

σ_v, σ_w are the standard deviations of images;

σ_{vw} is the cross-covariance for images v and w.

C_1, C_2, C_3, α, β and γ are the positive constants.

INTRODUCTION

In computer vision and computer graphics systems, there is a need to inpaint missing areas of the image. This problem arises in the case of faded colors or physical damage of the surface on which the image was located. In another case, there is a need for removing unwanted objects from the image in such a way that the resulted image looks realistic and fits the context. For example, when photographing, distracting objects in a scene such as strangers and objects that get in the way are usually unavoidable but at the same time undesirable for users. Before sharing a photo, users may want to make some changes, such as removing distracting elements from the scene or adjusting the position of objects in the image for a better composition. Image inpainting techniques provide automatic filling of missing regions of an image with a plausible hypothesis. These techniques are used in many real-world tasks, such as removing distracting objects, restoring damaged parts, and filling the missing areas of images [1, 2].

The object of research is inpainting of real scene images in computer vision and computer graphics systems.

Due to the development of modern technologies and the increase in computing power, a number of image inpainting methods based on deep learning CNNs have been elaborated, which can generate missing regions of the image with good global consistency and local fine textures. Thus, CNNs Context Encoder, MSNPS, GLCIC, DeepFill v1–2 differ in such characteristics as speed, the size of the processed image, and the quality of filling of image regions [3]. The disadvantage of the listed methods is that when using large masks, the result becomes unsatisfactory in terms of generating both image context and texture.

The subject of the research is methods of image inpainting using CNNs of deep learning.

Unlike the Context Encoder, MSNPS, GLCIC, DeepFill v1–2 methods, the LaMa-Fourier [4] is able to obtain a good result even when the missing areas occupy most of the image. In general, CNNs for image inpainting usually achieve better results by complicating the network architecture or by dividing it into sub-networks with separate tasks. LaMa-Fourier, on the contrary, uses a single network and a fewer variables [4].

The advantages of the LaMa-Fourier method are a higher speed of image processing and network training; better quality than other neural network methods when using narrow masks; better quality of large mask inpainting of spectral textures. The disadvantage of the LaMa-Fourier network is the insufficient quality of inpainting of fine details of images and edges of objects. To eliminate this shortcoming, it is appropriate to use a wavelet transform representing both global and local features of images.

The aim of the paper is to improve the quality of image inpainting in computer vision and computer graphics systems with applying wavelet transform in the LaMa-Fourier network architecture.

1 PROBLEM STATEMENT

The color natural image is represented as $I(x,y)=(I_R(x,y), I_G(x,y), I_B(x,y)\}$, where $x=1, \ldots, n$; $y=1, \ldots, m$. Then each pixel of the image is described by three features $I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ which take values from the interval [0, 255]. A mask is introduced to represent the missing areas of the image. This is a binary image $M(x, y)$ of the same size as each channel of the original image. The mask is element-by-element producted by image features. Then, using a mask, the image with missing areas is represented as $I_M(x,y)=(I_R(x,y)\circ M(x, y), I_G(x,y)\circ M(x, y), I_B(x,y)\circ M(x, y))$. It is necessary to transform the image $I_M(x,y)$ so as to fill missing areas. In this case, the resulting image should approximate the original one in the sense of some criterion [4].

Let the CNN $f_\theta=\{struct_{f\theta}, param_{f\theta}\}$ was preliminarily designed to inpaint the images. The set $struct_{f\theta}$ includes blocks with layers of the designed network. Taking $I_M(x,y)$ the inpainting network processes the input in a fully-convolutional manner, and produces $I_{in}(x,y) = f_\theta(I_M(x,y))$ which approximates the original image $I(x,y)$.

The problem of the refining CNN architecture is as follows. It is necessary to make structural changes to the existing architecture $struct_{f\theta}$ of the network $f_\theta(\bullet)$. These changes should improve the image inpainting performance compared to the initial $f_\theta(\bullet)$ network after training the parameters of the resulting network [5, 6]. The training is performed on a dataset of (image $I(x,y)$, mask $M(x, y)$) pairs obtained from natural images and synthetically generated masks.

2 REVIEW OF THE LITERATURE

The analysis of CNNs for image inpainting showed that such methods are primarily focused on the properties of processed images. Also, the architectures of the used CNNs are determined by the computing power available to the researcher and the quality requirements for the inpainted images.

Thus, the methods [7, 8] are recommended for the filling localized missing areas of small-sized images with not very high quality of the result. These methods use one CNN, which is quite easy to train on another class of images. It is not require relatively significant time and computing power. For example, in [7] the Context Encoder was designed on the basis of a generative-competitive network. This CNN includes a fully connected layer. Due to convolutional layers, all locations of spatial objects on the previous layer contribute to the location of spatial objects on the current layer. Thus, the

DOI 10.15588/1607-3274-2024-1-19

network can learn the relationship between the locations of objects. Also, the Context Encoder can be trained to understand the overall context of the entire image.

A significant number of image inpainting methods [7, 9–11] require the correct shape of the missing area. However, the PartialConv algorithm [8] is able to fill several areas of arbitrary shape at once. PartialConv is a Unet type network which differs by the applying of partial convolutions in the convolutional layers. When partial convolution is used, the processing image fragment is first multipled by a binary matrix element by element, and then only a filter mask is applied. The disadvantage of the PartialConv is the reducing of the quality of filling missing regions with few details lagging behind each other.

In contrast to CNNs that use one subnet, the MSNPS [9] fills the missing regions of images with the help of two CNNs applied sequentially. The first CNN is used to generate the global context, the second CNN is used to further add local texture. This approach improves the quality of the inpainted images, but significantly increases the training time. MSNPS is a further developing of [7], but differs in higher details of local textures due to the use of a separate CNN for their generation.

Methods [10–14] use ensembles of three or more CNN to achieve high-quality image inpainting. But they require the significant computing power and much training time.

In [10] the GLCIC consist of three CNNs to inpaint the images. One CNN is applied to fill missing regions of the image and two auxiliary networks is used as local and global discriminators. The latter networks are used only while training. Their role is to assess the realism of the resulting image by comparing the original image areas with the inpainted ones. At the same time, the generative network is learned to deceive the discriminators, and the discriminators are learned to better identify unrealistic images.

The GLCIC fills missing regions not only based on the current image, but also based on the images used while training. A fully connected layer is not used, which significantly reduces the time to inpaint an image as compared with [7, 9] and practically removes the limitation on the size of the input image. The disadvantage of GLCIC is that, in addition to the generative network, discriminators must also be trained.

In [11], the DeepFill v1 network was proposed as a sequential combination of the networks from [9, 10]. The peculiarity of the DeepFill v1 is that when searching for suitable areas for copying, not only similar areas are determined, but also the contribution of all visible objects to the missing area is estimated. As a result, a combination of the most significant visible objects is used to fill the missing region. This enhances the quality of the obtained result. However, for correct inpainting the missing area of the image must be square.

Instead of the filling of missing regions by global context generation and local texture generation EdgeConnect [12] is proposed to generate edge map and to inpaint an image based on the obtained edges. A discriminator is used while training of each of the two subnets of the EdgeConnect [12]. The EdgeConnect can fill missing regions of arbitrary shape and shows better results than previous methods when generating objects of complex shape. But the EdgeConnect is needed to train discriminators in addition to generative networks.

The DeepFill v2 network [13] is based on the DeepFill v1, EdgeConnect, and PartialConv networks. DeepFill v2 sequentially applies three CNNs. First network is designed to generate the global context. Second network generates a local texture based on the global context of the image. Third network is a discriminator assessing the realism of the resulting image. After training the DeepFill v2 network is able to fill missing regions not only on the basis of other parts of the image, but also to evaluate the contribution of surrounding objects to the content of the missing area. This network can process regions of arbitrary shape, and use an edge map when generating objects. The DeepFill v2 shows better performance in terms of inpainted image quality and processing time compared to [7–12]. However, the training of an ensemble of three CNNs with more than 4 million parameters requires significant time and computer resources.

The CoModGAN network [14] architecture is similar to DeepFill v2, but greatly enhanced. The connectivity of filled regions to the context is better compared to DeepFill v2, but visible artifacts are possible in the center of the inpainted region. The CoModGAN network on average shows better results than DeepFill v2, but due to increasing the number of parameters to 108 million. Then, the time of network training and image inpainting increases several times.

The considered CNNs improves of the quality of image inpainting by complicating the network architecture and/or by processing taking into account, in addition to the color components, other features of the images. In particular, the texture or the edges of objects are processed. The main feature of the LaMa-Fourier network compared to considered networks is the use of a new type of a convolutional layer which is FFC [4]. It allows to significantly increase the logical connectivity of the filling missing regions with known image regions and at the same time to reduce the number of network parameters several times.

The LaMa-Fourier uses the FFC, learning mask generator, and loss function different from previously proposed methods. Its architecture is simpler compared to DeepFill v2 and CoModGAN. LaMa-Fourier has 27 million parameters, and is faster than CoModGAN due to fewer layers. At the same time, it shows better results, the absence of visible artifacts and variations of the texture structure, especially when restoring large areas and spectral textures. However, the LaMa-Fourier requires more computational resources for the implementation of FFC compared to convolution. In addition, there is

DOI 10.15588/1607-3274-2024-1-19

sometimes a slight blurring of the areas inpainted by the LaMa-Fourier, which is a side effect of applying the Fourier transform [15].

Therefore, in the paper it is proposed to enhance the LaMa-Fourier network by applying of wavelet transform which can be considered as a generalization of the Fourier transform [16]. Wavelet decomposition is performed with the help of functions with a limited extension to get information about image details.

3 MATERIALS AND METHODS

The architecture of the LaMa-Fourier network is shown in Fig. 1 [4]. The network is inputted an image with missing areas and a mask with pixels need to be inpainted. Next, the image is reduced by a factor of 3 and passes through nine residual blocks (Fig. 1, a). After that, the image is enlarged to its original size and outputted [4].

In the residual block, the FFC is applied twice to the image and the result is added to the original image. FFC decomposes the image into local and global textures, which are further processed by convolution layers (Fig. 1, b). The global texture additionally passes through the spectral transform block. The outputs of the convolution layers are summed “cross over cross”. Then BN and the ReLU activation function are applied to them. The results of local and global texture processing are concatenated (Fig. 1, b) [4].

In the spectral transform block the image is Fourier transformed into the frequency domain, the real and imaginary parts are concatenated. Then the convolutional layer, BN and the ReLU activation function are applied sequentially (Fig. 1, c). The obtained result is splitted on the real and imaginary parts. Finally, the iFFT is applied, the result of which is the output of the block [4].

The LaMa-Fourier network is able to represent the general structure of images. But difficulties arise with the inpainting of fine high-frequency details and with the generation of the image edges. There may also be problems when reproducing complex textures, such as small leaves, thin fabric fibers, or detailed patterns (Fig. 1, d–f). Difficulties in ensuring similarity between the inpainted region and the existing texture may be related to the fact that the Fourier transform traditionally works better with low and medium frequencies than with very high ones [15].

As an alternative to the Fourier transform, to overcome the mentioned shortcomings, it is appropriate to use the wavelet transform [17, 18]. Then, it is necessary to define the block of the network to which changes will be made. Since the FFC decomposes the image into global and local texture, it was decided to improve only the network block, processing the global context of the image, namely, the spectral transform block. Applying the wavelet transform to the local context can increase the noise level in the image.

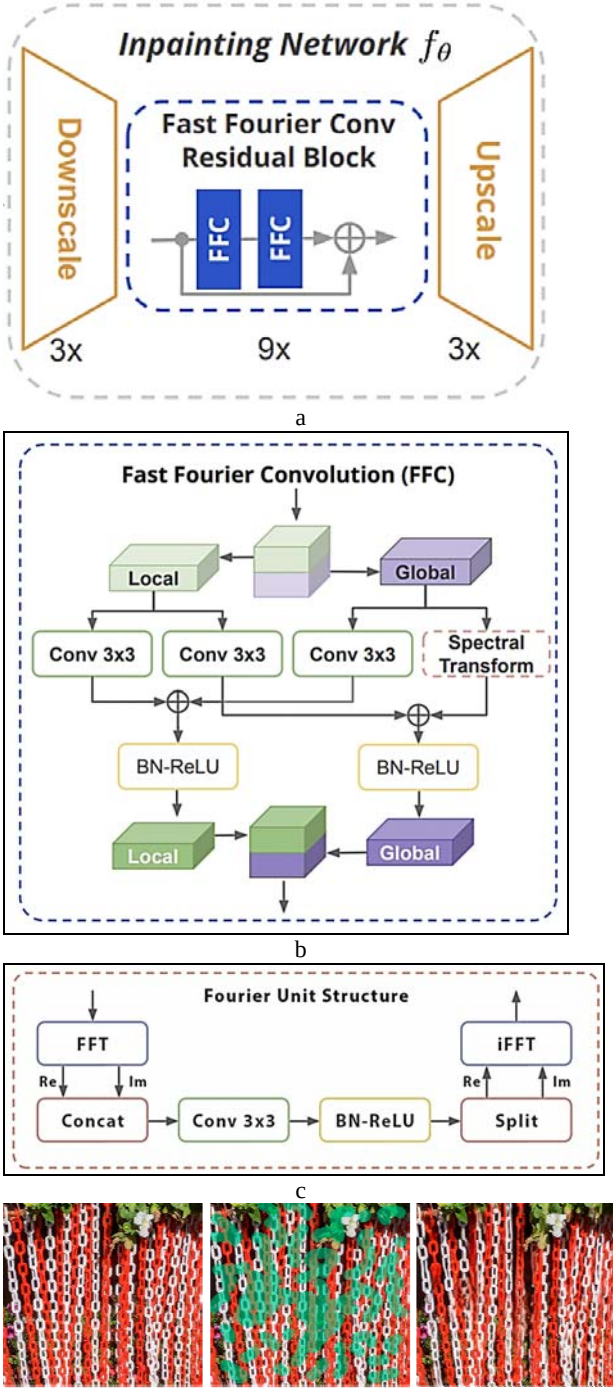


Figure 1 – LaMa-Fourier network architecture: a – Residual Block; b – FFC; c – Fourier Unit Structure [4]. LaMa-Fourier inpainting example: d – original image; e – original image and mask; f – inpainted image

To improve the block of spectral transform the Simple Wavelet Convolution Block elaborated by the authors is used instead of Fourier Unit Structure (Fig. 2). In this block, 3D wavelet transform of the image on two levels using the Daubechies wavelet db4 (Fig. 3, a) was initially performed.

DOI 10.15588/1607-3274-2024-1-19

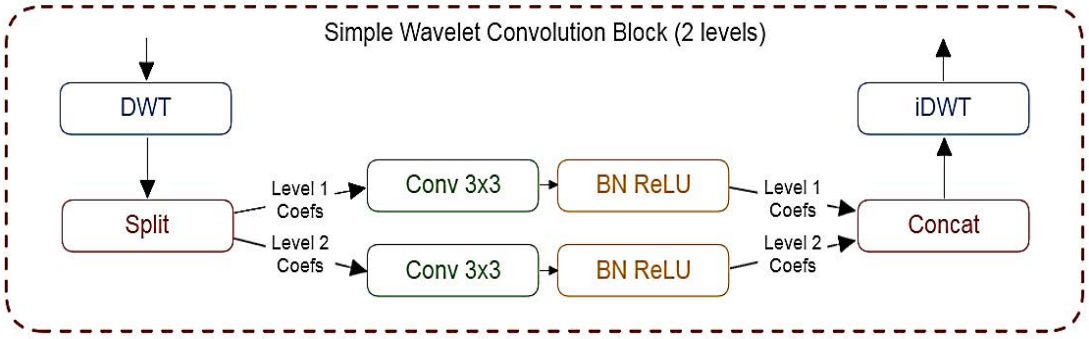


Figure 2 – Simple Wavelet Convolution Block

As a result of this transform, the eight subbands of coefficients at each level were obtained [19]. These are LLL, LLH, LHL, HLL, LHH, HLH, HHL, HHH subbands (Fig. 3, b). The main advantage of using a 3D wavelet transform, as opposed to applying a 2D transform separately for each image channel, is its ability to analyze inter-channel image correlation. This allowing to take into account color details on an inpainted image.

The obtained coefficients of 3D wavelet transform are splitted so that each subband represents a separate feature of the image. Convolutional layer, BN, and ReLU activation function are sequentially applied to the results of splitting of coefficients on each level of wavelet transform (Fig. 2). The number of convolutions in the convolutional layer was equal to the number of subbands at the corresponding level of the wavelet transform. After applying the ReLU activation function, the obtained subbands of wavelet coefficients are concatenated and the iDWT is applied to them, the result of which is the output of the block. Note that the wavelet coefficients at different levels were processed separately. This made it possible to reduce the computational complexity of calculating the network outputs while preserving the impact of each level to image inpainting.

The LaMa network uses the loss function, which is specially designed to solve the problem of filling large missing regions. This loss function L_{final} combines L_2, L_P and L_D to ensure the realism, semantic integrity and structural continuity of the inpainted regions, which corresponds to the human perception of image [4]:

$$L_{final} = kL_2 + aL_P + bL_D.$$

The gradient penalty [4] is not used to reduce amount of computation. The MSE between the original and restored images was used to estimate L_2 pixel loss [20]. For perceptual loss of L_P, LPIPS is used, which evaluates the perceptual similarity between the inpainted and original images using a pre-trained neural network [21].

The discriminator is used to estimate competition loss L_D. This additional CNN is trained in parallel with the basic network to distinguish between real and generated images. Based on this evaluation, the discriminator tunes the basic network coefficients to improve the realism of the generated images. Then, the L_D are estimation of the error in the global and local textures computed from the discriminator output [22].

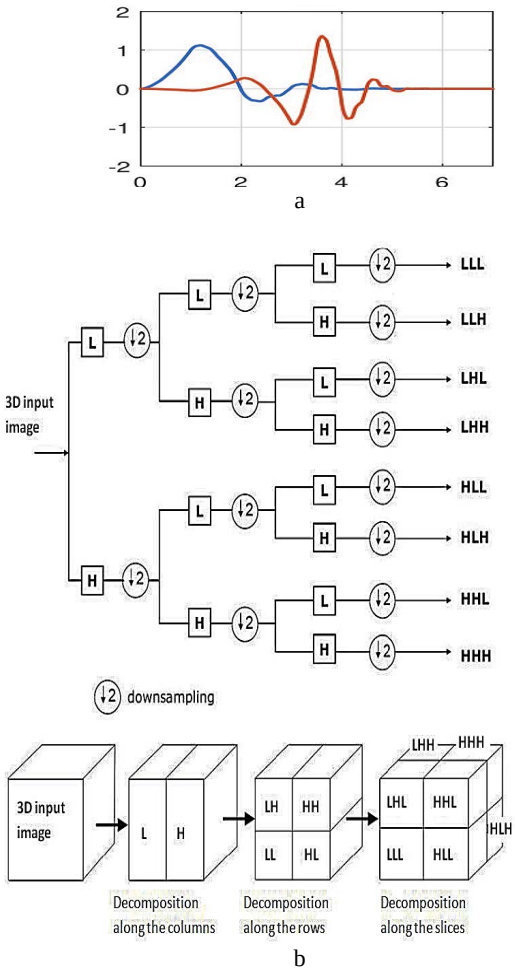


Figure 3 – Elements of LaMa-Wavelet network architecture: a – Daubechies db4 scaling function (blue) and wavelet (red) [16]; b – subbands of 3D wavelet transform coefficients [19]

4 EXPERIMENTS

At the first stage of the experiment, the LaMa-Wavelet network was trained to inpaint test database images. The Google Colab environment with a pre-configured NVIDIA Tesla T4 GPU, which has 16 GB of GDDR6 memory and 2,560 CUDA cores, was used. The T4 is optimized for machine learning computing and supports

DOI 10.15588/1607-3274-2024-1-19

NVIDIA Turing Tensor Cores to accelerate tensor operations. The Google Colab environment uses processors to process data and to interact with the GPU. The Google Colab environment provides about 60 GB of RAM. But no more than 4 GB was used for training, because the system is more demanding of video memory and most of the RAM is used only for mask generation. Google Drive data storage was used to store the training and testing datasets to easy access to them.

The LaMa-Wavelet was trained using the Adam optimizer with the following parameters of the L_{final} loss function: k=10, a=100, b=30.

While network training, an initial image and a mask are selected. A mask is a binary image on which black pixels correspond to pixels of the missing region of the initial image, and white pixels correspond to known pixels. Pixels, which will be inpainted later, were removed from the input image according to the mask. Next, the obtained image and the mask were fed to the input of the trained network. The image with filled missing regions was outputted by the network. The original image was superimposed on this generated image according to the mask. Thus, the result was an image in which the known pixels were copied from the original image, and the pixels of the missing regions were generated by the network.

The 16,000 images from databases [23, 24] were used to train the LaMa-Wavelet network. These images were scaled to a size of 256x256 pixels and randomly splitted into training and testing sets in the ratio of 95% to 5%. For each image, with a probability of 0.5, either a mask of 1–4 rectangles with sides of 30–150 pixels, or a mask of 1–5 straight lines 10–200 pixels long, 1–100 pixels wide and with a slope from 0 to 2π was generated. The sizes of the masks were variable, from narrow (10% of the image pixels) to large (80% of the image pixels). This ensured that the network was trained at different levels of inpainting complexity. Masks were generated with a random uniform distribution over the entire area to ensure uniform coverage of different image areas.

Evaluation of the results of the first stage of the experiment was performed by the FID score [25]. It was calculated using the Inception v3 network for two sets of images, specifically, real image set and set of images with generated regions. When this network obtained the features from the real and generated image sets, the FID is calculated as

$$FID=\| m_r - m_g\|^2 + \mathrm{tr}(R_r + R_g - 2(R_r R_g)^{1/2}).$$

FID measures the distance between the feature distributions of real images and images inpainted by the network. Lower FID value means that the feature distributions are closer, indicating more similarity between the generated and real images. This metric takes into account both the variability and the quality of the generated images [25].

The original LaMa-Fourier network is balanced in terms of image inpainting quality and processing time. Training of this model was completed according to a standard protocol, providing a reliable baseline for comparison [26]. After 128 epochs the training of the LaMa-Fourier network was completed. But the dependence of loss function from epoch for the LaMa-Wavelet still showed a downward trend. This indicated the possibility of further improvement of the loss provided the training is continued.

Continued training of the LaMa-Wavelet to 212 epochs was intended to approach or even exceed the FID value of the trained LaMa-Fourier network. Training throughout 212 epochs reduced the FID of the LaMa-Wavelet to about 8 on the training set and to 24 on the validation set. This equalizes it with the FID of the LaMa-Fourier network. The similarity of the FID for the LaMa-Fourier and LaMa-Wavelet networks indicates that the improvement in image inpainting by the LaMa-Wavelet can only be achieved through long training.

At the second stage of the experiment, the images of testing set are inpainted using the LaMa-Fourier and LaMa-Wavelet networks. Then the inpainted images were compared with the original images. For this, three separate categories of masks were formed, namely, narrow, medium, and large, covering 15%, 40%, 65% of the image area, respectively. Each of these mask categories represented a different level of image inpainting complexity. The test set included 2000 images from the Places2 dataset [23]. One mask from each category was generated for each image. After element-by-element multiplication of images on masks, the inpainting was performed using LaMa-Fourier and LaMa-Wavelet networks.

To compare the inpainted images with the original images, the FID was first calculated. It has been observed that the FID evaluates the overall similarity of the generated and original images, but does not focus on the recovering of edges or details. To solve this problem, two additional indexes, PSNR and SSIM, are applied [20].

PSNR compares the original and reconstructed image in terms of differences between them at the pixel level. In the context of image inpainting, PSNR indicates how well edges and fine details are reconstructed. It is estimated as the ratio between the maximum possible power of an original image and the MSE between original and inpainted images if minimum intensity level supposes to be 0 [20]:

$$PSNR = 10\log_{10}((L-1)^2/MSE).$$

SSIM provides a perceptually relevant estimation of image quality considering the differences in image structure, texture, and contrast. This is critical for preserving the natural appearance of image edges and textures. The SSIM is calculated based on the luminance term, contrast term and structural or correlation term as [20]:

DOI 10.15588/1607-3274-2024-1-19

$$SSIM(v, w)=l\ (v, w)^{\alpha}\ c\ (v, w)^{\beta}\ s\ (v, w)^{\gamma},$$
$$l\ (v, w) = (2m_v m_w + C_1)/(m_v^2 + m_w^2+C_1),$$
$$c\ (v, w) = (2\sigma_v \sigma_w +C_2)/(\sigma_v^2 + \sigma_w^2+ C_2),$$
$$s\ (x, y)=(\sigma_{vw} +C_3)/(\sigma_v \sigma_w +C_3),$$

If α=β=γ=1, then the index is in normalized scale with values between 0 to 1.

The PSNR and SSIM allowed a more detailed estimation of the inpainting performance, especially in terms of edge quality and structural integrity.

At the third stage of the experiment, the results of the inpainting of specific images by LaMa-Fourier and LaMa-Wavelet networks were compared. 3–5 images were selected with different complexity of background, namely, uniform background, background with structural texture (with repeating patterns) [26], complex background with repeating objects. To demonstrate how well the networks performed in texture, color, and edge recovery, each image was processed using the original LaMa-Fourier and LaMa-Wavelet network trained on 212 epochs. Inpainted images were evaluated visually and using PSNR and SSIM.

5 RESULTS

At the first stage of the experiment, the results of training of LaMa-Fourier and LaMa-Wavelet networks were evaluated using the FID score on training and validation sets. In addition, the training epoch time and image inpainting time were estimated. Image inpainting time is averaged for a set of 25 images of size 1024x1024 pixels (Table 1).

Table 1 – The LaMa-Fourier and LaMa-Wavelet training results

CNN	FID on training set	FID on validation set	Epoch time, minutes	Image inpainting time, seconds
LaMa-Fourier	8.2	25	40	2.2
LaMa-Wavelet	9.2	32	150	6.6

At the second stage of the experiment, the dependencies of the FID, PSNR, and SSIM from the size of the missing areas were researched (Table 2).

The PSNR of the CelebA-HQ [27] and Plases2 [23] datasets images inpainted by the methods known from the literature are given in Table 3 [28]. Note, however, that the results of Table 3 were obtained under significantly different experimental conditions and are used as collating data.

Table 2 – The LaMa-Fourier and LaMa-Wavelet testing results

CNN	Epochs	FID	PSNR	SSIM
Narrow masks				
LaMa-Fourier	128	21.8	25.68	0.7811
LaMa-Wavelet	128	24.7	26.58	0.8066
LaMa-Wavelet	212	21.3	26.82	0.8088
Medium masks				
LaMa-Fourier	128	24.3	25.04	0.8232
LaMa-Wavelet	128	31.1	25.88	0.8367
LaMa-Wavelet	212	24.8	26.19	0.8394
Large masks				
LaMa-Fourier	128	32.7	22.16	0.7857
LaMa-Wavelet	128	39.7	22.96	0.7973
LaMa-Wavelet	212	32.1	23.48	0.7999

Table 3 – The PSNR of images from the CelebA-HQ [27] and Plases2 [23] datasets inpainted by the methods known from the literature [28]

CNN, reference, publication year	CelebA-HQ images 256x256 pixels		
	Narrow masks	Medium masks	Large masks
LaMa-Fourier [4], 2021	22.7	34.1	27.8
CoModGAN [14], 2021	35.9	48.4	64.4
DeepFill v2 [13], 2019	37.0	45.3	43.0
EdgeConnect [12], 2019	29.2	40.5	34.7
	Places images 512x512 pixels		
LaMa-Fourier [4], 2021	12.7	11.7	12.0
CoModGAN [14], 2021	16.3	12.4	10.4
DeepFill v2 [13], 2019	17.9	18.3	22.1
EdgeConnect [12], 2019	18.9	21.9	30.5

At the third stage of the experiment, the results of the inpainting of specific images by the LaMa-Fourier and LaMa-Wavelet networks were obtained (Fig. 4–9). The corresponding values of the SSIM and PSNR are shown in Table 4.

Table 4 – The PSNR and SSIM of specific images inpainted by LaMa-Fourier and LaMa-Wavelet

Image	LaMa-Fourier		LaMa-Wavelet	
	PSNR	SSIM	PSNR	SSIM
Fig. 4, a	35.04	0.90	35.30	0.90
Fig. 4, d	34.58	0.87	35.82	0.89
Fig. 4, g	24.54	0.90	35.10	0.90
Fig. 4, j	38.20	0.98	40.01	0.99
Fig. 4, m	22.91	0.92	22.52	0.91
Fig. 4, p	32.88	0.97	33.19	0.96
Fig. 5, a	28.84	0.96	28.96	0.95
Fig. 5, d	24.89	0.94	24.78	0.93
Fig. 6, a	23.67	0.88	22.96	0.88
Fig. 6, d	19.43	0.78	22.07	0.72
Fig. 7, a	20.39	0.88	22.39	0.87
Fig. 7, d	19.59	0.88	19.47	0.87
Fig. 8	13.61	0.84	15.03	0.85

6 DISCUSSIONS

Analysing Table 1 it should be noted the follow. The LaMa-Wavelet network requires more time for training and image inpainting after training. Namely, the LaMa-Wavelet network requires 3.75 times more time per training epoch as compared with LaMa-Fourier. After training it takes 3 times longer to inpaint one image with LaMa-Wavelet than with LaMa-Fourier network. It is noticed that a significant part of the training time is spent on calculating the wavelet transform.

