UDC 004.896

METHOD FOR AGENT-ORIENTED TRAFFIC PREDICTION UNDER DATA AND RESOURCE CONSTRAINTS

Lovkin V. M. – PhD, Associate Professor, Associate Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Subbotin S. A. – Dr. Sc., Professor, Head of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Oliinyk A. O. – Dr. Sc., Professor, Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

ABSTRACT

Context. Problem of traffic prediction in a city is closely connected to the tasks of transportations in a city as well as air pollution detection in a city. Modern prediction models have redundant complexity when used for separate stations, require large number of measuring stations, long measurement period when predictions are made hourly. Therefore, there is a lack of method to overcome these constraints. The object of the study is a city traffic.

Objective. The objective of the study is to develop a method for traffic prediction, providing models for traffic quantification at measuring stations in the future under data and resource constraints.

Method. The method for agent-oriented traffic prediction under data and resource constraints was proposed in the paper. This method uses biLSTM models with input features, including traffic data obtained from agent, representing target station, and other agents, representing informative city stations. These agents are selected by ensembles of decision trees using Random Forest method. Input time period length is proposed to set using autocorrelation data.

Results. Experimental investigation was conducted on traffic data taken in Madrid from 59 measuring stations. Models created by the proposed method had higher prediction accuracy with lower values of MSE, MAE, RMSE and higher informativeness compared to base LSTM models.

Conclusions. Obtained models as study results have optimal number of input features compared to the known models, do not require complete system of city stations for all roads. It enables to apply these models under city traffic data and resource constraints. The proposed solutions provide high informativeness of obtained models with practically applicable accuracy level.

KEYWORDS: traffic, prediction, times series, LSTM, bidirectional LSTM.

ABBREVIATIONS

biLSTM is a bidirectional LSTM;
LSTM is a Long Short-Term Memory;
MAE is a Mean Absolute Error;
MSE is a Mean Squared Error;
RMSE is a Root Mean Square Error.

NOMENCLATURE

$auto_{\min}$ is a threshold value for autocorrelation;

B is a set of stations used for traffic quantification in a city or stations with data available for model creation;

B^S is a subset of stations selected from a set B based on the measurement of its impact on traffic at station A;

Cr is a limit number of stations possible to select to a subset B^S;

E is a set of factors, having impact on traffic values at station A during each of $(t + 1), \ldots, (t + h)$ hours;

E^K is a set of determined factors, having impact on traffic values at station A;

E^U is a set of undetermined factors, having impact on traffic values at station A;

f is a functional dependency to be determined by creation of traffic prediction model for station A;

H^F is a number of next hours from the current moment, which defines prediction horizon;

H^P is a number of previous hours from the current moment t, used for prediction;

t is a number of the current hour, i.e. moment in time when prediction is made;

T^{tr} is a length of training period;

tr_A^t is a traffic value at station A during the t-th hour;

v_e^t is a value of factor e from set tr_A^t during the t-th hour;

v_u^t is a value of factor u from subset of undetermined factors E^U during the t-th hour.

INTRODUCTION

Traffic is defined by number of vehicles moving through some location during some period of time [1]. Measuring stations are used to detect traffic and to quantify it at different locations.

Traffic is a structurally important concept, determining corresponding related processes in a modern city. Quality of life is closely connected to how traffic is managed in a city. Related flows should be clearly directed during city planning. Corresponding traffic rules as well as city construction principles should be set. Principles of construction of old cities did not take into account modern transport requirements, therefore transportations by vehicles in such cities are complicated. At the same time every modern city with correct planning restricts vehicle

DOI 10.15588/1607-3274-2023-4-10

movements according to certain principles, so it is not also totally free [2]. However, besides long-term traffic management, monitoring and informing about transport flows are important in short term as well [3].

Modern city traffic has influence on ecology. Cities set special restrictions on cars moving inside an area within city boundaries. These restrictions are gradually expanded. Strategic transition to electric cars is executed, but at the moment the majority of vehicles use fossil fuel, having negative influence on ecology.

Large part of emissions is caused by vehicles. Such air pollutants include nitrogen dioxide, benzol etc. In different cities percentage of influence of vehicles on air pollution varies. But the fact that vehicles are one of the main reasons of air pollution in modern cities is widely known. Besides, traffic is correlated with business activity. Therefore, absent or low traffic corresponds to low business activity and low air pollution at least in terms of concentration of pollutants closely connected to vehicle emissions. As a result, information about quantity of vehicles moving through some location in a city over a period of time helps to understand expected air pollution.

Prediction of traffic in a city may be valuable for every citizen while seeking for optimal path for transportation to the given location. It additionally emphasizes that city traffic has a significant influence on citizens in a city. Information about traffic in the short term allows citizens to plan their personal journeys in terms of its duration and in terms of impact on their health as well. The second factor is explained by causation between traffic and air pollution as well as air pollution and negative health impact even in the short term.

Taking it into account, it is obvious that traffic prediction problem is a significant practical problem. Besides, it is a significant scientific problem as well. A lot of factors have influence on traffic, so it is important to determine prediction models correctly.

The object of the study is a city traffic.

The subject of the study are traffic prediction models based on LSTM.

The objective of the study is to develop a method for traffic prediction, providing models for traffic quantification at measuring stations in the future under data and resource constraints.

1 PROBLEM STATEMENT

Problem of traffic prediction is a problem of traffic quantification at station *A* during each of H^F hours in the future using data on traffic at station *A* during previous H^P hours and data on other factors from set *E* defined by formula (1):

$$tr_A^{t+h} = f(tr_A^t, tr_A^{t-1}, ..., tr_A^{t-H^P-1}, v_e^t, v_e^{t-1}, ..., v_e^{t-H^P}\quad (1)$$
$$h = \overline{1, H^F}, e \in E.$$

Solving problem (1), H^F values of traffic tr_A^{t+h} at station *A* should be computed. Each value represents traffic over the (*t* + *h*)-th hour, determined by time period, starting at *t* + *h* hour 0 minutes and ending at *t* + *h* hour 59 minutes in the future. These computations should be made using functional dependence *f* and H^P values of traffic at station *A* in the past: $tr_A^t, tr_A^{t-1}, ..., tr_A^{t-H^P-1}$. As current moment defines finish of historical period, value of H^P comprises current hour with number *t*. Traffic at the moment *t* is quantified and known, so the last value (or called the first one otherwise) of input time period has number $H^P - 1$.

In this paper traffic is considered at hourly intervals. Therefore, every feature of observation should represent data collected over an hour. It includes data collected for all directions. If vehicles are moving in two directions through some location, then corresponding value of traffic is calculated as sum of number of vehicles moving towards a sensor called measuring station and in the opposite direction. If measuring station is located at crossroads, traffic should be computed as sum of all values determined for every direction. It enables to determine activeness of transport flows in general. If a vehicle moved twice or more times through some location in one or several directions over an hour, then it has to be calculated corresponding number of times. Therefore, this problem is connected with calculation of general number of vehicle drives through a location but not with identification of unique vehicles. Different approaches were proposed to organize work of measuring stations [4].

Set of factors *E* comprises subset of determined factors E^K, which impact was corroborated, and subset of undetermined factors E^U, representing impact of uncertainty. Solving problem (1) by creating traffic prediction model, subset E^U should be decreased at the expense of increasing subset E^K. But it is impossible to take into account all factors, which have influence on output, in the current state of development of world science. Therefore, influence of factors from subset E^U is active. It causes differences between output got by trained model and actual values of traffic for the same station over the same period of time.

2 REVIEW OF THE LITERATURE

Review of the literature was conducted based on the sequence of sources [5–14], taking into account results, represented in reviews of literature [5–6]. Results of the review demonstrate that big part of studies is directed at unification of models based on deep neural networks and models, applicable for taking into account spatial dependencies between measuring stations. This approach enables to create model, capable to process data from all stations simultaneously and to make predictions about future states of these stations. It is considered in the following studies in particular.

DOI 10.15588/1607-3274-2023-4-10

In the paper [7] it was proposed to use combined model based on graph convolutional neural network to extract characteristics of topological structure from traffic data, LSTM to extract characteristics of temporal structure and convolutional neural network to optimize general model. 5, 15 and 30 minutes were investigated as possible length of input time intervals. Model performance was evaluated by MAE, MSE and R^2. The longer input time interval was, the worse results were obtained. Experimental investigation was conducted on data collected in California (USA), using 39000 sensors.

In the paper [8] hybrid graph model was created. Unlike the previous model, it applies dynamic graph besides static one, representing topology of the traffic system as well as enabling to update it according to the current conditions. The proposed model combines graph neural network and convolutional neural network, applying attention mechanism. This solution enables to extract spatio-temporal characteristics. Experimental investigation is conducted on two datasets collected in California (USA). Obtained results were evaluated by MAE, RMSE and mean absolute percentage error as well.

Authors in the paper [9] proposed to use spatio-temporal graph convolutional network. Investigation was conducted for input time period of 60 previous minutes and prediction was executed for 15, 30 or 45 minutes. Results obtained on data collected in Beijing as well as in California demonstrate accuracy decrease for longer prediction periods. Accuracy for prediction period of 15 minutes is more than 1.5 times as much as accuracy for prediction period of 45 minutes.

Besides, there is a number of researches on traffic prediction where other traffic indicators are used. In particular speed prediction is researched in the paper [10]. It’s not strictly a problem considered, but similar structures are used for problem solving. The proposed model is based on convolutional neural network, LSTM, attention mechanism and 2 biLSTM. Similar solutions are proposed in the paper [11].

However practical applicability of these models is questionable in some cases. It demands high-performance computing, because resulting models have relatively complex structure, and data access which may be hard to provide. The last obstacle may be caused by absence of necessary historical data as it should have appropriate period which is possibly long. The more complex model structure, the longer period should be. Cities in developing countries may have fragmented traffic data, which do not cover all streets and crossroads or even a subsystem of roads. Therefore, it is hard to create a complete model. As a result, spatial dependencies are broken, because some locations in entire road system are absent. Besides, data may be inaccessible over a long period of time. For example, martial law can prohibit transmitting traffic data partially or completely, so open access services are restricted.

Moreover, prediction of traffic for every location in a city can be unnecessary. Then potential number of input features is far more than number of outputs. At the same time not all input features are necessary, so model complexity is excessive. It means that part of data is noisy and is separated by model structure. So, part of model structure is used not for prediction purposes but for separation of noisy data. On the other hand, when traffic prediction models give inputs for air pollution prediction, it should be taken into account that air pollution measuring stations are located at particular positions and its quantity is limited. Then approach proposed in papers [7]–[11] is inapplicable. In addition, it should be noted that results of conducted researches and other studies which were analysed demonstrate that traffic prediction for more than an hour is an individual problem and needs detailed investigation for improvement of results.

However, there is a number of studies which do not take into account spatial dependencies, being applicable for solving problem (1) under constraints on traffic data in a city. This approach is represented in the papers [12]–[14], and such technologies are applied for different problems, in particular [15–16].

In the paper [12] traffic is predicted based on LSTM models for 15, 30, 45, 60 minutes.

Traffic prediction for a period up to 60 minutes is investigated in the paper [13], using biLSTM models with 1 input feature and simulation data.

In the paper [14] biLSTM prediction is executed for 5 minutes, taking into account precipitation and visibility as additional features.

Therefore, it is necessary to develop results of researches [12–14]. Usage of additional features, principles of feature selection, investigation of input time period should be researched.

3 MATERIALS AND METHODS

Based on the problem statement (1), traffic prediction problem was analysed and general principles of problem solving were set. Time horizon for prediction was set equal to 6 hours in this study. However, the method for traffic prediction presented in this chapter uses principles of hourly data and making predictions for H^F hours ahead. So, it is applicable for other time periods.

Target measuring station *A* for which prediction is made has to be set as an input parameter for the method. All following stages are executed for this station. It should be presented by corresponding agent when prediction is made by for real-time data. Logic sequence of the proposed stages defines method for traffic prediction presented in Fig. 1.

At the initial stage of the method training dataset has to be created. Training dataset should contain data from all available stations from set *B*. All observations for each station have to represent each possible date and hour from training period. Every observation is defined by time series containing $H^P + H^F$ sequential hours.

Therefore, it is necessary to determine all hours in period of time defined as training to set training dataset. Training period is defined by time interval for which traffic data is available for all stations included in this data-

DOI 10.15588/1607-3274-2023-4-10

set. Values for some hours may be missed. For that reason, starting from the earliest available date and time (hour) every timestamp should be created with a step of 1 hour until the latest available date and time. Only when it is finished, each created timestamp should be connected with corresponding number of vehicles moved through location defined by each station over this period of time. After all available observations are set, missed values should be filled by linear interpolation.