DOI 10.15588/1607-3274-2024-1-19

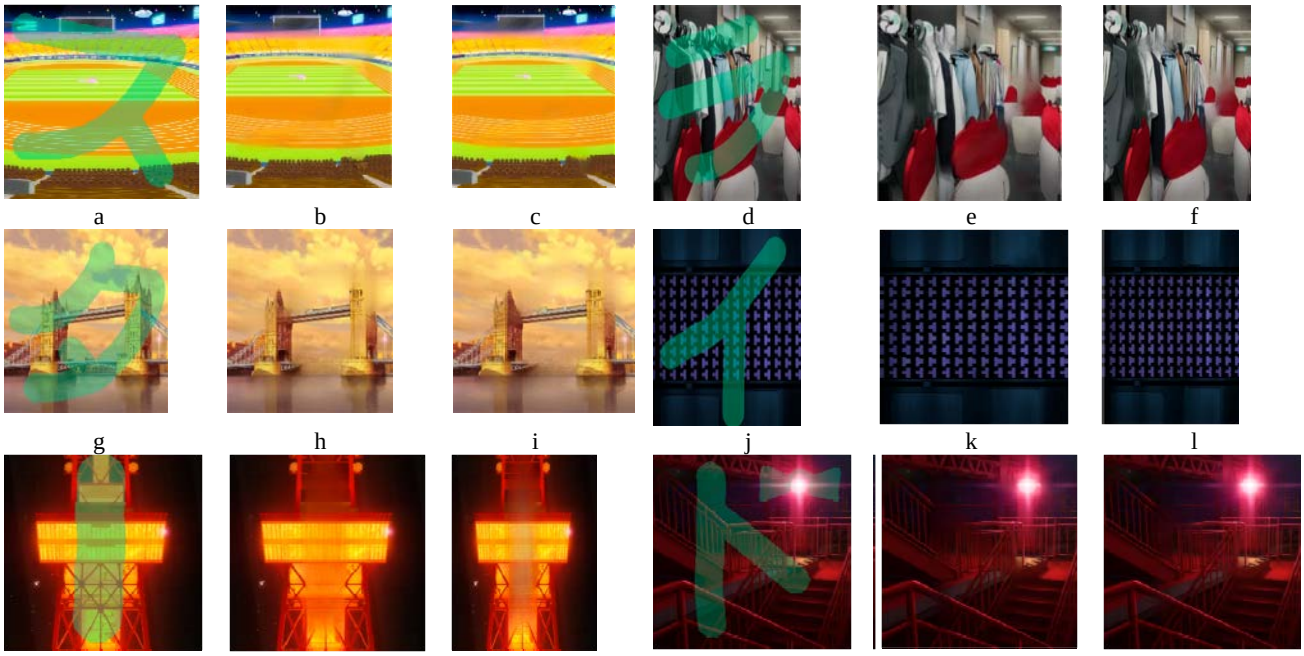


Figure 4 – Images with random masks: a, d, g, j, m, p – original image and mask; b, e, h, k, n, q – image inpainted with LaMa-Fourier; c, f, i, l, o, r – image inpainted with LaMa-Wavelet

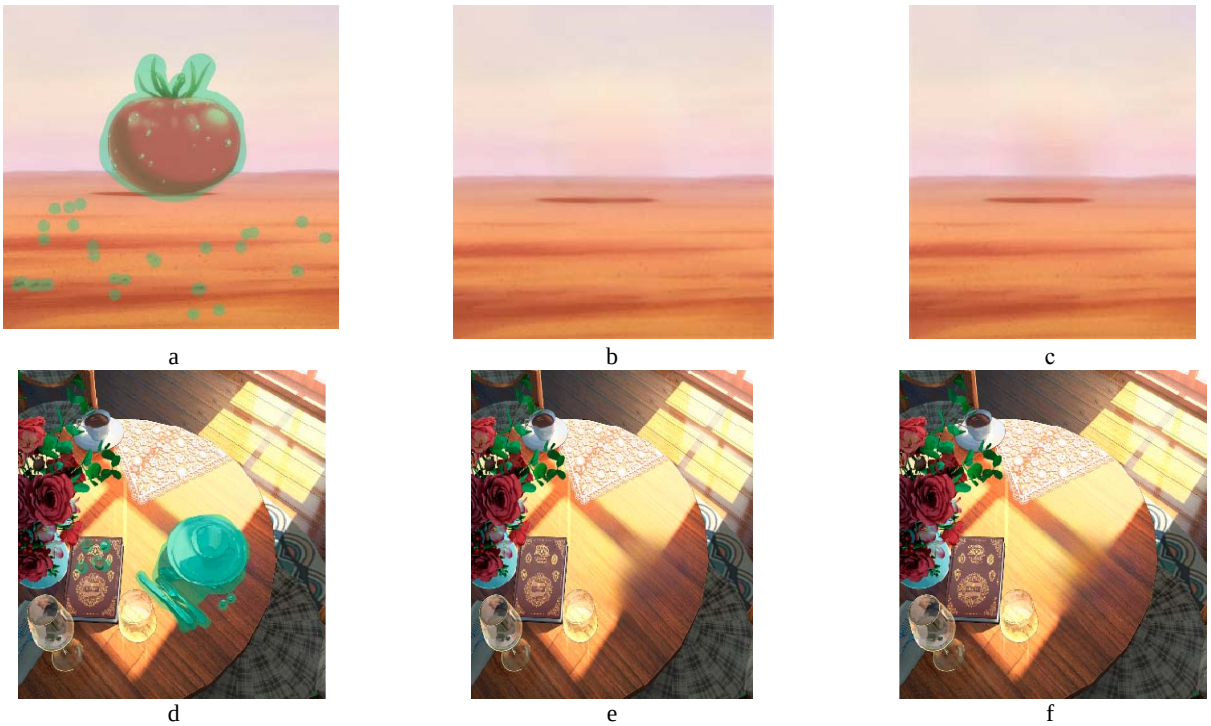
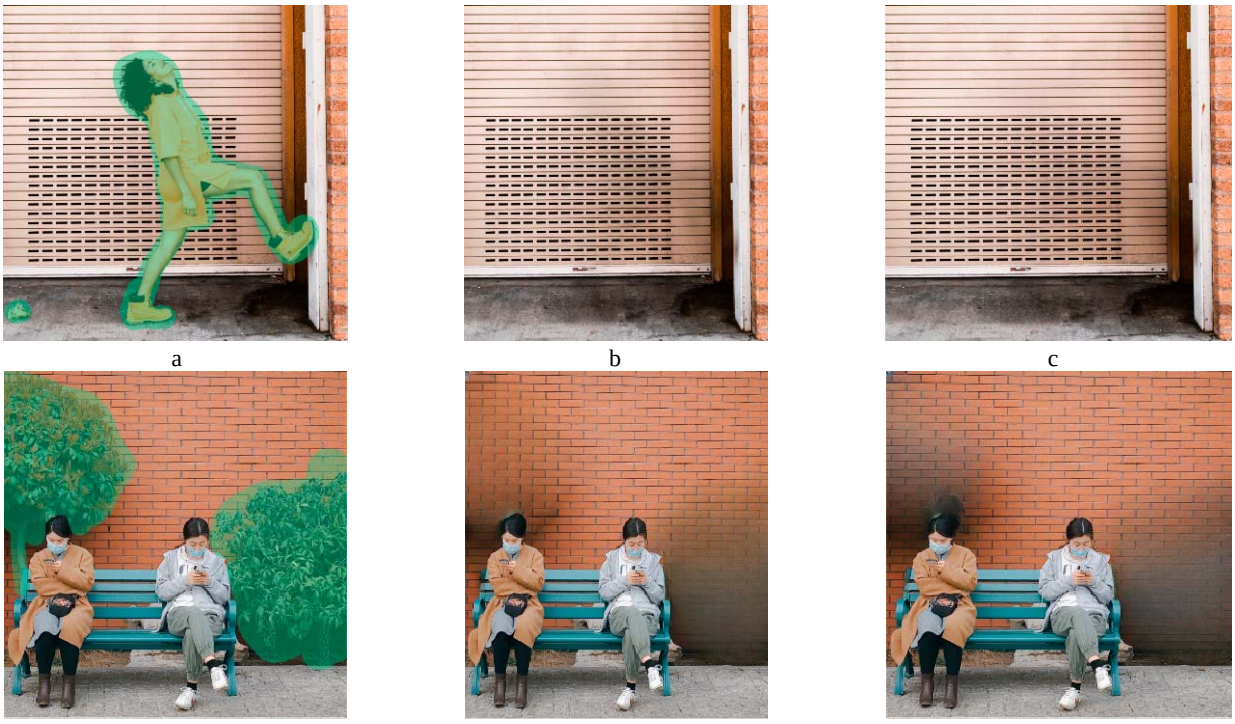


Figure 5 – Images with homogeneous background: a, d – original image and mask; b, e – image inpainted with LaMa-Fourier; c, f – image inpainted with LaMa-Wavelet

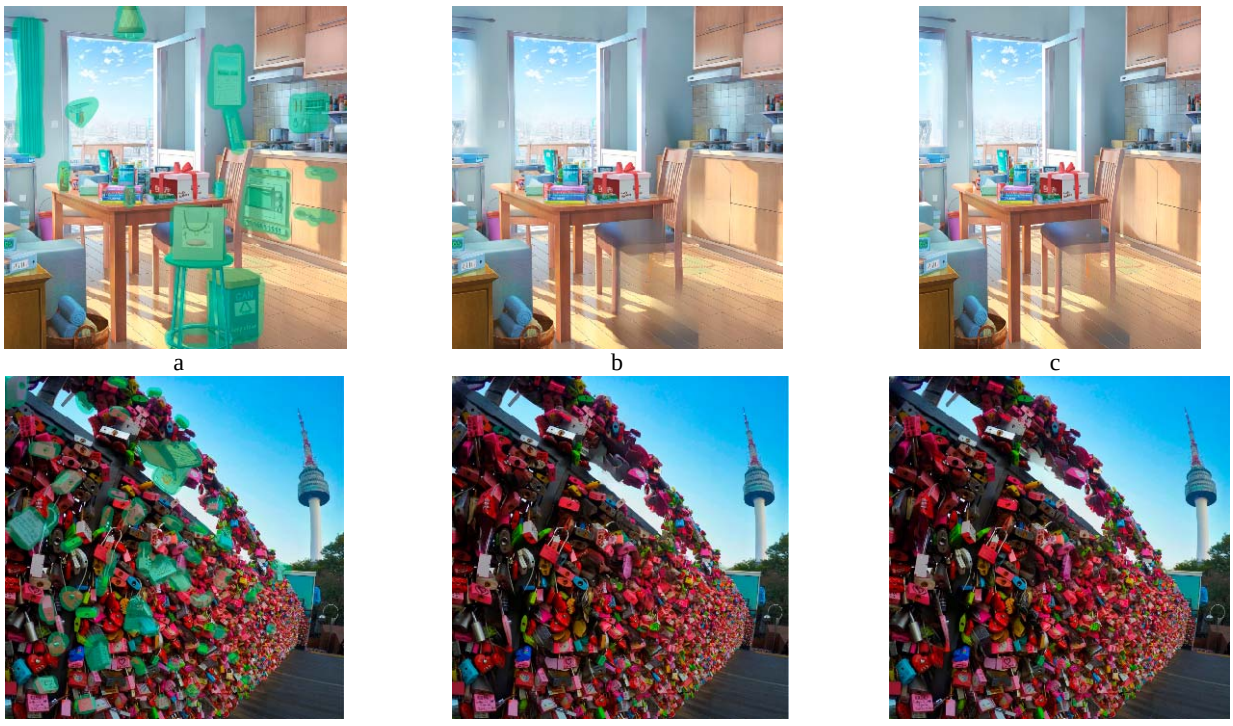
DOI 10.15588/1607-3274-2024-1-19



a b c

d e f

Figure 6 – Images with periodical background: a, d – original image and mask; b, e – image inpainted with LaMa-Fourier; c, f – image inpainted with LaMa-Wavelet

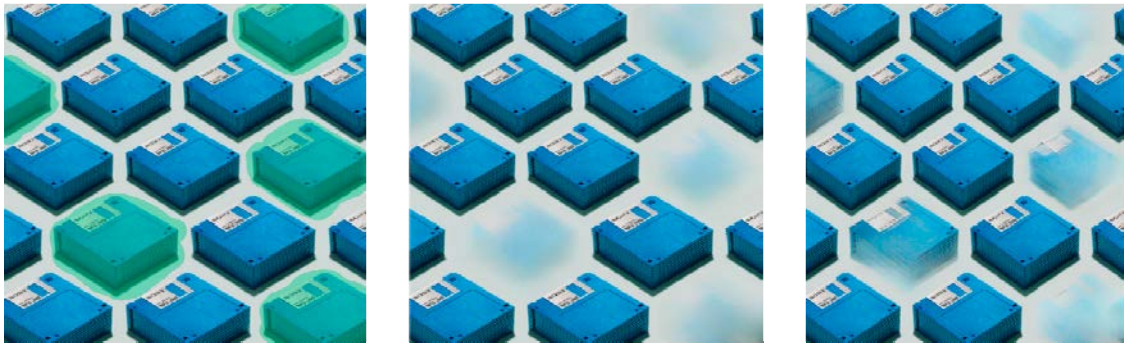


a b c

d e f

Figure 7 – Images with complex background: a, d – original image and mask; b, e – image inpainted with LaMa-Fourier; c, f – image inpainted with LaMa-Wavelet

DOI 10.15588/1607-3274-2024-1-19



a b c

Figure 8 – Image with generation artifacts: a – original image and mask; b – image inpainted with LaMa-Fourier; c – image inpainted with LaMa-Wavelet

The LaMa-Wavelet network is worse in terms of the training quality than LaMa-Fourier. Specifically, the FID on training and validation sets of images has increased by 12% and 28% respectively.

The analysis of the Table 2 showed that FID values obtained by the LaMa-Fourier and LaMa-Wavelet networks on test set of images are similar for narrow, medium, and large masks. Thus LaMa-Wavelet shows higher generalization ability than LaMa-Fourier. The PSNR of images inpainted using the LaMa-Wavelet exceeds the results obtained using the LaMa-Fourier network for narrow and medium masks in average by 1.15 dB (4.5%), for large masks in average by 1.3 dB (6%). The LaMa-Wavelet can enhance SSIM in average by 2–4% depending on a mask size.

In addition, it was noted that the quality improvement of inpainted images is largely determined by their content and properties. Thus, for structural textures [26] or objects against the background of such textures, the PSNR of images inpainted using the LaMa-Wavelet exceeds the results obtained using the LaMa-Fourier by 1.2–2.7 dB (3.5–14%). For objects on uniform background, the PSNR of images reconstructed with the LaMa-Wavelet is improved by 1.8–9.6 dB (9.8–41%) compared to the results obtained using the LaMa-Fourier. The improvement range of SSIM is less. This may mean that the LaMa-Wavelet is more capable of restoring fine details and edges of objects than inpainted the image structure.

Analysing of the LaMa-Wavelet network performance for different mask sizes it was noticed that there is a tendency for higher quality of inpainting of images with a low number of details and straight lines, especially with narrow masks. The inpainting of complex textures such as grass, leaves, branches or large numbers of people is difficult for both LaMa-Fourier and LaMa-Wavelet networks. The deformation of the image color in the large filled regions appears more frequently if the LaMa-Wavelet network has been used.

In the case of narrow masks, both networks show similar quality. However the LaMa-Wavelet shows a significant improvement in the inpainting of large missing areas relative to the LaMa-Fourier.

Let compares the results of the inpainting of specific images by the LaMa-Fourier and LaMa-Wavelet networks. At first several images with randomly selected narrow and medium masks were visually analyzed (Fig. 4). It was once again confirmed that, in general, LaMa-Wavelet shows better texture inpainting than LaMa-Fourier, but when the size of the masks increases, color deformation begins to appear. Also, the LaMa-Wavelet combines parallel lines less if they are close, and better preserves the bends of curved lines.

For images with a uniform background (Fig. 5) the numeric estimates are similar. There is a more pronounced background color defect at the place of the tomato and cup if the LaMa-Wavelet is applied.

The repeating background in the images (Fig. 6) is well reproduced by both networks. But when using larger masks, you can see, both visually and numerically, that LaMa-Wavelet generates better edges of texture elements, but loses in the generation of image color.

As the complexity of the image background increases, it can be seen that the LaMa-Wavelet network begins to lack context for the correct estimation of the expected generation (Fig. 7). In this regard, the quality of the edges of the inpainted regions reduces. However, under conditions of high complexity of the texture, the possibility of visual assessment is lost, since objects become, in principle, difficult to distinguish even for a person. Also, in such a scenario, the ability of the LaMa-Wavelet network to continue repeating textures becomes more of a problem than an advantage. This network starts mixing different textures in an attempt to continue them. This is seen in the example in Figure 8, where large duplicate objects need to be removed from an image. Instead of removing objects, the LaMa-Wavelet network generates an average between the background and the original texture in an attempt to restore the texture. It can also be seen that the LaMa-Wavelet network is more seek to forced inpainting of textures than the LaMa-Fourier.

Thus, analysis of specific image inpainting confirmed the practical effectiveness of LaMa-Wavelet network as compared with LaMa-Fourier. In particular, when removing objects on a homogeneous background, both networks show similar results. However, on complex backgrounds with repeating elements, the LaMa-Wavelet

DOI 10.15588/1607-3274-2024-1-19

is often more effective in restoring textures, despite some cases of texture mixing.

CONCLUSIONS

The actual scientific and applied problem of an inpainting of the fine details and object edges has been solved when missing regions of images are filled by CNN.

The scientific novelty is the proposed method of natural image inpainting with LaMa-Wavelet network. Due to applying wavelet transform, the image inpainting with large masks based on the LaMa network is improved. Specifically, the quality of reconstruction of image edges and fine details is increased.

The practical significance of obtained results is that the software realizing the proposed LaMa-Wavelet network is developed, as well as experiments to research its image inpainting performance are conducted. The experimental results allow to recommend the proposed LaMa-Wavelet for use in practice, as well as to determine effective conditions for the application of this network.

Prospects for further research is reducing the computing time by using fast transforms. It is also nessesary to identify classes of images for the inpainting of which it is advisable to use LaMa-Wavelet.

ACKNOWLEDGEMENTS

The authors express their deep gratitude to V. N. Krylov, Doctor of Technical Sciences, Professor of the Department of Applied Mathematics and Information Technologies, National University “Odessa Polytechnic” for valuable and constructive advice and comments while working on this paper.

REFERENCES

1. Ma Y., Liu X., Bai S. et al. Region-wise generative adversarial image inpainting for large missing areas, *IEEE Transactions on Cybernetics*, 2023, Vol. 53, № 8, pp. 5226–5239. DOI: 10.1109/TCYB.2022.3194149.
2. Petrov K. E., Kyrychenko V. V. Removal of rain components from single images using a recurrent neural network, *Radio Electronics, Computer Science, Control*, 2023, № 2, 91–102. DOI: 10.15588/1607-3274-2023-2-10
3. Kolodochka D. O., Polyakova M. V. Comparative analysis of convolutional neural networks for filling missing image regions, *Science and education: problems, prospects and innovations: 7th International Scientific and Practical Conference, Kyoto, Japan, 1–3 April, 2021: proceedings.* CPN Publishing Group, 2021, pp. 562–570.
4. Suvorov R., Logacheva E., Mashikhin A. et al. Resolution-robust large mask inpainting with Fourier convolutions, *Applications of Computer Vision: IEEE Workshop/Winter Conference, WACV, Waikoloa, Hawaii, 4–8 January, 2022 : proceedings*. IEEE, 2022, pp. 2149–2159. DOI: 10.1109/WACV51458.2022.00323
5. Leoshchenko S. D., Oliynyk A. O., Subbotin S. O., Hoffman E. O., Kornienko O. V. Method of structural adjustment of neural network models to ensure interpretability, *Radio electronics, computer science, management*, 2021, № 3, pp. 86–96. DOI: 10.15588/1607-3274-2021-3-8
6. Polyakova M. V. RCF-ST: Richer Convolutional Features network with structural tuning for the edge detection on natural images, *Radio electronics, computer science, management*, 2023, № 4, pp. 122–134. DOI: 10.15588/1607-3274-2023-4-12
7. Pathak D., Krahenbuhl P., Donahue J., Darrell T., Efros A. A. Context encoders: feature learning by inpainting, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June, 2016 : proceedings.* IEEE, 2016, pp. 2536–2544. DOI: 10.1109/CVRP.2016.278
8. Liu G., Reda F. A., Shih K. J. et al. Image inpainting for irregular holes using partial convolutions, *Computer Vision: European Conference, ECCV, Munich, Germany, 8–14 September, 2018 : proceedings.* IEEE: 2018, pp. 85–100. DOI: 10.1007/978-3-030-01252-6_6
9. Yang C., Lu X., Lin Z. et al. High-resolution image inpainting using multi-scale neural patch synthesis, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings.* IEEE, 2017, pp. 6721–6729. DOI: 10.1109/CVPR.2017.434
10. Iizuka S., Simo-Serra E., Ishikawa H. Globally and locally consistent image completion, *ACM Transactions on Graphics*, 2017, Vol. 36, № 4, pp. 107:1. DOI: 10.1145/3072959.3073659
11. Yu J., Yang J., Shen X., Lu X., Huang T. S. Generative image inpainting with contextual attention. *Computer Vision and Pattern Recognition Workshops: IEEE/CVF Conference, CVPRW, Salt Lake City, UT, USA, 18–22 June, 2018 : proceedings.* IEEE, 2018, pp. 5505–5514. DOI: 10.1109/CVPRW.2018.00577
12. Nazeri K., Ng E., Joseph T., Qureshi F., Ebrahimi M. EdgeConnect: structure guided image inpainting using edge prediction, *Computer Vision Workshop: IEEE/CVF International Conference, ICCVW, Seoul, Korea (South), 27–28 October, 2019 : proceedings.* IEEE, 2019, pp. 2462–2468. DOI: 10.1109/ICCVW.2019.00408
13. Yu J., Lin Z., Yang J. et al. Free-form image inpainting with gated convolution, *Computer Vision: IEEE/CVF International Conference, ICCV, Seoul, Korea (South), 27 October – 2 November, 2019 : proceedings.* IEEE, 2019, pp. 4471–4480. DOI: 10.1109/ICCV.2019.00457
14. Zhao S., Cui J., Sheng Y. et al. Large scale image completion via co-modulated generative adversarial networks, *Learning Representations: International Conference, ICLR, Vienna, Austria, 4 May 2021: processings* [Electronic resource]. Access mode: https://arxiv.org/abs/2103.10428. DOI: 10.48550/arXiv.2103.10428
15. Gonzalez R. C., Woods R. E. Digital Image Processing. NY, Pearson, 4th Edition, 2017, 1192 p.
16. Daubechies I. Ten Lectures on Wavelets, Philadelphia, SIAM Press, 1992, 352 p.
17. Li Q., Shen L., Guo S., Lai Z. WaveCNet: wavelet integrated CNNs to suppress aliasing effect for noise-robust image classification, *IEEE Transactions on Image Processing*, 2021, Vol. 30, pp. 7074–7089. DOI:10.1109/TIP.2021.3101395
18. Liu P., Zhang H., Zhang K., Lin L., Zuo W. Multi-level Wavelet-CNN for image restoration, *Computer Vision and Pattern Recognition Workshops: IEEE/CVF Conference, CVPRW, Salt Lake City, UT, USA, 18 – 22 June, 2018 : proceedings*. IEEE, 2018, pp. 2149–2159. DOI: 10.1109/CVPRW.2018.00121

DOI 10.15588/1607-3274-2024-1-19

19. Bobulski J. Multimodal face recognition method with two-dimensional hidden Markov model, *Bulletin of the Polish Academy of Sciences, Technical Sciences*, 2017, Vol. 65, № 1, pp. 121–128. DOI: 10.1515/bpasts-2017-0015
20. Sara U., Akter M., Uddin M. S. Image quality assessment through FSIM, SSIM, MSE and PSNR – a comparative study, *Journal of Computer and Communications*, 2019, Vol. 7, № 3, pp. 8–18. DOI: 10.4236/jcc.2019.73002
21. Johnson J., Alahi A., Fei-Fei L. Perceptual losses for real-time style transfer and super-resolution. *Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)*. Springer, Cham, 2016, Vol. 9906, pp. 694–711. DOI: 10.1007/978-3-319-46475-6_43
22. Wang T.-C., Liu M.-Y., Zhu J.-Y. et al. High-resolution image synthesis and semantic manipulation with conditional GANs, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Salt Lake City, UT, USA, 18–23 June, 2018 : proceedings*. IEEE, 2018, pp. 8798–8807. DOI: 10.1109/CVPR.2018.00917
23. Places365 Scene Recognition Demo [Electronic resource]. Access mode: http://places2.csail.mit.edu/
24. Safebooru [Electronic resource]. Access mode: https://safebooru.org/index.php?page=post&s=list&tags=no_humans+landscape
25. Heusel M., Ramsauer H., Unterthiner T., Nessler B., Hochreiter S. GANs trained by a two time-scale update rule converge to a local nash equilibrium, *Neural Information Processing Systems: 31st Annual Conference, NIPS, Long Beach, California, USA, 4–9 December, 2017 : proceedings.* Neural Information Processing Systems Foundation, Inc., 2017, pp. 6629–6640. DOI: 10.18034/ajase.v8i1.9
26. Polyakova M. V., Krylov V. N., Ishchenko A. V. Elaboration of the transform with generalized comb scaling and wavelet functions for the image segmentation, *Eastern-European Journal of Enterprise Technologies*, 2014, Vol. 2, № 2 (71), pp. 33–37. DOI: 10.15587/1729-4061.2014.27791
27. CelebA-HQ [Electronic resource]. Access mode: https://paperswithcode.com/dataset/celeba-hq
28. Supplementary material [Electronic resource]. Access mode: https://bit.ly/3zhv2rD/lama_supmat_2021.pdf

Received 11.01.2024.
Accepted 28.02.2024.

УДК 004.93

LAMA-WAVELET: РЕКОНСТРУКЦІЯ ЗОБРАЖЕНЬ З ВИСОКОЮ ЯКІСТЮ ВІДНОВЛЕННЯ ДЕТАЛЕЙ І КРАЇВ ОБ'ЄКТІВ

Колодочка Д. О. – студент Інституту комп'ютерних систем Національного університету «Одеська політехніка», Одеса, Україна.

Полякова М. В. – д-р техн. наук, доцент, професор кафедри прикладної математики та інформаційних технологій Національного університету «Одеська політехніка», Одеса, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто проблему реконструкції зображень в системах комп'ютерної графіки та комп'ютерного зору. Предметом дослідження є згорткові нейронні мережі глибокого навчання для реконструкції зображень.

Мета роботи. Покращення якості реконструйованих зображень в системах комп'ютерного зору та комп'ютерної графіки шляхом застосування вейвлет-перетворення в архітектурі нейронної мережі LaMa-Fourier.

Метод. Базова мережа LaMa-Fourier окремо обробляє глобальний та локальний контекст зображення. Пропонується вдосконалити для цієї мережі блок обробки глобального контексту зображення, а саме блок спектрального перетворення. Для цього замість Fourier Unit Structure використовується розроблений авторами Simple Wavelet Convolution Block, у якому спочатку виконується тривимірне вейвлет-перетворення зображення на двох рівнях. Отримані коефіцієнти розбиваються так, що кожна субполоса представляє окрему ознаку зображення. Згортковий шар, пакетна нормалізація та функція активації ReLU послідовно застосовуються до субполос коефіцієнтів на кожному рівні вейвлет-перетворення. Отримані субполоси вейвлет-коефіцієнтів конкатенуються і до них застосовується зворотне вейвлет-перетворення, результат якого передається на вихід блоку. Окрема обробка вейвлет-коефіцієнтів на різних рівнях зменшує обчислювальну складність, зберігаючи при цьому вплив контексту кожного рівня на реконструкцію зображення. Отриману нейронну мережу названо LaMa-Wavelet. Показники FID, PSNR, SSIM та візуальний аналіз були використані для оцінки якості зображень, реконструйованих мережею LaMa-Wavelet.

Результати. Запропоновану мережу LaMa-Wavelet програмно реалізовано та досліджено для вирішення проблеми реконструкції зображень. PSNR зображень, відновлених за допомогою мережі LaMa-Wavelet, перевищує результати, отримані за допомогою мережі LaMa-Fourier для малих і середніх масок у середньому на 4,5%, для великих масок – у середньому на 6%. Застосування LaMa-Wavelet може збільшити SSIM на 2–4% залежно від розміру маски. Але реконструкція одного зображення за допомогою LaMa-Wavelet займає в 3 рази більше часу, ніж за допомогою мережі LaMa-Fourier. Аналіз конкретних зображень демонструє, що обидві мережі показують схожі результати реконструкції однорідного фону. На складних фонах із повторюваними елементами LaMa-Wavelet часто ефективніше відновлює текстури.

Висновки. Отримана мережа LaMa-Wavelet дозволяє покращити відновлення великих областей зображень за рахунок застосування вейвлет-перетворення в архітектурі мережі LaMa. А саме, підвищується якість реконструкції країв зображення та дрібних деталей.

КЛЮЧОВІ СЛОВА: реконструкція зображення, вейвлет-перетворення, мережа LaMa, вейвлет Добеші, початкова відстань Фреше, вейвлет-згортка.