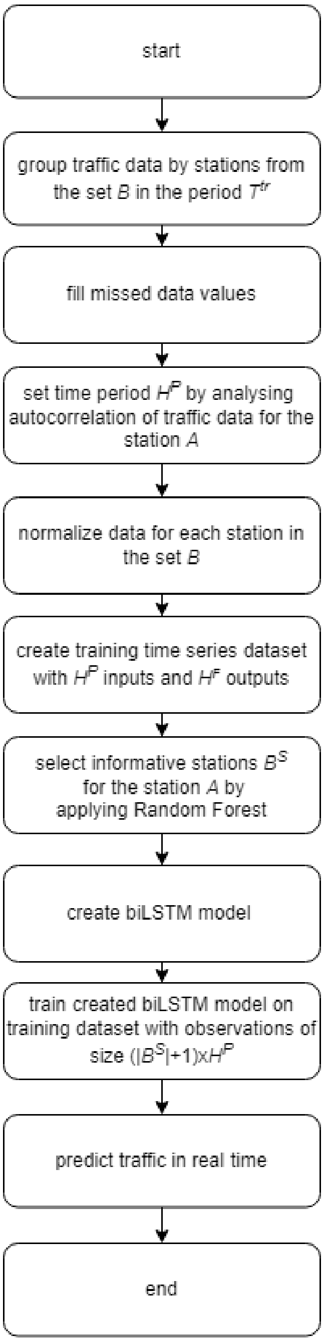


Figure 1 – Sequence of traffic prediction model training stages

Then value of H^P should be set. This parameter is influenced by value of H^F and it impacts on accuracy of predictions made by created models. So, this choice is important. The following procedure was defined to make this choice. Autocorrelation should be computed for values of traffic for station *A* with appropriate maximum lag. Maximum lag is a maximum possible value of H^P. Value of H^P should be set in such a way that autocorrelation with lag H^P is equal or higher than threshold value $auto_{\min}$. Threshold value $auto_{\min}$ should be set based on data analysis conducted for all available measuring stations. Otherwise it should be chosen to fulfill special requirements for predictions.

Then maximum and minimum values have to be got from training dataset for each station to normalize data. After normalization is finished, time series is created for every observation by appending elements for the next $H^P + H^F - 1$ hours. These elements are created by data shift.

Problem (1) is a problem of time series forecasting. So, values of traffic at a station over previous hours are used as values of input features. But every station is not totally separate. It is dependent on some other stations and it possibly has influence on some other stations. When entire set of stations is considered as input, it complicates model. However, it does not mean that existing dependence should not take into account at all. When optimal number of input features is investigated, dependence between values of traffic at station *A*, for which prediction is made, and at all other available stations from set *B* should be considered. As a result of this stage subset $B^S \subseteq B$ should be set. Subset B^S comprises stations selected from set *B* based on degree of influence of traffic at the selected stations on traffic at station *A*. This degree should be considered as significant.

Subset of stations B^S has to be created from set of stations *B* excluding station *A* at the following stage. It is proposed to use ensembles of decision trees using Random Forest method for this purpose.

Limit (maximum) number of stations *Cr* has to be set while method is applied. This number determines maximum number of additional features for final model and power of set B^S. Set of power $|B^S|$ has to be got as an output. It has to be not larger than the given limit number of stations *Cr*.

At the next stage biLSTM model [17–18] has to be created for station *A* with the next structure:

– input layer which corresponds to the structure described below;

– the first hidden bidirectional layer;

– dropout;

– the second hidden bidirectional layer;

– fully connected layer with H^F neurons for getting output values.

This model should detail dependence defined by the problem statement (1) in a way defined by formula (2):

DOI 10.15588/1607-3274-2023-4-10

$$tr_A^{t+h} = f(tr_A^t, tr_A^{t-1}, ..., tr_A^{t-H^P-1}, tr_b^t, tr_b^{t-1}, ..., tr_b^{t-H^P-1}, ..., v_u^t, v_u^{t-1}, ..., v_u^{t-H^P-1}), \quad (2)$$
$$h = \overline{1, H^F}, b \in B^S, B^S \subseteq B, e \in E.$$

So, input data for this model has to be determined by matrix of size $(|B^S|+1) \times H^P$. Rows of matrix contain traffic data from $|B^S|$ stations and from station A, for which model is created.

At the next stage model created at the previous stage has to be trained. Early stopping should be applied for training procedure to prevent overfitting.

When model is applied for real-time predictions, data is collected from agents, representing stations. These steps should be applied every hour. Every agent enables access to data, collected for the last hour. Agent should aggregate data, collected by sensor or any appropriate device located at station. When request is received, agent should send aggregated data to agent requested data. Connections are created between agents representing relevant stations. When data from all relevant stations is received, it should be united with data from target station and necessary historical data. Then prediction should be made using trained model. An observation presented as matrix of size $(|B^S|+1) \times H^P$ has to be used as input data. Every agent represents one station and has trained model for predictions. Corresponding sequence of real-time traffic prediction stages is presented in Fig. 2.

Results of the method include a trained model applicable for traffic prediction as well as number of hours H^P and subset of stations B^S.

Figure 2 – Sequence of real-time traffic prediction stages

4 EXPERIMENTS

Experimental investigation was conducted on dataset created from traffic data taken in Madrid (Spain) from 59 measuring stations and published in open access at Open data portal [19] of Madrid City Council [20]. The data was collected at 60 stations in Madrid. Data from 1.01.2019 until 30.09.2022 was used for investigation. As data analysis revealed absence of data at Calle Arenal station for the given period, it was excluded from the following investigation. Data from 59 stations was used for experimental investigation as a result. Each station provided quantification of traffic in two directions. Therefore, downloaded dataset was processed and aggregated, accumulating values in two directions for obtaining each station's hourly values.

Every file of initial dataset contains traffic data collected over a month. Every row in a file has a structure with the following columns:

– datemark, determining date when data observation was made;

– number of measuring station;

– additional mark (the first part of the day and forward traffic direction, the first part of the day and reverse traffic direction, the second part of the day and forward traffic direction, the second part of the day and reverse traffic direction);

– sequence of columns presenting traffic values for 12 hours separately.

Data was aggregated and saved to the united dataset. The dataset contains data for each station separately as well as a list of timestamps presenting date and time (hour and minutes) when corresponding observation was made. Every observation for every station contains normalized quantity of vehicles moved through location defined by measuring station in forward and reverse directions during time moment defined by corresponding timestamp. Quantity of vehicles moved through location was computed for hours until 12 as a sum of number of vehicles moved through location in forward and reverse directions per corresponding hour of the first part of a day (the first and the second marks were applied). The same procedure was used for hours of the second part of a day (the third and the fourth marks were applied).

80 % of data observations from the final dataset were used to create training dataset, other 20 % were used for test dataset. This separation was executed, taking into account batch size. The value of correspondent parameter BATCH_SIZE was set for training. The value of this parameter was set to 32. Therefore, number of observations in training dataset was coordinated with batch size to be divided without a remainder. If it wasn't true from the beginning, then number of observations in dataset was decreased by adding observations to test dataset. After this procedure was finished, number of observations in training dataset had to become divided by batch size without a remainder.

Creation of prediction models was based on the number of common principles: every model had an input layer, 2 hidden layers with dropout between layers, fully

DOI 10.15588/1607-3274-2023-4-10

connected layer for getting output value. Optimization was done using Adam optimizer. Loss function was determined by MSE. Maximum number of training iterations was set to 500. Early stopping criterion was determined for preventing model overfitting: if 40 iterations in a row didn't improve training results (loss function value didn't decrease), then training had to be stopped.

Final biLSTM models were created with 32 cells in two hidden bidirectional layers. Dropout was set to 0.1. Limit number *Cr* was set to 2 when Random Forest method was applied for selection of relevant stations. That is why maximum number of input features model could have was 3 with corresponding length of input time period.

Training dataset was divided into the part used for training directly (75 % of training dataset or 60 % of the united dataset) and the part used for validation (25 % of training dataset or 20 % of the united dataset).

The following models were investigated:

– LSTM models which use traffic data values for the previous 6 hours collected at target station as an input feature;

– biLSTM models which use traffic data values for the previous 6 hours collected at target station as an input feature;

– biLSTM models which use traffic data values for the previous 6 hours collected at target station and at selected stations considered relevant for target station as input features;

– biLSTM models which use traffic data values for the previous 24 hours collected at target station and at selected stations considered relevant for target station as input features;

– final biLSTM models created by the proposed method which use traffic data values for the optimal number of previous hours collected at target station and at selected stations considered relevant for target station as input features (marked by 3 features, adjusted 24/6 previous hours).

The following indicators were used for evaluation of model performance: MSE, RMSE, MAE, R^2.

Evaluation was executed using output values of each hour predicted by model separately. These values were compared with corresponding values from test dataset to compute values of model performance indicators (MSE, RMSE, MAE). Besides average values of model performance evaluation indicators for each station, average values of MSE were computed for each hour separately. Using computed values of model performance evaluation indicators for each station, minimum, maximum and average values were computed for each indicator.

5 RESULTS

All results were obtained only on test dataset. Observations from training dataset were used only for model training.

Distribution of MSE values is important for interpretation of results. Overall distribution of MSE values for biLSTM models (3 features, 6 hours), containing all 6 hours, is presented in Fig. 3. These models have 3 input features based on data traffic determined for the previous 6 hours. 3 features include target station and 2 additional stations selected as relevant for station of the first feature.

Each histogram defines number of models with certain characteristics created and trained for different stations (it should be represented as number of stations or models for these stations).

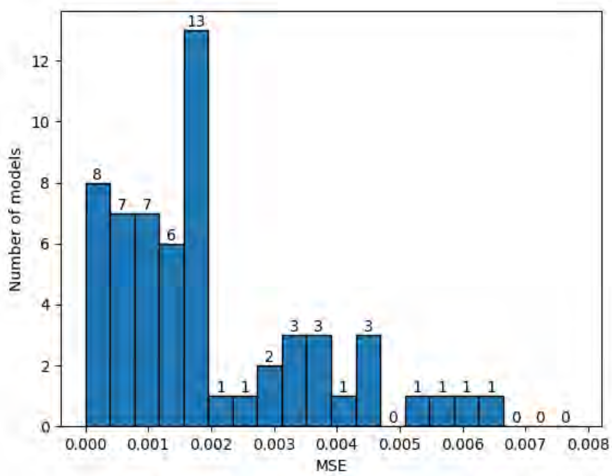


Figure 3 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations

The corresponding distribution of MSE values for models, created by the proposed method (with adjusted time period) for different stations, is demonstrated by histogram in Fig. 4.

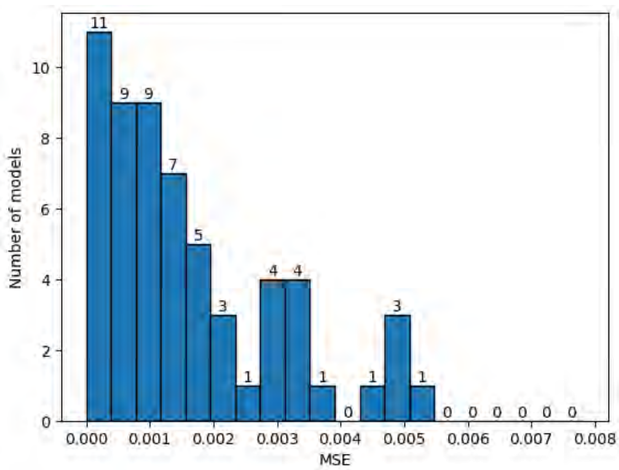


Figure 4 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations

The previous tables and figures contain results concerning all hours for which prediction was made. But there could be different trends in prediction for different step (number of corresponding hour). As prediction horizon was 6 hours, distribution of MSE values for biLSTM models (3 features, 6 hours) is presented for each hour in Fig. 5–10.

DOI 10.15588/1607-3274-2023-4-10

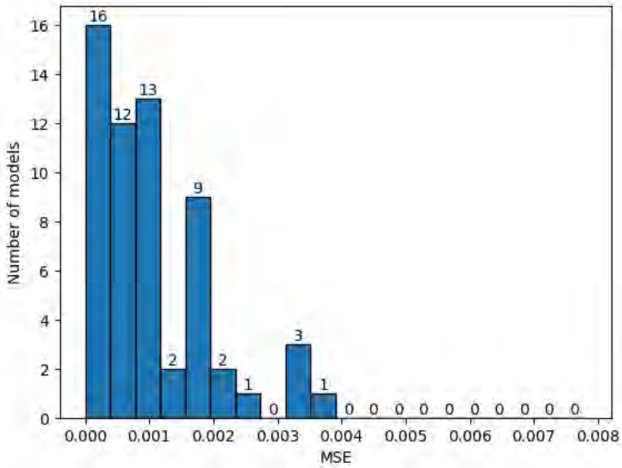


Figure 5 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in an hour

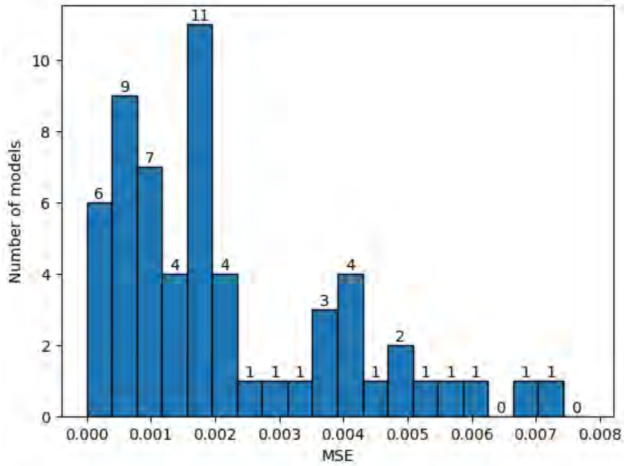


Figure 8 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in 4 hours

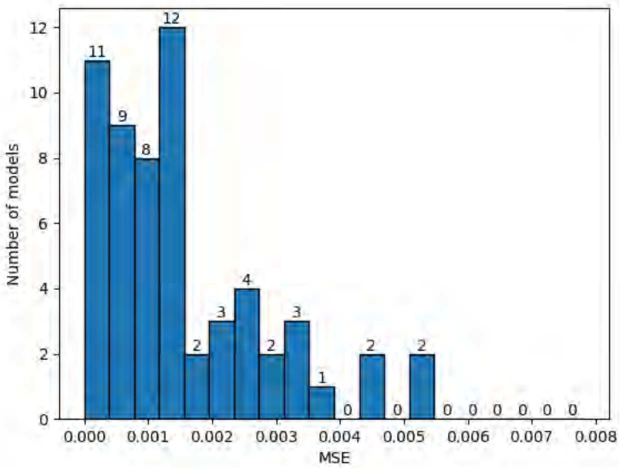


Figure 6 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in 2 hours

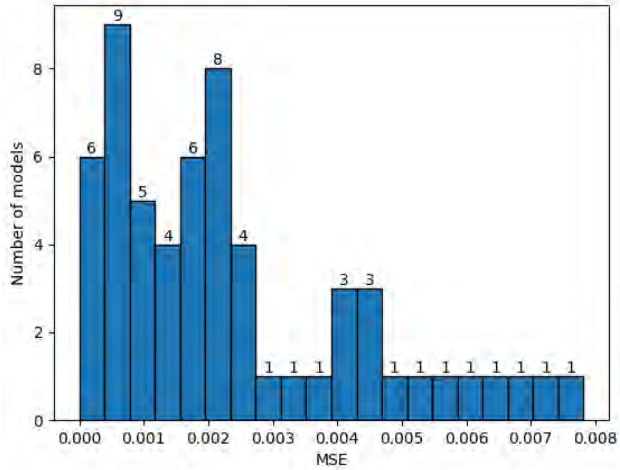


Figure 9 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in 5 hours

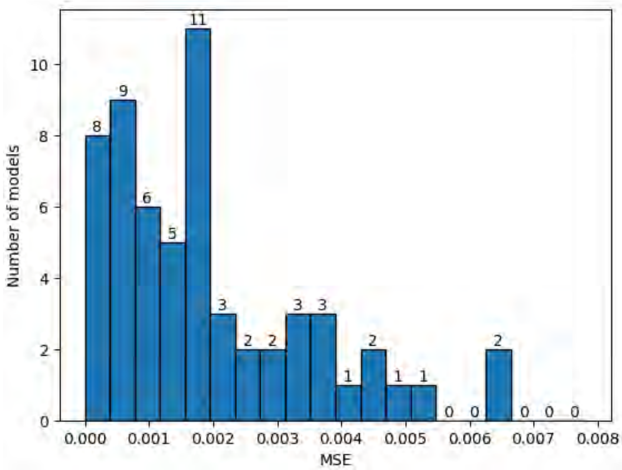


Figure 7 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in 3 hours

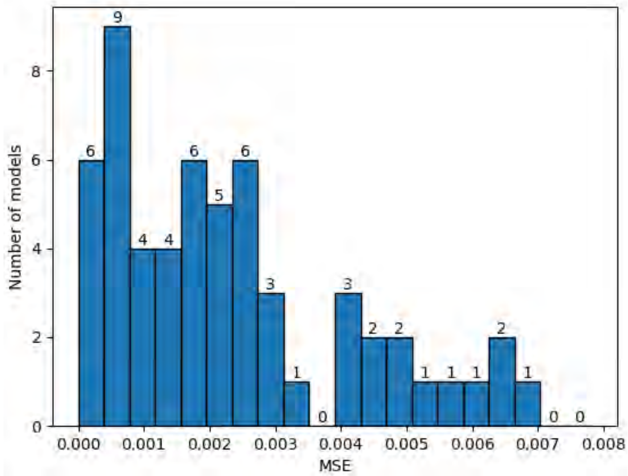


Figure 10 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, 6 hours) for all stations in 6 hours

DOI 10.15588/1607-3274-2023-4-10

Distribution of MSE values for biLSTM models (3 features, adjusted 24/6 previous hours) is presented in Fig. 11–16 hour by hour.

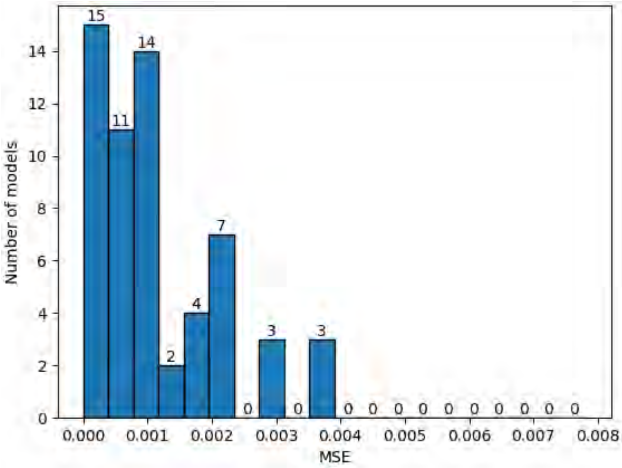


Figure 11 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in an hour

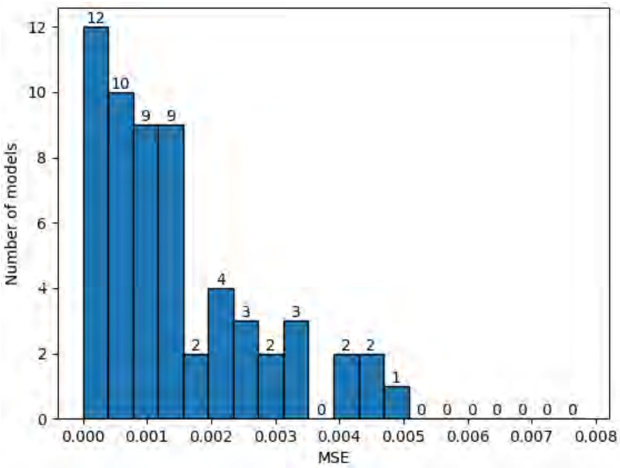


Figure 12 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in 2 hours

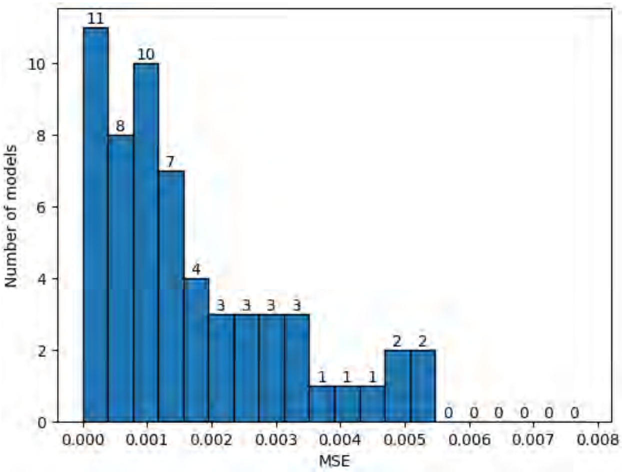


Figure 13 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in 3 hours

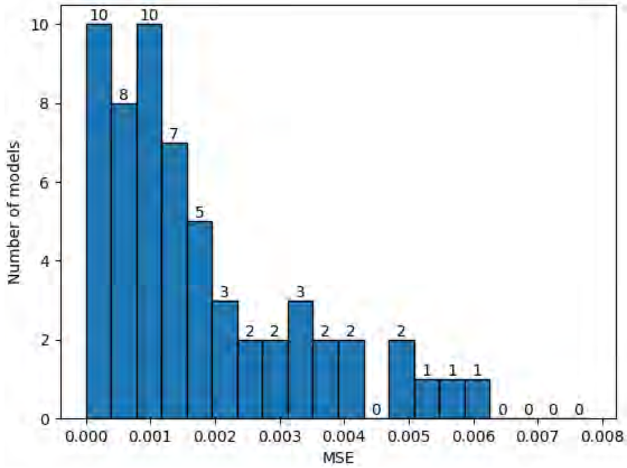


Figure 14 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in 4 hours

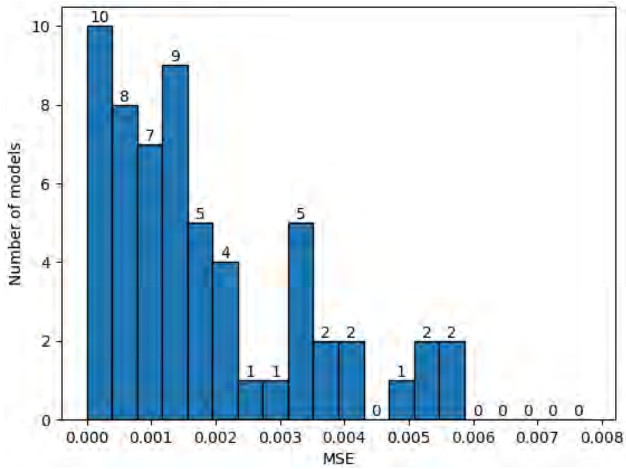


Figure 15 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in 5 hours

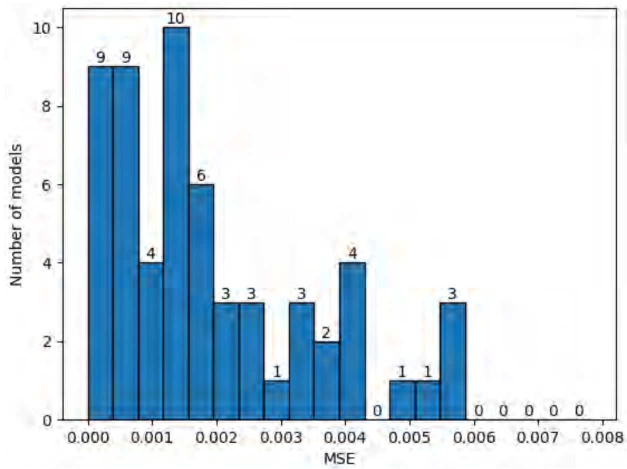


Figure 16 – Histogram visualizing distribution of MSE values obtained by biLSTM models (3 features, adjusted 24/6 hours) for all stations in 6 hours

Each histogram in Fig. 5–16 represents data for an hour.

DOI 10.15588/1607-3274-2023-4-10

Histograms were built with a step computed as 0.00821 (maximum observed value of MSE), divided by 21 to create 20 intervals of MSE values.

Results, representing values of MSE, MAE, RMSE (accuracy), R^2 (informativeness), for each of 59 stations were aggregated in Table 1 by computing average values of each indicator for all models being investigated.

Table 1 – Average results of traffic prediction by different models

Model characteristics	MSE	MAE	RMSE	R^2
LSTM (1 feature, 6 hours)	0.002908	0.032192	0.049139	0.818415
biLSTM (1 feature, 6 hours)	0.002125	0.027964	0.041744	0.863663
biLSTM (3 features, 6 hours)	0.001924	0.026616	0.039862	0.873622
biLSTM (3 features, 24 hours)	0.001636	0.024937	0.036757	0.884319
biLSTM (3 features, adjusted 24/6 hours)	0.001631	0.024942	0.036556	0.891

As absolute values of each model performance evaluation indicator may not completely reflect a relative difference in values for some stations even when normalized, results obtained by biLSTM models with different characteristics were compared to base LSTM model (1 feature, 6 hours) for each station. The results were computed as percentage of these comparisons. Average obtained values of the indicators are presented in Table 2 grouped by biLSTM models with different characteristics (number of input features and length of time period).

Table 2 – Average relative results of traffic prediction by different models compared to LSTM model

Model characteristics	Relative MSE, %	Relative MAE, %	Relative RMSE, %	Relative R^2, %
biLSTM (1 feature, 6 hours)	26.41	12.54	14.61	6.02
biLSTM (3 features, 6 hours)	32.69	16.72	18.43	7.19
biLSTM (3 features, 24 hours)	30.9	22.45	22.06	8.65
biLSTM (3 features, adjusted 24/6 hours)	43.26	22.24	25.26	9.41

6 DISCUSSION

Average results (Table 1) computed over all 59 stations demonstrated that biLSTM models with 3 features and input time period set to 24 hours for 56 stations or to 6 hours for 3 stations allowed to decrease MSE by 43.91 % compared to base LSTM models and MAE by 22.52 %. Informativeness of models increased by 8.87 % on average. So average results of final models emphasize significant improvement in terms of all model performance evaluation indicators.

biLSTM models with input time period of 24 hours allowed to decrease MSE by 14.97 %, MAE by 6.31 %, RMSE by 7.79 % and increase informativeness by 1.22 % compared to biLSTM models with shorter period. Usage of relevant stations as 2 additional input features allowed to decrease MSE by 9.46 %, MAE by 4.82 % and RMSE by 4.51 % compared to biLSTM model with 1 input feature on average. Informativeness of these models was higher by 1.15 %.