ЛІТЕРАТУРА

1. Region-wise generative adversarial image inpainting for large missing areas / [Y. Ma, X. Liu, S. Bai et al.] // IEEE Transactions on Cybernetics. – 2023. – Vol. 53, № 8. – P. 5226–5239. DOI: 10.1109/TCYB.2022.3194149.
2. Петров К. Е. Видалення компонентів дощу з одиночних зображень з використанням рекурентної нейронної

DOI 10.15588/1607-3274-2024-1-19

мережі / К. Е. Петров, В. В. Кириченко // Радіоелектроніка, інформатика, управління. – 2023. – № 2. – С. 91–102. DOI: 10.15588/1607-3274-2023-2-10
3. Колодочка Д. О. Порівняльний аналіз згорткових нейронних мереж для реконструкції відсутніх областей зображень / Д. О. Колодочка, М. В. Полякова // Science and education: problems, prospects and innovations: 7th International Scientific and Practical Conference, Kyoto, Japan, 1–3 April, 2021: proceedings. – CPN Publishing Group, 2021. – P. 562–570.
4. Resolution-robust large mask inpainting with Fourier convolutions / [R.Suvorov, E. Logacheva, A. Mashikhin et al.] // Applications of Computer Vision: IEEE Workshop/Winter Conference, WACV, Waikoloa, Hawaii, 4 – 8 January, 2022 : proceedings. – IEEE, 2022. – P. 2149–2159. DOI: 10.1109/WACV51458.2022.00323
5. Метод структурного доналаштування нейромережевих моделей для забезпечення інтерпретабельності / [С. Д. Леощенко, А. О. Олійник, С. О. Субботін та ін.] // Радіоелектроніка, інформатика, управління. – 2021. – № 3. – С. 86–96. DOI: 10.15588/1607-3274-2021-3-8
6. Polyakova M. V. RCF-ST: Richer Convolutional Features network with structural tuning for the edge detection on natural images / M. V. Polyakova // Radio electronics, computer science, management. – 2023. – № 4. – P. 122–134. DOI: 10.15588/1607-3274-2023-4-12
7. Context encoders: feature learning by inpainting / [D. Pathak, P. Krahenbuhl, J. Donahue et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June, 2016 : proceedings. – IEEE, 2016. – P. 2536–2544. DOI: 10.1109/CVRP.2016.278
8. Image inpainting for irregular holes using partial convolutions / [G. Liu, F. A. Reda, K. J. Shih et al.] // Computer Vision: European Conference, ECCV, Munich, Germany, 8 – 14 September, 2018 : proceedings. – IEEE: 2018. – P. 85–100. DOI: 10.1007/978-3-030-01252-6_6
9. High-resolution image inpainting using multi-scale neural patch synthesis / [C. Yang, X. Lu, Z. Lin et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 6721–6729. DOI: 10.1109/CVPR.2017.434
10. Iizuka S. Globally and locally consistent image completion / S. Iizuka, E. Simo-Serra, H. Ishikawa // ACM Transactions on Graphics. – 2017. – Vol. 36, № 4. – P. 107:1. DOI: 10.1145/3072959.3073659
11. Generative image inpainting with contextual attention / [J. Yu, J. Yang, X. Shen et al.] // Computer Vision and Pattern Recognition Workshops: IEEE/CVF Conference, CVPRW, Salt Lake City, UT, USA, 18–22 June, 2018 : proceedings. – IEEE, 2018. – P. 5505–5514. DOI: 10.1109/CVPRW.2018.00577
12. Edge Connect: structure guided image inpainting using edge prediction / [K. Nazeri, E. Ng, T. Joseph et al.] // Computer Vision Workshop: IEEE/CVF International Conference, ICCVW, Seoul, Korea (South), 27–28 October, 2019 : proceedings. – IEEE, 2019. – P. 2462–2468. DOI: 10.1109/ICCVW.2019.00408
13. Free-form image inpainting with gated convolution / [J. Yu, Z. Lin, J. Yang et al.] // Computer Vision: IEEE/CVF International Conference, ICCV, Seoul, Korea (South), 27 October – 2 November, 2019 : proceedings. – IEEE, 2019. – P. 4471 – 4480. DOI: 10.1109/ICCV.2019.00457
14. Large scale image completion via co-modulated generative adversarial networks / [S. Zhao, J. Cui, Y. Sheng et al.] // Learning Representations: International Conference, ICLR, Vienna, Austria, 4 May 2021: processings [Electronic resource]. – Access mode: https://arxiv.org/abs/2103.10428. DOI: 10.48550/arXiv. 2103.10428
15. Gonzalez R. C. Digital Image Processing / R. C. Gonzalez, R. E. Woods. – NY : Pearson, 4th Edition, 2017. – 1192 p.
16. Daubechies I. Ten Lectures on Wavelets / I. Daubechies. – Philadelphia, SIAM Press, 1992. – 352 p.
17. WaveCNet: wavelet integrated CNNs to suppress aliasing effect for noise-robust image classification / [Q. Li, L. Shen, S. Guo, Z. Lai] // IEEE Transactions on Image Processing. – 2021. – Vol. 30. – P. 7074–7089. DOI:10.1109/TIP.2021.3101395
18. Multi-level Wavelet-CNN for image restoration / [P. Liu, H. Zhang, K. Zhang et al.] // Computer Vision and Pattern Recognition Workshops: IEEE/CVF Conference, CVPRW, Salt Lake City, UT, USA, 18 – 22 June, 2018 : proceedings. – IEEE, 2018. – P. 2149–2159. DOI: 10.1109/CVPRW.2018.00121
19. Bobulski J. Multimodal face recognition method with two-dimensional hidden Markov model / J. Bobulski // Bulletin of the Polish Academy of Sciences, Technical Sciences. – 2017. – Vol. 65, № 1. – P. 121–128. DOI: 10.1515/bpasts-2017-0015
20. Sara U. Image quality assessment through FSIM, SSIM, MSE and PSNR – a comparative study / U. Sara, M. Akter, M. S. Uddin // Journal of Computer and Communications. – 2019. – Vol. 7, № 3. – P. 8–18. DOI: 10.4236/jcc.2019.73002
21. Johnson J. Perceptual losses for real-time style transfer and super-resolution / J. Johnson, A. Alahi, L. Fei-Fei // Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds). – Springer, Cham, 2016. – Vol. 9906. – P. 694–711. DOI: 10.1007/978-3-319-46475-6_43
22. High-resolution image synthesis and semantic manipulation with conditional GANs / [T.-C. Wang, M.-Y. Liu, J.-Y. Zhu et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Salt Lake City, UT, USA, 18–23 June, 2018 : proceedings. – IEEE, 2018. – P. 8798–8807. DOI: 10.1109/CVPR.2018.00917
23. Places365 Scene Recognition Demo [Electronic resource]. – Access mode: http://places2.csail.mit.edu/
24. Safebooru [Electronic resource]. – Access mode: https://safebooru.org/index.php?page=post&s=list&tags=no_humans+landscape
25. GANs trained by a two time-scale update rule converge to a local nash equilibrium / [M. Heusel, H. Ramsauer, T. Unterthiner et al.] // Neural Information Processing Systems: 31st Annual Conference, NIPS, Long Beach, California, USA, 4–9 December, 2017 : proceedings. – Neural Information Processing Systems Foundation, Inc., 2017. – P. 6629–6640. DOI: 10.18034/ajase.v8i1.9
26. Polyakova M. V. Elaboration of the transform with generalized comb scaling and wavelet functions for the image segmentation / M. V. Polyakova, V. N. Krylov, A. V. Ishchenko // Eastern-European Journal of Enterprise Technologies. – 2014. – Vol. 2, № 2 (71). – P. 33–37. DOI: 10.15587/1729-4061.2014.27791
27. CelebA-HQ [Electronic resource]. – Access mode: https://paperswithcode.com/dataset/celeba-hq
28. Supplementary material [Electronic resource]. – Access mode: https://bit.ly/3zhv2rD/lama_supmat_2021.pdf

DOI 10.15588/1607-3274-2024-1-19

UDC 004.94

PROACTIVE HORIZONTAL SCALING METHOD FOR KUBERNETES

Rolik O. I. – Dr. Sc., Professor, Head of the Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

Omelchenko V. V. – Postgraduate student of the Department of Information Systems and Technologies, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

ABSTRACT

Context. The problem of minimizing redundant resource reservation while maintaining QoS at an agreed level is crucial for modern information systems. Modern information systems can include a large number of applications, each of which uses computing resources and has its own unique features, which require a high level of automation to increase the efficiency of computing resource management processes.

Objective. The purpose of this paper is to ensure the quality of IT services at an agreed level in the face of significant dynamics of user requests by developing and using a method of proactive automatic application scaling in Kubernetes.

Method. This paper proposes a proactive horizontal scaling method based on the Prophet time series prediction algorithm. Prometheus metrics storage is used as a data source for training and validating forecasting models. Based on the historical metrics, a model is trained to predict the future utilization of computation resources using Prophet. The obtained time series is validated and used to calculate the required number of application replicas, considering deployment delays.

Results. The experiments have shown the effectiveness of the proposed proactive automated application scaling method in comparison with existing solutions based on the reactive approach in the selected scenarios. This method made it possible to reduce the reservation of computing resources by 47% without loss of service quality compared to the configuration without scaling.

Conclusions. A method for automating the horizontal scaling of applications in Kubernetes is proposed. Although the experiments have shown the effectiveness of this solution, this method can be significantly improved. In particular, it is necessary to consider the possibility of integrating a reactive component for atypical load patterns.

KEYWORDS: dynamic resource provisioning, Kubernetes, autoscaling, horizontal scaling, proactive scaling, Prophet, Horizontal Pod Autoscaler.

ABBREVIATIONS

CPU is central processor unit;
HPA is horizontal pod autoscaler;
HTTP is hypertext transfer protocol;
LSTM is namely long-term short-term memory;
MAPE is mean average percentage error;
QoS is quality of service;
RAM is random access memory.

NOMENCLATURE

C_x is a x-th type of computing resource, such as CPU time or RAM;

R_{Cx} is a deployment request for computing resource C_x;

$I_{MAX}(t)$ is a a function of the number of deployment instances with regard to all requested computational resources;

$I_{Cx}(t)$ is a a function of the number of deployment instances in the context of the computing resource C_x;

$W_{Cx}(t)$ is a a function of the total workload (actual usage) of the resource C_x;

D_{UP} is an upscaling delay for a deployment;

D_{DOWN} is a downscaling delay for a deployment.

INTRODUCTION

The emergence and use of orchestrators such as Kubernetes, Nomad, EC2, and others for managing IT infrastructure resources has dramatically simplified many aspects of deploying and managing computing resources in cloud environments. These tools take developers to the next level of abstraction with new challenges, including compute resource management. Under-provisioning of computing resources can lead to a deterioration in QoS. Meanwhile, over-provisioning wastes both computing and financial resources that could be used to solve other computing tasks. Balancing these two aspects is the task of dynamic resource allocation [1]. Modern information systems can include thousands of different applications, each with its own unique features and resource requirements, which makes this task much more difficult. The reactive scaling approach, when resources are added after QoS constraints are violated, is an essential part of any IT infrastructure management system. Nevertheless, the reactive approach has a number of disadvantages associated with the irrational use of computing resources and systematic violation of QoS requirements. These problems are solved by using proactive management methods, when the amounts of computing resources required for the operation of applications in accordance with the defined QoS constraints are managed in advance, taking into account the dynamics of changes in the values of QoS indicators [2].

The object of study is the management of computing resources in information systems to maintain the quality of services provided by applications deployed in the IT infrastructure at an agreed level.

The subject of study is a method of proactive horizontal scaling of computing resources allocated to applications.

The purpose of the work is to develop a method and technical solution for automating proactive horizontal scaling in the Kubernetes environment to maintain QoS at an agreed level. This solution should be universal,

DOI 10.15588/1607-3274-2024-1-20

namely, without the need for significant manual configuration, without the requirement for significant prior knowledge of the features of the applications and their resource requirements, and with the ability to adapt to the features of each application in an automatic mode.

1 PROBLEM STATEMENT

Suppose that for some application, the R_{Cx} requests for computing resources are set to constant values. This application can be horizontally scalable, and the number of application instances is determined by the function $I(t)$.

Then, to minimize the use of the computing resource C_x while maintaining a given level of service quality, the following equality must be satisfied at any time t:

$$I_{C_x}(t) = ceil(\frac{W_{C_x}(t)}{R_{C_x}}).$$

Since for the application to work correctly, it must be provided with all types of necessary computing resources, we obtain the following equality:

$$I_{MAX}(t) = ceil(\max\left\{\frac{W_{C_x}(t)}{R_{C_x}} : x = 1...X\right\}).$$

In real-world IT infrastructures, there is always a delay between determining the need to allocate additional resources after a decrease in QoS and the actual allocation of additional computing resources or application instances. Accordingly, to ensure the correct operation of the application and, therefore, ensure the specified QoS indicators, it is necessary to take into account this delay in D_{up} scaling up. However, when scaling down, resources should not be reduced in advance:

$$I_{MAX}(t) = ceil(\begin{cases} \max\left\{\frac{W_{C_x}(t+D_{up})}{R_{C_x}} : x = 1..X\right\}, W_{C_x}(t) \le W_{C_x}(t+D_{up}), \\ \max\left\{\frac{W_{C_x}(t+D_{down})}{R_{C_x}} : x = 1..X\right\}, W_{C_x}(t) > W_{C_x}(t+D_{down}) \end{cases}).$$

Accordingly, to obtain accurate values for the number of instances, it is necessary to obtain accurate predictions for the workload function $W_{Cx}(t)$, which includes collecting metrics, aggregating and processing them, evaluating the accuracy of the resulting prediction models, and selecting the best one. In addition, it is necessary to decide when and how to apply the obtained $I_{MAX}(t)$ values.

2 REVIEW OF THE LITERATURE

Proactive scaling methods can be divided into the following groups: threshold-based rules, reinforcement learning, queuing theory, control theory, and time series analysis [3]. The methods discussed in this paper are based on time series analysis, so this section compares the methods of this group.

One of the concepts for predictive scaling is based on finding the most similar load pattern in the past and extrapolating it to the current state. For example, in the work [4], the authors propose a solution in which the historical time series is analyzed for patterns and, based on them, the most similar load pattern to the current one is searched for. The identified patterns may differ in scale, but the correlation between the elements of the identified pattern and the current pattern should be similar. The resulting patterns are interpolated using weighted interpolation. The most similar patterns will have the highest weight in the resulting time series. The main idea is to find the most similar load situation in the past and adapt it to the current load. Among the advantages of this method is the ability to predict non-trivial time series, in which there is no seasonality, but typical load segments can occur at random moments. The paper also compares it with other methods such as RightScale, linear regression, and autoregression. On different data, the accuracy rates were both significantly better than the alternatives and significantly worse, depending on the testing application and experimental conditions.

The next paper considers predictive scaling based on ARMA/ARIMA [5]. The authors propose an approach in which there are several levels of confidence that are selected depending on the application features. The authors also evaluate the accuracy of the resulting model and assess the impact on service quality indicators. In particular, the best obtained accuracy of MAPE is 0.09, which is approximately equal to the accuracy level of such algorithms as Prophet [6] and GreyKite [7]. In addition, in the worst-case scenario, the share of rejected requests was 5%, which is acceptable. The data used to evaluate the accuracy were taken from open sources, namely historical load data for Wikipedia. It is worth noting that the historical data does not contain complex seasonality, so the evaluation is based on a simple time series. The article also notes that finding the coefficients p and q, which specify the order of the ARIMA model, requires quite significant computing resources for this approach.

In the paper [8], the authors proposed a two-component PRESS design for predictive scaling. The first component, signature-driven, is used for load patterns that contain examples of repeating loads. Fast Fourier Transform is used to process this type of pattern, and Pearson's criterion is used to compare the similarity of current and past examples. If the criterion does not provide the required similarity index, the second component, state-driven, is used. This component is based on Markov chains, where all possible variants are evenly distributed among a given number of baskets. After that, a graph of possible states and a probability matrix are built. The authors suggest that this solution can be easily scaled and is suitable for massive systems.

In [9], neural networks, namely LSTM are used to manage the quality of services. LSTM networks are a particular type of recurrent neural networks capable of learning long-term dependencies. The main feature of LSTM networks is their ability to find and store information over long sequences or periods of time, which makes LSTM networks good at solving time series forecasting tasks. In addition, this algorithm is supplemented with Reinforcement Learning to obtain more accurate predictions. The proposed solution is tested on NASA datasets and shows better accuracy compared to linear approaches and LSTM-based approaches without augmentation.

The solution proposed in this paper does not require significant computing resources for its operation, which

DOI 10.15588/1607-3274-2024-1-20

allows it to handle a large number of loads in a cluster. It also supports working with time series containing complex seasonality, trends, and anomalies. The developed solution is integrated into the Kubernetes ecosystem, which allows to test its effectiveness in a real environment and compare it with existing solutions.

3 MATERIALS AND METHODS

Modern IT infrastructure management systems use various platforms for orchestrating containerized applications. The most common orchestration platform is Kubernetes, which provides tools for flexible management of computing resources. A Kubernetes cluster operates on nodes, each of which has its own set amount of computing resources $\{C_1, C_2, \ldots, Cn\}$, including CPU time and memory. Applications in the form of deployments contain information about requests $\{R_{C1}, R_{C2}, \ldots, R_{Cx}\}$ for the resources required for their correct operation. The specification of each deployment also contains the required number of application instances. Kubernetes places application instances in the form of pods on the cluster nodes in such a way that the sum of all requests of the placed instances does not exceed the total volume of the nodes. This ensures that each instance of the application will have enough resources to operate and ensure the specified QoS indicators. Suppose an instance uses more of a specific C_x resource than specified in the R_{Cx} specification. In that case, it will either crash or be artificially slowed down, depending on the type of resource.

Horizontal scaling is the adaptation of the number of deployment instances depending on the current demand for computing resources. By increasing the number of instances, the overall capacity to process web requests or tasks from the queue increases. Adding a new instance of an application can require significant D_{up} time, which includes network latency, image look-up and downloading, scaling the Kubernetes cluster itself, and application initialization. A proactive approach to QoS management, with predictions for the workload functions $W_{Cx}(t)$, has the ability to scale deployments in advance to maintain the required QoS level.

Without losing the meaning of the research and to simplify the experiments and presentation of the results, this paper considers only the management of the CPU time of computing resources. In general, this solution has the ability to work with any metric, such as memory and network bandwidth.

Figure 1 shows a structural simplified diagram of the integration of the proposed solution into a Kubernetes cluster. This diagram shows a user sending a request to an application hosted in the cluster. A header-based network proxy redirects the request to the target application's balancer, which distributes all incoming requests among the deployment instances. When a new application instance is added, this balancer ensures that it is included in the load balancing processing without additional network settings.

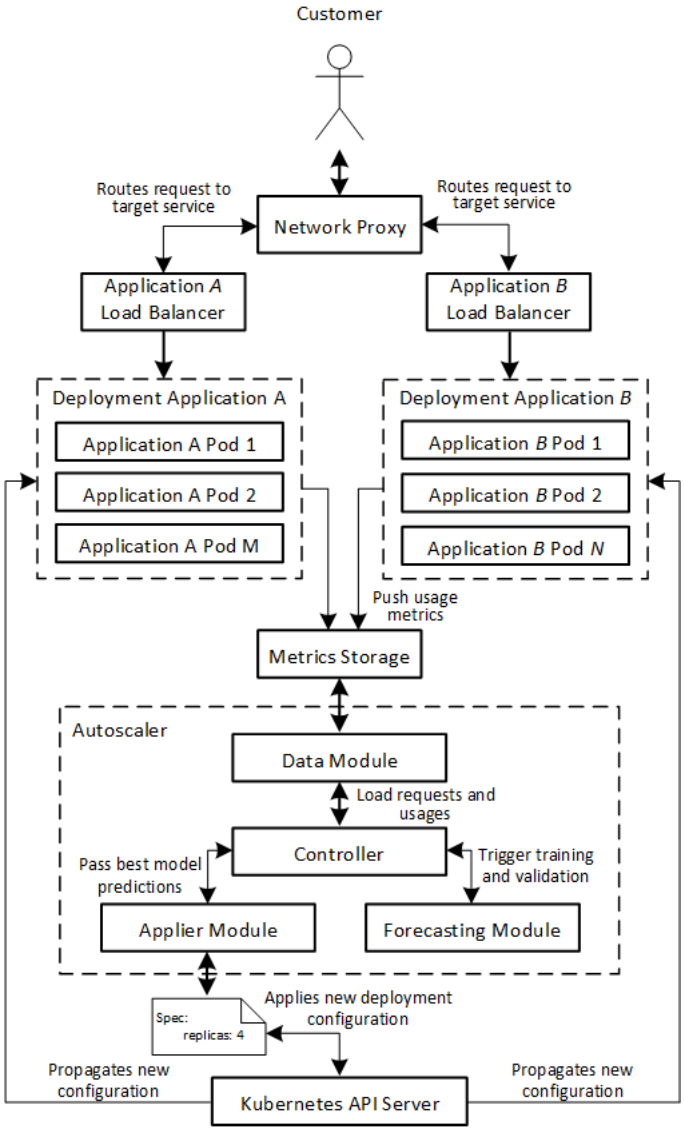


Figure 1 – Structure diagram of integration in Kubernetes

Processing a request requires the use of computing resources, such as CPU time, memory, or disk. At regular intervals, a monitoring system in Kubernetes, such as Prometheus, collects data on the total usage of computing resources from the pods. After aggregation, this information is stored in a special storage – metrics storage – for time series and is available for analysis.

The solution proposed in this paper relies on Prometheus as a source of historical data. Prometheus is a de facto standard in Kubernetes and offers long-term data storage and a specialized language for querying and aggregating data.

Having historical data about the application workload, it is necessary to calculate future values of the workload in order to set the required number of application replicas. Therefore, a crucial part of the proposed solution is a prediction algorithm. The modern generation of prediction algorithms, such as Prophet, Greykite, and TBATS, are accurate and capable of detecting seasonality, trends, and anomalies in an automatic mode. For this work, Prophet

DOI 10.15588/1607-3274-2024-1-20

was chosen, but any other method from the list can be used in the future.

Prophet is a time series prediction library developed by Facebook [7]. The main goal of the development was to create a simple, transparent, and understandable model generation algorithm that would simplify getting reliable predictions quickly. Prophet provides convenient tools for analyzing time-series and cross-validating the resulting model and has a user-friendly API.

The Forecasting module is responsible for the work with predictions and provides a unified API regardless of which prediction algorithms are used. This module requires the history of computing resource usage for previous periods of operation as input. In the current implementation, historical data is divided into training and validation data. On the training data, a set of models is trained, each with different input parameters. Then, the accuracy of the predictions is checked on the validation data. First, this allows us to choose the model with the best parameters and, accordingly, with the best accuracy. Secondly, this approach allows us to assess the accuracy of the final model as a whole and the appropriateness of its application.

When the proposed solution is initialized to automate the scaling of the target application, there may be no previous usage metrics. In this case, the proposed solution sets the number of `defaultReplicas` replicas set by the administrator until a reliable prediction can be obtained. Reliability of resource utilization prediction is calculated using the mean absolute percentage error. If the model error is less than the confidence parameter, then the proposed solution, relying on the obtained values, sets the calculated optimal number of replicas by changing the replicas field in the Kubernetes deployment manifest [10].

Historical data is obtained through requests to the Prometheus server, which is the data module's responsibility. To get historical CPU usage metrics, the query of the form `sum(rate(container_cpu_usage_total {container!=""}[60s])) by (pod)` is used. To get the current specified requests, the Kubernetes API is used, namely the `spec.container[0].requests.cpu` field in the deployment manifest. The selected Prophet prediction algorithm is capable of automatically detecting seasonality, trends, and anomalies in time series, so it does not require additional configuration. However, the administrator can specify base seasonalities with dedicated configuration parameters to improve the accuracy of predictions. The model is trained and evaluated with a specified `trainEvaluatePeriod` period. When the model's accuracy drops, this solution sets the number of replicas to `defaultReplicas` until the required accuracy is obtained. Internal or external load anomalies, incorrect operation of the monitoring subsystem, or network problems in the cluster can cause accuracy drops. This approach ensures the correct operation of the application until it is possible to get accurate predictions of resource utilization again.

When designing a horizontal scaling solution, it is necessary to take into account that applications need some time to initialize. For example, in the case of a Redis database, the application needs to read the last snapshot into memory and establish connections to other cluster components. In addition, after the command to increase the number of application replicas, it takes some time to initialize the pod, namely, to deploy it on an available node, download the image, and connect the volumes. Therefore, the proposed solution has an additional parameter, `applicationTimeToStart`, to accurately calculate the moment when it is necessary to increase the number of application replicas. In addition, the process of determining the `applicationTimeToStart` parameter for an application in a cluster can be automated.

Accordingly, the proposed solution checks the need to scale the application at short intervals. When scaling up, the Applier Module component checks the predicted utilization values at the `currentTime + applicationTimeToStart` time to set the required number of replicas in advance and not affect the QoS performance. When scaling down, only the value at the current time is checked so as not to reduce the number of replicas ahead of time. Therefore, we have a simplified formula for calculating the required number of replicas based on the forecast – `max(forecastedUsage[currentTime + timeToStart], actualUsage[currentTime])`.

This solution is placed in a Kubernetes cluster as a deployment and has direct access to the Prometheus server and the Kubernetes API using the Data Module.

4 EXPERIMENTS

The proposed solution is compared with two other configurations for resource management.

The first configuration uses HPA to compare a proactive and reactive approach. In order for the comparison to be valid, it is necessary to minimize the delays related to collecting system metrics.

The second configuration is to provide the test application with the required amount of resources and compare whether the developed solution affects the QoS indicators.

To test the resulting solution and compare it with other configurations, we use a Kubernetes cluster based on minikube [11]. This cluster includes a single node that has six processor cores with a clock frequency of 3.6GHz and 16 gigabytes of RAM. Any network communication does not go beyond a single machine. This cluster contains an installed Prometheus for monitoring using the kube-prometheus stack.

A scenario with a periodic load was chosen for testing. The load period is 300 seconds. The total minimum load is 100 millicores or 100m, the maximum is 550m.

Load testing is performed using the specialized locust utility [12] for sending HTTP requests, an instance of which is also deployed in the cluster. The load pattern of requests consists of periodic oscillations. The oscillation period is 300s, the minimum request frequency is ten requests per second, and the maximum is 50.

The selected test application can be scaled both horizontally and vertically. The test application is a web server that performs some CPU-intensive work for each request. The number of application replicas or the amount of allocated resources does not affect the application's performance. The application is initialized for a specified

DOI 10.15588/1607-3274-2024-1-20

time before becoming available for requests and before the readiness probe is considered successful. In our experiments, this time is 5s.

In this work, scaling is performed only for the CPU resource. The request for this resource for the test application is 200m. If this value is exceeded, trotting will be applied to the application, which may affect the test results. A limit of 250m was also set for this resource. The initial number of replicas is five.

HPA is a built-in solution for automating horizontal scaling in Kubernetes. HPA is a representative of the reactive approach, so it does not contain any prediction algorithms. The concept is to maintain the value of the average load per instance – `targetAverageUtilization` – set by the administrator by adjusting the number of replicas. The main difference is that the decision to scale is made based on the current values of computing resource utilization.

Since the reactive approach is highly dependent on delays in obtaining current data, it is necessary to minimize this impact on the results of the experiment. HPA uses metrics-server as a data source, so the resolution setting for metrics-server is set to the lowest possible value – 15s. Also, the sync-period, cpu-initialization-period, initial-readiness-delay settings were set to the minimum value to speed up the HPA's response time to changes. In addition, the limits for downscaling and upscaling rates were removed in the behavior scaling policies.

5 RESULTS

Figure 2 shows the results of the test scenario without using automatic scaling. This pair of graphs contains data on the actual total utilization, CPU time reservation, and the resulting request processing time. In this case, the application has enough instances to process the received requests. At the maximum load, the total actual CPU time usage is 570m. At the same time, the response time at peak times increases from 12 to 90 milliseconds, which is actually a QoS indicator in this experiment.

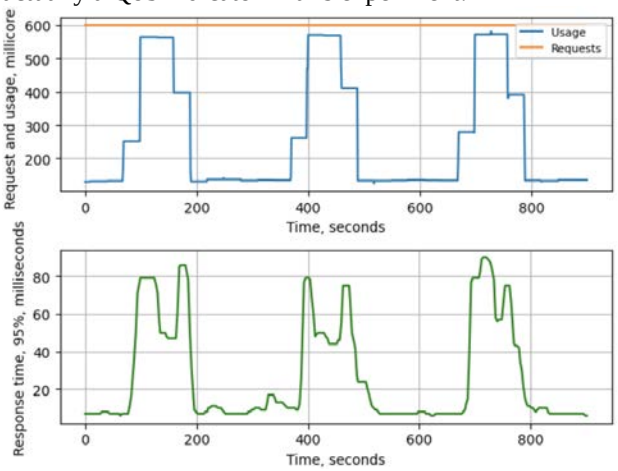


Figure 2 – No-autoscaling result

Figure 3 shows the results of testing the developed solution. This graph shows that the response time of the test application is similar to the first experiment – from 12 to 90 milliseconds so that this behavior can be interpreted as a feature of the application. Upscaling occurred ahead of time, and downscaling occurred after the peak load was passed.

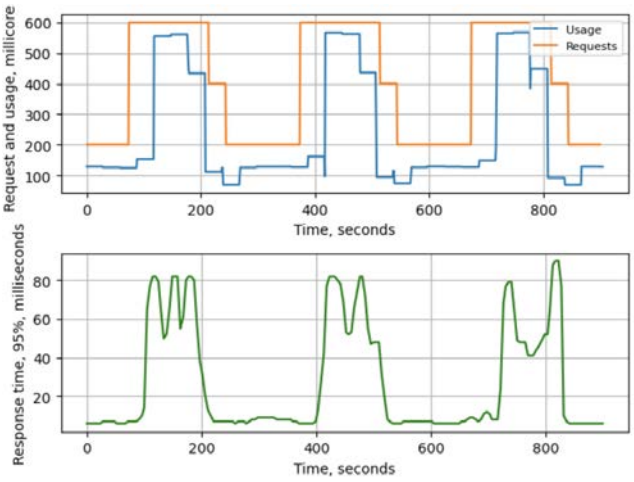


Figure 3 – Proactive scaling results

Figure 4 shows the results of HPA testing with the configuration described earlier. It should be noted that the metric for reserved resources is the sum of the requests of all pods that are in the Ready status. Therefore, in this figure, you can see that there is a slight delay between the increase in actual resource utilization and the resource provided. Because of this, there is a short-term deterioration in QoS at the time of this delay, namely an increase in response time from 90 in previous experiments to 1700 milliseconds.

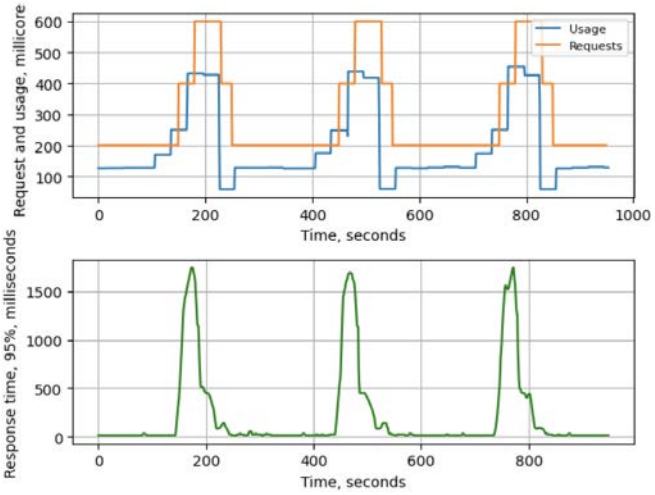


Figure 4 – Reactive scaling results

Figure 5 compares response times for all three configurations.

DOI 10.15588/1607-3274-2024-1-20

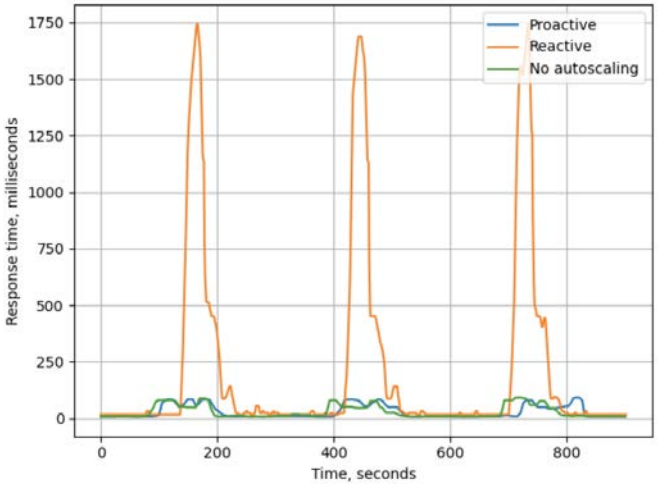


Figure 5 – Response time comparison

Table 1 shows the average response time and the 95th and 99th percentiles. In particular, on average, all response times for the proactive and reactive configurations differ by a factor of 8 in favor of the former but reserve 26% more resources. Also, the response times for this solution and management without autoscaling are similar, but the resource reservation is reduced by 47%. It is worth noting that this experiment's conditions aim to demonstrate the advantage of the proactive approach under fast-growing load patterns, which is a significant problem with the reactive approach.

Table 1 – Response time comparison

Approach	Response time		
	Average, ms	*95%, ms*	*99%, ms*
Proactive	23	87	190
No autoscaling	22	88	188
Reactive	160	770	1350

6 DISCUSSION

The results obtained indicate the efficiency and effectiveness of the developed solution. However, the experiments were conducted for one type of load – processor time. Future studies should also take into account more complex patterns, such as those containing several seasonalities or trends.

Also, this solution has not been tested for another essential computing resource, memory. This type has its own specifics of resource allocation since if the set requests or limits are exceeded, the application may crash. In addition, unlike CPU time, some part of the memory is used to store the code and initial data for work regardless of the load. This means that the calculation of the total amount of memory during horizontal scaling should include the constant component described above.

In the above architecture, the D_{UP} and D_{DOWN} delays are set by the user, but these parameters can also be determined from historical data.

Vertical and hybrid scaling can also be built based on this architecture since the main components – data modules, prediction, and application – are similar.

CONCLUSIONS

In this paper, we have developed a solution for predictive horizontal scaling in Kubernetes. The obtained experimental results allow us to conclude that in the selected scenarios, the developed architecture allows to significantly reduce the reservation of computing resources while maintaining a high level of QoS compared to HPA. That is, the proposed method uses computing resources more efficiently.

The scientific novelty. We have proposed a relatively simple architecture for horizontal scaling in Kubernetes, which can be easily adapted to different types of loads or types of scaling. In addition, the use of new time series prediction methods for processing workloads was proposed.