Comparison of results demonstrate that bidirectional architecture of LSTM models allowed to decrease MSE by 26.93 %, MAE by 13.13 %, RMSE by 15.05 %, and to increase R^2 by 5.53 %.

However average results do not reflect all broadness of information about efficiency of models. Despite data normalization adjusting input data to the same interval (traffic varies from station to station), values of MSE, RMSE, MAE and R^2 significantly differ from station to station. In some cases, average results can't distinguish improvements for some stations. These cases are characterized by low error if compared to the majority of stations. That's why even significant decrease in such an error may be unnoticed. So, it was important to analyse not only average absolute results of each indicator but its distribution in general, hour by hour and relative results as well.

Therefore, relative changes of indicator values between stations were considered, using base LSTM model as a basis for comparison. In this case difference in values of indicator is computed as percentage. Having percentages for each station and each indicator, it was possible to compute average (Table 2) results. In this case results represent relative values, so difference between error levels for various stations does not affect results. It makes possible to compare results between different architectures and structures of models.

In general results in Table 2 are close to the results in Table 1. But biLSTM (3 features, 24 hours) has relative MSE of 30.9 %, and percentage of change between its value in Table 1 and the same value of base LSTM equals to 43.74 %. It emphasizes significant change, taking into account that all other indicators for all other models have difference in values less than 1 % and only in one case it is slightly higher. Such a difference is explained by the worst case, when biLSTM (3 features, 24 hours) was worse than base LSTM model. It impacts on average relative value in Table 2 but it is impossible to detect such a situation in Table 1, as absolute value of MSE for this station is lower than for other stations. It is additionally emphasized by values of RMSE and R^2. At the same time even the worst value of accuracy obtained by resulting model was better than the one obtained by base LSTM model. So, usage of static input interval in some cases can reflect in results worse than base LSTM model. It makes application of more complex model unnecessary, so its usage for certain station in the case of this dataset or some stations in general is in doubt. But biLSTM models with adjusted input time period have more stable results. It means that resulting biLSTM models allow to obtain better results in terms of MSE, MAE, RMSE, R^2 not only in general but also for individual stations.

Histograms were used for comparative analysis of two biLSTM models with 3 features both but with different length of input time period: with static length of 6 hours and with dynamic adjustment. Distribution of results in terms of MSE according to Fig. 3–4 comes to our notice

DOI 10.15588/1607-3274-2023-4-10

that dynamic adjustment of input time period length for biLSTM models allows to improve obtained results by slightly moving distribution towards less values of error. It additionally underlines and corroborates statements made using average results.

When distribution of MSE is analysed hour by hour, it is noticeably that values of MSE are increasing hour by hour for both variants of biLSTM model. The curve representing changes in columns of histogram is becoming right-skewed as a result. So, the bigger the number of hour is, the lower prediction accuracy is.

But detailed comparison of distribution of MSE for biLSTM models with static input period of 6 hours and with dynamic input period hour by hour comes to our notice some differences. Comparison of Fig. 5 and Fig. 11 demonstrates that there are slightly better results in the first case (Fig. 5). It is noticeable when values in each column for both variants of models are compared from left to right. Number of models (stations) for the first variant is slightly bigger for left intervals and otherwise. But when histograms for the next hours are compared between both variants of models, the trend is different. Dynamic adjustment of input time period (actually between 6 and 24 hours) allowed to decrease MSE in reverse to the first statement. So, the first statement is completely true only for the first hour of prediction.

Therefore, in cases when predictions are made for some number of hours in the future but accuracy of predictions for the first hour is critical, input time period should be decreased and static length of 6 hours should be used. When it is not critical, length of input time period should be adjusted dynamically at the stage of the method described in chapter 3. At the same time, it is worth noting that value of $auto_{\min}$ enables to impact on this choice in an appropriate way.

CONCLUSIONS

The problem of traffic prediction was investigated in the paper. Modern traffic prediction methods are characterized by the complexity of models created. These models have large number of input features, require complete system of traffic measuring stations for spatial recognition based on road system in a city. However, not all cities have complete system of traffic measuring stations. Otherwise data may be inaccessible. Besides, resource restrictions should be taken into account for models created. Therefore, this powerful toolkit is not applicable to all practical cases where traffic is predicted.

Method for traffic prediction was proposed in the paper. This method is applicable under data and resource restrictions. It is based on biLSTM models with additional input features determined by other stations in a city considered relevant. These stations create a subset of the most informative stations selected by ensembles of decision trees using Random Forest method. When real-time predictions are made, data should be collected from different stations. The proposed prediction procedure is based on agent-oriented principles. It represents all stations by software agents. Every agent collects data for its target station, uses trained model for predictions and requests data of relevant stations from corresponding agents. These requests can be realized in parallel. Input time period length is proposed to set using autocorrelation data as a stage of the proposed method.

Experimental investigation was conducted on traffic data taken in Madrid from 59 measuring stations. The obtained results demonstrate significant improvement in traffic prediction using models, created by the proposed method, in terms of accuracy as well as informativeness. Improvement was achieved in terms of MSE (the value is 43 % lower compared to base LSTM model), MAE (22 % lower), RMSE (25 % lower) and R^2 (8–9 % higher). Besides, improvement in all indicators for biLSTM models compared to base LSTM was corroborated as well as for additional input features determined by traffic data of other measuring stations and increasing input time period. Experimental investigation demonstrated that in cases when predictions are made for some number of hours in the future but accuracy of predictions for the first hour is critical, the method should be adjusted to decrease input time period.

Models created by the method application have more optimal number of input features compared to the known models, therefore need less data and do not require complete system of city stations for all roads. It enables to apply these models under city traffic data and resource constraints. The proposed solutions provide high informativeness of obtained models with accuracy level which is significantly higher than accuracy of LSTM models in particular.

The scientific novelty of the obtained results is in the proposed method of traffic prediction.

The practical significance of the obtained results is in the created and trained models enabling to predict traffic at measuring stations for the next 6 hours based on the previous 24 hours or 6 hours in some cases.

Prospects for the further research are to integrate the proposed method for air pollution prediction.

ACKNOWLEDGEMENTS

The work was done as part of the state budget research project “Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects” (state registration number 0121U107499) of Software Tools Department of National University “Zaporizhzhia Polytechnic”.

REFERENCES

1. Bruntlett C., Bruntlett M. Curbing Traffic: The Human Case for Fewer Cars in Our Lives. Washington, Island Press, 2021, 240 p. DOI: 10.1080/07352166.2022.2155440.
2. Boltze M., Tuan V. A. Approaches to Achieve Sustainability in Traffic Management, *Procedia Engineering*, 2016, Volume 142, pp. 205–212. DOI: 10.1016/j.proeng.2016.02.033.
3. Moumen I., Abouchabaka J., Najat R. Adaptive traffic lights based on traffic flow prediction using machine learning models, *International Journal of Electrical and Computer*

DOI 10.15588/1607-3274-2023-4-10

Engineering (IJECE), 2023, Volume 13, No. 5, pp. 5813–5823. DOI: 10.11591/ijece.v13i5.pp5813–5823.
4. Tan D.M., Kieu L.-M. TRAMON: An automated traffic monitoring system for high density, mixed and lane-free traffic, *IATSS Research*, Volume 47, Issue 4, pp. 468–481. DOI: 10.1016/j.iatssr.2023.10.001.
5. Medina-Salgado B., Sanchez-DelaCruz E., Pozos-Parra P., Sierra J. Urban traffic flow prediction techniques: A review, *Sustainable Computing: Informatics and Systems*, 2022, Volume 35, 100739. DOI: 10.1016/j.suscom.2022.100739.
6. Vlahogianni E., Karlaftis M., Golias J. Short-term traffic forecasting: Where we are and where we're going, *Transportation Research Part C*, 2014, Volume 43, pp. 3–19. DOI: 10.1016/j.trc.2014.01.005.
7. Wu Z., Huang M., Zhao A., lan Z. Traffic prediction based on GCN-LSTM model, *Journal of Physics: Conference Series*, 2021, Volume 1972, 012107. DOI: 10.1088/1742-6596/1972/1/012107.
8. Chen R., Yao H. Hybrid Graph Models for Traffic Prediction, *Applied Sciences*, 2023, Volume 13 (15), 8673. DOI: 10.3390/app13158673.
9. Yu B., Yin H., Zhu Z. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting, *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pp. 3634–3640. DOI: 10.48550/arXiv.1709.04875.
10. Bai L., Yao L., Li C., Wang X., Wang C. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting, *34th Conference on Neural Information Processing Systems (NeurIPS2020)*. Vancouver, Canada, 2020, pp. 17804–17815. DOI: 10.48550/arXiv.2007.02842.
11. Hu X., Liu T., Hao X. Attention-based Conv-LSTM and Bi-LSTM networks for large-scale traffic speed prediction, *The Journal of Supercomputing*, 2022, Volume 78, pp. 12686–12709. DOI: 10.1007/s11227-022-04386-7.
12. Tian Y., Pan L. Predicting Short-term Traffic Flow by Long Short-Term Memory Recurrent Neural Network, *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. Chengdu, China, 2015, pp. 153–158. DOI: 10.1109/SmartCity.2015.63.
13. Abduljabbar R., Dia H., Tsai P.-W. Development and evaluation of bidirectional LSTM freeway traffic forecasting models using simulation data, *Scientific Reports*, 2021, Volume 11, 23899. DOI: 10.1038/s41598-021-03282-z.
14. Kouziokas G. Deep Bidirectional and Unidirectional LSTM Neural Networks in Traffic Flow Forecasting from Environmental Factors, *Advances in Intelligent Systems and Computing*, 2020, Volume 1278, pp. 171–180. DOI: 10.1007/978-3-030-61075-3_17.
15. Lovkin V., Oliinyk A., Lukashenko Y. Air Pollution Prediction as a Source for Decision Making Framework in Medical Diagnosis [Electronic resource], *IntelITSIS'2021: 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security*, 2021, Khmelnytskyi, Ukraine, 2021, pp. 295–302.
16. Lovkin V., Oliinyk A., Fedoronchak T., Lukashenko Y. Information Model of Outdoor Air Pollution Prediction for Medical Diagnosis System, *4th IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021*. Lviv, LPNU, 2021, pp. 141–144. DOI: 10.1109/AICT52120.2021.9628981.
17. Aggarwal C. Neural Networks and Deep Learning: A Textbook, Springer, 2018, 520 p. DOI: 10.1007/978-3-319-94463-0.
18. Goodfellow I., Bengio Y., Courville A. Deep Learning, The MIT Press, 2016, 800 p. DOI: doi.org/10.1007/s10710-017-9314-z.
19. En portada – Portal de datos abiertos del Ayuntamiento de Madrid [Electronic resource]. 2023. Access mode: https://datos.madrid.es/portal/site/egob.
20. Aforos de trafico en la ciudad de Madrid permanentes – Portal de datos abiertos del Ayuntamiento de Madrid [Electronic resource]. 2023. Access mode: https://datos.madrid.es/sites/v/index.jsp?vgnextoid=fabbf3e1de124610VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD.

Received 26.09.2023.
Accepted 30.10.2023.

УДК 004.896

МЕТОД АГЕНТНО-ОРІЄНТОВАНОГО ПРОГНОЗУВАННЯ АВТОМОБІЛЬНОГО ТРАФІКУ В УМОВАХ ОБМЕЖЕНОСТІ ДАНИХ ТА РЕСУРСІВ

Льовкін В. М. – канд. техн. наук, доцент, доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Субботін С. О. – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Олійник А. О. – д-р техн. наук, професор, професор кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

АНОТАЦІЯ

Актуальність. Проблема прогнозування автомобільного трафіку в місті пов'язана з розв'язанням одночасно і завдань переміщення в місті, і визначення забрудненості повітря в місті. Сучасні моделі прогнозування мають надмірну складність при використанні для окремих станцій, потребують великої кількості станцій спостереження, тривалого періоду спостереження зокрема у випадку погодинного прогнозування. Тому існує потреба в створенні відповідного метода, який дозволить подолати ці обмеження. Об'єктом роботи є автомобільний трафік у місті.

Мета роботи – розробити метод прогнозування автомобільного трафіку, який дозволить створити моделі для визначення трафіку за станціями у майбутньому в умовах обмеженості даних та ресурсів.

Метод. У статті запропоновано метод агентно-орієнтованого прогнозування автомобільного трафіку в умовах обмеженості наявних даних і обчислювальних ресурсів. Даний метод ґрунтується на використанні двонаправлених LSTM моделей з вхідними ознаками, якими є дані, отримані від агента, що відповідає станції прогнозування, та від агентів, що представля-

DOI 10.15588/1607-3274-2023-4-10

ють інші інформативні станції у місті, які обираються на основі використання ансамблів дерев рішень за допомогою методу Random Forest. Довжина вхідного часового інтервалу в методі обирається на основі даних автокореляції.