The practical orientation of the study. The main part of this research is the development of an automated subsystem for horizontal scaling in Kubernetes. The resulting solution is a ready-to-use component that is fully integrated into the orchestrator ecosystem.

Prospects for further research. In future research, it makes sense to consider integrating the proposed method with a reactive component. In addition, this architecture can be used for vertical and hybrid scaling.

REFERENCES

1. Rolik O. I., Telenik S. F., Yasochka M. V. Upravlinnya korporativnoyu infrastrukturoyu. Kyiv, Naukova Dumka, 2018, 576 p.
2. Omelchenko V. V., Rolik O. I. Automation of resource management in information systems based on reactive vertical scaling, *Adaptive systems of automatic control*, 2022 Vol. 2, No. 4, pp. 65–78. DOI: 10.20535/1560-8956.41.2022.271344.
3. Lorido-Botran T., Miguel-Alonso J., Lozano J. A. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments, *Journal of Grid Computing: Springer Science and Business Media LLC*, 2014,Vol. 12, No. 4, pp. 559–592. DOI: 10.1007/s10723-014-9314-7.
4. Caron E., Desprez F., Muresa A. Pattern Matching Based Forecast of Non-periodic Repetitive Behavior for Cloud Clients, *Journal of Grid Computing: Springer Science and Business Media LLC*, 2011, Vol. 9, No. 1, pp. 49–64. DOI: 10.1007/s10723-010-9178-4.
5. Calheiros R. N., Masoumi E., Ranjan R., Buyya R. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS, *IEEE Transactions on Cloud Computing: Institute of Electrical and Electronics Engineers*, 2015, Vol. 3, No. 4, pp. 449–458. DOI: 10.1109/tcc.2014.2350475.
6. Hosseini R., Chen A., Yang K., Patra S. Greykite: Deploying Flexible Forecasting at Scale at LinkedIn, *arXiv*. 2022. DOI: 10.48550/ARXIV.2207.07788.
7. Taylor S. J., Letham B. Forecasting as scale, *PeerJ*, 2017. DOI: 10.48550/ARXIV.2111.15397.
8. Zhenhuan G., Xiaohui G., Wilkes J. PRESS: PRedictive Elastic ReSource Scaling for cloud systems, *2010 International Conference on Network and Service Management. IEEE*, 2010. DOI: 10.1109/cnsm.2010.5691343.
9. Zhong J., Duan S., Li Q. Auto-Scaling Cloud Resources using LSTM and Reinforcement Learning to Guarantee Service-Level Agreements and Reduce Resource Costs,

DOI 10.15588/1607-3274-2024-1-20

Journal of Physics: Conference Series, 2019, Vol. 1237, No. 2, pp. 22–33. DOI: 10.1088/1742-6596/1237/2/022033.
10. “Deployment Controllers” Kubernetes Documentation. [Online]. Available: https://kubernetes.io/docs/concepts/workloads/controllers/deployment/.
11. “Minikube Documentation,” Minikube Documentation. [Online]. Available: https://minikube.sigs.k8s.io/docs/.
12. Locust, “Locust GitHub Repository,” GitHub. [Online]. Available: https://github.com/locustio/locust.

Received 07.12.2023.
Accepted 12.02.2024.

УДК 004.94

МЕТОД ПРОАКТИВНОГО ГОРИЗОНТАЛЬНОГО МАСШТАБУВАННЯ В KUBERNETES

Ролік О. І. – д-р техн. наук, професор, завідувач кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.
Омельченко В. В. – аспірант кафедри інформаційних систем та технологій, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

АНОТАЦІЯ

Актуальність. Інформаційні системи можуть включати велику множину додатків, кожен з яких використовує обчислювальні ресурси та має свої унікальні особливості роботи, що вимагає високого рівня автоматизації задля підвищення ефективності виконання процесів управління обчислювальними ресурсами. Проблема мінімізації збиткового резервування ресурсів при підтриманні показників QoS на узгодженому рівні є важливою для сучасних інформаційних систем.
Мета роботи. Метою даної роботи є забезпечення якості IT-сервісів на узгодженому рівні в умовах суттєвої динаміці запитів користувачів шляхом розробки та використання методу проактивного автоматичного масштабування додатків в Kubernetes.
Метод. В даній роботі пропонується метод проактивного горизонтального масштабування на основі алгоритму передбачення часових рядів Prophet. Як джерело даних пропонується використовувати сховище метрик Prometheus. На основі історичних метрик використання обчислювальних ресурсів отримується модель для передбачення майбутніх об’ємів використання за допомогою Prophet. Отримані значення валідуються, після чого застосовуються для обрахунку необхідної кількості реплік додатку з врахуванням затримок розгортання подів.
Результати. Проведені досліди показали ефективність запропонованого методу для проактивного автоматичного масштабування додатків у порівнянні з існуючими рішеннями з використанням реактивного методу в обраних сценаріях. Даний метод дозволив зменшити резервування обчислювальних ресурсів на 47% без втрати в якості обслуговування у порівнянні з конфігурацією без масштабування.
Висновки. Запропоновано метод автоматизації горизонтального масштабування додатків в Kubernetes. Хоча проведені досліди показали ефективність даного рішення, даний метод може бути значно доповнений. Зокрема, необхідно розглянути можливість інтеграції реактивної складової для нетипових шаблонів навантаження.

КЛЮЧОВІ СЛОВА: динамічне виділення ресурсів, Kubernetes, автомасштабування, горизонтальне масштабування, проактивне масштабування, Prophet, Horizontal Pod Autoscaler.

ЛІТЕРАТУРА

1. Ролік О. І. Управління корпоративною інфраструктурою / О. І. Ролік, С. Ф. Теленик, М. В. Ясочка. – Київ, Наукова Думка, 2018. – 576 с.
2. Omelchenko V. V. Automation of resource management in information systems based on reactive vertical scaling / V. V. Omelchenko, O. I. Rolik // Adaptive systems of automatic control. – 2022 – Vol. 2, No. 4. – P. 65–78. DOI: 10.20535/1560-8956.41.2022.271344.
3. Lorido-Botran T. A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments / T. Lorido-Botran, J. Miguel-Alonso, J. A. Lozano // Journal of Grid Computing: Springer Science and Business Media LLC. – 2014. – Vol. 12, No. 4. – P. 559–592. DOI: 10.1007/s10723-014-9314-7.
4. Caron E. Pattern Matching Based Forecast of Non-periodic Repetitive Behavior for Cloud Clients / E. Caron, F. Desprez, A. Muresa // Journal of Grid Computing: Springer Science and Business Media LLC – 2011. – Vol. 9, No. 1. – P. 49–64. DOI: 10.1007/s10723-010-9178-4.
5. Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications’ QoS / [R. N. Calheiros, E. Masoumi, R. Ranjan, R. Buyya] // IEEE Transactions on Cloud Computing: Institute of Electrical and Electronics Engineers. – 2015. – Vol. 3, No. 4. – P. 449–458. DOI: 10.1109/tcc.2014.2350475.
6. Greykite: Deploying Flexible Forecasting at Scale at LinkedIn / [R. Hosseini, A. Chen, K. Yang, S. Patra] // arXiv. – 2022. DOI: 10.48550/ARXIV.2207.07788.
7. Taylor S. J. Forecasting as scale / S. J. Taylor, B. Letham // PeerJ. – 2017. DOI: 10.48550/ARXIV.2111.15397.
8. Zhenhuan G. PRESS: PRedictive Elastic ReSource Scaling for cloud systems / G. Zhenhuan, G. Xiaohui, J. Wilkes // 2010 International Conference on Network and Service Management. IEEE. – 2010. DOI: 10.1109/cnsm.2010.5691343.
9. Zhong J. Auto-Scaling Cloud Resources using LSTM and Reinforcement Learning to Guarantee Service-Level Agreements and Reduce Resource Costs / J. Zhong, S. Duan, Q. Li // Journal of Physics: Conference Series. – 2019 – Vol. 1237, No. 2. – P. 22–33. DOI: 10.1088/1742-6596/1237/2/022033.
10. “Deployment Controllers” Kubernetes Documentation. [Online]. Available: https://kubernetes.io/docs/concepts/workloads/controllers/deployment/.
11. “Minikube Documentation,” Minikube Documentation. [Online]. Available: https://minikube.sigs.k8s.io/docs/.
12. Locust, “Locust GitHub Repository,” GitHub. [Online]. Available: https://github.com/locustio/locust.

DOI 10.15588/1607-3274-2024-1-20

UDC 004.222.5:681.142.01

METHOD FOR DETERMINING THE BIT GRID OVERFLOW OF A COMPUTER SYSTEM OPERATING IN THE SYSTEM OF RESIDUAL CLASSES

Yanko A. S. – PhD, Associate Professor, Associate Professor of the Department of Computer and Information Technologies and Systems, National University "Yuri Kondratyuk Poltava Polytechnic", Poltava, Ukraine.

Krasnobayev V. A. – Dr. Sc., Professor, Professor of Department of Electronics and Control Systems, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.

Nikolsky S. B. – PhD, Associate Professor, Associate Professor of the Department of Informatics, Municipal establishment "Kharkiv humanitarian-pedagogical academy" of Kharkiv regional council, Kharkiv, Ukraine.

Kruk O. O. – Post-graduate student of the Department of Automation, Electronics and Telecommunications, National University "Yuri Kondratyuk Poltava Polytechnic".

ABSTRACT

Context. Consideration of a set of examples of practical application of the procedure for identifying overflow of the bit grid of a computer system operating in a non-positional number system in residual classes. The object of the study is the process of processing data represented in the residual class system.

Objective. The goal of the work is to consider and analyze examples of the bit grid overflow definition of a computer system when implementing the operation of adding two numbers in a system of residual classes based on the application of a method for determining the bit grid overflow, based on the use of the concept of number rank.

Method. The specificity of the functioning of a computer system in a system of residual classes requires the implementation of not only modular operations, but also requires the implementation of additional, so-called non-modular operations. Non-modular operations include the operation of determining the overflow of the bit grid of a computer system in the system of residual classes. In a non-positional number system in residual classes, implementing the process of detecting overflow of the bit grid of a computer system is a difficult task to implement. The method considered in the work for determining the overflow of the bit grid is based on the use of positional features of a non-positional code of numbers in the system of residual classes, namely the true and calculated ranks of a number. The process of determining the overflow of the result of the operation of adding two numbers in the system of residual classes has been studied, since this arithmetic operation is the main, basic operation performed by a computer system.

Results. The developed methods are justified theoretically and studied when performing arithmetic modular operations of addition, subtraction and multiplication using tabular procedures.

Conclusions. The main advantage of the presented method is that the process of determining the overflow of the bit grid can be carried out in the dynamics of the computing process of the computer system, i.e. without stopping the solution of the problem. This circumstance makes it possible to reduce the unproductive expenditure of the computer system in the system of residual classes. In addition, this method can be used to control the operation of adding two numbers in the residual class system. This increases the reliability of obtaining the true result of the operation of adding two numbers in the system of residual classes.

KEYWORDS: arithmetic operation of modular addition, bit grid overflow, comparison operation, computer system, non-positional code, rank of the number, system of residual classes, zeroing procedure.

ABBREVIATIONS

CS is a computer system;
MDBGO is method for determining bit grid overflow;
PFNC is a positional feature of a non-positional code;
PNS is a positional number system;
SCS is a specialized computer system;
SRC is a system in residual classes;
ZC is a zeroing constant.

NOMENCLATURE

p_i is a base (module) SRC, $(i = \overline{1, n})$;

X is a number in the SRC, represented by a set of residues x_i modulo p_i;

Y is a number in the SRC, represented by a set of residues y_i modulo p_i;

R_X is a rank of number X;

$R_X^{(T)}$ is a true rank of number X;

$R_X^{(C)}$ is a calculated rank of number X;

B_i is a orthogonal basis of the SRC;

e_i is a weight of the i-th orthogonal basis B_i;

P is a numerical range of CS in the SRC;

$R_{X+Y}^{(C)}$ is a calculated rank of the sum of two numbers X and Y;

$z^{(i)}$ is a minimum ZC for module p_i;

X_n is a zeroing number X (the value of number X as a result of the zeroing procedure);

q_i is a number of additions of type $X + z^{(i)}$;

δ_i is a known quantity that is determined sequentially in the process of transformation (in the process of zeroing) of the original number X into the number X_n, $(i = \overline{1, n})$.

DOI 10.15588/1607-3274-2024-1-21

INTRODUCTION

Solving a wide class of computational problems by the CS operating in a non-positional number system in the SRC requires additional implementation of non-modular (positional) operations. Positional operations the SRC are those operations that require knowledge of the magnitudes of numbers in the binary PNS [1].Such operations primarily include the following operations: arithmetic and algebraic comparison of numbers, determining the sign of a number, determining the location of a number on the number axis, dividing numbers, operations with the fractional part of numbers, rounding numbers, determining whether the digit grid is overflowed, diagnostics, control and correction of data in the SRC, etc [2]. Accounting for grid overflow is one of the most common problems in the field of computer science and programming. An overflow occurs when the result of a calculation cannot be represented in the current bit grid size, resulting in loss of precision and incorrect values. The consequences of overflowing the bit grid can be catastrophic. Incorrect values can lead to software crashes, data loss, unpredictable behavior, and other problems [3].

Knowledge of the fact that the bit grid is overflowed is also important when implementing in the SRC not only modular, but also various positional operations, since in the SRC the number of bits used to represent numbers is limited. For example, when adding two numbers with the same signs, its sum modulo may be greater in modulus than the maximum number that can be written with a given number of digits and the result of the addition will be incorrect. Also, the availability of information about the overflow of the bit grid is important when determining the true value of the number in the PNS [4–6].

The unsolved problem of effectively determining the overflow of the bit grid in the SRC requires the development and study of MDBGO. Therefore, the scientific task of determining the overflow of the CS bit grid in the SRC is important and relevant. The solution to this problem will contribute to the further theoretical and practical development of machine arithmetic in the non-positional number system in the SRC. This will make it possible to widely use SRC to create ultra-fast, reliable and fault-tolerant specialized CS [7].

The object of study is the process of determining the overflow of the result of the operation of adding two numbers in the SRC. The process of determining the overflow of the bit grid when implementing various operations, especially the addition operation in the CS operating in the SRC, affects various aspects of the calculations (implementation complexity, calculation accuracy).

The subject of study is the MDBGO. The method consists of a set of the following operations. The values of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of the summands X and Y are determined, and the calculated value of the rank $R_{X+Y}^{(C)}$ of the result $X+Y$ of the operation of adding two numbers is also determined. By means of orthogonal bases B_i SRC, the true value of the rank $R_{X+Y}^{(T)}$ of the result $X+Y$ of the operation of adding two numbers is determined. A comparison is made between the calculated and true ranks of numbers. A conclusion is made about the presence or absence of overflow of the result of the addition operation.

The purpose of the work is to consider and analyze examples of determining the overflow of the bit grid of the CS when implementing the operation of adding two numbers in the SRC based on the use of a method based on the use of the concept of rank of the number. To achieve the goal of the work, the following tasks are formulated and solved: to formulate the problem, to formulate a criterion for assessing the fact of overflow of the CS bit grid based on the analysis of the ranks of the summands of numbers X and Y in the SRC, to give general and specific (for a given SRC) examples of solving the problem of determining the overflow, to carry out analyze the results obtained and draw conclusions.

1 PROBLEM STATEMENT

To implement positional operations in the SRC, including determining the fact of overflow of the bit grid of the CS, various PFNC can be used [1, 8]. The rank R_X of number $X = (x_1 \| x_2 \| ... \| x_n)$ in the SRC, represented by the set of residues x_i from dividing number X itself by the set of bases p_i $(i = \overline{1, n})$ in the SRC, will be used as the PFNC. The SRC defines two types of rank of the number: the true $R_X^{(T)}$ and the calculated $R_X^{(C)}$ ranks of number X. The true rank $R_X^{(T)}$ is a natural number that shows how many times the numerical range $P = \prod_{i=1}^{n} p_i$ of the CS in the SRC was exceeded during the transition from the representation of the number X in the SRC to its representation in the PNS through a system of orthogonal bases of the form $B_i = \frac{e_i \cdot P}{p_i}$ $(i = \overline{1, n})$, where the value of e_i determines the weight of the i-th orthogonal basis B_i SRC [1, 9].

Let the SRC be given by its bases p_i $(i = \overline{1, n})$. This SRC uniquely corresponds to a system of orthogonal bases B_i $(i = \overline{1, n})$, for which the equality $X_{PNS} = \left\{ \sum_{i=1}^{n} x_i \cdot B_i \right\} \bmod P$ holds. This ratio can also be represented as $X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P$. The rank of the number, which is the result of an arithmetic operation, obtained from the ranks of numbers, is called the calculated rank of the number.

To achieve the result of the study, it is necessary to consider specific examples of identifying the fact of over-

DOI 10.15588/1607-3274-2024-1-21

flow of the CS bit grid based on the ranks of numbers *X* and *Y* in the SRC. In turn, the task of determining the overflow of the CS bit grid in the SRC is implemented by determining and comparing the calculated $R_{X+Y}^{(C)}$ and true $R_{X+Y}^{(T)}$ ranks of the number *X*+*Y* in the SRC. If the condition $R_{X+Y}^{(C)} = R_{X+Y}^{(T)}$, is satisfied then it is considered that there is no overflow. Otherwise, i.e. when $R_{X+Y}^{(C)} \neq R_{X+Y}^{(T)}$, there is an overflow of the CS bit grid in the SRC.

2 REVIEW OF THE LITERATURE

One of the reasons for overflowing the bit grid is the use of large numerical values in the implemented operations. For example, when adding two numbers, if the result exceeds the maximum allowed value, an overflow occurs. Overflow can also occur when implementing other arithmetic operations. To solve the problem of eliminating the negative influence of the bit grid overflow process, there are various approaches [10–12].

At the same time, the problem of overflow of the bit grid has not yet been completely solved, since most methods for determining overflow lead to an increase in the amount of memory and slow down the calculation process in the CS [13]. Depending on the specific requirements and characteristics of the problem, the choice of method for solving the scientific problem of bit grid overflow may vary. It is important to consider both accuracy and performance and find a balance between them to achieve optimal results. Due to the relevance and unsolved nature of this problem, computer scientists are in search of effective methods and procedures for determining and eliminating the consequences of the process of overflowing the CS bit grid.

Modern publications increasingly contain various innovative methods for determining overflow when implementing various operations in a CS. Software-based architectural bound-checking based on boundary bits (bounds checking bits) that detects and prevents buffer overflows [14] results in increased memory requests to dynamically check object bindings using a boundary bit. This leads to an increase in the amount of allocated memory and affects the performance of the computer system, since one of the key aspects of increasing the performance of any software system is the efficient allocation of memory and the release of resources.

Also, to solve the problem of overflowing the bit grid, a method based on the use of modified codes is widely used [16–17]. Modification of codes consists in introducing an additional digit, which is located before the sign one. This bit is often called an overflow bit. When using various algorithms, modified codes may contain two sign bits. Article [18] presents a bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters. The disadvantage of all methods based on various modified codes is the expansion of the bit grid by at least one bit.

The possibility of using hardware methods to solve the overflow problem is being widely explored. In [19] considers using an *N* bit result integer multiplier with overflow detector indicating an *N* bit multiplication result and overflow status with an *N* bit multiplier and multiplicand input. The overflow is determined by the lower *N* bit result of multiplication and the number of leading sign bits of the multiplier and the multiplicand. The proposed method to prevent the bit grid from overflowing when implementing a multiplication (exponentiation) operation, with a slight decrease in performance during this operation.

To prevent overflow of the bit grid, it is necessary to take measures to optimize existing methods and develop new ones [20]. An analysis of publications in this area has shown that it is necessary to develop methods and procedures aimed at identifying and eliminating the negative consequences of overflows when implementing various operations in a computer system, which does not reduce the overall performance of the computer system, which depends on the speed of execution of these operations. This task also applies to the CS in the SRC.

3 MATERIALS AND METHODS

Since in this work the rank of a number as a PFNC was used, therefore, in the process of identifying overflow of the bit grid of the CS, it is very important to calculate the ranks of numbers *X* and *Y*. Based on this, an important task is to consider specific examples of identifying facts of overflow of the CS bit grid based on the use of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers *X* and *Y*.

Let's consider the procedure for determining the calculated rank of the sum of two numbers in the SRC [1]. If two numbers $X = (x_1 \| x_2 \| ... \| x_n)$ and $Y = (y_1 \| y_2 \| ... \| y_n)$ are given in the SRC with the corresponding calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers, then the calculated rank $R_{X+Y}^{(C)}$ of the sum of two numbers *X*+*Y* is determined as follows:

$$R_{X+Y}^{(C)} = R_X^{(C)} + R_Y^{(C)} - \sum_{i=1}^{n} \left[\frac{x_i + y_i}{p_i} \right] \cdot e_i. \quad (1)$$

Let's show the correctness of expression (1). Let's write expressions for determining the numbers *X* and *Y* in the PNS using true ranks:

$$X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P, \quad (2)$$

$$Y_{PNS} = \sum_{i=1}^{n} y_i \cdot B_i - R_Y^{(T)} \cdot P. \quad (3)$$

Let's add two expressions (2) and (3) and get:

DOI 10.15588/1607-3274-2024-1-21

$$X_{PNS}+Y_{PNS}=\sum_{i=1}^{n}x_i\cdot B_i-R_X^{(T)}\cdot P+\sum_{i=1}^{n}y_i\cdot B_i-R_Y^{(T)}\cdot P,$$

or

$$X_{PNS}+Y_{PNS}=\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-(R_X^{(T)}+R_Y^{(T)})\cdot P. \quad (4)$$

On the other hand, based on the rule for calculating the sum of two numbers in the SRC for each corresponding SRC base can be written that:

$$X+Y=\left\{\left(x_1+y_1-\left[\frac{x_1+y_1}{p_1}\right]\cdot e_1\right),\right.$$
$$\left(x_2+y_2-\left[\frac{x_2+y_2}{p_2}\right]\cdot e_2\right),\ldots \quad (5)$$
$$\left.\ldots,\left(x_n+y_n-\left[\frac{x_n+y_n}{p_n}\right]\cdot e_n\right)\right\}$$

Expression (5) when using expression (2) can be represented as:

$$X+Y=\sum_{i=1}^{n}\left(\left\{(x_i+y_i)-\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\right\}\right)\cdot B_i-R_{X+Y}^{(T)}\cdot P. \quad (6)$$

Let's transform expression (6), taking into account the fact that the orthogonal basis SRC is represented as $B_i=\frac{e_i\cdot P}{p_i}$. As a result, the following expression can be obtained:

$$X+Y=\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot B_i-R_{X+Y}^{(T)}\cdot P, \quad (7)$$

or

$$X+Y=\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot\frac{e_i\cdot P}{p_i}-R_{X+Y}^{(T)}\cdot P. \quad (8)$$

Let's compare the right-hand sides of expressions (4) and (8) to check the correctness of expression (1):

$$\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-\left(R_X^{(T)}+R_Y^{(T)}\right)\cdot P=$$
$$=\sum_{i=1}^{n}(x_i+y_i)\cdot B_i-\sum_{i=1}^{n}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i\cdot P-R_{X+Y}^{(T)}\cdot P,$$

that is, we have that expression (1) is satisfied (fairly).

Expression (1) is the main analytical expression that allows us to determine the calculated rank of the sum of two numbers *X* and *Y* from the values of the calculated ranks of the summands *X* and *Y*.

It's obvious that:

if $x_i+y_i\ge p_i$ then the integer part of the expression is equal to $\left[\frac{x_i+y_i}{p_i}\right]=1$;

if $x_i+y_i<p_i$ then the integer part of the expression is equal to $\left[\frac{x_i+y_i}{p_i}\right]=0$.

The procedure for determining the calculated rank $R_X^{(C)}$ of number $X=(x_1\|x_2\|\ldots\|x_n)$ is as follows. First, it needs to present the original number $X=(x_1\|x_2\|\ldots\|x_n)$ to a zeroable number of the form $X_n=(0\|0\|\ldots\|0)$. To do this, to the initial number $X=(x_1\|x_2\|\ldots\|x_n)$ in the SRC, the rank $R_X^{(C)}$ of which must be determined, the so-called ZC, in the form of minimum numbers of the form $z^{(i)}=(0\|0\|\ldots\|0\|z_i\|z_{i+1}\|\ldots\|z_n)$ $(i=\overline{1,n})$, are sequentially added. In this case, this value in the PNS is equal to the value $z_{PNS}^{(i)}=p_1\cdot p_2\cdot\ldots\cdot p_{i-1}$.

In particular, we find that the ZC has the following form:

$$z^{(1)}=\min(z_1^{(1)}\|z_2^{(1)}\|\ldots\|z_n^{(1)})=(1\|1\|\ldots\|1),$$
$$z^{(2)}=\min(0\|z_2^{(2)}\|\ldots\|z_n^{(2)})=(0\|p_1\|p_2\|\ldots\|p_{i-1}),$$
$$z^{(3)}=\min(0\|0\|z_3^{(3)}\|z_4^{(3)}\|\ldots\|z_n^{(3)})=\{(0\|0\|p_1\cdot p_2(\mathrm{mod}\,p_3)\|$$
$$\|p_1\cdot p_2(\mathrm{mod}\,p_4)\|\ldots\|p_1\cdot p_2(\mathrm{mod}\,p_n)\},$$

etc., where $z^{(n)}=(0\|0\|\ldots\|0\|z_n)$.

Let's show the procedure for obtaining the value of $X_n=(0\|0\|\ldots\|0)$. At the beginning of the procedure, let's add the ZC $z^{(1)}=(z_1^{(1)}\|z_2^{(1)}\|\ldots\|z_n^{(1)})$ to the initial number *X* as many times as necessary to satisfy the condition $x_1=0$. Let this require q_1 additions of type $X+z^{(1)}$. In this case, we get that $X_1=X+q_1\cdot z^{(1)}$. As a result, the resulting number X_1 has an intermediate calculated rank $R_{X_1}^{(C)}$ (the intermediate calculated rank is the calculated rank of the number, which is sequentially formed in the process of obtaining the value $X_n=(0\|0\|\ldots\|0)$).

Then, we obtain that $R_{X_1}^{(C)}=R_X^{(C)}+\delta_1$, where δ_1 is a known value. Next, we add q_2 times the value of the ZC $z^{(2)}=(0\|z_2^{(2)}\|\ldots\|z_n^{(2)})$ with the number X_1 until we obtain a zero residue to the base p_2, i.e. we obtain $x_2=0$. So we have a number $X_2=X_1+q_2\cdot z^{(2)}$ with an intermediate rank $R_{X_2}^{(C)}=R_{X_1}^{(C)}+\delta_2$, where δ_2 is a known value. The algorithm for obtaining the number $X_n=(0\|0\|\ldots\|0)$ can be represented by the following expressions:

DOI 10.15588/1607-3274-2024-1-21

$$\begin{cases} X_1 = X + q_1 \cdot z^{(1)}, R_{X_1}^{(C)} = R_X^{(C)} + \delta_1; \\ X_2 = X_1 + q_2 \cdot z^{(2)}, R_{X_2}^{(C)} = R_{X_1}^{(C)} + \delta_2; \\ \dots \\ X_i = X_{i-1} + q_i \cdot z^{(i)}, R_{X_i}^{(C)} = R_{X_{i-1}}^{(C)} + \delta_i; \\ \dots \\ X_n = X_{n-1} + q_n \cdot z^{(n)}, R_{X_n}^{(C)} = R_{X_{n-1}}^{(C)} + \delta_n. \end{cases} \quad (9)$$

Continuing the procedure for all remainders of the number X, the result is the number $X_n = (0 \| 0 \| \dots \| 0) = P$. In accordance with expression (2) we have that:

$$\begin{gathered} X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P, \\ X_n = \sum_{i=1}^{n} x_i \cdot B_i - R_{X_n}^{(T)} \cdot P, \\ (0 \| 0 \| \dots \| 0) = \sum_{i=1}^{n} x_i \cdot B_i - R_{X_n}^{(T)} \cdot P, \\ P = 0 - R_{X_n}^{(T)} \cdot P, \\ R_{X_n}^{(T)} = -1. \end{gathered} \quad (10)$$

Thus, the true rank $R_{X_n}^{(T)}$ of the zeroable number $X_n = (0 \| 0 \| \dots \| 0)$ is equal to – 1. On the other hand, it was shown in expression (9) that the calculated rank $R_{X_n}^{(C)}$ of the zeroable number X_n is equal to $R_{X_{n-1}}^{(C)} + \delta_n$. Since the value of the calculated rank $R_{X_n}^{(C)}$ must coincide with the true rank $R_{X_n}^{(T)}$, then the last expression of the ratio (9) and (10) must coincide:

$$R_{X_n}^{(C)} = R_{X_n}^{(T)} \Rightarrow R_{X_n}^{(C)} = -1 \quad (11)$$

or expression (11) can be written as:

$$R_{X_{n-1}}^{(C)} + \delta_n = -1 \Rightarrow R_{X_{n-1}}^{(C)} = -1 - \delta_n. \quad (12)$$

4 EXPERIMENTS

The theoretical basis for creating an experimental research base is scientific material, which is presented in the relevant sections of number theory and is also presented as the result of the proof of the Chinese remainder theorem [21]. In this case, the initial data for conducting the experiment are presented in the form of a set of bases (modules) of the SRC. SRC bases are a set of mutually prime numbers. As an experiment, this work presents the content and discusses the description of the structure of the method for determining the overflow of the CS bit grid in the SRC.

The general scheme of the experiment to determine the overflow of the bit grid of the CS, when implementing the operation of adding two numbers in the SRC [22], is presented in the following form:

1. Using the zeroing procedure based on the minimum ZCs $z^{(i)}$ SRC, the values of the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers are determined. A comparison is made between the calculated and true ranks of numbers.
2. Using the minimum ZCs $z^{(i)}$ SRC and calculated ranks values $R_X^{(C)}$ and $R_Y^{(C)}$ of numbers X and Y, according to expression (1), the calculated value $R_{X+Y}^{(C)}$ of the rank of the result $X+Y$ of the operation of adding two numbers is determined.
3. By means of orthogonal bases B_i SRC, the true value of the rank $R_{X+Y}^{(T)}$ of the result of the operation of adding two numbers $X+Y$ is determined.
4. A comparison is made between the calculated and true ranks of numbers. A conclusion is made about the fact of the bit grid overflow of the result of the addition operation according to the rank comparison criterion.

The work, as an experiment, provides a set of specific examples of the use of MDBGO. The results of the experiment showed the practical significance of the method under consideration. In addition, based on the use of the main results of the experiments, the State Patent of Ukraine for a utility model No. 129125, G06F 11/08 “Device for monitoring the result $A+B$ of the addition of two numbers A and B in the system of residual classes” (published 10.25.2018, Bull. No. 20) was obtained, authors: Krasnobayev V. A., Yanko A. S. et al. This device contains blocks for determining the calculated and true values of the ranks of the result of the operation of adding two numbers. Also this device also contains a block for comparing the calculated and true ranks of numbers in the SRC. The invention is based on the use of MDBGO. The purpose of this invention is to reduce the control time of the modular operation of addition of two numbers presented in the SRC. The goal is achieved by combining in time data processing operations in blocks for determining the calculated and true values of the ranks of numbers. This increases the efficiency of monitoring the implementation of the modular operation of adding of numbers. Thus, the above described allows, in addition to implementing the main function, to additionally use MDBGO to control the procedure for adding numbers in the SRC. This indicates the versatility of using the method discussed in the article. The presence of a patent confirms the global novelty and practical significance of some of the scientific results obtained in the article. Some results obtained in the article are an undoubted contribution to the theory and practice of non-positional machine arithmetic. The results obtained in the article can be used when creating a CS in the SRC.

5 RESULTS

In accordance with the procedure for determining the rank of a number, let’s consider examples of determining the calculated rank $R_X^{(C)}$ of number $X = (x_1 \| x_2 \| \dots \| x_n)$ presented in a specific SRC. To the initial number $X = (x_1 \| x_2 \| \dots \| x_n)$ in the SRC, the calculated rank $R_X^{(C)}$

DOI 10.15588/1607-3274-2024-1-21

of which must be determined, let's successively add the minimum ZCs $z^{(i)}$ until we ultimately obtain the zeroable number $X_n = (0 \| 0 \| ... \| 0)$, the intermediate calculated rank $R_{X_n}^{(C)}$, which from expression (11) is equal to $R_{X_n}^{(C)} = -1$. Next, using orthogonal bases, the true rank of the number is determined.

Let's give examples of determining the calculated rank $R_X^{(C)}$ of a number *X*. Table 1 presents the SRC bases $\{p_i\}$, $i = \overline{1,3}$, orthogonal bases B_i of the bases and its weights e_i. In Table 2, for a given SRC, the minimum ZCs $z^{(i)}$ and its ranks $R_{z^{(i)}}$ are given. For the SRC under consideration, the volume of the range of representable numbers is equal to $P = \prod_{i=1}^{3} p_i = 3 \cdot 5 \cdot 7 = 105$.

Table 1 – Values of the SRC bases and orthogonal bases

$p_1 = 3$	$p_2 = 5$	$p_3 = 7$
$e_1 = 2$	$e_2 = 1$	$e_3 = 1$
$B_1 = 70$	$B_2 = 21$	$B_3 = 15$

Table 2 – Values of minimum ZCs and its ranks

$z^{(1)} = (1 \| 1 \| 1)$	$z^{(2)} = (0 \| 3 \| 3)$	$z^{(3)} = (0 \| 0 \| 1)$
$R_{z^{(1)}} = 1$	$R_{z^{(2)}} = 1$	$R_{z^{(3)}} = 0$

The ranks $R_{z^{(i)}}$ of the minimum ZCs $z^{(i)}$ are calculated in advance using expression (2). Let's determine the values of the minimum constants for the SRC specified in Table 1:

$$z^{(1)} = (1 \| 1 \| 1) = 1 \cdot B_1 + 1 \cdot B_2 + 1 \cdot B_3 = 1 \cdot 70 + 1 \cdot 21 + 1 \cdot 15 =$$
$$= (70 + 21 + 15) \bmod 105 = 106 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(1)}} \cdot P = 106 - R_{z^{(1)}} \cdot P = 106 - 1 \cdot 105.$$

to comply with the above equality, it comes out to $R_{z^{(1)}} = 1$ (Table 2).

$$z^{(2)} = (0 \| 3 \| 3) = 0 \cdot B_1 + 3 \cdot B_2 + 3 \cdot B_3 =$$
$$= 0 \cdot 70 + 3 \cdot 21 + 3 \cdot 15 = (0 + 63 + 45) \bmod 105 = 108 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(2)}} \cdot P = 108 - R_{z^{(2)}} \cdot P = 108 - 1 \cdot 105.$$

to comply with the above equality, it comes out to $R_{z^{(2)}} = 1$ (Table 2).

$$z^{(3)} = (0 \| 0 \| 1) = 0 \cdot B_1 + 0 \cdot B_2 + 1 \cdot B_3 =$$
$$= 0 \cdot 70 + 0 \cdot 21 + 1 \cdot 15 = (0 + 0 + 15) \bmod 105 = 15 \bmod 105 =$$
$$= \sum_{i=1}^{n} z_i \cdot B_i - R_{z^{(3)}} \cdot P = 15 - R_{z^{(3)}} \cdot 105 = 15 - 0 \cdot 105,$$

to comply with the above equality, it comes out to $R_{z^{(3)}} = 0$ (Table 2).

Example 1. Determine the calculated rank $R_X^{(C)}$ of the number $X = (2 \| 1 \| 1) = 71$.

First stage. Determination of the calculated rank of the number in the SRC.

Let's reset the residue $x_1 = 2$ to zero according to the first module $p_1 = 3$. Let's add the number *X* and minimum ZC $z^{(1)} = (1 \| 1 \| 1)$ and get:

$$X_1 = X + z^{(1)} = (2 \| 1 \| 1) + (1 \| 1 \| 1) = (0 \| 2 \| 2).$$

The rank of the sum will be determined by expression (1), where instead of the rank of the number *Y* the value of the rank of the minimum ZC $z^{(i)}$ will be used:

$$R_{X_i}^{(C)} = R_{X_{i-1}}^{(C)} + R_{z^{(i)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(i)}}{p_i} \right] \cdot e_i, \qquad (13)$$

where $[k]$ is the integer part of the number *k*, not less than it; $R_{z^{(i)}}$ is the calculated rank of minimum ZC $z^{(i)}$ (Table 2); $z_j^{(i)}$ is the value of the *j*-th residue, $j = \overline{1,n}$ (in this case $j = \overline{1,3}$) of the *i*-th minimum ZC.

Based on expression (13), the following calculated value of the rank of the number X_1 can be obtained:

$$R_{X_1}^{(C)} = R_X^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(1)}}{p_i} \right] \cdot e_i = R_X^{(C)} + R_{z^{(1)}} -$$
$$- \left\{ \left[\frac{x_1 + z_1^{(1)}}{p_1} \right] \cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2} \right] \cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3} \right] \cdot e_3 \right\} =$$
$$= \left(R_X^{(C)} + 1 \right) - \left\{ \left[\frac{2+1}{3} \right] \cdot 2 + \left[\frac{1+1}{5} \right] \cdot 1 + \left[\frac{1+1}{7} \right] \cdot 1 \right\} =$$
$$= R_X^{(C)} + 1 - (1 \cdot 2 + 0 \cdot 1 + 0 \cdot 1) = R_X^{(C)} + 1 - 2 = R_X^{(C)} - 1.$$

In this case, there was one transition through the first base p_1 (expression (9)).

Let's reset the residue $x_2 = 2$ to zero according to the second module $p_2 = 5$ of the number $X_1 = (0 \| 2 \| 2)$. Let's add the number X_1 and minimum ZC $z^{(2)} = (0 \| 3 \| 3)$ and get:

$$X_2 = X_1 + z^{(2)} = (0 \| 2 \| 2) + (0 \| 3 \| 3) = (0 \| 0 \| 5).$$

The calculated rank of the number X_2 is determined as:

DOI 10.15588/1607-3274-2024-1-21

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$

$$+R_{z^{(2)}} - \left\{\left[\frac{x_1 + z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)} - 1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{2+3}{5}\right]\cdot 1 + \left[\frac{2+3}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

In this case, there was one transition through the second base p_2.

Let's reset the residue $x_3 = 5$ to zero according to the third module $p_3 = 7$ of the number $X_2 = (0 \| 0 \| 5)$. Let's add the number X_2 and minimum ZC $z^{(3)} = (0 \| 0 \| 1)$ and get:

$$X_3 = X_2 + z^{(3)} = (0 \| 0 \| 5) + (0 \| 0 \| 1) = (0 \| 0 \| 6).$$

The calculated rank of the number X_3 is determined as:

$$R_{X_3}^{(C)} = R_{X_2}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$

$$+R_{z^{(3)}} - \left\{\left[\frac{x_1 + z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)} - 1\right) + 0 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 0 - 0 - 0 - 0 = R_X^{(C)} - 1.$$

Since the residue $x_3 = 6$ of the number $X_3 = (0 \| 0 \| 6)$ has not been reset to zero, let's add the value $z^{(3)}$ again:

$$X_n = X_3 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

The calculated rank of the zeroable number X_n is determined as:

$$R_{X_n}^{(C)} = R_{X_3}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$

$$+R_{z^{(3)}} - \left\{\left[\frac{x_1 + z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$

$$= \left(R_X^{(C)} - 1\right) + 0 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{6+1}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} - 1 + 0 - 0 - 0 - 1\cdot 1 = R_X^{(C)} - 2.$$

In this case, there was one transition through the third base p_3.

In accordance with expression (11) we have that:

$$R_{X_n}^{(C)} = -1 \to R_X^{(C)} - 2 = -1 \to R_X^{(C)} = 1.$$

Second stage. Determination of the true rank of the number in the SRC.

The true rank $R_X^{(T)}$ of number X is determined using orthogonal bases B_i (Table 1) and expression (2):

$$X = (2 \| 1 \| 1) = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 =$$

$$= 2 \cdot B_1 + 1 \cdot B_2 + 1 \cdot B_3 = 2 \cdot 70 + 1 \cdot 21 + 1 \cdot 15 =$$

$$= (2 \cdot 70 + 1 \cdot 21 + 1 \cdot 15) \bmod 105 = 176 \bmod 105 =$$

$$= X_{PNS} = \sum_{i=1}^{n} x_i \cdot B_i - R_X^{(T)} \cdot P =$$

$$= 176 \bmod 105 = 176 - R_X^{(T)} \cdot P =$$

$$= 176 - 1 \cdot 105 = 176 - 105 = 71.$$

Thus, the true of the number $X = (2 \| 1 \| 1) = 71$ is $R_X^{(T)} = 1$.

Third stage. Checking the reliability of obtaining the rank of the number in the SRC.

Let's compare the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X. Obviously, $R_X^{(C)} = R_X^{(T)} = 1$.

Conclusions. So, since the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X are equal, then the calculated rank $R_X^{(C)}$ is determined correctly.

Example 2. Determine the calculated rank $R_X^{(C)}$ of the number $X = (1 \| 1 \| 5) = 61$.

First stage. Determination of the calculated rank of the number in the SRC.

Let's reset the residue $x_1 = 1$ to zero according to the first module $p_1 = 3$. Let's add the number X and minimum ZC $z^{(1)} = (1 \| 1 \| 1)$ and get:

$$X_1 = X + z^{(1)} = (1 \| 1 \| 5) + (1 \| 1 \| 1) = (2 \| 2 \| 6).$$

The calculated rank of the number X_1 will be determined by expression (13):

$$R_{X_1}^{(C)} = R_X^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(1)}}{p_i}\right]\cdot e_i = R_X^{(C)} + R_{z^{(1)}} -$$

$$-\left\{\left[\frac{x_1 + z_1^{(1)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3}\right]\cdot e_3\right\} =$$

$$= R_X^{(C)} + 1 - \left\{\left[\frac{1+1}{3}\right]\cdot 2 + \left[\frac{1+1}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right\} =$$

$$= R_X^{(C)} + 1 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)} + 1.$$

Since the residue x_1 of the number has not been reset to zero, let's add the value $z^{(1)}$ again:

$$X_2 = X_1 + z^{(1)} = (2 \| 2 \| 6) + (1 \| 1 \| 1) = (0 \| 3 \| 0).$$

DOI 10.15588/1607-3274-2024-1-21

The calculated rank of the number X_2 is determined as:

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(1)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} + 1\right) + R_{z^{(1)}} -$$
$$-\left\{\left[\frac{x_1 + z_1^{(1)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} + 1\right) + 1 - \left\{\left[\frac{2+1}{3}\right]\cdot 2 + \left[\frac{2+1}{5}\right]\cdot 1 + \left[\frac{6+1}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} + 1 + 1 - 1\cdot 2 - 0\cdot 1 - 1\cdot 1 = R_X^{(C)} - 1.$$

In this case, two transitions took place through the first base p_1 and through the third base p_3.

Let's reset the residue $x_2 = 2$ to zero according to the second module $p_2 = 5$ of the number $X_2 = (0 \| 3 \| 0)$. Let's add the number X_2 and minimum ZC $z^{(2)} = (0 \| 3 \| 3)$ and get:

$$X_3 = X_2 + z^{(2)} = (0 \| 3 \| 0) + (0 \| 3 \| 3) = (0 \| 1 \| 3).$$

The calculated rank of the number X_3 is determined as:

$$R_{X_3}^{(C)} = R_{X_2}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(2)}} - \left\{\left[\frac{x_1 + z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{3+3}{5}\right]\cdot 1 + \left[\frac{0+3}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1$$

So, since the residue x_2 of the number $X_3 = (0 \| 1 \| 3)$ has not been reset to zero, then let's add the value $z^{(2)}$ again:

$$X_4 = X_3 + z^{(2)} = (0 \| 1 \| 3) + (0 \| 3 \| 3) = (0 \| 4 \| 6).$$

The calculated rank of the number X_4 is determined as:

$$R_{X_4}^{(C)} = R_{X_3}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(2)}} - \left\{\left[\frac{x_1 + z_1^{(2)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(2)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(2)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{1+3}{5}\right]\cdot 1 + \left[\frac{3+3}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)}.$$

There are no transitions along the bases. Since the residue x_2 of the number $X_4 = (0 \| 4 \| 6)$ has not been reset to zero, let's add the value $z^{(2)}$ again:

$$X_5 = X_4 + z^{(2)} = (0 \| 4 \| 6) + (0 \| 3 \| 3) = (0 \| 2 \| 2).$$

The calculated rank of the number X_5 is determined as:

$$R_{X_5}^{(C)} = R_{X_4}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i =$$
$$= R_X^{(C)} + 1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{4+3}{5}\right]\cdot 1 + \left[\frac{6+3}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} + 1 - 0\cdot 2 - 1\cdot 1 - 1\cdot 1 = R_X^{(C)} + 1 - 2 = R_X^{(C)} - 1.$$

Since the residue x_2 has not been reset to zero, the operation of adding two numbers X_5 and $z^{(2)}$ is implemented again:

$$X_6 = X_5 + z^{(2)} = (0 \| 2 \| 2) + (0 \| 3 \| 3) = (0 \| 0 \| 5).$$

The calculated rank of the number X_6 is determined as:

$$R_{X_6}^{(C)} = R_{X_5}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(2)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$
$$+1 - \left\{\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{2+3}{5}\right]\cdot 1 + \left[\frac{2+3}{7}\right]\cdot 1\right\} =$$
$$= R_X^{(C)} - 1 + 1 - 0\cdot 2 - 1\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

In this case, there was one transition through the second base p_2.

Let's reset the residue $x_3 = 5$ to zero according to the third module $p_3 = 7$ of the number $X_6 = (0 \| 0 \| 5)$. Let's add the number X_6 and minimum ZC $z^{(3)} = (0 \| 0 \| 1)$ and get:

$$X_7 = X_6 + z^{(3)} = (0 \| 0 \| 5) + (0 \| 0 \| 1) = (0 \| 0 \| 6).$$

The calculated rank of the number X_7 is determined as:

$$R_{X_7}^{(C)} = R_{X_6}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3}\left[\frac{x_i + z_j^{(3)}}{p_i}\right]\cdot e_i = \left(R_X^{(C)} - 1\right) +$$
$$+R_{z^{(3)}} - \left\{\left[\frac{x_1 + z_1^{(3)}}{p_1}\right]\cdot e_1 + \left[\frac{x_2 + z_2^{(3)}}{p_2}\right]\cdot e_2 + \left[\frac{x_3 + z_3^{(3)}}{p_3}\right]\cdot e_3\right\} =$$
$$= \left(R_X^{(C)} - 1\right) + 0 - \left(\left[\frac{0+0}{3}\right]\cdot 2 + \left[\frac{0+0}{5}\right]\cdot 1 + \left[\frac{5+1}{7}\right]\cdot 1\right) =$$
$$= R_X^{(C)} - 1 + 0 - 0\cdot 2 - 0\cdot 1 - 0\cdot 1 = R_X^{(C)} - 1.$$

DOI 10.15588/1607-3274-2024-1-21

Since the residue x_3 has not been reset to zero, the operation of adding two numbers is implemented again:

$$X_n = X_7 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

The calculated rank of the zeroable number X_n is determined as:

$$R_{X_n}^{(C)} = R_{X_7}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$

$$= \left(R_X^{(C)} - 1 \right) + 0 - \left(\left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{6+1}{7} \right] \cdot 1 \right) =$$

$$= R_X^{(C)} - 1 + 0 - 0 \cdot 2 - 0 \cdot 1 - 1 \cdot 1 = R_X^{(C)} - 2.$$

In this case, there was one transition through the third base p_3.

In accordance with expression (11) we have that:

$$R_{X_n}^{(C)} = -1 \to R_X^{(C)} - 2 = -1 \to R_X^{(C)} = 1.$$

Second stage. Determination of the true rank of the number in the SRC.

The true rank $R_X^{(T)}$ of number X is determined using orthogonal bases B_i (Table 1) and expression (2):

$$X = (1 \| 1 \| 5) = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 =$$

$$= 1 \cdot B_1 + 1 \cdot B_2 + 5 \cdot B_3 = 1 \cdot 70 + 1 \cdot 21 + 5 \cdot 15 =$$

$$= (1 \cdot 70 + 1 \cdot 21 + 5 \cdot 15) \bmod 105 =$$

$$= 166 \bmod 105 = 166 - R_X^{(T)} \cdot P =$$

$$= 166 - 1 \cdot 105 = 61.$$

Thus, the true of the number $X = (1 \| 1 \| 5) = 71$ is $R_X^{(T)} = 1$.

Third stage. Checking the reliability of obtaining the rank of the number in the SRC.

Let's compare the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X. Obviously, $R_X^{(C)} = R_X^{(T)} = 1$.

Conclusions. So, since the calculated $R_X^{(C)}$ and true $R_X^{(T)}$ ranks of the number X are equal, then the calculated rank $R_X^{(C)}$ is determined correctly.

Example 3. Carry out control of the arithmetic operation of addition of two numbers $X = (x_1 \| x_2 \| x_3) = (2 \| 4 \| 4) = 74$ and $Y = (y_1 \| y_2 \| y_3) = (2 \| 3 \| 1) = 8$ presented in the SRC.

In accordance with the procedure described above for determining the rank of a number in the SRC, it is initially necessary to determine the calculated ranks $R_X^{(C)}$ and $R_Y^{(C)}$ of the summands of the numbers X and Y.

First, in accordance with the control method, let's determine the calculated rank $R_X^{(C)}$ of the number $X = (2 \| 4 \| 4) = 74$.

Let's zero the number X to the first base by adding the value of minimum ZC $z^{(1)}$:

$$X_1 = X + z^{(1)} = (2 \| 4 \| 4) + (1 \| 1 \| 1) = (0 \| 5 \| 5).$$

Using expression (13), we obtain:

$$R_{X_1}^{(C)} = R_X^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(1)}}{p_i} \right] \cdot e_i = R_X^{(C)} + R_{z^{(1)}} -$$

$$- \left\{ \left[\frac{x_1 + z_1^{(1)}}{p_1} \right] \cdot e_1 + \left[\frac{x_2 + z_2^{(1)}}{p_2} \right] \cdot e_2 + \left[\frac{x_3 + z_3^{(1)}}{p_3} \right] \cdot e_3 \right\} =$$

$$= R_X^{(C)} + 1 - \left\{ \left[\frac{2+1}{3} \right] \cdot 2 + \left[\frac{4+1}{5} \right] \cdot 1 + \left[\frac{4+1}{7} \right] \cdot 1 \right\} =$$

$$= R_X^{(C)} + 1 - 1 \cdot 2 - 1 \cdot 1 - 0 \cdot 1 = R_X^{(C)} + 1 - 3 = R_X^{(C)} - 2.$$

So, since the residue $x_3 = 5$ has not been reset to zero, then let's add the value $z^{(3)} = (0 \| 0 \| 1)$:

$$X_2 = X_1 + z^{(3)} = (0 \| 5 \| 5) + (0 \| 0 \| 1) = (0 \| 0 \| 6).$$

Using expression (13) let's determine the calculated rank of the number $X_2 = (0 \| 0 \| 6)$:

$$R_{X_2}^{(C)} = R_{X_1}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$

$$= \left(R_X^{(C)} - 2 \right) + R_{z^{(3)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{5+1}{7} \right] \cdot 1 \right\} =$$

$$= R_X^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_X^{(C)} - 2.$$

So, since the residue $x_3 = 6$ has not been reset to zero, then let's add the value $z^{(3)}$ again:

$$X_n = X_2 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

For the zeroable number $X_n = (0 \| 0 \| 0)$ according to expression (13) we have the calculated rank equal to:

$$R_{X_n}^{(C)} = R_{X_2}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{x_i + z_j^{(3)}}{p_i} \right] \cdot e_i =$$

$$= \left(R_X^{(C)} - 2 \right) + R_{z^{(3)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{6+1}{7} \right] \cdot 1 \right\} =$$

$$= R_X^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 1 \cdot 1 = R_X^{(C)} - 3.$$

Based on expression (11), we can determine the calculated rank $R_X^{(C)}$ of the number $X = (2 \| 4 \| 4) = 74$:

DOI 10.15588/1607-3274-2024-1-21

$$R_{X_n}^{(C)} = -1 \to R_X^{(C)} - 3 = -1 \to R_X^{(C)} = 2.$$

Checking. Let's calculate the true rank $R_X^{(T)}$ of number $X = (2 \| 4 \| 4) = 74$ using orthogonal bases B_i SRC (Table 1). We have that in the PNS:

$$X_{PNS} = x_1 \cdot B_1 + x_2 \cdot B_2 + x_3 \cdot B_3 - R_X^{(T)} \cdot P = \\ = 2 \cdot 70 + 4 \cdot 21 + 4 \cdot 15 - 2 \cdot 105 = 74.$$

So the true rank of the number $X = (2 \| 4 \| 4) = 74$ is $R_X^{(T)} = 2$.

Conclusions. So, as the calculated rank $R_X^{(C)}$ of the number X is equal to the true rank $R_X^{(T)}$ of the number X, i.e. $R_X^{(C)} = R_X^{(T)} = 2$, then the calculated rank is determined correctly.

Let's determine the calculated rank $R_Y^{(C)}$ of the second summand $Y = (2 \| 3 \| 1) = 8$. First, as for the first summand X, let's reduce the number Y to the form $Y_n = (0 \| 0 \| 0)$, i.e. let's zero the number Y according to the first base $p_1 = 3$, adding the minimum ZC $z^{(1)}$ to the original number Y:

$$Y_1 = Y + z^{(1)} = (2 \| 3 \| 1) + (1 \| 1 \| 1) = (0 \| 4 \| 2).$$

Using expression (13) we determine the calculated rank of the number $Y_1 = (0 \| 4 \| 2)$:

$$R_{Y_1}^{(C)} = R_Y^{(C)} + R_{z^{(1)}} - \sum_{i=1}^{3} \left[\frac{y_i + z_j^{(1)}}{p_i} \right] \cdot e_i = R_Y^{(C)} + 1 - \\ - \left\{ \left[\frac{y_1 + z_1^{(1)}}{p_1} \right] \cdot e_1 + \left[\frac{y_2 + z_2^{(1)}}{p_2} \right] \cdot e_2 + \left[\frac{y_3 + z_3^{(1)}}{p_3} \right] \cdot e_3 \right\} = \\ = R_Y^{(C)} + 1 - \left\{ \left[\frac{2+1}{3} \right] \cdot 2 + \left[\frac{3+1}{5} \right] \cdot 1 + \left[\frac{1+1}{7} \right] \cdot 1 \right\} = \\ = R_Y^{(C)} + 1 - 1 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_Y^{(C)} + 1 - 2 = R_Y^{(C)} - 1.$$

So, since the residue $y_2 = 4$ has not been reset to zero, then let's add the value of the minimum ZC $z^{(2)} = (0 \| 3 \| 3)$:

$$Y_2 = Y_1 + z^{(2)} = (0 \| 4 \| 2) + (0 \| 3 \| 3) = (0 \| 2 \| 5).$$

$$R_{Y_2}^{(C)} = R_{Y_1}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3} \left[\frac{y_i + z_j^{(2)}}{p_i} \right] \cdot e_i = \left(R_Y^{(C)} - 1 \right) + R_{z^{(2)}} - \\ - \left\{ \left[\frac{y_1 + z_1^{(2)}}{p_1} \right] \cdot e_1 + \left[\frac{y_2 + z_2^{(2)}}{p_2} \right] \cdot e_2 + \left[\frac{y_3 + z_3^{(2)}}{p_3} \right] \cdot e_3 \right\} = \\ = \left(R_Y^{(C)} - 1 \right) + 1 - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{4+3}{5} \right] \cdot 1 + \left[\frac{2+3}{7} \right] \cdot 1 \right\} = \\ = R_Y^{(C)} - 1 + 1 - 0 \cdot 2 - 1 \cdot 1 - 0 \cdot 2 = R_Y^{(C)} - 1.$$

After carrying out the stage of zeroing the residue modulo $p_2 = 5$, we obtain the value $y_2 = 2$. Thus, it is necessary to carry out one more time zeroing the residue $y_2 = 2$ of the number $Y_2 = (0 \| 2 \| 5)$ modulo $p_2 = 5$. Let's add the minimum ZC $z^{(2)}$ again:

$$Y_3 = Y_2 + z^{(2)} = (0 \| 2 \| 5) + (0 \| 3 \| 3) = (0 \| 0 \| 1).$$

$$R_{Y_3}^{(C)} = R_{Y_2}^{(C)} + R_{z^{(2)}} - \sum_{i=1}^{3} \left[\frac{y_i + z_j^{(2)}}{p_i} \right] \cdot e_i = \\ \left(R_Y^{(C)} - 1 \right) + R_{z^{(2)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{2+3}{5} \right] \cdot 1 + \left[\frac{5+3}{7} \right] \cdot 1 \right\} = \\ = R_Y^{(C)} - 1 + 1 - 0 \cdot 2 - 1 \cdot 1 - 1 \cdot 1 = R_Y^{(C)} - 2.$$

Let's reset the residue $y_3 = 1$ to zero according to the third module $p_3 = 7$ of the number $Y_3 = (0 \| 0 \| 1)$. Let's add the number Y_3 and minimum ZC $z^{(3)} = (0 \| 0 \| 1)$ and get:

$$Y_4 = Y_3 + z^{(3)} = (0 \| 0 \| 1) + (0 \| 0 \| 1) = (0 \| 0 \| 2).$$

$$R_{Y_4}^{(C)} = R_{Y_3}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{y_i + z_j^{(3)}}{p_i} \right] \cdot e_i = \left(R_Y^{(C)} - 2 \right) + R_{z^{(3)}} - \\ - \left\{ \left[\frac{y_1 + z_1^{(3)}}{p_1} \right] \cdot e_1 + \left[\frac{y_2 + z_2^{(3)}}{p_2} \right] \cdot e_2 + \left[\frac{y_3 + z_3^{(3)}}{p_3} \right] \cdot e_3 \right\} = \\ = \left(R_Y^{(C)} - 2 \right) + 0 - \left(\left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{1+1}{7} \right] \cdot 1 \right) = \\ = R_Y^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 1 - 0 \cdot 1 = R_Y^{(C)} - 2.$$

After adding four more times with minimum ZC $z^{(3)}$ we get the number $Y_8 = (0 \| 0 \| 6)$. Add to this number the value of ZC $z^{(3)}$:

$$Y_n = Y_8 + z^{(3)} = (0 \| 0 \| 6) + (0 \| 0 \| 1) = (0 \| 0 \| 0).$$

Let's determine the calculate rank for the zeroable number $Y_n = (0 \| 0 \| 0)$:

$$R_{Y_n}^{(C)} = R_{Y_8}^{(C)} + R_{z^{(3)}} - \sum_{i=1}^{3} \left[\frac{y_i + z_j^{(3)}}{p_i} \right] \cdot e_i = \\ \left(R_Y^{(C)} - 2 \right) + R_{z^{(3)}} - \left\{ \left[\frac{0+0}{3} \right] \cdot 2 + \left[\frac{0+0}{5} \right] \cdot 1 + \left[\frac{6+1}{7} \right] \cdot 1 \right\} = \\ = R_Y^{(C)} - 2 + 0 - 0 \cdot 2 - 0 \cdot 2 - 1 \cdot 1 \\ = R_Y^{(C)} - 3.$$

According to expression (11) we have that:

$$R_{Y_n}^{(C)} = -1 \to R_Y^{(C)} - 3 = -1 \to R_Y^{(C)} = 2.$$

DOI 10.15588/1607-3274-2024-1-21

Thus, the calculated rank of the number $Y=(2\|3\|1)=8$ is $R_Y^{(C)}=2$.

Checking. In the PNS, the value of the second summand $Y=(2\|3\|1)=8$ is equal to the value:

$$Y_{PNS}=\sum_{i=1}^{3}y_i\cdot B_i=2\cdot 70+3\cdot 21+1\cdot 15=$$
$$=218-R_Y^{(T)}\cdot 105=218-2\cdot 105=8.$$

So the true rank of the number $Y=(2\|3\|1)=8$ is $R_Y^{(T)}=2$.

Conclusions. Since the calculated rank of the number Y is equal to the true rank of the number Y, i.e. $R_Y^{(C)}=R_Y^{(T)}=2$, then the calculated rank is determined correctly.

Let's determine the sum of two numbers $X+Y$:

$$X+Y=(2\|4\|4)+(2\|3\|1)=(1\|2\|5).$$

According to expression (1), the calculated rank $R_{X+Y}^{(C)}$ of the sum of two numbers $X+Y$ is equal to:

$$R_{X+Y}^{(C)}=R_X^{(C)}+R_Y^{(C)}-\sum_{i=1}^{3}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i=$$
$$=R_X^{(C)}+R_Y^{(C)}-\left\{\left[\frac{2+2}{3}\right]\cdot 2+\left[\frac{4+3}{5}\right]\cdot 1+\left[\frac{4+1}{7}\right]\cdot 1\right\}=$$
$$=2+2-1\cdot 2-1\cdot 1-0\cdot 1=1.$$

Checking. $(X+Y)_{SRC}=(1\|2\|5)$ and $(X+Y)_{PNS}=$ $=1\cdot 70+2\cdot 21+5\cdot 15-R_{X+Y}^{(T)}\cdot P=187-1\cdot 105=82.$

Conclusion. The true rank $R_{X+Y}^{(T)}$ of the number $X+Y=(1\|2\|5)$ is equal to the calculated $R_{X+Y}^{(T)}=R_{X+Y}^{(C)}=1$. Therefore, there was no overflow when performing the addition operation.

Example 4. Check for overflow when adding two numbers $X=Y=(2\|4\|4)=74$.

Let's determine the sum of two numbers $X+Y$:

$$X+Y=(2\|4\|4)+(2\|4\|4)=(1\|3\|1).$$

Considering that in example 3, the calculated rank $R_X^{(C)}$ of the number $X=(2\|4\|4)=74$ was calculated, since the numbers X and Y are the same, its calculated ranks are also the same: $R_X^{(C)}=R_Y^{(C)}=2$.