Результати. Експериментальне дослідження проводилося на основі даних про трафік у місті Мадрид, використовуючи дані, зібрані за 59 станціями спостереження. У результаті застосування створених на основі запропонованого методу моделей було отримано підвищену точність прогнозування, яку було підтверджено зменшенням значень MSE, MAE, RMSE, та підвищену інформативність порівняно з базовими LSTM-моделями.

Висновки. Отримані в результаті проведеного дослідження моделі відрізняються оптимальною кількістю вхідних ознак порівняно з відомими, не потребують використання цілісної системи станцій у місті на всіх автомобільних дорогах. Це дозволяє використовувати дані моделі в умовах обмеження ресурсів та обмеженої доступності даних про трафік у місті. При цьому забезпечується достатньо висока інформативність створених моделей з придатним для застосування на практиці рівнем точності прогнозування.

КЛЮЧОВІ СЛОВА: трафік, прогнозування, часовий ряд, LSTM, двонаправлена LSTM.

ЛІТЕРАТУРА

1. Bruntlett C. Curbing Traffic : The Human Case for Fewer Cars in Our Lives / C. Bruntlett, M. Bruntlett. – Washington : Island Press, 2021. – 240 p. DOI : 10.1080/07352166.2022.2155440.
2. Boltze M. Approaches to Achieve Sustainability in Traffic Management / Manfred Boltze, Vu Anh Tuan // Procedia Engineering. – 2016. – Volume 142. – P. 205–212. DOI : 10.1016/j.proeng.2016.02.033.
3. Moumen I. Adaptive traffic lights based on traffic flow prediction using machine learning models / I. Moumen, J. Abouchabaka, R. Najat // International Journal of Electrical and Computer Engineering (IJECE). – 2023. – Volume 13, No. 5. – P. 5813–5823. DOI : 10.11591/ijece.v13i5.pp5813-5823.
4. Tan, D.M. TRAMON : An automated traffic monitoring system for high density, mixed and lane-free traffic / Dang Minh Tan, Le-Minh Kieu // IATSS Research. – Volume 47, Issue 4. – P. 468–481. DOI : 10.1016/j.iatssr.2023.10.001.
5. Urban traffic flow prediction techniques [Electronic resource] : A review / Boris Medina-Salgado, Eddy Sanchez-DelaCruz, Pilar Pozos-Parra, Javier E. Sierra // Sustainable Computing : Informatics and Systems. – 2022. – Volume 35. – Article 100739. DOI : 10.1016/j.suscom.2022.100739.
6. Vlahogianni E. Short-term traffic forecasting : Where we are and where we're going / E. Vlahogianni, M. Karlaftis, J. Golias // Transportation Research Part C . – 2014. – Volume 43. – Pp. 3–19. – DOI : 10.1016/j.trc.2014.01.005.
7. Traffic prediction based on GCN-LSTM model [Electronic resource] / Zhizhu Wu, Mingxia Huang, Aiping Zhao and Zhixun lan // Journal of Physics : Conference Series. – 2021. – Volume 1972. – Article 012107. DOI : 10.1088/1742-6596/1972/1/012107.
8. Chen R. Hybrid Graph Models for Traffic Prediction [Electronic resource] / R. Chen, H. Yao // Applied Sciences. – 2023. – № 13 (15). – Article 8673. DOI : 10.3390/app13158673.
9. Yu B. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting / Bing Yu, Haoteng Yin, Zhanxing Zhu // Proceedings of the 27th International Joint Conference on Artificial Intelligence. – 2018. – P. 3634–3640. DOI : 10.48550/arXiv.1709.04875.
10. Adaptive Graph Convolutional Recurrent Network for Traffic Forecasting / Lei Bai, Lina Yao, Can Li, Xianzhi Wang, Can Wang // 34th Conference on Neural Information Processing Systems (NeurIPS2020), Vancouver, Canada. – Vancouver, 2020. – P. 17804–17815. – DOI : 10.48550/arXiv.2007.02842.
11. Attention-based Conv-LSTM and Bi-LSTM networks for large-scale traffic speed prediction / X. Hu, T. Liu, X. Hao et al. // The Journal of Supercomputing. – 2022. – Volume 78. – P. 12686–12709. DOI : 10.1007/s11227-022-04386-7.
12. Tian Y. Predicting Short-term Traffic Flow by Long Short-Term Memory Recurrent Neural Network / Yongxue Tian, Li Pan // 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). – Chengdu, China, 2015. – P. 153–158. DOI : 10.1109/SmartCity.2015.63.
13. Abduljabbar R. Development and evaluation of bidirectional LSTM freeway traffic forecasting models using simulation data [Electronic resource] / Rusul L. Abduljabbar, Hussein Dia, Pei-Wei Tsai // Scientific Reports. – 2021. – Volume 11. – Article 23899. DOI : 10.1038/s41598-021-03282-z.
14. Kouziokas G. Deep Bidirectional and Unidirectional LSTM Neural Networks in Traffic Flow Forecasting from Environmental Factors / G. Kouziokas // Advances in Intelligent Systems and Computing. – 2020. – Volume 1278. – P. 171–180. – DOI : 10.1007/978-3-030-61075-3_17.
15. Lovkin V. Air Pollution Prediction as a Source for Decision Making Framework in Medical Diagnosis [Electronic resource] / V. Lovkin, A. Oliinyk, Y. Lukashenko // IntelITSIS'2021 : 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security, March 24–26, 2021, Khmelnytskyi, Ukraine. – Khmelnytskyi : KhNU, 2021. – P. 295–302.
16. Information Model of Outdoor Air Pollution Prediction for Medical Diagnosis System / V. Lovkin, A. Oliinyk, T. Fedoronchak, Y. Lukashenko // 4th IEEE International Conference on Advanced Information and Communication Technologies (AICT) – 2021, September 21–25, 2021, Lviv, Ukraine. – Lviv : LPNU, 2021. – P. 141–144. DOI : 10.1109/AICT52120.2021.9628981.
17. Aggarwal C. Neural Networks and Deep Learning : A Textbook / C. Aggarwal. – Springer, 2018. – 520 p. DOI : 10.1007/978-3-319-94463-0.
18. Goodfellow, I. Deep Learning / Ian Goodfellow, Yoshua Bengio, Aaron Courville. – The MIT Press, 2016. – 800 p. – DOI : doi.org/10.1007/s10710-017-9314-z.
19. En portada – Portal de datos abiertos del Ayuntamiento de Madrid [Electronic resource]. – Access mode : https://datos.madrid.es/portal/site/egob.
20. Aforos de trafico en la ciudad de Madrid permanentes – Portal de datos abiertos del Ayuntamiento de Madrid [Electronic resource]. – Access mode : https://datos.madrid.es/sites/v/index.jsp?vgnextoid=fabbf3e1de124610VgnVCM2000001f4a900aRCRD&vgnextchannel=374512b9ace9f310VgnVCM100000171f5a0aRCRD.

DOI 10.15588/1607-3274-2023-4-10

UDC 519.6

PARALLEL AND DISTRIBUTED COMPUTING TECHNOLOGIES FOR AUTONOMOUS VEHICLE NAVIGATION

Mochurad L. I. – PhD, Associate Professor, Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine.

Mamchur M. V. – Student, Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. Autonomous vehicles are becoming increasingly popular, and one of the important modern challenges in their development is ensuring their effective navigation in space and movement within designated lanes. This paper examines a method of spatial orientation for vehicles using computer vision and artificial neural networks. The research focused on the navigation system of an autonomous vehicle, which incorporates the use of modern distributed and parallel computing technologies.

Objective. The aim of this work is to enhance modern autonomous vehicle navigation algorithms through parallel training of artificial neural networks and to determine the optimal combination of technologies and nodes of devices to increase speed and enable real-time decision-making capabilities in spatial navigation for autonomous vehicles.

Method. The research establishes that the utilization of computer vision and neural networks for road lane segmentation proves to be an effective method for spatial orientation of autonomous vehicles. For multi-core computing systems, the application of parallel programming technology, OpenMP, for neural network training on processors with varying numbers of parallel threads increases the algorithm's execution speed. However, the use of CUDA technology for neural network training on a graphics processing unit significantly enhances prediction speeds compared to OpenMP. Additionally, the feasibility of employing PyTorch Distributed Data Parallel (DDP) technology for training the neural network across multiple graphics processing units (nodes) simultaneously was explored. This approach further improved prediction execution times compared to using a single graphics processing unit.

Results. An algorithm for training and prediction of an artificial neural network was developed using two independent nodes, each equipped with separate graphics processing units, and their synchronization for exchanging training results after each epoch, employing PyTorch Distributed Data Parallel (DDP) technology. This approach allows for scalable computations across a higher number of resources, significantly expediting the model training process.

Conclusions. The conducted experiments have affirmed the effectiveness of the proposed algorithm, warranting the recommendation of this research for further advancement in autonomous vehicles and enhancement of their navigational capabilities. Notably, the research outcomes can find applications in various domains, encompassing automotive manufacturing, logistics, and urban transportation infrastructure. The obtained results are expected to assist future researchers in understanding the most efficient hardware and software resources to employ for implementing AI-based navigation systems in autonomous vehicles. Prospects for future investigations may encompass refining the accuracy of the proposed parallel algorithm without compromising its efficiency metrics. Furthermore, there is potential for experimental exploration of the proposed algorithm in more intricate practical scenarios of diverse nature and dimensions.

KEYWORDS: computer vision, neural networks, navigation methods, CUDA technology, PyTorch DDP technology.

ABBREVIATIONS

NN is a Neural Network;
OpenMP is an Open Multi-Processing;
CUDA is a Compute Unified Device Architecture;
DDP is a Distributed Data Parallel;
LiDAR is a Light Detection and Ranging;
ACO is an Ant Colony Optimization;
CNN is a Convolutional Neural Network;
IoT is an Internet of Things;
FPN is a Feature Pyramid Network;
CPU is a central processing unit;
GPU is a graphics processing unit.

NOMENCLATURE

N is the number of records in the dataset;
p is the cores count;
$T_1()$ is an execution time of a sequential algorithm;
$T_p()$ is an execution time of a parallel algorithm.

INTRODUCTION

With the advancement of autonomous vehicles, the demand for high-precision navigation systems and efficient algorithms is becoming increasingly crucial. Optimization and enhancement of existing artificial intelligence methods to improve navigation accuracy, along with the application of parallel computing to boost algorithm speed, have the potential to unlock new opportunities and contribute significantly to the development of the autonomous transportation sector [1, 2]. Algorithm and navigation method optimization can contribute to ecological and economic development as autonomous vehicles have the potential to reduce fuel costs and facilitate efficient infrastructure utilization.

Improvements in navigation algorithms can have a positive impact on various sectors, including logistics, automated warehouses, and robotics, where precise localization and navigation are critically important for effective operations [3, 4].

Since autonomous vehicles must react to real-time road situations, parallel computing helps ensure the swift execution of algorithms, which is vital for road safety and efficiency [5].

DOI 10.15588/1607-3274-2023-4-11

This work investigates the effectiveness of parallelizing the training and prediction algorithm of an artificial neural network for road lane segmentation across various devices: processor, graphics processing unit (GPU), and two GPUs simultaneously. Different distributed and parallel computing technologies are considered, including OpenMP [6], CUDA [7], and PyTorch DDP [8].

The object of this research is the navigation system of an autonomous vehicle, which encompasses various contemporary distributed and parallel computing technologies.

The subject of investigation comprises existing algorithms utilized for enhancing spatial navigation in autonomous vehicles, as well as parallel computing technologies aimed at improving the performance of these algorithms.

The purpose of this work is to enhance current navigation algorithms for autonomous vehicles in space by means of parallel training of artificial neural networks, and to determine the optimal combination of technologies and devices to increase speed and enable real-time decision-making capabilities.

1 PROBLEM STATEMENT

Let $D_{train} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training dataset, where x_i denotes the input data and y_i denotes the corresponding ground truth lane segmentation labels for S samples, i.e., $x_i = \{x_{i,s}\}_{s=1}^{S}$ and $y_i = \{y_{i,s}\}_{s=1}^{S}$, by analogy let D_{test} be the test dataset.

Suppose θ represents the parameters of the artificial neural network model, C represents the configuration space, encompassing parameters such as thread counts and parallel programming methodologies.

The functions $J(D_{train}, \theta, C)$ and $J(D_{test}, \theta, C)$ quantify the optimization criterion for the training and test dataset, encapsulating metrics that measure the efficiency of the parallelized algorithm. These metrics include aspects such as training duration, and prediction error.

In summary, the problem is to determine optimal parameters of the artificial neural network model θ and configuration space C that lead to the most efficient parallelization of the artificial neural network-based lane segmentation on test dataset, so that $J(D_{test}, \theta, C) \to \min$.

2 REVIEW OF THE LITERATURE

During the analysis of scientific papers and sources, a list of facts that served as the foundation for this research was identified. The first fact is that artificial intelligence methods are finding increasing applications in autonomous vehicles. This is due to their potential to significantly enhance road safety and make vehicle control more efficient and comfortable for drivers. According to a study, the use of autonomous vehicles with a 50% share can reduce the number of road accidents by 29% [9]. The second fact is that spatial navigation in autonomous vehicles relies on a combination of various technologies, including computer vision and artificial neural networks [10]. The third fact is that different parallel computing technologies are employed for training artificial neural networks [11, 12]. Other navigation methods for autonomous vehicles that utilize artificial intelligence include deep learning and image recognition-based techniques, such as the use of LiDAR sensors and cameras to gather environment data and create a 3D road map. This enables the vehicle to determine its location and identify surrounding objects [10].