According to expression (1), the calculated rank $R_{X+Y}^{(C)}$ of the sum $X+Y$ of two numbers in the SRC is equal to:

$$R_{X+Y}^{(C)}=R_X^{(C)}+R_Y^{(C)}-\sum_{i=1}^{3}\left[\frac{x_i+y_i}{p_i}\right]\cdot e_i=$$
$$=2+2-\left\{\left[\frac{2+2}{3}\right]\cdot 2+\left[\frac{4+4}{5}\right]\cdot 1+\left[\frac{4+4}{7}\right]\cdot 1\right\}=$$
$$=2+2-1\cdot 2-1\cdot 1-1\cdot 1=0.$$

Thus, the calculated rank of the sum of two numbers $X+Y$ is equal to the value $R_{X+Y}^{(C)}=0$.

Checking. $(X+Y)_{SRC}=(1\|3\|1)$ and $(X+Y)_{PNS}=$ $=1\cdot 70+3\cdot 21+1\cdot 15-R_{X+Y}^{(T)}\cdot P=148-R_{X+Y}^{(T)}\cdot 105=148-$ $-1\cdot 105=43$. Thus, $R_{X+Y}^{(T)}=1$.

Conclusion. It is obvious that the true rank $R_{X+Y}^{(T)}=1$ doesn't coincide with the calculated rank $R_{X+Y}^{(C)}=0$. The inequality $R_{X+Y}^{(C)}\neq R_{X+Y}^{(T)}$ of the rank values of the number $X+Y$ shows that there was an overflow during the operation. Therefore, the sum of two numbers $X+Y$ has the wrong value: 43, not 148.

6 DISCUSSION

When solving CS computational problems, it becomes necessary to take into account the overflow of the bit grid that occurs during data processing. Analysis of these processes showed the following. To solve the problem associated with overflow of the CS bit grid, there are various approaches. One of them is the use of a wider bit grid to represent the meanings of numbers, i.e. Data processing is carried out on computers with a relatively large bit grid [23]. This allows you to increase the range of values of processed numbers that can be represented in the CS without taking into account the consequences of the overflow factor. However, this requires more memory and can increase the time it takes to solve a calculation problem, which is especially critical for real-time CS [24, 25]. The MDBGO proposed in the article is intended for use in a CS that operates in the SRC. The properties of the SRC (independence, equality and low-bit residues, the totality of which determines the non-positional code structure) and their use in creating the structure of the CS determine the interpretation of the CS in the SRC as a set of individual low-bit computers [7, 26]. Each computer operates using a specific SRC module [27, 28]. In this case, eliminating the consequences of overflowing the bit grid is carried out without interrupting the computational process, i.e. during the operation of the CS, without stopping the calculations.

The reliability and significance of the results obtained are due to the following factors:

– the research was carried out using the modern proven mathematical apparatus of number theory and the basic theoretical principles of machine arithmetic in the residual classes;

– the consistency of the results obtained with both the scientific provisions of the general theory of constructing the structures of positional CSs, and the theoretical provisions of the creation of CSs operating in the SRC;

– the coincidence of some theoretical conclusions with existing modern provisions on the prospects for the development of real-time CS;

– the results of the analysis of the given specific examples of the use of MDBGO for various initial data of the SRC.

DOI 10.15588/1607-3274-2024-1-21

This problem, solved in the article, was directly or indirectly considered in the monographs: Aksushskyi I. Ya. and Yuditskyi D. I. "Machine arithmetic in residual classes" [1] and Torgashov V. A. "System of residual classes and reliability of digital computers" [8]. These monographs provide directions for further research in the direction of improving real-time CS structures in the SRC. In particular, some theoretical further research is presented.

CONCLUSIONS

The current scientific problem of using MDBGO in the CS operating in the SRC has been solved. The use of MDBGO to detect the fact of overflow of the CS bit grid is shown using specific examples of the implementation of the operation of adding two numbers in the SRC. The MDBGO considered in the article is based on the use of positional feature of a non-positional code of numbers in the SRC, namely on the calculation and use of the true and calculated ranks of numbers.

The scientific novelty of the results obtained lies in the fact that when implementing MDBGO, the procedure for determining the rank of a number is carried out directly in the process of performing the operation of adding two numbers, being an essential part of it. This circumstance makes it possible to reduce the time it takes to detect the fact of overflow of the CS bit grid in the SRC. In addition, a feature of the presented method for detecting the fact of overflow of the CS bit grid in the SRC is that MDBGO can simultaneously be used to organize the process of monitoring the operation of adding numbers modulo. This expands the functionality of the MDBGO.

Practical significance of the results. To confirm the practical feasibility of the procedure, examples are given of determining the overflow of the result of the operation of adding two numbers in the SRC. A set of examples is given of the specific implementation of the operation of overflowing the bit grid using the MDBGO method, while simultaneously implementing control of the addition of two numbers for a given SRC, which confirm the effectiveness of using the considered method.

Prospects for further research are as follows. In the SRC, using the basic properties of the class of residues, control of arithmetic operations can be carried out in the dynamics of the computational process, i.e. without stopping the calculation process. This makes it possible, firstly, to fully use the main property of the SRC – the high speed of the CS execution of arithmetic modular operations. Secondly, it is possible to reduce the amount of the CS equipment required to implement positional operations in the SRC. The research results obtained in the article are recommended for use in on-board digital computers of ballistic missiles, in the use of unmanned aerial vehicle computers and in the use of specialized computers for a wide class of various non-recoverable disposable aircraft operating in the SRC. The feasibility of further research in the field of using non-positional code structures in the SRC, in particular, expanding the area of practical use of MDBGO, is due to the fact that positive research results will make it possible to create ultra-fast and fault-tolerant real-time specialized CS.

REFERENCES

1. Akushskij I. Ya., Yudickij D. I. Mashinnaya arifmetika v ostatochnyx klassax. Moscow, Sovet. radio, 1968, 440 p.
2. Krasnobayev V., Yanko A., Hlushko A. Information Security of the National Economy Based on an Effective Data Control Method, *Journal of International Commerce, Economics and Policy*, 2023, No. 14(3), Article no. 2350021. DOI: 10.1142/S1793993323500217
3. Butt M. A., Ajmal Z., Khan Z. I. , Idrees M., Javed Y. An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques, *Applied Sciences*, 2022, No. 12(13), P. 6702. DOI: 12. 6702. 10.3390/app12136702
4. Gavrylenko S. Y., Chelak V. V., Semenov S. G. Development of method for identification the computer system state based on the decision tree with multi-dimensional nodes, R*adio Electronics, Computer Science, Control*, 2022, No. (2), pp. 113–121. DOI: https://doi.org/10.15588/1607-3274-2022-2-11
5. Laktionov O., Lievi L., Tretia A., Movin M. Investigation of combined ensemble methods for diagnostics of the quality of interaction of human-machine systems, *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, 2023, No. 4, pp. 138–143. DOI: https://doi.org/10.33271/nvngu/2023-4/138
6. Salnikov D., Karaman D., Krylova V. Highly reconfigurable soft-CPU based peripheral modules design, *Advanced Information Systems*, 2023, No. 7(2), pp. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13
7. Yanko A., Koshman S., Krasnobayev V. Algorithms of data processing in the residual classes system, *International Scientific-Practical Conference: Problems of Infocommunications. Science and Technology (PIC S&T). Kharkov*, 2017, pp. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
8. Torgashev V. A. Sistema ostatochnyx klassov i nadezh-nost' CVM. Moscow, Sovet. radio, 1973, 118 p.
9. Mohan P. V. A. Residue Number Systems: Theory and Applications. Birkhäuser Basel, Springer International Publishing, Switzerland, 2016, 351 p. DOI: 10.1007/978-3-319-41385-3_2 9
10. Colbert I., Pappalardo A., Petri-Koenig J. A2Q: Accumulator-Aware Quantization with Guaranteed Overflow Avoidance Supplementary Material, *2023 IEEE/CVF International Conference on Computer Vision (ICCV). Paris*, France, 2023, pp. 16943–16952. DOI: 10.1109/ICCV51070.2023.0155
11. Bo L., Ruifeng Z., Jiangang L., Wenxin G., Yang L. Control on Abnormal Data Overflow of Distribution Network Management Platform, *Journal of Physics: Conference Series 1748*, 2021, No. 032064, pp. 1–5. DOI: 10.1088/1742-6596/1748/3/032064
12. Meakin R. L. Adaptive spatial partitioning and refinement for overset structured grids, *Computer Methods in Applied Mechanics and Engineering*, 2000, No. 189 (4), pp. 1077–1117. DOI: https://doi.org/10.1016/S0045-7825(99)00369-2
13. Hui H. W., Zhou C. C., Xu S. G., Lin F. H. A Novel Secure Data Transmission Scheme in Industrial Internet of Things, *China Communications*, 2020, No. 17 (1), pp. 73–88. DOI: 10.23919/JCC.2020.01.006
14. Chiamwongpaet S., Piromsopa K. Boundary Bit: Architectural Bound Checking for Buffer-Overflow Protection, *ECTI Transactions on Computer and Information Technology (ECTI-CIT)*, 2020, No. 14, pp. 162–173. DOI:10.37936/ecti-cit.2020142.212338
15. Tsukada M., Matsutani H. An Overflow/Underflow-Free Fixed-Point Bit-Width Optimization Method for OS-ELM Digital Circuit, *IEICE Trans. Fundamentals, Vol. E105–A*, No. 3, 2022, pp. 437–447. DOI: 10.1587/transfun.2021VLP0017
16. Xie H., Song Y., Cai L., Li M. Overflow Aware Quantization: Accelerating Neural Network Inference by Low-bit Multiply-Accumulate Operations, *29th International Joint Conference on*

DOI 10.15588/1607-3274-2024-1-21

Artificial Intelligence. Japan, 2020, pp. 868–875. DOI: https://doi.org/10.24963/ijcai.2020/121
17. Lee S., Shin D. J. Overflow-Detectable Floating-Point Fully Homomorphic Encryption, *IEEE Access*, 2024, pp. 1–21. DOI: 10.1109/ACCESS.2024.3351738
18. Li T., Wang Y., Li R., Li K. A bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters, *2009 IEEE 8th International Conference on ASIC*. Changsha, Hunan, 2009, pp. 238–241. DOI: 10.1109/ASICON.2009.5351482
19. Cha Y., Cho G., Choi H., Song H. N bit result integer multiplier with overflow detector, *Electronics Letters*, 2001, No.(37), pp. 940–942. DOI: 10.1049/el:20010633.
20. Ivohin E. V. Gavrylenko V. V., Ivohina K. E. On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method, *Radio Electronics, Computer Science, Control*, 2023, No. (3), pp. 141–147. DOI: https://doi.org/10.15588/1607-3274-2023-3-14
21. Ulman Z., Czyzak M. and Zurada J. Effective RNS scaling algorithm with the Chinese remainder theorem decomposition, *Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing*, 1993, Vol. 2, pp. 528–531 DOI: 10.1109/PACRIM.1993.407305
22. Krasnobayev V. A., Yanko A. S., Kovalchuk D. M. Methods for tabular implementation of arithmetic operations of the residues of two numbers represented in the system of residual classes, *Radio Electronics, Computer Science, Control*, 2022, No. 4, pp. 18–27. DOI: https://doi.org/10.15588/1607-3274-2022-4-2
23. Subbotin S. A. Data clustering based on inductive learning of neuro-fuzzy network with distance hashing, *Radio Electronics, Computer Science, Control*, 2022, No. 4, pp. 71–85. DOI: https://doi.org/10.15588/1607-3274-2022-4-6
24. Krasnobayev V. Koshman S., Yanko A., Martynenko A. Method of Error Control of the Information Presented in the Modular Number System, *International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (PIC S&T)*. Kharkov, 2018, pp. 39–42. DOI: 10.1109/INFOCOMMST.2018.8632049
25. Chao Huang, Peterson D., Rauch H., Teague J., Fraser D. Implementation of a fast digital processor using the residue number system, *IEEE Transactions on Circuits and Systems*, 1981, January, Vol. 28, № 1, pp. 32–38. DOI: 10.1109/TCS.1981.1084905
26. Szabo N. S., Tanaka R. I. Residue Arithmetic and Its Applications to Computer Technology. New York, McGraw-Hill, 1967, 236 p.
27. Piestrak S. J. Design of residue generators and multioperand modular adders using carry-save adders, *Proceedings 10th IEEE Symposium on Computer Arithmetic*, 26–28 June, 1991, pp. 100–107. DOI: 10.1109/ARITH.1991.145540
28. Salnikov D., Karaman D., Krylova V. Highly reconfigurable soft-CPU based peripheral modules design, *Advanced Information Systems*, 2023, No. 7(2), pp. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13

Received 19.01.2024.
Accepted 28.02.2023.

УДК 004.222.5:681.142.01

МЕТОД ВИЗНАЧЕННЯ ПЕРЕПОВНЕННЯ РОЗРЯДНОЇ СІТКИ КОМП'ЮТЕРНОЇ СИСТЕМИ, ЩО ФУНКЦІОНУЄ В СИСТЕМІ ЗАЛИШКОВИХ КЛАСІВ

Янко А. С. – канд. техн. наук, доцент, доцент кафедри комп'ютерних та інформаційних технологій і систем Національного університету «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна.

Краснобаєв В. А. – д-р техн. наук, професор, професор кафедри електроніки та управляючих систем Харківського національного університету імені В. Н. Каразіна, Харків, Україна.

Нікольський С. Б. – канд. техн. наук, доцент, доцент кафедри інформатики комунального закладу «Харківської гуманітарно-педагогічної академії» Харківської обласної ради, Харків, Україна.

Крук О. О. – аспірант кафедри автоматики, електроніки та телекомунікацій Національного університету «Полтавська політехніка імені Юрія Кондратюка», Полтава, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто метод визначення переповнення розрядної сітки, а також комплекс прикладів практичного застосування процедури ідентифікації переповнення розрядної сітки комп'ютерної системи, що функціонує в непозиційній системі числення в залишкових класах. Об'єктом дослідження є процес обробки даних, представлених у системі залишкових класів. Мета роботи – розглянути та проаналізувати приклади визначення переповнення розрядної сітки комп'ютерної системи при реалізації операції додавання двох чисел у системі залишкових класів на основі застосування методу визначення переповнення розрядної сітки, заснованого на використанні поняття рангу числа.

Метод. Специфіка функціонування комп'ютерної системи у системі залишкових класів вимагає виконання як модульних операцій, так й реалізації додатково, так званих, немодульних операцій. До немодульних операцій належить операція визначення переповнення розрядної сітки комп'ютерної системи у системі залишкових класів. У непозиційній системі числення в залишкових класах реалізація процесу виявлення переповнення розрядної сітки комп'ютерної системи є важко реалізованим завданням. Розглянутий у статті метод визначення переповнення розрядної сітки грунтується на використанні позиційних ознак непозиційного коду чисел у системі залишкових класів, а саме істинного та розрахункового рангів числа. Досліджено процес визначення переповнення розрядної сітки результату операції додавання двох чисел у системі залишкових класів, оскільки саме виконання арифметичної операції додавання є основною, базовою операцією комп'ютерної системи.

Результати. Наведено приклади використання методу визначення переповнення результату операції додавання двох чисел у системі залишкових класів, в основу якого покладено модульні операції визначення розрахункового та істинного рангів безпосередньо доданків та рангу суми двох доданків. Аналіз отриманих результатів показав практичну застосовність розглянутого методу.

Висновки. Основним перевагою представленого методу є те, що визначення переповнення розрядної сітки можна здійснювати у динаміці обчислювального процесу комп'ютерної системи, тобто без зупинки розв'язання задачі. Ця обставина дозволяє знизити непродуктивні витрати комп'ютерної системи в системі залишкових класів. Крім цього, цей метод можна використовувати для контролю операції додавання двох чисел в системі залишкових класів. Це підвищує достовірність отримання істинного результату операції додавання двох чисел в системі залишкових класів.

DOI 10.15588/1607-3274-2024-1-21

КЛЮЧОВІ СЛОВА: арифметична операція модульного додавання, переповнення розрядної сітки, операція порівняння, комп'ютерна система, непозиційний код, ранг числа, система залишкових класів, процедура нулевізації.

ЛІТЕРАТУРА

1. Акушский И.Я. Машинная арифметика в остаточных классах / И. Я. Акушский, Д. И. Юдицкий. – Москва : Совет. радио, 1968. – 440 с.
2. Krasnobayev V. Information Security of the National Economy Based on an Effective Data Control Method / V. Krasnobayev, A. Yanko, A. Hlushko // Journal of International Commerce, Economics and Policy. – 2023. – No. 14(3). – Article no. 2350021. DOI: 10.1142/S1793993323500217
3. Butt M. A. An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques / [M. A. Butt, Z. Ajmal, Z. I. Khan et al.] // Applied Sciences. – 2022. – No. 12(13), P. 6702. DOI: 12. 6702. 10.3390/app12136702
4. Gavrylenko S. Y. Development of method for identification the computer system state based on the decision tree with multi-dimensional nodes / S. Y. Gavrylenko, V. V. Chelak, S. G. Semenov // Radio Electronics, Computer Science, Control. – 2022. – No. (2). – P. 113–121. DOI: https://doi.org/10.15588/1607-3274-2022-2-11
5. Investigation of combined ensemble methods for diagnostics of the quality of interaction of human-machine systems / [O. Laktionov, L. Lievi, A.Tretia, M. Movin] // Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu. – 2023. – No. (4). – P. 138–143. DOI: https://doi.org/10.33271/nvngu/2023-4/138
6. Salnikov D. Highly reconfigurable soft-CPU based peripheral modules design / D. Salnikov, D. Karaman, V. Krylova // Advanced Information Systems, 7(2). – 2023. – P. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13
7. Yanko A. Algorithms of data processing in the residual classes system / A. Yanko, S. Koshman, V. Krasnobayev // International Scientific-Practical Conference: Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov. – 2017. – P. 117–121. DOI: 10.1109/INFOCOMMST.2017.8246363
8. Торгашев В. А. Система остаточных классов и надежность ЦВМ / В. А. Торгашев. – Москва : Совет. радио, 1973. – 118 с.
9. Mohan P. V. A. Residue Number Systems: Theory and Applications / P. V. A. Mohan. – Birkhäuser Basel: Springer International Publishing, Switzerland, 2016. – 351 p. DOI: 10.1007/978-3-319-41385-3_2 9
10. Colbert I. A2Q: Accumulator-Aware Quantization with Guaranteed Overflow Avoidance Supplementary Material / I. Colbert, A. Pappalardo, J. Petri-Koenig // 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France. – 2023. – P. 16943–16952. DOI: 10.1109/ICCV51070.2023.0155
11. Control on Abnormal Data Overflow of Distribution Network Management Platform / [L. Bo, Z. Ruifeng, L. Jiangang et al.] // Journal of Physics: Conference Series 1748. –2021. – No. 032064. – P. 1–5. DOI: 10.1088/1742-6596/1748/3/032064
12. Meakin R. L. Adaptive spatial partitioning and refinement for overset structured grids / R. L. Meakin // Computer Methods in Applied Mechanics and Engineering. – 2000. – No. 189 (4). – P. 1077–1117. DOI: https://doi.org/10.1016/S0045-7825(99)00369-2
13. A Novel Secure Data Transmission Scheme in Industrial Internet of Things / [H. W. Hui, C. C. Zhou, S. G. Xu, F. H. Lin] // China Communications. – 2020. – No. 17 (1). – P. 73–88. DOI: 10.23919/JCC.2020.01.006
14. Chiamwongpaet S. Boundary Bit: Architectural Bound Checking for Buffer-Overflow Protection / S. Chiamwongpaet, K. Piromsopa // ECTI Transactions on Computer and Information Technology (ECTI-CIT). – 2020. – No. 14. – P. 162–173. DOI:10.37936/ecti-cit.2020142.212338
15. Tsukada M. An Overflow/Underflow-Free Fixed-Point Bit-Width Optimization Method for OS-ELM Digital Circuit / M. Tsukada, H. Matsutani // IEICE Trans. Fundamentals, 2022. – Vol. E105–A, No. 3. – P. 437–447. DOI: 10.1587/transfun.2021VLP0017
16. Overflow Aware Quantization: Accelerating Neural Network Inference by Low-bit Multiply-Accumulate Operations / [H. Xie, Y. Song, L. Cai, M. Li] // 29th International Joint Conference on Artificial Intelligence, Japan, 2020. – P. 868–875. DOI: https://doi.org/10.24963/ijcai.2020/121
17. Lee S. Overflow-Detectable Floating-Point Fully Homomorphic Encryption / S. Lee, D. J. Shin // IEEE Access. – 2024. –P. 1–21. DOI: 10.1109/ACCESS.2024.3351738
18. A bidirectional overflow digital correction algorithm with a single bit redundancy used in the pipeline A/D converters / [T. Li, Y. Wang, R. Li, K. Li] // 2009 IEEE 8th International Conference on ASIC, Changsha, Hunan, 2009. –P. 238–241. DOI: 10.1109/ASICON.2009.5351482
19. N bit result integer multiplier with overflow detector / [Y. Cha, G. Cho, H. Choi, H. Song] // Electronics Letters, 2001. – No.(37). – P. 940–942. DOI: 10.1049/el:20010633.
20. Ivohin E. V. On the recursive algorithm for solving the traveling salesman problem on the basis of the data flow optimization method / E. V. Ivohin, V. V. Gavrylenko, K. E. Ivohina // Radio Electronics, Computer Science, Control. – 2023. – No. (3). – P. 141–147. DOI: https://doi.org/10.15588/1607-3274-2023-3-14
21. Ulman Z. Effective RNS scaling algorithm with the Chinese remainder theorem decomposition / Z. Ulman, M. Czyzak and J. Zurada // Proceedings of IEEE Pacific Rim Conference on Communications Computers and Signal Processing. – 1993. – Vol. 2. – P. 528–531 DOI: 10.1109/PACRIM.1993.407305
22. Krasnobayev V. A. Methods for tabular implementation of arithmetic operations of the residues of two numbers represented in the system of residual classes / V. A. Krasnobayev, A. S. Yanko, D. M. Kovalchuk // Radio Electronics, Computer Science, Control. – 2022. – No.(4). – P. 18–27. DOI: https://doi.org/10.15588/1607-3274-2022-4-2
23. Subbotin S. A. Data clustering based on inductive learning of neuro-fuzzy network with distance hashing / S. A. Subbotin // Radio Electronics, Computer Science, Control. –2022. – No. (4). – P. 71–85. DOI: https://doi.org/10.15588/1607-3274-2022-4-6
24. Method of Error Control of the Information Presented in the Modular Number System / [V. Krasnobayev, S. Koshman, A. Yanko, A. Martynenko] // International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (PIC S&T). – Kharkov, 2018. – P. 39–42. DOI: 10.1109/INFOCOMMST.2018.8632049
25. Chao Huang Implementation of a fast digital processor using the residue number system / [Chao Huang, D. Peterson, H. Rauch et al.] // IEEE Transactions on Circuits and Systems. 1981. – January, Vol. 28, № 1. – P. 32–38. DOI: 10.1109/TCS.1981.1084905
26. Szabo N. S. Residue Arithmetic and Its Applications to Computer Technology / N. S. Szabo, R. I. Tanaka. – New York : McGraw-Hill, 1967. – 236 p.
27. Piestrak S. J. Design of residue generators and multioperand modular adders using carry-save adders / S. J. Piestrak // Proceedings 10th IEEE Symposium on Computer Arithmetic, 26–28 June, 1991. – P. 100–107. DOI: 10.1109/ARITH.1991.145540
28. Salnikov D. Highly reconfigurable soft-CPU based peripheral modules design / D. Salnikov, D. Karaman, V. Krylova // Advanced Information Systems. – 2023. – No. 7(2). P. 92–97. DOI: https://doi.org/10.20998/2522-9052.2023.2.13

DOI 10.15588/1607-3274-2024-1-21

УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

CONTROL IN TECHNICAL SYSTEMS

UDC 62.505:629.524

STEWART PLATFORM DYNAMICS MODEL IDENTIFICATION

Zozulya V. A. – PhD, Associate Professor of the Department of Digital Economy and System Analysis, State University of Trade and Economics, Kyiv, Ukraine.

Osadchy S. I. – Dr. Sc., Professor, of the Department of Aircraft Construction, Aircraft Engines, and Airworthiness Maintenance, Flight Academy of the National Aviation University, Kropyvnytskyi, Ukraine.

Nedilko S. N. – Dr. Sc., Professor, Acting Director of the Flight Academy of the National Aviation University, Kropyvnytskyi, Ukraine.

ABSTRACT

Context. At the present stage, with the current demands for the accuracy of motion control processes for a moving object on a specified or programmable trajectory, it is necessary to synthesize the optimal structure and parameters of the stabilization system (controller) of the object, taking into account both real controlled and uncontrolled stochastic disturbing factors. Also, in the process of synthesizing the optimal controller structure, it is necessary to assess and consider multidimensional dynamic models, including those of the object itself, its basic components, controlled and uncontrolled disturbing factors that affect the object in its actual motion.

Objective. The aim of the research, the results of which are presented in this article, is to obtain and assess the accuracy of the Stewart platform dynamic model using a justified algorithm for the multidimensional moving object dynamics identification.

Method. The article employs a frequency-domain identification method for multidimensional stochastic stabilization systems of moving objects with arbitrary dynamics. The proposed algorithm for multidimensional moving object dynamics model identification is constructed using operations of polynomial and fractional-rational matrices addition, multiplication, Wiener factorization, Wiener separation, and determination of dispersion integrals.

Results. As a result of the conducted research, the problem of identifying the dynamic model of a multidimensional moving object is formalized, illustrated by the example of a test stand based on the Stewart platform. The outcomes encompass the identification of the dynamic model of the Stewart platform, its transfer function, and the transfer function of the shaping filter. The verification of the identification results confirms the sufficient accuracy of the obtained models.

Conclusions. The justified identification algorithm allows determining the order and parameters of the linearized system of ordinary differential equations for a multidimensional object and the matrix of spectral densities of disturbances acting on it under operating conditions approximating the real functioning mode of the object prototype. The analysis of the identification results of the dynamic models of the Stewart platform indicates that the primary influence on the displacement of the center of mass of the moving platform is the variation in control inputs. However, neglecting the impact of disturbances reduces the accuracy of platform positioning. Therefore, for the synthesis of the control system, methods should be applied that enable determining the structure and parameters of a multidimensional controller, considering such influences.

KEYWORDS: Identification, transfer function matrix, spectral density, quality functional, Stewart platform.

ABBREVIATIONS

AHRS is an attitude and heading reference systems;
IKP is an inverse kinematics problem;
IMU is an inertial measurement unit;
LMS is a linear movement system;
ONS are sensors of orientation and navigation;
RT is a target Real-Time;
SLM are sensors of linear movements;
TIG is a trajectory and interpolation generator;
WS is a working surface.

NOMENCLATURE

D is a result of the Wiener factorization of the transposed matrix $S'_{\zeta\zeta}$;

E_n is a $n\times n$ unit matrix;

J is a quality functional;

M is a matrix of dimension $m\times n$, the elements of which are polynomials from the differentiation operator s;

m is a number of signals at the output of the control system;

n is a local system inputs number;

$O_{m\times n}$ is a zero matrix of size $m\times n$;

P is a polynomial matrix of dimension $m\times m$;

R is an additionally defined weight matrix;

r_0 is a vector of program signals;

$S'_{r_0r_0}$ is a transposed spectral density matrix of the vector r_0;

DOI 10.15588/1607-3274-2024-1-22

$S'_{x_1x_1}$ is a transposed spectral density matrix of the vector x_1;

$S'_{\zeta x}$ is a transposed matrix of mutual spectral densities between the generalised input vector ζ and the vector x_1;

$S'_{x\zeta}$ is a transposed matrix of mutual spectral densities between the vectors x_1 and ζ;

$S'_{\Delta\Delta}$ is a transposed matrix of spectral densities of uncorrelated white noise of single intensity;

$S'_{\varepsilon\varepsilon}$ is a transposed matrix of spectral densities of the vector of identification errors ε;

$S'_{\psi_{ob}\psi_{ob}}$ is a transposedmatrix of spectral densities of the disturbing influence;

$S'_{\zeta\zeta}$ is a transposed spectral density matrix of the generalized input vector;

T_0 is a matrix of results of dividing the polynomials of the numerators by the polynomials of the denominator of the product on the right side of the expression;

T_+ is a matrix of fractional rational functions whose poles are located in the left half-plane of the complex plane;

T_- is a matrix of fractional rational functions with poles in the right half-plane;

W_i are controllers;

W_1 is an optimal structure of the matrix is the transfer function of the identification object;

W_2 is an optimal structure of the matrix is the transfer function of the shaping filter Ψ;

x_0 is a vector of movement of the working surface in the working space;

x_1 is a vector of signals at the output of the control system;

x_{id} is an estimate of the vector of the WS movement, obtained through the identification process;

Δ is an uncorrelated white noise of single intensity;

Φ is a block matrix of transfer functions of size $n\times(n+m)$;

ε is a vector of identification errors;

φi is a vector of measurement noise;

Ψ is a transfer function of the shaping filter;

ψ_{ob} is a vector of centred stationary random disturbances in the control object;

ψ_{WS} is a vector of centred stationary random disturbances in the working surface;

ζ is a generalised vector of input influences.

INTRODUCTION

At the present stage of creating and operating moving objects of various purposes, spatial mechanisms with parallel structures, and a range of complex and responsible controlled technological processes, the issues of ensuring the competitiveness of products being created have become crucial. Their competitiveness is mainly influenced by the extent to which they achieve high quality and efficient utilization, as well as the ultimate goals of functioning in responsible operating modes. As stated in references [1, 2], the operation of these products is affected by a multitude of stochastic factors, both deterministic and random, which considerably complicate the processes of attaining set goals and the ultimate results in each specific responsible operating mode of moving objects. Determining the dynamics models of input-output stochastic signal vectors of a control object or its prototype in respective operating modes allows for structural identification of the dynamics models of such an object. The practical methods and algorithms for structural identification should enable the determination of dynamic models for both the control object itself in the mode of interest and the uncontrolled stochastic disturbances acting on the object under these conditions.

This relevance is driven by the practical requirements to align identification procedures with the conditions of designing closed-loop control systems. Modern methods and algorithms should be based on new approaches in creating computational procedures that exhibit enhanced accuracy and reliability in computation, reducing the growth of orders in the results. This enables the determination of the order and parameters of the linearized system of ordinary differential equations for a multidimensional object and the matrix of spectral densities of disturbances acting on it in conditions approximating the real operating mode of the experimental object.

The object of study in this paper is the Stewart platform working surface motion closed-loop control system. The Stewart platform is a spatial mechanism with a parallel kinematic structure, consisting of six identical kinematic chains (actuators) [3]. Such mechanisms can be used as machining centers (machines), coordination-measuring centers, vibration platforms (test stands), motion simulators, and stabilization platforms. The Stewart platform has six degrees of freedom for the motion of the mobile platform. By programmatically adjusting the Stewart platform drives lengths, it is possible to control the moving base position, move it in vertical and horizontal directions, as well as rotate it in three planes.

The subject of study is the algorithm for identifying the Stewart platform dynamic model, its transfer function, and the shaping filter transfer function.

The purpose of the work is to obtain the Stewart platform dynamic model using a justified algorithm for the multidimensional moving object dynamics identification.

1 PROBLEM STATEMENT

The basis of the theory of constructing mathematical models (identification) is the information-algorithmic approach. Under conditions of a priori uncertainty, the information component begins to play a dominant role, as its analysis largely determines the application of certain algorithmic procedures and formalization methods that allow the object under study mathematical description synthesis [4].

Structural identification allows establishing the interaction of system individual components in the process of

DOI 10.15588/1607-3274-2024-1-22

forming reactions. In this case, the system configuration is considered known, or assumptions are made about the class of the functional description relative to it, while the parameters that characterize the system are treated as unknown. The identification task boils down to search for solutions in the space of the sought parameters of the system [1, 5].

The analysis carried out made it possible to propose a structural diagram of the Stewart platform motion control system, which is built according to the principle of two-loop tracking systems (Fig. 1).

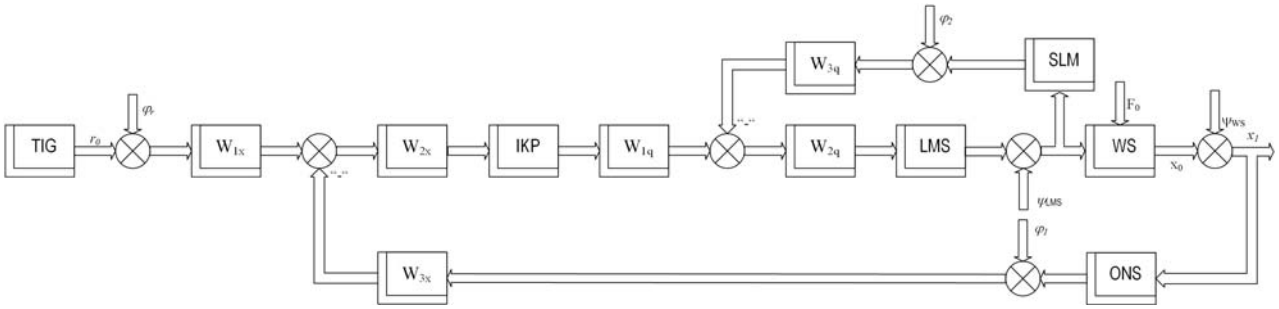


Figure 1 – Structural diagram of the motion control system of the working surface of Stewart platform

The programme signal vector r_0 is received from the trajectory and interpolation generator block, i.e. the set trajectory of movement of the working surface:

$$r_0 = [\xi_0 \quad \eta_0 \quad \zeta_0 \quad \psi_0 \quad \vartheta_0 \quad \gamma_0]^{/},$$

where ξ_0, η_0, ζ_0 – given coordinates of the rotation center of the working surface relative to the coordinate system associated with the base of Stewart platform; ψ_0 is a specified yaw angle, ϑ_0 is a specified pitch angle, γ_0 is a specified roll angle of the working surface; index $^{/}$ is a transposition sign.

The position of the working surface relative to the base characterizes the vector of movement of the working surface x_0 in the working space, of the form:

$$x_0 = [\xi_{out} \quad \eta_{out} \quad \zeta_{out} \quad \psi_{out} \quad \vartheta_{out} \quad \gamma_{out}]^{/},$$

where ξ_{out}, η_{out}, ζ_{out} – the output coordinates of the rotation center of the working surface relative to the coordinate system associated with the base of Stewart platform; ψ_{out} is a yaw output angle, ϑ_{out} is an output pitch angle, γ_{out} is and output roll angle of the working surface.

Vector of real values of the movement of the working surface x_0:

$$x_1 = x_0 + \psi_{WS},$$

where ψ_{WS} is the vector of disturbances affecting the working surface, or

$$x_1 = [\xi_1 \quad \eta_1 \quad \zeta_1 \quad \psi_1 \quad \vartheta_1 \quad \gamma_1]^{/}.$$

As a result of the structural transformation of the WS motion control scheme of Stewart platform in Fig. 1 and considering that we can measure the *m*-dimensional vector of control signals r_0 and the vector of actual values of the motion of the working surface x_1, the scheme can transform to the form depicted in Fig. 2 in cases where feedback cannot be established.

Furthermore, it is common for the origin of disturbances Δ and the vector of control signals r_0 to have distinct physical natures. Therefore, the assumption of their independence is typically adopted.

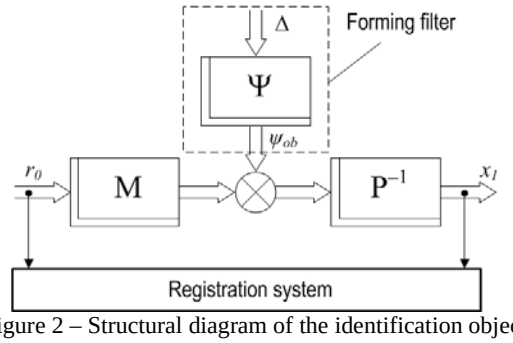


Figure 2 – Structural diagram of the identification object

Fig. 2 shows *P*, *M* – polynomial matrices from the differentiation operator *s* of the appropriate size, which characterize the dynamics of a closed system, ψ_{ob} is a vector of centered stationary random disturbances, which characterizes the action of all disturbances and noises in the control system (Fig. 1) and n is a measurable stationary random process with zero mathematical expectation and the unknown matrix of spectral densities $S_{\psi_{ob}\psi_{ob}}$.

Reviewing the types of tasks addressed by mechanisms utilizing the Stewart platform [3], one can affirm that the working surface undergoes minor movements around the center of rotation. Subsequently, in the first approximation, it is possible to formulate an ordinary differential equation representing the dynamics of the control object. This equation reflects the correlation between the system inputs r_0, ψ_{ob}, and the output x_1 (Fig. 2):

DOI 10.15588/1607-3274-2024-1-22

$$Px_1 = Mr_0 + \psi_{ob} .$$

The output x_1 and one of the inputs r_0 can be measured, but the input ψ_{ob} cannot be measured.

The vector ψ_{ob} is formed from the noise vector Δ by a linear stable filter with a matrix of transfer functions Ψ and can be represented as:

$$\psi_{ob} = \Psi\Delta .$$

The task of obtaining dynamic characteristics of a multidimensional moving object and the disturbance vector involves the following: based on the measured input vector r_0 and output vector x_1, it is necessary to determine the order and parameters of the matrices P, M, and Ψ that minimize the following identification quality criterion.

$$J = \frac{1}{j}\int_{-j\infty}^{j\infty} tr\left(S'_{\varepsilon\varepsilon}R\right)ds , \quad (1)$$

where R is a symmetric positive definite weight matrix; $S'_{\varepsilon\varepsilon}$ is the transposed matrix of spectral densities of the vector of identification errors ε.

The identification error vector, denoted as ε, is defined as the difference between the vector of measured actual values of the movement of the WS x_1 and the estimate of the vector of the WS movement, x_{id}, obtained through the identification process:

$$\varepsilon = x_1 - x_{id} .$$

Similarly, the identification error vector ε, can be expressed as:

$$\varepsilon = x_1 - \Phi\zeta . \quad (2)$$

where Φ represents the block matrix comprising the transfer functions of the identification object, defined as $\Phi = [W_1, W_2]$; $\zeta = \begin{bmatrix} r_0 \\ \Delta \end{bmatrix}$ extended vector of input signals.

Then the task of identifying the dynamics model of the Stewart platform is simplified to minimizing either the error vector ε (2) or the identification quality indicator (1) by determining the two transfer function matrices W_1 and W_2.

The solution to the task was found as a result of three stages of work:

– development of the algorithm for structural identification of a multidimensional dynamic object with stochastic input signals;

– gathering and processing experimental data of vectors r_0 and x_1;

– assessing variations in the variance of the output coordinates vector of the working surface of the Stewart platform x_1 with random alterations in vectors r_0 and ψ_{ob}.

2 REVIEW OF THE LITERATURE

Given the modern requirements for the accuracy of motion control processes of a moving object along a specified or programmable trajectory, it is necessary to synthesize optimal structures and parameters for the object stabilization system (controller). This synthesis should take into account the real controlled and uncontrollable stochastic disturbing factors that act on the object in each specific mode of its operation. In today's demand for precise control processes of a moving object along a specified or programmable trajectory, there is a need to design the optimal structure and parameters of the object's stabilization system (controller). This involves considering real controllable and uncontrollable stochastic disturbing factors, which operate on the object in each specific operational mode. Furthermore, in the process of synthesizing the optimal controller structure, it is also necessary to evaluate and consider multidimensional dynamic models of the object itself, its basic components, as well as controlled and uncontrollable disturbing factors that influence the object in its actual motion [6].

As a rule, the mentioned dynamic models of moving objects, corresponding to real operational modes of motion, are either unknown or known very imprecisely [7, 8]. Such situations in the modern stage of technological development do not meet the requirements for the competitiveness of motion control systems for existing or newly created objects. Due to the lack of knowledge of the required methods and algorithms for processing and the target application of the results of processing stochastic information that can be obtained during testing, convenient and necessary 'real' models of the dynamics of objects, their parts, and disturbing factors are practically absent at the present time. For instance, considering the Stewart platform as an object controlled reveals several peculiarities, the main of which is that, for many technological tasks, the parameters of disturbing influences applied to the working part, individual axes, are not predetermined, and there is complexity in constructing an adequate analytical mathematical model [7, 9].

However, special full-scale (semi-full-scale) studies of prototypes of future mobile objects allow solving problems of structural identification [1] of the dynamics models of complex objects.

Today, there is a fairly wide range of methods for identifying dynamic models of control objects that operate under conditions of uncontrolled concentrated stationary random influences [1, 4, 5, 10], based on the 'input-output' data. The 'input-output' method [10] involves combining signals acting at the input and output of the system into a single signal vector. This vector is considered the output of an imaginary dynamic system, to the input of which a virtual test signal with known dynamics, such as "white noise", is applied. However, almost all of them are designed for use in conditions where there is no mutual influence between external disturbance and control signal. At the same time, the presence of feedback that cannot be unlocked during identification makes it impossible to accept the hypothesis even for different

DOI 10.15588/1607-3274-2024-1-22

sources of disturbances and control signals. In such cases, specialized, sophisticated identification technologies are required.

The method of identifying the dynamics of multidimensional control objects, as described in sources [11, 12], overcomes these drawbacks but limits the class of useful signals, disturbances, and interferences acting during the experiment. All the mentioned signals must belong to centered stationary random processes or to the additive mixture of a stationary random process and a deterministic time function.

The article in [13] presents a method for identifying multidimensional stochastic stabilization systems for moving objects with arbitrary dynamics in the frequency domain. Initial information about changes in the "input-output" signals is obtained from passive experiments during field trials, which is distorted by the imperfections of measuring instruments and the recording system. This method is employed with the requirement that the dynamic models of external influences on the system, which come into play during the identification experiment, need to be explicitly defined.

In such conditions, the range of identification methods is significantly narrowed. For instance, in the article [14], algorithms for identifying the dynamics of elements in a multidimensional stabilization system are presented. It asserts that under the conditions where the signals in the control loop fall within the category of centered stationary random processes and sensor noises stem from diverse sources introducing disturbances to the system, it becomes feasible to distinctly identify the matrices of fractional-rational functions associated with the disturbance generator. Moreover, it allows determining a system of ordinary differential equations of minimal order that characterizes the dynamics of the controlled object. Nevertheless, there is no empirical verification for the application of this identification algorithm through real-world experiments on either a dynamic object or its prototype.

3 MATERIALS AND METHODS

To attain the objective, as outlined in reference [1], the task of developing an identification algorithm for the dynamic model of the multi-dimensional moving object – the Stewart platform – was formulated and solved.

Using expression (2) and the Wiener-Khinchin theorem in vector form [15], it is possible to form the transposed matrix of spectral densities of the vector of random identification errors $S'_{\varepsilon\varepsilon}$:

$$S'_{\varepsilon\varepsilon} = S'_{x_1x_1} - S'_{\zeta x_1}\Phi_* - \Phi S'_{x_1\zeta} + \Phi S'_{\zeta\zeta}\Phi_*,$$

where $S'_{\zeta\zeta}$ is a transposed matrix of spectral densities of the extended vector of input signals:

$$S'_{\zeta\zeta} = \begin{bmatrix} S_{r_0r_0} & O_{6\times6} \\ O_{6\times6} & S_{\Delta\Delta} \end{bmatrix}, \quad (3)$$

where $S'_{x_1\zeta}$ is a transposed matrix of mutual spectral densities between vector random processes ζ and x_1:

$$S'_{\zeta x_1} = \begin{pmatrix} S'_{r_0x_1} & S'_{\Delta x_1} \end{pmatrix}. \quad (4)$$

To solve the task, it is necessary to express the matrix of mutual spectral densities $S_{\Delta x_1}$ in terms of the output data. For this purpose, taking into account matrices (3), (4), and the system structure, and performing some transformations, the relationship equation is obtained:

$$S_{x_1\Delta}S_{\Delta\Delta}^{-1}S_{\Delta x_1} = S_{x_1x_1} - S_{x_1r_0}S_{r_0r_0}^{-1}S_{r_0x_1}. \quad (5)$$

To find the function $S_{\Delta x_1}$, it is necessary to factorize the matrix (5), taking into account the specificity of the vector Δ as a vector of unit "white" noises [16]. Knowing the matrix $S_{\Delta x_1}$, we begin to solve the identification problem, which is equivalent to minimizing the functional (1) over the class of functions Φ physically realizable and having analytical variation $\delta\Phi$ only in the right half-plane.

The task of finding the function Φ that minimizes the functional is solved using the Wiener-Kolmogorov method. The first variation of the identification quality functional δJ is expressed as:

$$\delta J = \frac{1}{j}\int_{-j\infty}^{j\infty} tr\Big[R_{0_*}\Big(-R_0S'_{\zeta x_1}D_*^{-1} + R_0\Phi D\Big)D_*\delta\Phi_* + R_0\Big(-S'_{x_1\zeta}R_0D^{-1} + S'_{\zeta\zeta}\Phi_*R_{0_*}D_*\Big)\delta\Phi D\Big]ds. \quad (6)$$

The condition of identical equality to zero of the variation (6) under the assumption of only stable variations of the functions is as follows:

$$\Phi = R_0^{-1}[T_0 + T_+]D^{-1}, \quad (7)$$

where R_0^{-1} is the result of Wiener factorization [16, 17] of a positively definite polynomial matrix R, which can be expressed as the product of Hermitian conjugate polynomial matrices:

$$R = R_0R_{0^*}, \quad (8)$$

where the determinant of the matrix R_0 has only zeros with negative real parts; $T_0 + T_+$ are a matrix with poles in the left half-plane of the complex variable, which is the result of the separation [16] of the following product

$$T = T_0 + T_+ + T_- = S'_{\zeta x_1}D_*^{-1}, \quad (9)$$

where D is the result of the Wiener factorization [17] of the transposed matrix $S'_{\zeta\zeta}$:

$$DD_* = S'_{\zeta\zeta}, \quad (10)$$

$$D = \begin{bmatrix} d_u & O_{6\times6} \\ O_{6\times6} & E_{6\times6} \end{bmatrix}. \quad (11)$$

DOI 10.15588/1607-3274-2024-1-22

Using algorithm (7), optimal structures of the transfer function matrices of the identification object W_1 and the filter Ψ, which shapes the dynamic characteristics of the disturbance brought to the system output, are determined – W_2.

Determining the dynamic properties of matrix P by applying the single-side pole removal operation to the matrices W_1 and W_2 [16, 18], after which the matrix M is found:

$$M = \tilde{P} W_1; \tag{12}$$

Calculation of the matrix of spectral densities of disturbing influence:

$$S'_{\psi_{ob}\psi_{ob}} = \tilde{P} W_1 W_2 \tilde{P}_*. \tag{13}$$

To determine the value of the minimum identification error variance, it is necessary to substitute the matrix D from expression (11) and matrices (7) and R_0 from (8) into the functional (6).

The algorithm for identifying the dynamics model of a multidimensional moving object mentioned above is developed through addition operations, multiplication of polynomial and fractional-rational matrices, Wiener factorization, Wiener separation of fractional-rational matrices, and determination of variance integrals.

4 EXPERIMENTS

The authors in [19] developed a research prototype of a machine tool based on the Stewart platform (Fig. 3), applicable for the physical modeling of the movement of various technical objects in space. During the research, a software and hardware system was developed to collect experimental data for identifying the dynamics model of the Stewart platform.

The software and technical components of the experimental data collection system were developed using LabVIEW, utilizing FPGA, SoftMotion, and Real-Time modules. To ensure determinism, the time-critical task is offloaded from Windows and transferred to the real-time kernel or the target Real-Time (RT) system. Therefore, two systems are in use. The first system with the Windows OS is called the Host. Application development takes place on the Host system. The developed application is loaded into the processor of the second system, referred to as the Target RT system. The Target RT system executes the software code, manages input/output devices, and exchanges data with the Host system. The Host PC and the Target Real-Time platform are connected via an Ethernet network but operate independently. The hardware chosen for the target RT system includes a personal computer with a multifunctional reconfigurable I/O device based on the NI PCI-7833R with Virtex-II 3M Gate FPGA. On this personal computer, the Venturcom Phar Lap Embedded Tool Suite was installed to create the target Real-Time platform. It is a 32-bit real-time operating system based on x86 architecture and built upon the Win32 API by Microsoft. The PCI-7833R board enables the creation of comprehensive measurement or control systems, featuring up to 8 channels of analog input (AI), 8 channels of analog output (AO), and 96 digital channels (DIO) (Fig. 3).

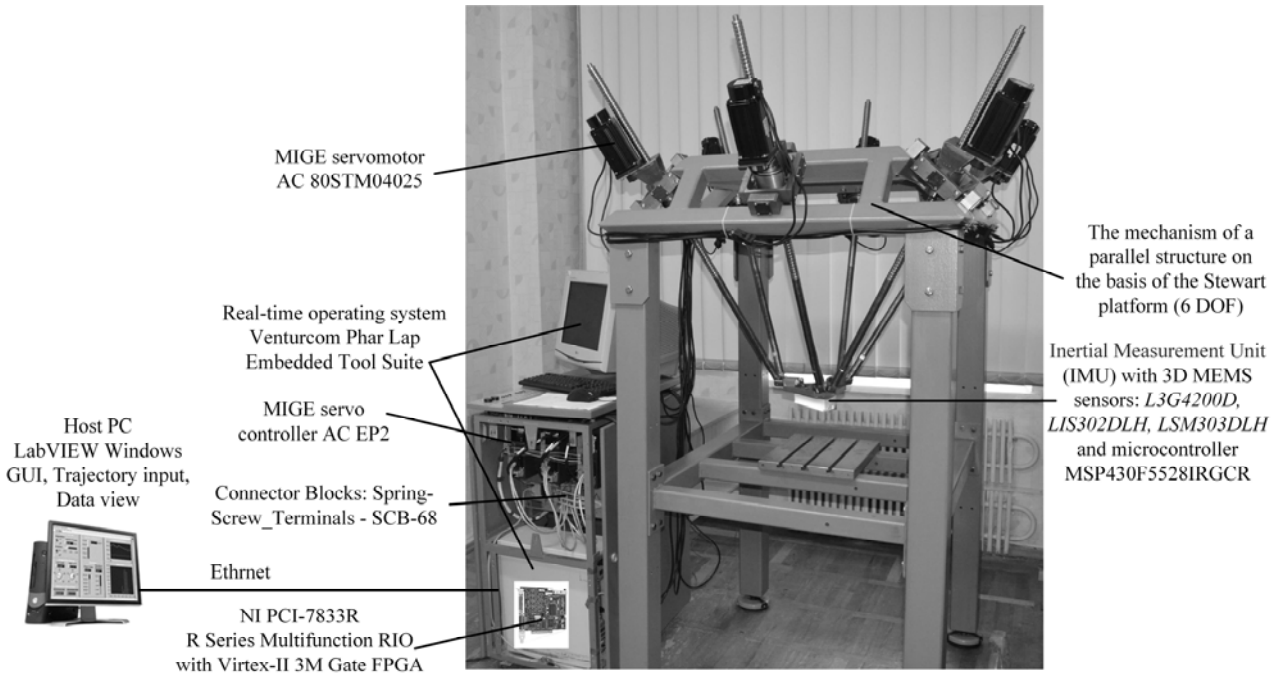


Figure 3 – Technical implementation of the motion control system for the working surface of the Stewart platform using National Instruments technology

DOI 10.15588/1607-3274-2024-1-22

To determine the current coordinates, velocity, and acceleration of the working surface of the Stewart platform, an orientation and navigation sensor (ONS) was developed based on an inertial measurement unit (IMU) with MEMS sensors. The unit includes the following set of primary information sources: a three-axis microelectromechanical gyroscope L3G4200D, a three-axis microelectromechanical accelerometer LIS302DLH, a three-axis magnetic field sensor (magnetic compass) LSM303DLH, and a computational block based on the MSP430F5528IRGCR microcontroller (Fig. 4). The information obtained from the primary information sources is processed using the Attitude and Heading Reference Systems (AHRS) position determination algorithm implemented in LabView.

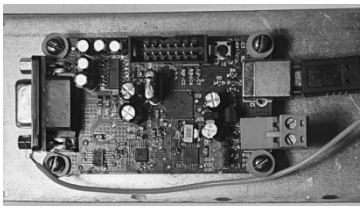


Figure 4 – Technical implementation of the inertial measurement unit based on 3D MEMS sensors

The electric drive and IMU were connected to the NI PCI-7833R board using Spring-Screw Terminals – SCB-68. A multifunctional AC servo drive from the EP2 series with an 80ST-M04025 servo motor from HANGZHOU MIGE ELECTRIC CO., LTD is used as the electric drive (Fig. 3).

To conduct an active experiment, with such identification, the formation of a vector of programmed control signal for moving the center of rotation of the Stewart platform WP is possible with a multidimensional filter based on the standard Dryden model as a universal tool for forming a stochastic external influence [20]. Based on this, a vector of the programmed signal r_0, was generated (Fig. 5) using the Dryden model implemented in Matlab/Simulink [21]. The range of this signal was constrained by the size of the working zone of the physical prototype of the Stewart platform [17].

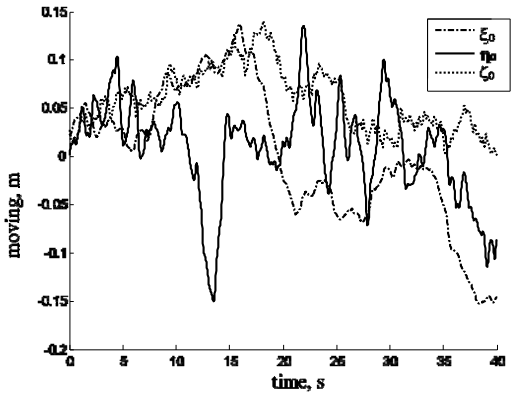


Figure 5 – Input program signal r_0 values

The experiment was constructed as follows: a realization of the vector of program signals r_0, simulating the change in the position of the center of rotation of the Stewart platform, was applied to the inputs of the control system of the physical prototype. The acquisition of the displacement vector of the Stewart platform's working surface, based on a full-scale experiment, was performed using a physical prototype of the Stewart platform with an IMU installed at the center of rotation. Before this, the transfer function matrix and spectral densities of measurement disturbances of the developed IMU were estimated using structural identification methods [22]. This contributed to improved accuracy through optimal stochastic signal filtration [23]. After the preliminary processing using the AHRS algorithm, an array of records of the displacement vector signal for the center of rotation of the working surface of the Stewart platform x_1, was obtained, the graph of which is depicted in Fig. 6.

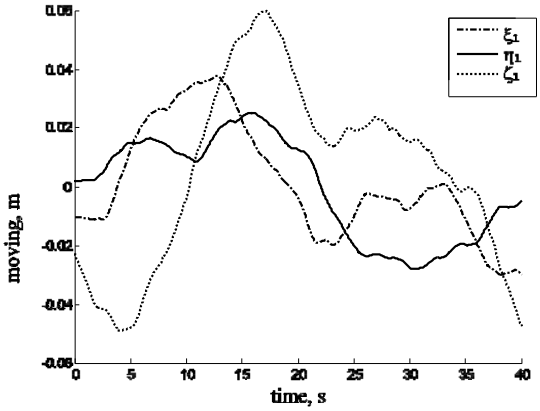


Figure 6 – Output signal x_1 values

As a result of the full-scale experiment, two arrays of points were obtained: one for the input signal r_0 and the other for the reaction of the Stewart platform to this signal x_1. These arrays were input into the Matlab program for calculating the dynamic model.

5 RESULTS

As a result of the conducted research, the task of identifying the dynamics model of a multidimensional moving object was formalized, illustrated by the example of a test specimen of a mechanism based on the Stewart platform (Fig. 3). Vectors of useful signals and interference are multidimensional stationary random processes with zero mathematical expectations and fractional-rational matrices of spectral and mutual spectral densities $S_{x_1x_1}$, $S_{r_0r_0}$, $S_{x_1r_0}$, $S_{r_0x_1}$ which have already been obtained as a result of experimental processing. Measurement disturbances are uncorrelated with each other and with the useful signal, while the disturbance ψ_{ob} is uncorrelated with the control signal r_0. As an example, here are some of the fractional-rational matrices of spectral and mutual spectral densities:

– spectral density of the input signal $S_{r_0r_0}$:

DOI 10.15588/1607-3274-2024-1-22

$$S'_{r_0r_0}=\begin{bmatrix} \dfrac{0.049}{\left|(s+0.6)(s+0.7)\right|^2} & \dfrac{0.03}{(-s+0.6)(s+0.7)(s+1.7)(s+2)} & \dfrac{-0.01(s+2)}{(s+0.6)(s+0.7)(s-0.68)(s-0.85)} \\ \dfrac{0.03}{(-s+2)(-s+1.7)(-s+0.7)(s+0.6)} & \dfrac{0.68}{\left|(s+2)(s+1.7)\right|^2} & \dfrac{0.005}{(s+1.7)(-s+0.68)(-s+0.85)} \\ \dfrac{-0.01(-s+2)}{(s-0.7)(s-0.6)(s+0.68)(s+0.85)} & \dfrac{0.005}{(-s+1.7)(s+0.68)(s+0.85)} & \dfrac{-0.005\left|s+2\right|^2}{\left|(s+0.68)(s+0.85)\right|^2} \end{bmatrix};$$

– mutual spectral density of the signal $S_{r_0x_1}$:

$$S'_{r_0x_1}=\begin{bmatrix} \dfrac{9^{-5}(s-1.37)(s+0.16)(s^2+7.9s+54.14)}{\left|(s+0.7)(s+0.6)\right|^2(s-1.7)z_1} \\ \dfrac{-4.8^5(s+2.22)(s-16.75)(s^2+1.36s+0.93)(s^2-2.02s+1.84)}{\left|(s+0.7)(s+0.6)\right|^2(s-1.7)(s-2)z_1} \\ \dfrac{-29.9^5(s+0.95)(s-3.22)(s-8.92)(s^2+0.04s+0.25)}{\left|(s+0.7)(s+0.6)\right|^2(s-2)(s^2+0.11s+0.04)(s^2+2s+1.09)} \\ \dfrac{3.46^{-5}(s+0.14)(s+0.02)(s-331.8)}{\left|s+1.7\right|^2(s+2)(s^2+0.11s+0.04)(s^2+0.63s+0.15)} \\ \dfrac{-1.84^{-5}(s+8.36)(s+142.1)}{\left|(s+1.7)(s+2)\right|^2(s^2+0.63s+0.15)} \\ \dfrac{-11.48^{-5}(s+1.1)(s+0.87)(s+3.08)(s-28.41)(s^2+0.61s+0.09)(s^2+1.37s+1.37)}{\left|(s+1.7)(s+2)\right|^2(s+0.7)z_2} \\ \dfrac{3.99^{-5}(s+0.16)(s^2+3.38s+2.96)(s^2-2.08s+31.39)}{(s+0.6)(s+1.7)(s-0.85)(s-0.68)z_1} \\ \dfrac{-2.12^{-5}(s+1.42)(s-6.81)(s^2+1.36s+0.89)(s^2+3.87s+4.77)}{\left|s+0.85\right|^2(s+0.6)(s+1.7)(s-0.68)z_1} \\ \dfrac{-13.24^{-5}(s+1.78)(s-6.87)(s^2+0.5s+0.14)(s^2+0.005s+0.2)}{(s+0.6)(s-0.85)(s-0.68)(s^2+0.11s+0.04)(s^2+2s+1.09)} \end{bmatrix},$$

where $z_1=(s^2+0.63s+0.15)(s^2+2s+1.09)$, $z_2=(s^2+0.11s+0.04)z_1$.

Matrices $S_{x_1x_1}$, $S_{r_0r_0}$, $S_{x_1r_0}$, $S_{r_0x_1}$ were subjected to reduction using the method of typical logarithmic frequency characteristics [8]. As a result, estimates of these densities were determined.

The transposed matrix of spectral densities of the extended vector of input signals $S'_{\zeta\zeta}$ is obtained according to the expression (3) of the identification algorithm, using Wiener factorization of polynomial matrices, substituting the spectral density of the input signal $S_{r_0r_0}$, and assuming $S_{\Delta\Delta}=1$.

Accordingly, with the expression (5), we obtained the matrices $S_{x_1\Delta}S_{\Delta\Delta}^{-1}S_{\Delta x_1}$, and by performing its factorization, we obtained:

$$S'_{\Delta\zeta x_1}=\begin{bmatrix} \dfrac{0.03}{z_3} & 0 & 0 \\ \dfrac{0.018}{z_3} & \dfrac{0.0245}{z_3} & 0 \\ \dfrac{0.01}{z_3} & \dfrac{0.02}{z_3} & \dfrac{0.054}{z_3} \end{bmatrix},$$

where $z_3=(s+0.45)(s+0.55)$.

Subsequently, by inserting $S'_{\Delta x_1}$ and $S'_{r_0x_1}$ into equation (4), we derive the matrix containing mutual spectral

DOI 10.15588/1607-3274-2024-1-22

densities $S'_{\zeta x_1}$. Using the expression (10) of the identification algorithm and executing the factorization of the matrix $S'_{\zeta\zeta}$, we have achieved:

$$D=\begin{bmatrix} d_u & O_{3\times3} \\ O_{3\times3} & E_{3\times3} \end{bmatrix},$$

where

$$d_u=\begin{bmatrix} \frac{0.22\left(s^2+3.49s+3.28\right)}{(s+0.7)z_4} & \frac{0.24819(s+0.78)}{(s+0.7)z_4} & \frac{0.008(s+1.14)}{(s+0.7)z_4} \\ \frac{-0.15(s+0.29)}{z_4} & \frac{0.81(s+0.61)}{z_4} & \frac{-0.027}{z_4} \\ \frac{-0.054(s-1.92)}{(s+0.68)(s+0.85)} & \frac{0.005}{(s+0.68)(s+0.85)} & \frac{-0.005|s+2|^2}{(s+0.68)(s+0.85)} \end{bmatrix},$$

where $z_4=(s+0.6)(s+1.7)(s+2)$.