Convolutional neural networks are effective when working with input data like images, including those from autonomous vehicle cameras [13]. Convolutional encoder-decoder architectures, on the other hand, are used to reduce the dimensionality of images and are often employed for image segmentation tasks [14].

For training neural networks used in autonomous vehicles, large datasets are required, particularly images of roads and road markings. One way to increase data volume is through data augmentation, which involves applying random transformations to images, such as scaling and rotation [15]. When using neural networks for vehicle navigation, factors like changes in lighting and atmospheric conditions must be considered. To address this, neural network models with additional layers responsible for data normalization (Batch Normalization) and the introduction of random noise to input data can be used [16]. To combine data from different sensors, Kalman filters and enhanced versions of these filters can be employed [17].

In [18], a method for lane boundary detection is presented, which operates by extracting candidate lane segments from an image and subsequently selecting the most prominent lane using dynamic programming. The authors utilize real road videos for experiments, demonstrating the effectiveness of their proposed approach. However, this method is considered outdated and does not account for factors such as changes in lighting and atmospheric conditions.

In [19], authors propose a hybrid approach based on Ant Colony Optimization (ACO) for line detection in images, using the Canny edge detection method and Hough transform for line extraction. The proposed system operates quickly but, as noted by the authors, is confidently applicable only on straight roads.

In [20], authors address road scene segmentation for RGB images using recent advancements in semantic segmentation through convolutional neural networks (CNN) and convolutional encoder-decoders. They introduce several architecture improvements that balance segmentation quality and computational speed. Experimental results indicate that their model provides accurate lane predictions in the original image size..

In [21], the authors addressed the problem of lane detection using an Internet of Things (IoT) system for

DOI 10.15588/1607-3274-2023-4-11

interaction between different modules, including the car module, cloud module, and remote car controller. The method for lane detection and tracking is executed initially on the car module and then transmitted to the cloud module for additional processing. The authors achieved a processing speed of approximately 31 ms per frame. An explicit drawback of this approach is the requirement for the car to be within the cellular network coverage area and have access to the internet, which is not always guaranteed, particularly on remote highways.

In [22], the authors utilize deep learning to tackle the lane segmentation problem, employing deep convolutional neural networks. The system achieves a respectable accuracy of 96%, but it requires 132 ms for processing a single frame.

One of the limitations of the previously presented algorithm and many others analyzed by us is that authors consider training and operating neural networks on a single powerful node (video processor) for lane segmentation, which can significantly limit the speed of learning and predictions of such networks in real-world scenarios. Our approach involves multiple independent nodes with separate video processors for parallel training and prediction, ensuring greater scalability and speed. Additionally, our approach utilizes PyTorch DDP technology for efficient communication and synchronization between nodes.

In summary, while the literature review reveals several promising approaches to lane segmentation in autonomous driving, most of them do not explore multi-node (video processor) training. Our proposed algorithm, based on parallel training and prediction on multiple independent nodes with separate video processors and PyTorch DDP communication, represents significant progress in terms of efficiency and scalability.

3 MATERIALS AND METHODS

To solve image processing tasks, CNNs (Convolutional Neural Networks) are widely used, including for object segmentation in images [13]. CNN consists of a sequence of convolutional and pooling layers, allowing the model to automatically identify important features of images at different levels.

Convolutional layers perform the convolution operation, which involves moving filters (of varying sizes and shapes) over the image to extract different image features such as edges, corners, textures, and more. The result of convolution is a feature map that highlights important regions of the image. Pooling layers reduce the size of the feature map and retain the most important features from each region, reducing the number of network parameters and preventing overfitting. CNNs leverage internal pixel relationships within the image for effective object segmentation. Operations of this type form the basis of convolutional encoders-decoders and Feature Pyramid Networks (FPN), which are used to address the lane segmentation task in the present study.

This network was proposed in 2017 with the aim of improving the object segmentation process in images. FPN (Feature Pyramid Network) consists of several convolutional layers that interact with the object in the image at different scales [23].

For the segmentation task, a slightly modified version of the FPN network is used, where each FPN level is gradually increased using convolutional functions and bilinear upscaling until it reaches a scale of 1/4. Then, these outputs are added and finally transformed into pixel-level output [24]. In general, the use of the FPN network for segmentation tasks allows for improved accuracy of results by optimally utilizing features at different scales of image resolution.

The time complexity of the convolutional encoder-decoder (FPN network) depends on the size of the input and output data, as well as the number of layers and filters in the network. In general, the time complexity can be expressed as a function of the number of operations required to process the input data.

Assuming that the convolutional encoder-decoder has L layers, filters with a size of $F \times F$ at each layer, and input data with a size of $D \times D$, then the general time complexity of the algorithm is:

$$O(N * L * D^2 * F^2).$$

Due to the fact that for each layer, the input data is processed by filters of size $F \times F$, each of size and this process is repeated L times (number of layers).

During parallelization using the CPU, the time complexity decreases proportionally to the number of physical threads engaged in computing mathematical operations during training (for example, calculating activation functions), where $N = N / THREADS$.

During parallelization using a GPU, the time complexity decreases proportionally to the number of computational units (CUDA cores) on the graphics processor and their speed, where $N = N / GRID\_SIZE$.

During parallelization using GPU and additionally PyTorch DDP, the time complexity decreases proportionally to the CUDA cores and is further divided by the number of nodes with video processors used, since the data (N) is shared among them for processing, where $N = N / (GRID\_SIZE * N\_DEVIES)$.

For training the neural network, the PyTorch library was utilized, which achieves parallelization of the training process through OpenMP. During the course of the research, the specific directives that were employed and the manner in which basic operations are parallelized during training were analyzed, specifically:

– To determine the number of used threads, the function omp_set_num_threads() is employed;

– The #pragma omp parallel for directive is used for parallelizing the ReLU activation function;

DOI 10.15588/1607-3274-2023-4-11

– The #pragme omp parallel for schedule(static) directive is utilized for parallelizing the loop of the Adam optimizer;

– The #pragma omp parallel directive is applied for parallelizing the operations of the convolutional layer;

– Matrix multiplication operations (batch_matmul) are parallelized using the #pragma omp parallel for collapse(2) directive.

Regarding the parallelization of the algorithm using CUDA technology, in our case, we utilized the CUDA kernel implementation in PyTorch, which, in turn, leverages existing NVIDIA solutions such as CUBLAS and CUDNN.

Since we have a single kernel, PyTorch by default employs the following values for constructing the grid:

```
THREADS_PER_BLOCK = 512;
BLOCKS_PER_SM = 4;
```

As a result of parallelizing the algorithm using CUDA technology:

– For parallelizing the operations of the convolutional layer, cudnnConvolutionBackwardFilter was used, which is a part of CUDNN;

– For parallelizing pooling operations, we utilize cudnnPoolingForward, which is provided with an array of tensors;

– For batch normalization, we employ cudnnBatchNormalizationForward and cudnnBatchNormalizationBackward.

The PyTorch DistributedDataParallel (DDP) technology allows distributing computations across multiple devices, such as servers or GPUs, to accelerate the training speed of models. PyTorch DDP employs an asynchronous approach to data and computation distribution among devices. It utilizes collective communication mechanisms, enabling each device to exchange data with others in the group. Additionally, PyTorch DDP ensures automatic parameter synchronization among devices during training. With PyTorch DDP, computations can be distributed across multiple servers or nodes, thereby enhancing computational power and reducing model training time. Moreover, PyTorch DDP supports automatic scaling of computational resources based on demand, facilitating efficient utilization of limited resources. DDP follows the CUDA algorithm, with the only difference being that the dataset is evenly split between two nodes, and the weights are synchronized using gradient aggregation at the end of each epoch, resulting in a 2x acceleration.

In the process of training machine learning models, input data plays a significant role. The quality and quantity of data can impact the accuracy and performance of the model. Therefore, it is crucial to properly select and prepare input data for model training.

One of the most popular applications for creating datasets for autonomous driving models is the CARLA Simulator [25]. This open-source software allows simulating urban traffic and autonomous vehicles. CARLA enables the creation of diverse scenarios for model training and testing, including simulations of various weather conditions, road traffic, pedestrian behavior, and other objects on the road.

The dataset created using the CARLA Simulator consists of images of road lanes with markings and other road elements. Each image is sized 1024x512 pixels and is presented in color. The total size of the dataset is 2.05 GB.

The images for training are captured by a camera mounted on a simulated vehicle. The annotated images provide segmentation masks. Each pixel in the annotated image is classified as:

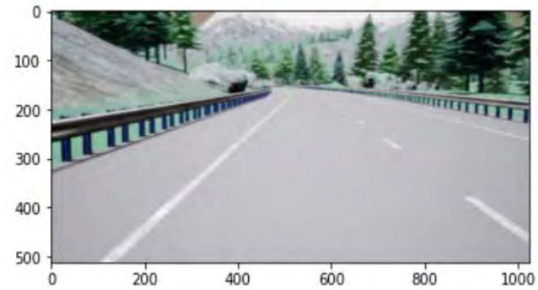
– part of the left lane boundary;

– part of the right lane boundary;

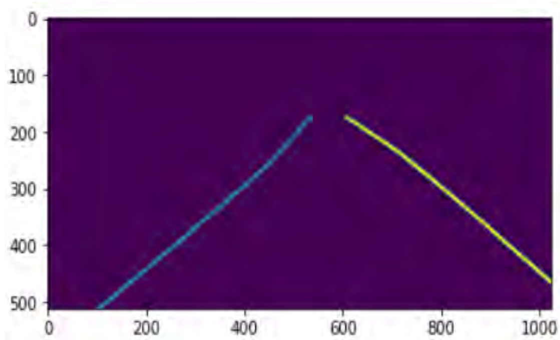
– none of the above (background).

The dataset was intentionally divided into a training and validation set: 3075 images for training and 129 for validation, along with 3075 and 129 corresponding mask images, respectively.

The challenge associated with this dataset is to train the model to accurately predict the segmentation masks for the validation dataset (see Fig. 1).



a



b

Figure 1 – The example of training samples:
a – original image; b – corresponding image-mask

DOI 10.15588/1607-3274-2023-4-11

Data augmentation can be used to increase the amount of training data and improve the quality of the model [26]. For example, images can be altered using techniques such as cropping, different positioning, color adjustments, resizing, and other transformations. Data augmentation can be particularly useful when working with limited data. In cases where the dataset lacks sufficient data for a specific task, augmentation can create new data by manipulating existing examples. This can help prevent overfitting, provide a more diverse dataset, and enhance the model's generalization ability.

One tool for data augmentation is the Python library "imgaug". It offers a variety of functions for image transformations, including rotation, scaling, color changes, noise addition, and more. Additionally, there are other libraries and tools available for data augmentation that can be used to enhance the quality of both data and machine learning models.

The developed algorithm performs data augmentation before training the network to enhance the accuracy of network training (an example of this is illustrated in Figure 2). The operations used in this sequence are as follows:

– **ShiftScaleRotate:** shifts, scales, and rotates the image with random parameters.

– **IAAAdditiveGaussianNoise:** adds Gaussian noise to the image with a probability of 0.2.

– **CLAHE:** applies the Contrast Limited Adaptive Histogram Equalization algorithm to enhance image contrast.

– **RandomBrightness:** changes the brightness of the image by a random amount.

– **RandomGamma:** adjusts the gamma of the image to a random value.

– **IAASharpen:** sharpens the image.

– **Blur:** applies blurring to the image to reduce sharpness.

– **MotionBlur:** adds motion blur to the image.

– **RandomContrast:** changes the contrast of the image by a random amount.

– **HueSaturationValue:** changes the hue and saturation of the image to random values.

After applying this sequence of operations, the images will slightly differ from each other, which allowed us to improve the model's performance on the validation data-set.

Step-by-Step Description of the Proposed Algorithm:

1. Import necessary libraries.

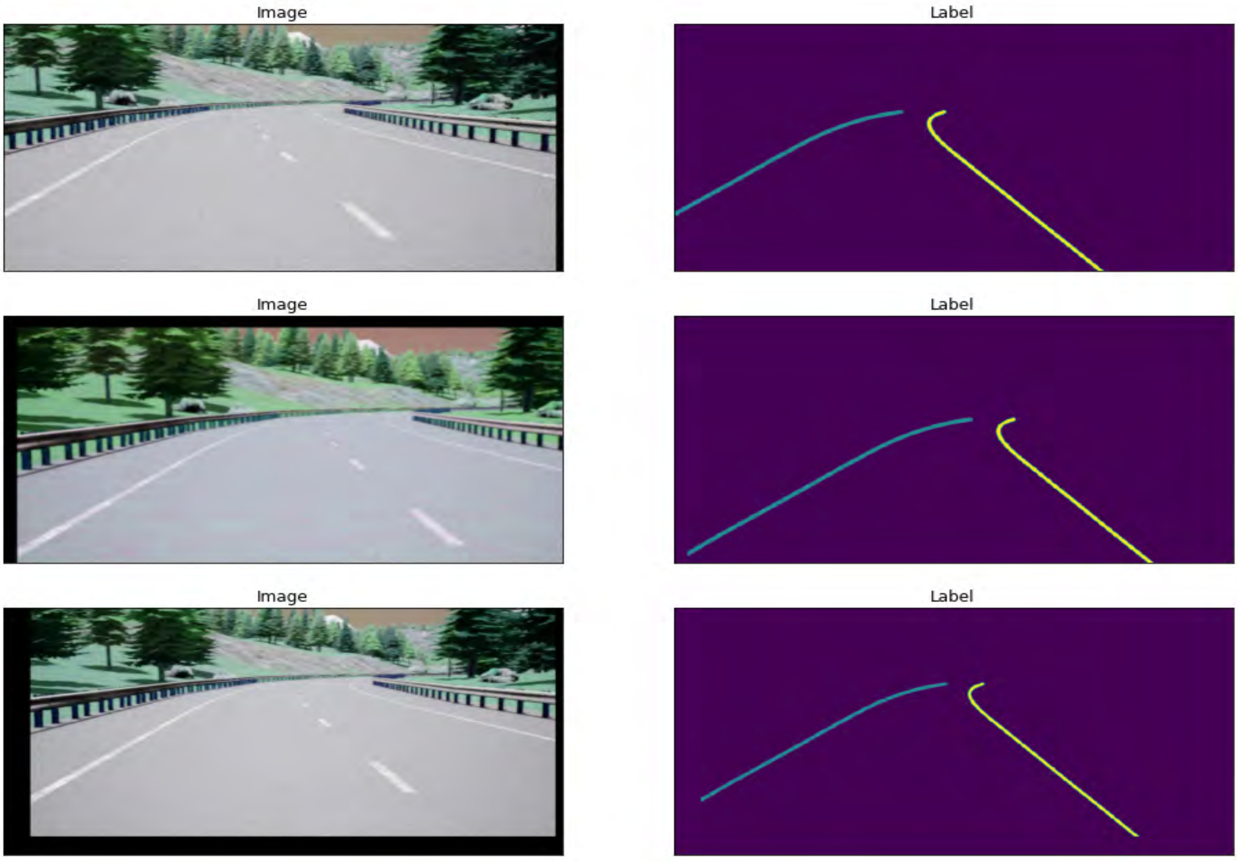


Figure 2 – Resulting augmented input image (left) and corresponding masks (right)

DOI 10.15588/1607-3274-2023-4-11

2. Declare the class CarlaLanesDataset with methods `__init__`, `__getitem__`, and `__len__`, which will be used to retrieve and preprocess the data.

3. Load the dataset into memory (training and validation sets).

4. Declare methods get_training_augmentation, get_validation_augmentation, and get_preprocessing to define transformations for data augmentation and preprocessing.

5. Apply augmentation to the loaded training dataset.

6. Initialize the artificial neural network model object.

7. Initialize the loss function and optimizer objects.

8. Parallel execution of the training process:

– On CPU using OpenMP technology.

– On GPU using CUDA technology.

– On multiple GPUs using CUDA technology and PyTorch DDP.

9. Validate on the testing dataset.

Next step taken is to calculate the theoretical acceleration and efficiency metrics for parallel algorithms using different numbers of threads when parallel computations are performed on the CPU. Additionally, we calculated the acceleration metrics for parallel algorithms on the GPU. To compute these metrics, we used the following formulas:

$$S_p(N) = \frac{T_1(N)}{T_p(N)}, \quad (1)$$

$$E_p(N) = \frac{S_p(N)}{p}. \quad (2)$$

Here, Equation (1) is used to calculate the speedup, and Equation (2) – the efficiency.

First, let's perform a theoretical speedup estimation for various numbers of processors used for training the neural network. It should be noted that calculating the theoretical speedup for training a model on the GPU is analytically impossible since the number of graphic cores used during training is unknown.

$$S_2(N) = \frac{T_1(3075)}{T_2(3075)} = \frac{O(3075 * L * D^2 * F^2)}{O\left(\frac{3075}{2} * L * D^2 * F^2\right)} = 2,$$

$$S_4(N) = \frac{T_1(3075)}{T_4(3075)} = \frac{O(3075 * L * D^2 * F^2)}{O\left(\frac{3075}{4} * L * D^2 * F^2\right)} = 4,$$

$$S_8(N) = \frac{T_1(3075)}{T_8(3075)} = \frac{O(3075 * L * D^2 * F^2)}{O\left(\frac{3075}{8} * L * D^2 * F^2\right)} = 8.$$

Let's derive the theoretical estimates of efficiency:

$$E_2(N) = \frac{S_2(N)}{2} = 1,$$

$$E_4(N) = \frac{S_4(N)}{4} = 1,$$

$$E_8(N) = \frac{S_8(N)}{8} = 1.$$

It should be noted that the provided estimates apply to a system with p processor (core) computational units.

4 EXPERIMENTS

During the research, the training of the neural network was conducted on a CPU using the OpenMP technology. For this purpose, a computer with an Intel(R) Xeon(R) CPU @ 2.20GHz processor with 2 cores and 4 threads was utilized. The network training was carried out over 5 epochs.

Parallelization was achieved through the inter-op functionality of OpenMP – a specific thread pool is allocated for performing individual tasks, such as processing one of the input parameters. Inter-op allows us to handle micro-operations like pooling, batch operations, or matrix multiplication by dividing sub-tasks among threads. As a result of inter-op, the tasks of an iteration are synchronized, marking its completion.

The results of training the neural network on the CPU using OpenMP technology revealed a test dataset accuracy of approximately 96%. The network's prediction results are depicted in Fig. 3.

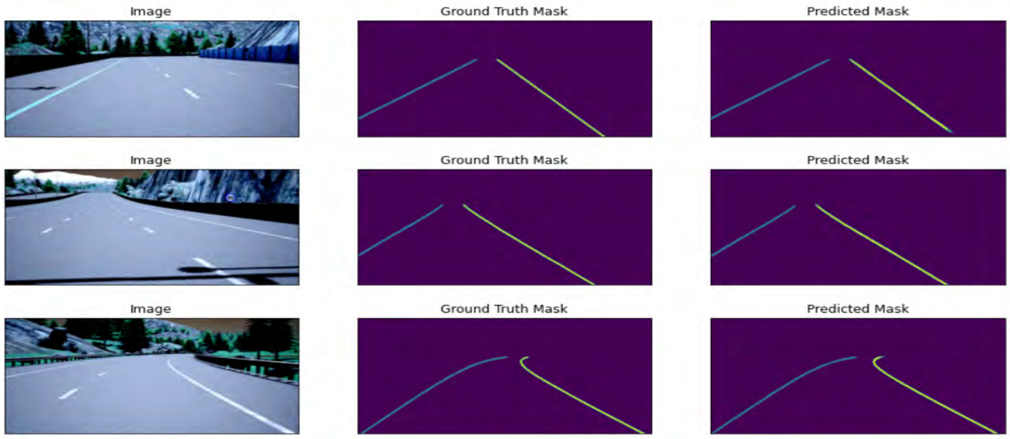


Figure 3 – The prediction results of the trained neural network. The first column displays images from the test dataset, the middle column shows the corresponding ground truth lane masks, and the left column presents the predicted lane masks

DOI 10.15588/1607-3274-2023-4-11

The training algorithm was also parallelized on GPU using CUDA technology. Specifically, the network training was conducted on an NVIDIA T4 graphics card, a professional GPU released in 2018 designed for high-performance computing and artificial intelligence applications. It is built on the Nvidia Turing architecture and boasts 2560 CUDA cores, enabling it to handle large datasets and perform tasks such as deep learning and machine learning with high precision. The GPU comes equipped with 16 gigabytes of fast video memory.

PyTorch leverages pre-existing NVIDIA cores along with CUBLAS and CUDNN frameworks. These cores receive requests for executing intra-op tasks from CUBLAS and inter-op tasks from the CUDNN framework, and then perform these operations using available CUDA cores.

By utilizing PyTorch DDP technology, we were able to utilize a second device with a similar GPU module, effectively harnessing a total of 5120 CUDA cores for training. This parallel execution approach not only allows us to combine different GPUs but also avoids being restricted to a single physical device [27], which might limit the number of GPUs that can be attached. This approach enables us to scale the network to any desired size. The primary device serves as a synchronization point, while other nodes are launched with specified IP addresses, and they receive work ranges and necessary computation data from the controlling device to perform calculations.

5 RESULTS

In Table 1, we will present the time costs of sequential and parallel implementations on CPU using OpenMP technology, on GPU with CUDA, and on GPU with PyTorch DDP technology. The results of Table 1 are also visualized in Fig. 4.

Table 1 – Training Time of Neural Network on CPU, Single GPU, and Dual GPUs (in minutes)

Sequential excution	CPU + OpenMP			GPU + CUDA	2xGPU + DDP
	2	4	8		
333.5	144	123.6	144.6	9.1	4.8

From Table 1 and Figure 4, it is evident that the results of multi-threaded training demonstrate that increasing the number of threads up to 4 leads to improved performance, followed by a decline at 8 threads. Transitioning from 1 to 2 threads doubles the speed due to the presence of only 2 physical cores, indicating the validity of the obtained results. Moving from 2 to 4 threads results in a slight speed increase since the number of physical cores remains the same, but the logical cores provide additional cache memory for the threads. However, utilizing 8 threads depletes the available cache memory, prompting the processor to reload data to allow both threads to share a cache memory, resulting in cache miss penalties.

6 DISCUSSION

Considering the achieved speedup through the use of OpenMP technology, it can be concluded that employing OpenMP for neural network training on CPU is an effective method to reduce training time, especially in cases where GPU usage is not feasible.

Training the neural network using GPU is 36.6 times more efficient than sequential CPU training and 14 times more efficient than CPU training with 4 threads. With two GPUs, the training time per epoch is reduced to 4.8

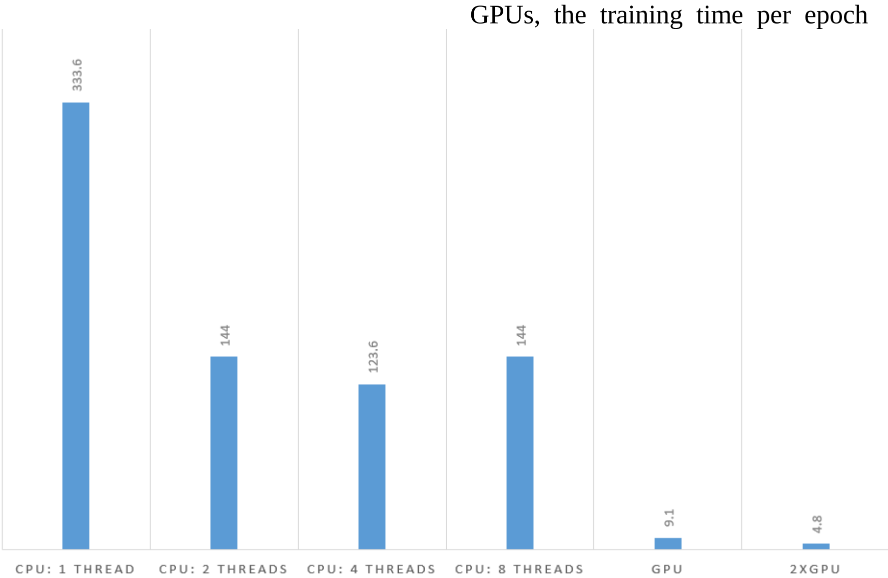


Figure 4 – Comparative diagram illustrating the training time dependency of a neural network on CPU with the involvement of threads, on a single GPU, and on two GPUs

DOI 10.15588/1607-3274-2023-4-11

minutes, which is 1.86 times more efficient compared to using a single GPU. However, the acceleration is not precisely twofold, as each device has independent video memory, and we need to synchronize training results between the devices in our local network of compute nodes after each epoch, which consumes some time.

Table 2 – Actual acceleration metrics of the parallel algorithm depending on the number of utilized threads on CPU, as well as the parallel algorithm on GPU and on two GPUs simultaneously

CPU, number of threads			GPU	2xGPU
2	4	8		
2.33	2.69	2.31	36.6	69.5

From Table 2, we observe that we achieved higher acceleration metrics for two threads compared to the theoretical values, which is due to the specifics of the Linux kernel scheduler prioritizing tasks with multiple active threads, thereby resulting in a single-threaded program having significantly lower computational speed than expected.

Since the process with one thread resulted in reduced performance, parallel execution with two threads yielded higher acceleration metrics.

When transitioning from 2 to 4 threads, the efficiency growth is marginal, as the number of physical cores remains unchanged, but the logical cores provide additional cache memory for threads.

With the utilization of 8 threads, the available cache memory is exhausted, prompting the processor to reload data to allow both threads to use the same cache memory, resulting in cache miss penalties.

Consequently, training the neural network using DDP is 69.5 times more efficient than sequential CPU training and 25.8 times more efficient than CPU training with 4 threads. Furthermore, it is 1.86 times more efficient than training with a single GPU.