Next, in accordance with expression (9) of the identification algorithm, by multiplying the fractional-rational matrices $S'_{\zeta x_1}$ and D and performing the Wiener separation, the resulting matrix is obtained:

$$T_0+T_+=\begin{bmatrix} \frac{0.012(s+1.15)\left(s^2+4.03s+4.12\right)}{(s+0.7)z_4} & \frac{0.013(s+2.27)\left(s^2+2.56s+1.7\right)}{(s+0.7)z_4} & \frac{0.015(s+1)\left(s^2+4.09s+4.4\right)}{(s+0.7)z_4} \\ \frac{0.0079(s+0.73)(s+2.23)}{z_4} & \frac{-0.0034(s+0.45)(s+3.52)}{z_4} & \frac{0.01(s+0.67)(s+2.9)}{z_4} \\ \frac{0.004(s-1.96)}{(s+0.68)(s+0.85)} & \frac{0.0075(s-1.34)}{(s+0.68)(s+0.85)} & \frac{0.004(s-2.47)}{(s+0.68)(s+0.85)} \end{bmatrix}\cdot$$
$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Entering the notation R_0=1 and following expression (7) of the identification algorithm, multiplying the fractional-rational matrices (T_0+T_+) and D^{-1}, the matrix $\Phi=[W_1\ W_2]$ is obtained, where W_1 is the optimal structure of the matrix of the transfer function of the identification object:

$$W_1=\frac{10^{-2}}{z_2}\begin{bmatrix} 1.3(s+5.4)(s+0.83)(s+0.15)\left(s^2+0.3s+0.067\right) & -1.6(s-2.12)(s+0.14)(s-0.029)\left(s^2+1.89s+1.04\right) \\ (s+2.1)\left(s^2+0.22s+0.03\right)\left(s^2+1.35s+0.83\right) & -0.4(s+9.3)\left(s^2+1.95s+0.99\right)\left(s^2+0.066s+0.09\right) \\ 4.9(s+0.96)\left(s^2+0.5s+0.14\right)\left(s^2-0.018s+0.2\right) & 0.6(s+0.19)\left(s^2+1.43s+0.54\right)\left(s^2+1.23s+4.09\right) \end{bmatrix}$$
$$;$$
$$\begin{bmatrix} -0.8(s-2.1)(s+0.76)(s+0.19)\left(s^2+0.25s+0.08\right) \\ 0.4(s+1.96)\left(s^2+0.22s+0.055\right)\left(s^2+1.06s+0.67\right) \\ 2.5(s+0.94)\left(s^2-0.2s+0.068\right)\left(s^2+0.59s+0.17\right) \end{bmatrix}$$

W_2 is the optimal structure of the matrix of transfer function of the filter, which forms the dynamic characteristics of the disturbance brought to the output of the system:

$$W_2=\begin{bmatrix} \frac{0.03}{z_3} & 0 & 0 \\ \frac{0.018}{z_3} & \frac{0.0245}{z_3} & 0 \\ \frac{0.01}{z_3} & \frac{0.02}{z_3} & \frac{0.054}{z_3} \end{bmatrix}.$$

DOI 10.15588/1607-3274-2024-1-22

Further, applying the operation of one-sided removal of the poles of matrices W1 and W2, a polynomial matrix P is obtained which is equal to:

$$P = \begin{bmatrix} z_2 & 0 & 0 \\ 0 & z_2 & 0 \\ 0 & 0 & z_2 \end{bmatrix}.$$

After that, there is matrix M according to (12):

$$M = \begin{bmatrix} 0.013(s+5.4)(s+0.83)(s+0.15)(s^2+0.3s+0.067) & -0.016(s-2.1)(s+0.14)(s-0.027)(s^2+1.89s+1) \\ 0.01(s+2.1)(s^2+0.22s+0.03)(s^2+1.35s+0.83) & -0.004(s+9.3)(s^2+1.95s+1)(s^2+0.067s+0.09) \\ 0.05(s+0.96)(s^2+0.5s+0.14)(s^2-0.018s+0.2) & 0.006(s+0.19)(s^2+1.4s+0.54)(s^2+1.23s+4.09) \end{bmatrix}$$

$$\begin{bmatrix} -0.008(s-2.1)(s+0.76)(s+0.19)(s^2+0.24s+0.085) \\ 0.004(s+1.95)(s^2+0.22s+0.055)(s^2+1.06s+0.67) \\ 0.025(s+0.94)(s^2-0.2s+0.067)(s^2+0.59s+0.17) \end{bmatrix}.$$

According to expression (13) of the identification algorithm, applying the operation of entering zeros to the left, the matrix of spectral densities of the disturbing influence was obtained:

$$S'_{\psi_{ob}\psi_{ob}} = 10^{-4} z_5 \begin{bmatrix} 9 & 5.3 & 3.2 \\ 5.3 & 9 & 6.7 \\ 3.2 & 6.7 & 34.4 \end{bmatrix},$$

where

$$z_5 = \left| \frac{(s^2+0.1s+0.04)(s^2+0.6s+0.15)(s^2+2s+1)}{z_3} \right|^2.$$

Thus, the research goal has been achieved. The results include the identification of the dynamic model of the Stewart platform, its transfer function and the transfer function of the shaping filter.

6 DISCUSSION

To validate the identification results, simulations were conducted to assess the accuracy of the dynamic model identification of the Stewart platform. It was done using the Simulink simulation tool in the Matlab 6.5 environment.

The principle of checking the accuracy of identification consists of comparing the vector of the change in the coordinates of the rotation center of WS of Stewart platform x_1 (Fig. 6) measured during the full-scale experiment with the sum of the vector formed when the software control signal vector r_0 (Fig. 5) passes through the transfer function of the identification object W_1 and vector of stationary random disturbances ψ_{ob}, which is formed when "white noise" signals pass through the transfer function of the forming disturbance filter W_2.

A number of relevant blocks are presented on the diagram of the simulation model (Fig. 7), which implements this principle. Block r_1 is designed to form a set of changes in the control signal r_0. At the output of block x_1, the vector x_1 of the values of the coordinates of the center of rotation WS of the Stewart platform recorded during the live experiment is formed. To generate the vector of stationary random disturbances ψ_{ob}, blocks of "white noise" generators with unit intensity were used. These generators are combined into the ψ_{ob} block. All data necessary for the operation of these blocks are presented in the workspace of the Matlab engineering calculation system in the format of *iddata* structures. Also, all components of these vectors were centred using the Constant block. Blocks *W1* and *W2* are designed to store matrices of transfer functions W_1 and W_2.

The diagram of the simulation model (Fig. 7) also presents demultiplexers for extracting vector components and multiplexers for combining data into vectors, as well as Scope oscilloscopes used to display simulation results for viewing and in the workspace.

As a result of the simulation to assess the accuracy of the identification of the dynamic model of the Stewart platform, in the *ksi_delta*, *ita_delta*, *sigma_delta* blocks comparison plots were obtained. These plots depict the vector formed at the output of the identification object x_{id} compared to the output vector x_1 in linear coordinates ξ, η, ζ (Fig. 8). Also, graphs in the delta block illustrating the change in the identification error were obtained (Fig. 9).

DOI 10.15588/1607-3274-2024-1-22

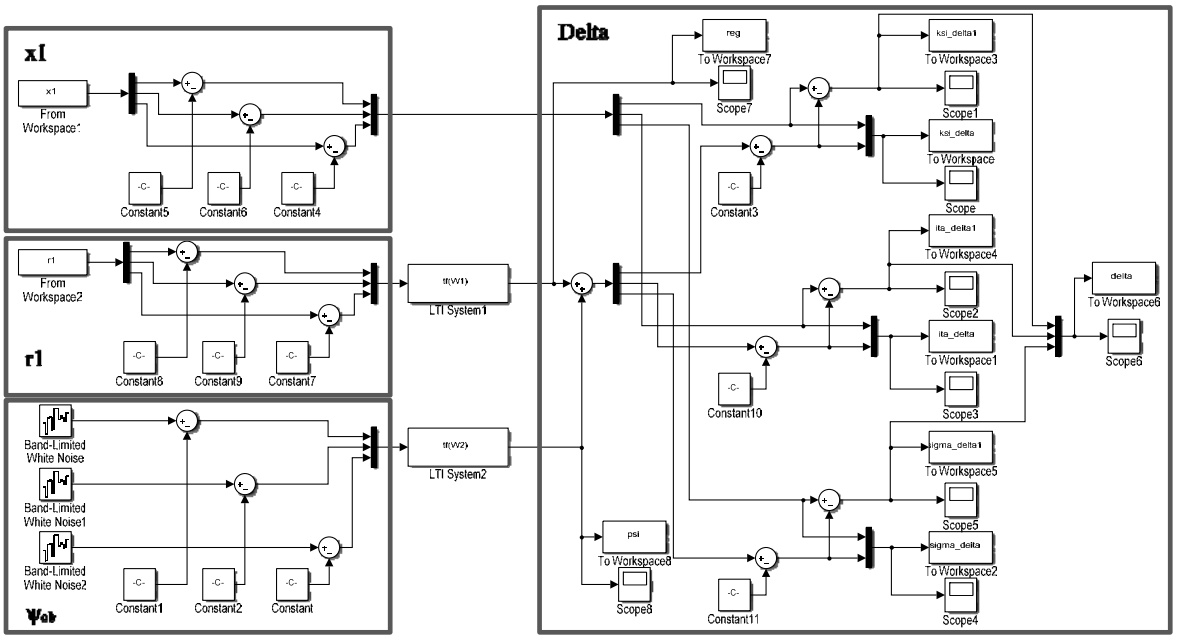
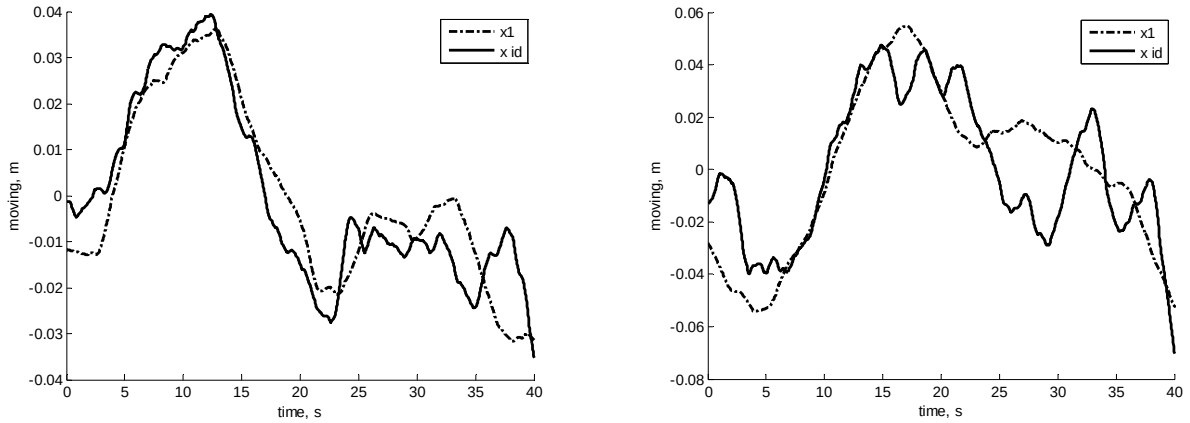
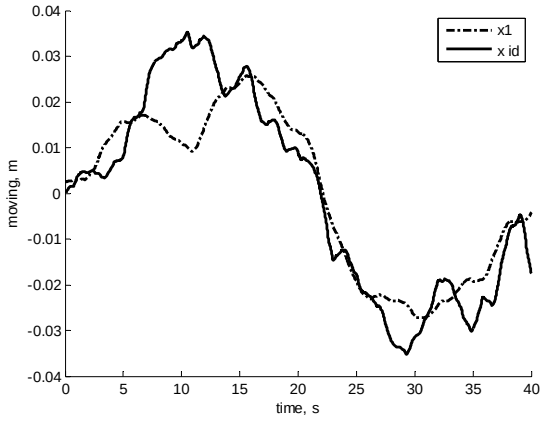


Figure 7 – Schematic of a simulation model for verifying the accuracy of mechanism identification based on Stewart platform



a b



c

Figure 8 – Graphs of change in the coordinates of the rotation center of WS according to the results of the full-scale experiment x_1 and the estimates obtained as a result of the identification x_{id}, according to linear coordinates: a – ξ, b – η, c – ζ

DOI 10.15588/1607-3274-2024-1-22

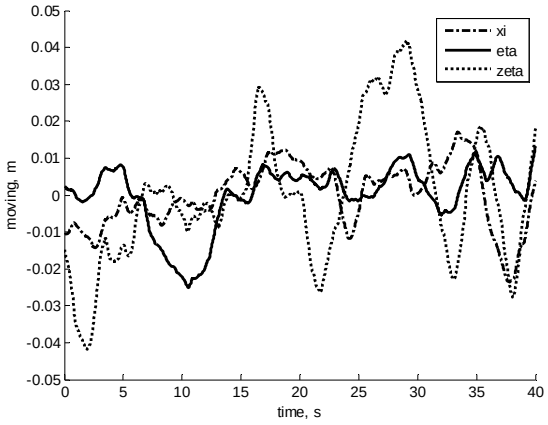


Figure 9 – Graphs of change in the identification error by linear coordinates ξ, η, ζ

On the basis of the analysis of the graphs of the identification error (Fig. 9), it is possible to estimate the root mean square deviation of the error relative to its mathematical expectation based on the unbiased estimate of its dispersion along the linear coordinates ξ, η, ζ, which are respectively equal to: $\sigma_{\xi}^{id} = 0.0086$, $\sigma_{\eta}^{id} = 0.0085$, $\sigma_{\zeta}^{id} = 0.00188$ and $D_{\xi}^{id} = 7.485 \cdot 10^{-5}$, $D_{\eta}^{id} = 7.3389 \cdot 10^{-5}$, $D_{\zeta}^{id} = 3.5458 \cdot 10^{-4}$.

As a result, the analysis of simulation results confirms the sufficient accuracy of identifying dynamic models of mechanisms based on the Stewart platform. Additionally, the main influence on the movement of the center of mass of the moving platform has changes in control actions. However, neglecting the influence of disturbances reduces the precision of platform positioning. Therefore, for the synthesis of the control system, methods should be applied that allow determining the structure and parameters of a multi-dimensional controller, taking into account such influences.

CONCLUSIONS

In the paper, an algorithm for identifying the dynamics model of a Stewart platform is substantiated based on known frequency algorithms of structural identification. It allows finding the order and parameters of the linearized system of ordinary differential equations for a multidimensional object and the matrix of spectral densities of disturbances acting on it under operating conditions close to the real mode of operation of the experimental object.

The scientific novelty of the obtained results lies in the fact that the above-justified algorithm for identifying the dynamics model of a multidimensional moving object demonstrates enhanced accuracy and reliability in computational performance. These improvements were achieved through the introduction of a novel approach to the processes of polynomial matrix factorization. At its core is the improvement of algorithms for multiplying polynomial matrices to minimize the loss of significant digits. This was achieved through the appropriate ordering and ranking of elementary operands, and the multiplication of fractional-rational matrices to mitigate the increase in order within the results.

The practical significance of the obtained results lies in the fact that the verification of the identification results confirms sufficient accuracy in identifying dynamic models of the mechanism based on the Stewart platform and the forming filter. Root-mean-square deviations of the error relative to its mathematical expectation were obtained based on an unbiased estimate of its variance along linear coordinates. Additionally, it was determined that the primary influence on the displacement of the center of mass of the moving platform is the variation in control inputs.

Prospects for further research involve the development of a method and algorithm for synthesizing the control system of the motion of the WS of the Stewart platform. These should allow for determining the structure and parameters of a multidimensional controller, taking into account that the primary influence on the displacement of the center of mass of the moving platform is the variation in control inputs.

ACKNOWLEDGEMENTS

The team of authors expresses their sincere gratitude to Professors Blokhin L.M. and Ladaniuk A.P. for numerous suggestions and recommendations for improving approaches to identifying models of dynamics of multidimensional systems, which are widely used both in aviation and technology production.

REFERENCES

1. Azarskov V. N., Blokhin L. N., Zhitetsky L.S.; eds.: L. N. Blokhin Methodology of designing optimal systems of stochastic stabilisation: Monograph. Kyiv, Book publishing house NAU, 2006, 440 p. ISBN 966-598-325-3.
2. Kondratenko Y. P., Kuntsevich V. M., Chikrii A. A., Gubarev V. F. Advanced Control Systems: Theory and Applications. River Publishers Series in Automation, Control and Robotics, 2021, 300 p.
3. Merlet J.-P. Parallel Robots. Springer, 2nd edition, 2006, 394 p.
4. Gubarev V. F. Modeling and identification of complex systems. Kyiv, Naukova Dumka, 2019, 248 p.
5. Levchuk I. L., Manko H. I., Tryshkin V. Ya., Korsun V. I. Theory and practice of identification of controled objects: Monograph. Dnipro, SHEI USUCT, 2019, 203 p. ISBN 978-617-7478-46-0
6. Lutska N. M., Ladaniuk A. P. Optimal and robust control systems for technological objects: monograph. Kyiv, LiraK Publishing House, 2015, 288 p.
7. Taghirad H. D. Parallel Robots. Mechanics and Control. Taylor & Francis Group, CRC Press., 2013, 533 p.
8. Blokhin L. M., Burichenko M. Y., Bilak N. V. et al.Statistical dynamics of control systems. Kyiv, NAU, 2014, 300 p.
9. Osadchy S., Zozulya V., Timoshenko A. Advances in Intelligent Robotics and Collaborative Automation – Robots, *Book. – River Publishers, 2015. – Chapter 2: The Dynamic Characteristics of the Manipulator with Parallel Kinematic*

DOI 10.15588/1607-3274-2024-1-22

Structure Based on Experimental Data, pp. 27–48 https://doi.org/10.1201/9781003337119.
10. Afrooz Ebadat. Experiment Design for Closed-loop System Identification with Applications in Model Predictive Control and Occupancy Estimation: thesis ... doctor of philosophy / Afrooz Ebadat. Royal Institute of Technology (KTH). Stockholm, Sweden, 2017, 231 p. ISSN 1653-5146. ISBN 978-91-7729-464-1.
11. Osadchyi S. I., Vikhrova L. H. Structural identification in the problem of linearisation of the model of dynamics of longitudinal gliding of the transom of a supercavitating object, *Bulletin of the National Technical University "Kharkiv Polytechnic Institute". Collection of scientific papers. Thematic issue: Computing and modelling: NTU "KhPI"*, 2011, № 36, pp. 128–134.
12. Jianhong W., Ramirez-Mendoza R. A. The practical analysis for closed-loop system identification. Cogent Engineering, 2020. DOI:10.1080/23311916.2020.1796895
13. Osadchyi S. I., Zozulia V. A., Kalich V. M. et al. The frequency method for optimal identification of close-loop system elements, *Radio Electronics, Computer Science, Control*, 2023, No. 4, pp. 193–205. DOI 10.15588/1607-3274-2023-4-18
14. Melnichenko M. M., Osadchy S. I., Zozulya V. A. Identification of the signals in position control circuits of a hexapod platform, *Conference: Methods and Systems of Navigation and Motion Control (MSNMC 2016): proceedings.* Kyiv: IEEE, KNAU, 2016, pp. 51–57 https://doi.org/10.18372/1990-5548.50.11387.
15. Korn G., Korn T. Handbook of Mathematics (for scientists and engineers). Moscow, Nauka, 1977, 831 p.
16. Aliev F. A., Bordyug V. A., Larin V. B. Factorisation of polynomial matrices with respect to imaginary axis and unit circle, *Avtomatika*, 1989, No. 4, pp. 51–58.
17. Davis M.C. Factoring the spectral matrix, *IEEE Trans. Automat. Cointr.*, 1963, AC-8, N 4, pp. 296–305.
18. Kucera Vladimir. The H2 control problem: a general transfer-function solution, *International Journal of Control*, 2007, Vol. 80, №5, pp. 800–815 DOI:10.1080/00207170701203590.
19. Development of a physical model of a lathe based on a parallel structure mechanism with a control system for the displacement drives of the working body: Report on Research Work (NDDKR) Kirovohrad National Technical University. Kirovohrad, 2011, 176 p. No. DR 0109U00210. registration No. 0211U005056
20. Tunik A. A., Rydlo K., Savchenko O. V. et al. Practical Experience of Intellectual UAV Attitude Stabilization System Computer-Aided Design, *Electronics and Control Systems*, 2015, No. 1(43), pp. 17–25. https://doi.org/10.18372/1990-5548.43.8842.
21. Patankar P., Kulkarni S. MATLAB and Simulink In-Depth. BPB Publications, 2022, 602 p.
22. Osadchy S. I., Zozulya V. A., Rudiuk G. I. The Dynamics of 3-dimensional micro-mechanic sensor of angle motions of a robot-hexapod, *Conference: Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2015): proceedings*. Warsaw, IEEE, 2015, Vol. 2, pp. 908–912 https:/doi.org/10.1109/IDAACS.2015.7341435.
23. Osadchy S. I., Zozulya V. A.Optimal Filtering of Hexapod Acceleration Data Obtained Under Action of Electromagnetic Interference, *Conference: Methods and Systems of Navigation and Motion Control (MSNMC 2014): proceedings*. Kyiv, IEEE, KNAU, 2014, pp. 21–23. https:/doi.org/10.1109/MSNMC.2014.6979719.

Received 05.01.2024.
Accepted 28.02.2024.

УДК 62.505:629.524

ІДЕНТИФІКАЦІЯ МОДЕЛІ ДИНАМІКИ ПЛАТФОРМИ СТЮАРТА

Зозуля В. А. – канд. техн. наук доцент кафедри цифрової економіки та системного аналізу Державного торговельно-економічного університету, Кропивницкий, Україна.

Осадчий С. І. – д-р техн. наук, професор кафедри конструкції повітряних суден, авіадвигунів та підтримання льотної придатності Льотної академії Національного авіаційного університету, Кропивницкий, Україна.

Неділько С. М. – д-р техн. наук, професор, виконуючий обов'язки директора Льотної академії Національного авіаційного університету, Кропивницкий, Україна.

АНОТАЦІЯ

Актуальність. За сучасних вимог до точності процесів керування рухом рухомого об'єкта на заданій або програмованій траєкторії руху необхідно синтезувати оптимальні структуру та параметри системи стабілізації (регулятора) об'єкта з урахуванням як реальних контрольованих, так і неконтрольованих стохастичних збурюючих факторів. Також у процесі синтезу оптимальної структури регулятора необхідно оцінювати і враховувати багатовимірні моделі динаміки як самого об'єкта, його базових частин, контрольованих і неконтрольованих збурюючих факторів, що впливають на об'єкт у його реальному русі.

Мета роботи. Метою дослідження, результати якого представлені у цій статті, є отримання та оцінка точності моделі динаміки платформи Стюарта за допомогою обгрунтованого алгоритму ідентифікації динаміки багатовимірного рухомого об'єкта.

Метод. У статті використано метод ідентифікації в частотній області багатовимірних стохастичних систем стабілізації рухомих об'єктів з довільною динамікою. Обгрунтований алгоритм ідентифікації моделі динаміки багатовимірного рухомого об'єкта, побудований за допомогою операцій додавання, множення поліноміальних та дробово – раціональних матриць, вінеровської факторизації, вінеровської сепарації дробово – раціональних матриць, знаходження дисперсійних інтегралів.

Результати. В результаті проведених досліджень формалізовано задачу ідентифікації моделі динаміки багатовимірного рухомого об'єкта на прикладі дослідного зразка верстата на основі платформи Стюарта. Результати включають ідентифікацію динамічної моделі платформи Стюарта, її передатної функції та передатної функції формуючого фільтра та верифікація результатів ідентифікації яка підтверджує достатню точність отриманих моделей.

DOI 10.15588/1607-3274-2024-1-22

Висновки. Обгрунтований алгоритм ідентифікації дозволяє знаходити порядок та параметри лінеаризованої системи звичайних диференціальних рівнянь багатовимірного об'єкта та матриці спектральних щільностей збурень, які діють на нього в умовах роботи наближених до реального режиму функціонування дослідного зразка об'єкту. Аналіз результатів ідентифікації динамічних моделей платформи Стюарта показує, що основний вплив на переміщення центру мас рухомої платформи має зміна керуючих впливів. Однак нехтування впливом збурень знижує точність позиціонування платформи. Тому для синтезу системи керування слід застосовувати методи, які дозволяють визначити структуру та параметри багатовимірного регулятора з урахуванням таких впливів.

КЛЮЧОВІ СЛОВА: ідентифікація, матриця передавальних функцій, спектральна щільність, функціонал якості, платформа Стюарта.

ЛІТЕРАТУРА

1. Azarskov V.N. Metodologija konstruirovanija optimal'nyh sistem stohasticheskoj stabilizacii: Monografija / V. N. Azarskov, L. N. Blohin, L. S. Zhiteckij, pod red. Blohina L. N. – K. : Knizhnoe izdatel'stvo NAU, 2006. – 440 p. ISBN 966-598-325-3.
2. Advanced Control Systems: Theory and Applications. / [Y. P. Kondratenko, V. M. Kuntsevich, A. A. Chikrii, V. F. Gubarev]. – River Publishers Series in Automation, Control and Robotics. 2021. – 300 p.
3. Merlet J.-P. Parallel Robots / J.-P. Merlet. – Springer, 2nd edition, 2006. – 394 p.
4. Gubarev V. F. Modeljuvannja ta identifikacija skladnih sistem. – K. : Naukova dumka, 2019. – 248 p.
5. Teorija i praktika identifikacii ob'єktiv upravlinnja : Monografija / [I. L. Levchuk, G. I. Manko, V. Ja. Trishkin, V. I. Korsun]. – Dnipro : DVNZ UDHTU, 2019. – 203 p. ISBN 978-617-7478-46-0
6. Luc'ka N.M. Optimal'ni ta robasni sistemi keruvannja tehnologichnimi ob'єktami: monografija / N. M. Luc'ka, A. P. Ladanjuk. – K. : Vidavnictvo LiraK, 2015. – 288 p.
7. Taghirad H. D. Parallel Robots. Mechanics and Control / H. D. Taghirad. – Taylor & Francis Group, CRC Press., 2013. – 533 p.
8. Statistichna dinamika sistem upravlinnja /[L.M. Blohin, M.Ju. Burichenko, N.V. Bilak ta in.]. – K.: NAU, 2014. – 300s.
9. Osadchy S. Advances in Intelligent Robotics and Collaborative Automation – Robots / S. Osadchy, V. Zozulya, A. Timoshenko // Book. – River Publishers, 2015. – Chapter 2: The Dynamic Characteristics of the Manipulator with Parallel Kinematic Structure Based on Experimental Data. – P. 27–48. https://doi.org/10.1201/9781003337119.
10. Afrooz Ebadat. Experiment Design for Closed-loop System Identification with Applications in Model Predictive Control and Occupancy Estimation: thesis ... doctor of philosophy / Afrooz Ebadat. – Royal Institute of Technology (KTH). Stockholm, Sweden, 2017. – 231 p. ISSN 1653-5146. ISBN 978-91-7729-464-1.
11. Osadchij S.I. Strukturnaja identifikacija v zadache linearizacii modeli dinamiki prodol'nogo glissirovanija tranca superkavitacionnogo ob'ekta / S. I. Osadchij, L. G. Vihrova // Visnik Nacional'nogo tehnichnogo universitetu „Hrkivskij politehnichnij institut". Zbirnik naukovih prac'. Tematichnij vipusk: Informatika i modeljuvannja. – Harkiv : NTU "HPI". – 2011. – №36. – P. 128–134.
12. Jianhong W. The practical analysis for closed-loop system identification / W. Jianhong, R. A. Ramirez-Mendoza. – Cogent Engineering. – 2020. DOI:10.1080/23311916.2020.1796895
13. The frequency method for optimal identification of close-loop system elements / [S. I. Osadchyi, V. A. Zozulia, V. M. Kalich et al.] // Radio Electronics, Computer Science, Control. – 2023. – No. 4. – P. 193–205. DOI 10.15588/1607-3274-2023-4-18
14. Melnichenko M. M. Identification of the signals in position control circuits of a hexapod platform / M. M. Melnichenko, S. I. Osadchy, V. A. Zozulya // Conference: Methods and Systems of Navigation and Motion Control (MSNMC 2016): proceedings. – Kyiv : IEEE, KNAU, 2016. – P. 51–57. https://doi.org/10.18372/1990-5548.50.11387.
15. Korn G. Handbook of Mathematics (for scientists and engineers) / G. Korn, T. Korn. – Moscow : Nauka, 1977. – 831 p.
16. Aliev F. A. Faktorizacija polinomial'nyh matric otnositel'no mnimoj osi i edinichnoj okruzhnosti / F. A. Aliev, V. A. Bordjug, V. B. Larin // Avtomatika. – 1989. – № 4. – P. 51–58.
17. Davis M. C. Factoring the spectral matrix/ M. C. Davis // IEEE Trans. Automat. Cointr. – 1963. – AC-8, N 4. – P. 296–305.
18. Kucera V. The H2 control problem: a general transfer-function solution / V. Kucera // International Journal of Control – 2007. – Vol. 80, №5. – P. 800–815. DOI:10.1080/00207170701203590.
19. Rozrobka fizichnoї modeli verstata na osnovi mehanizmu paralel'noї strukturi z sistemoju keruvannja privodami peremishhennja robochogo organa: Zvit po NDDKR Kirovograds'kij nacional'nij tehnichnij universitet. № DR 0109U00210, oblik. № 0211U005056. – Kirovograd, 2011. 176 p.
20. Practical Experience of Intellectual UAV Attitude Stabilization System Computer-Aided Design / [A. A. Tunik, K. Rydlo, O. V. Savchenko et al.] // Electronics and Control Systems. – 2015. – No. 1(43). – P. 17–25 https://doi.org/10.18372/1990-5548.43.8842.
21. Patankar P. MATLAB and Simulink In-Depth / P. Patankar, S. Kulkarni. – BPB Publications, 2022. – 602 p.
22. Osadchy S. I. The Dynamics of 3-dimensional micromechanic sensor of angle motions of a robot-hexapod / S. I. Osadchy, V. A. Zozulya, G. I. Rudiuk // Conference: Intelligent Data Acquisition and Advanced Computing Systems (IDAACS'2015) : proceedings. – Warsaw : IEEE, 2015. – Vol. 2. – P. 908–912 https:/doi.org/10.1109/IDAACS.2015.7341435.
23. Osadchy S. I. Optimal Filtering of Hexapod Acceleration Data Obtained Under Action of Electromagnetic Interference / S. I. Osadchy, V. A. Zozulya // Conference: Methods and Systems of Navigation and Motion Control (MSNMC 2014): proceedings. – Kyiv : IEEE, KNAU, 2014. – P. 21–23. https:/doi.org/10.1109/MSNMC.2014.6979719.

DOI 10.15588/1607-3274-2024-1-22

Наукове видання

Радіоелектроніка,

інформатика,

управління

№ 1/2024

Науковий журнал

Головний редактор – д-р техн. наук С. О. Субботін

Заст. головного редактора – д-р техн. наук Д. М. Піза

Комп’ютерне моделювання та верстання С. В. Зуб
Редактор англійських текстів С. О. Субботін

Оригінал-макет підготовлено у редакційно-видавничому відділі НУ «Запорізька політехніка»

Свідоцтво про державну реєстрацію
КВ № 24220-14060 ПР від 19.11.2019.

Підписано до друку 04.03.2024. Формат 60×84/8.
Папір офс. Різогр. друк. Ум. друк. арк. 29,76.
Тираж 300 прим. Зам. № 203.

69063, м. Запоріжжя, НУ «Запорізька політехніка», друкарня, вул. Жуковського, 64

Свідоцтво суб’єкта видавничої справи
ДК № 6952 від 22.10.2019.