Table 3 – Actual efficiency metrics of the parallel algorithm depending on the number of threads used on CPU

Number of threads		
2	4	8
1.16	0.67	0.29

Analyzing the results of Table 3, it can be observed that the actual efficiency metrics do not align with the theoretical ones. This discrepancy arises from the fact that the considered CPU has only 2 physical cores, but 4 logical cores provide additional cache memory for threads. However, as the number of threads increases to 8, the overhead of supporting these threads becomes predominant. Hence, the obtained results are reliable.

CONCLUSIONS

The paper analyzes contemporary approaches and methods for solving the problem of road lane segmentation to localize vehicles. During the analysis of scientific articles and sources, a list of facts was identified upon which this research is based.

It has been found that the utilization of modern parallel and distributed computing technologies on both CPU and GPU can significantly reduce the training time of neural networks for addressing the problem outlined in this study.

The scientific novelty of the obtained results lies in the introduction of a parallel algorithm for solving the road lane segmentation task using multiple GPUs with CUDA technology and PyTorch DDP. It has been established that the use of DDP expands computational capabilities by adding new independent nodes that can utilize both GPUs and CPUs. Therefore, this technology allows bypassing the limitations of calculations on a single device and achieving acceleration by orders of magnitude, sacrificing time only for exchanging intermediate training results between nodes.

In this work, based on the proposed algorithm, it was possible to achieve approximately a 90% increase in acceleration when using training on two nodes with NVIDIA T4 GPUs compared to one node. This is around 25 times faster compared to using the OpenMP technology for multi-core computer systems.

Furthermore, it was found that the time required for lane prediction for a single road frame by the model reached 19 ms, which is 1.63 times faster than in [20] and 6 times faster than in [21].

The algorithm employed in this study enabled achieving an accuracy of 96%, which is similar to [22]. However, it can be confidently stated that without compromising accuracy, significant acceleration of solving the road lane segmentation task for vehicle localization was achieved, specifically by a factor of 7.

The practical significance of the obtained results lies in the development of software that implements the proposed algorithm, as well as conducting a series of numerical experiments aimed at comparing the use of modern distributed and parallel computing technologies for autonomous vehicle navigation. The findings of this research can have a positive impact on road safety, cost-effectiveness, environmental friendliness, and transportation accessibility. Furthermore, they can contribute to the advancement of smart cities, integration of transportation systems, and enhance the competitiveness of automotive manufacturers. This research can also provide insights into the most efficient hardware and software tools to employ for implementing AI-based navigation systems in autonomous vehicles, depending on the situation [28].

The prospects for further research involve exploring the proposed parallel algorithm for a wide range of practical tasks.

DOI 10.15588/1607-3274-2023-4-11

REFERENCES

1. Varsi A., Taylory J., Kekempanosz L., Pyzer-Knapp E., Maskell S. A Fast Parallel Particle Filter for Shared Memory Systems, *IEEE Signal Process. Lett*, 2020, Vol. 27, pp. 1570–1574.
2. Huang K., Cao J. Parallel Differential Evolutionary Particle Filtering Algorithm Based on the CUDA Unfolding Cycle, *Wireless Communications and Mobile Computing*, 2021, Vol. 2021, pp. 1–12.
3. Kretzschmar H., Kuderer M., Burgard W. Learning to predict trajectories of cooperatively navigating agents, *2014 IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China, 2014, pp. 4015–4020.
4. Wang P., Yang J., Zhang Y., Wang Q., Sun B., Guo D. Obstacle-Avoidance Path-Planning Algorithm for Autonomous Vehicles Based on B-Spline Algorithm, *World Electr. Veh. J*, 2022, Vol. 13, №12:233, pp. 1–15.
5. Xu L., Ouyang Y., Ford J., Banerjee S., Chen M. Real-time 3D traffic scene understanding from moving vehicles, *IEEE Transactions on Intelligent Transportation Systems*, 2018, Vol. 19, № 9, pp. 2827–2838.
6. Mochurad L. Optimization of Regression Analysis by Conducting Parallel Calculations, *COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems*. Kharkiv, Ukraine, 22–23 April, 2021, pp. 982–996.
7. Mochurad L. Canny Edge Detection Analysis Based on Parallel Algorithm, Constructed Complexity Scale and CUDA, *Computing and Informatics*, 2022, Vol. 41, № 4, pp. 957–980.
8. Li Sh., Zhao Yanli, Varma Rohan, Salpekar Omkar, Noordhuis Pieter, Li Teng, Paszke Adam, Smith Jeff, Vaughan Brian, Damania Pritam, Chintala Soumith PyTorch Distributed: Experiences on Accelerating Data Parallel Training, *Proceedings of the VLDB Endowment*, 2020, Vol. 13, № 12, pp. 3005–3018.
9. Petrović Đ., Mijailovic R., Pešić D. Traffic Accidents with Autonomous Vehicles: Type of Collisions, Manoeuvres and Errors of Conventional Vehicles' Drivers, *Transportation Research Procedia*, 2020, Vol. 45, pp. 161–168.
10. Haris M., Glowacz A. Navigating an Automated Driving Vehicle via the Early Fusion of Multi-Modality, *Sensors*, 2022, Vol. 22, № 4, pp. 1–18.
11. Hoffmann R. B., Löff J. , Griebler D. et al. OpenMP as runtime for providing high-level stream parallelism on multicores, *J. Supercomput*, 2022, Vol. 78, pp. 7655–7676.
12. Sierra-Canto X. Madera-Ramirez F., V. Uc-Cetina//Parallel Training of a Back-Propagation Neural Network Using CUDA, *Ninth International Conference on Machine Learning and Applications*. Washington, DC, USA, IEEE, 2010, pp. 307–312.
13. Sewak M., Karim Md. R., Pujari P. Implement advanced deep learning models using Python. *Practical convolutional neural networks*. Birmingam-Mumbai:Packt Publishing, 2018, 218 p.
14. Badrinarayanan V., Kendall A., Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, Vol. 39, № 12, pp. 2481–2495.
15. Shijie J., Ping W., Peiyi J. and Siping H. Research on data augmentation for image classification based on convolution neural networks, 2017 *Chinese Automation Congress (CAC)*. Jinan, China, IEEE, 2017, pp. 4165–4170.
16. Bjorck N. Gomes Carla P, Bart Selman, and Kilian Q Weinberger Understanding batch normalization, *In Advances in Neural Information Processing Systems: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*. Montréal, Canada, 2018, pp. 7694–7705.
17. Chen J., Chen K., Ding C., Wang G., Liu Q. and Liu X. An Adaptive Kalman Filtering Approach to Sensing and Predicting Air Quality Index Values, *in IEEE Access*, 2020, Vol. 8, pp. 4265–4272.
18. Kang D.-J. Jung M.-H. Road lane segmentation using dynamic programming for active safety vehicles, *Pattern Recognition Letters*, 2003, Vol. 24, Issue 16, pp. 3177–3185.
19. Daigavane P. M., Bajaj P. R. Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization, *3rd International Conference on Emerging Trends in Engineering and Technology*. Goa, India, 2010, pp. 76–80.
20. Oliveira G. L., Burgard W., Brox T. Efficient deep models for monocular road segmentation, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Daejeon, Korea (South), 2016, pp. 4885–4891.
21. Ghanem S., Kanungo P., Panda G. et al. Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure, *Complex Intell. Syst.*, 2023, Vol. 9, pp. 3601–3612.
22. Kortli Ya., Gabsi Souhir, Lew F. C., Lew Ya. V., Jridi M., Merzougui M., Atri M. Deep embedded hybrid CNN-LSTM network for lane detection on NVIDIA Jetson Xavier NX, *Knowledge-Based Systems*, 2022, Vol. 240, pp. 1–33.
23. Hu M., Li Y., Fang L., Wang S. A2-FPN: Attention Aggregation Based Feature Pyramid Network for Instance Segmentation, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15343–15352.
24. Zapata M.A.D., Erkent Ö., Laugier Ch. YOLO-based Panoptic Segmentation Network, *COMPSAC 2021 – Intelligent and Resilient Computing for a Collaborative World 45th Anniversary Conference*. Madrid, Spain, Jul 2021, pp. 1–5.
25. Gómez-Huélamo C., Del Egido J., Bergasa L. M. et al. Train here, drive there: ROS based end-to-end autonomous-driving pipeline validation in CARLA simulator using the NHTSA typology, *Multimed Tools Appl.*, 2022, Vol. 81, pp. 4213–4240.
26. Kostrikov I., Yarats D., Fergus R.. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels [Electronic resource]. Access mode: https://arxiv.org/abs/2004.13649.
27. Li Sh. et al. Pytorch distributed: Experiences on accelerating data parallel training [Electronic resource]. Access mode: https://arxiv.org/abs/2006.15704.
28. Severino A., Curto S., Barberi S., Arena F., Pau G. Autonomous Vehicles: An Analysis Both on Their Distinctiveness and the Potential Impact on Urban Transport Systems, *Appl. Sci.*, 2021, Vol. 11, № 8:3604, pp. 1–17.

Received 18.08.2023.
Accepted 21.11.2023.

DOI 10.15588/1607-3274-2023-4-11

УДК 519.6

ТЕХНОЛОГІЇ ПАРАЛЕЛЬНИХ І РОЗПОДІЛЕНИХ ОБЧИСЛЕНЬ ДЛЯ АВТОНОМНОЇ НАВІГАЦІЇ ТРАНСПОРТНИХ ЗАСОБІВ

Мочурад Л. І. – канд. техн. наук, доцент, доцент кафедри систем штучного інтелекту національного університету «Львівська політехніка», Львів, Україна.

Мамчур М. В. – студент кафедри систем штучного інтелекту національного університету «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Автономні автомобілі стають все більш популярними і одним з важливих сучасних завдань розробки таких автомобілів є забезпечення ефективної навігації останніх у просторі та їх руху у своїй виділеній проїзній смузі. У даній роботі розглянуто метод орієнтування у просторі автомобіля за допомогою комп'ютерного зору та штучних нейронних мереж. Об'єктом дослідження була система навігації автономного автомобіля, що включає в себе використання сучасних технологій розподілених та паралельних обчислень.

Мета роботи – вдосконалення сучасних алгоритмів навігації автономного автомобіля у просторі на основі паралельного навчання штучних нейронних мереж та визначення найоптимальнішої комбінації технологій та пристроїв для збільшення швидкості та можливості отримання рішення в режимі реального часу.

Метод. У роботі встановлено, що використання комп'ютерного зору та нейронних мереж для сегментації смуги дорожнього руху є ефективним методом орієнтації автономного автомобіля у просторі. При цьому для багатоядерних обчислювальних систем застосування технології паралельного програмування OpenMP для тренування нейронної мережі на процесорі з різним числом паралельних потоків збільшує швидкість виконання алгоритму. Проте використання технології CUDA для навчання нейронної мережі на відеопроцесорі дозволило значно збільшити швидкість передбачень в порівнянні з OpenMP. Також досліджено можливість використання технології PyTorch DDP для навчання нейронної мережі на декількох відеопроцесорах (вузлах) одночасно, що , в свою чергу, ще більш покращило час виконання передбачень в порівнянні з використанням одного відеопроцесора.

Результати. Розроблено алгоритм навчання та передбачення штучної нейронної мережі на двох незалежних вузлах з окремими відеопроцесорами та їх синхронізацією задля обміну результатами навчання після кожної епохи із використанням технології PyTorch DDP, що дозволяє масштабувати розрахунки при наявності більшої кількості потужностей і значно пришвидшити навчання моделі.

Висновки. Проведені експерименти підтвердили ефективність запропонованого алгоритму і дозволяють рекомендувати дане дослідження для подальшого розвитку автономних автомобілів та покращення їх навігаційних можливостей. Зокрема результати дослідження можуть знайти застосування у різних сферах, включаючи автомобільну транспортну промисловість, логістику та транспортну інфраструктуру міст. Отримані результати повинні допомогти наступним дослідникам зрозуміти, які апаратні та програмні засоби найефективніше використовувати для реалізації навігаційних систем на основі штучного інтелекту в автономних автомобілях. Перспективами подальших досліджень може бути покращення точності запропонованого паралельного алгоритму не погіршуючи показників ефективності, а також експериментальне дослідження запропонованого алгоритму на більш складних практичних задачах різної природи та розмірності.

КЛЮЧОВІ СЛОВА: комп'ютерний зір, нейронні мережі, методи навігації, технологія CUDA, технології PyTorch DDP.

ЛІТЕРАТУРА

1. A Fast Parallel Particle Filter for Shared Memory Systems / A. Varsi, J. Taylory, L. Kekempanosz et al.] // IEEE Signal Process. Lett. – 2020. –Vol. 27. – P. 1570–1574.
2. Huang K. Parallel Differential Evolutionary Particle Filtering Algorithm Based on the CUDA Unfolding Cycle / K. Huang, J. Cao // Wireless Communications and Mobile Computing. – 2021. – Vol. 2021. – P. 1–12.
3. Kretzschmar H. Learning to predict trajectories of cooperatively navigating agents / H. Kretzschmar, M. Kuderer, W. Burgard // 2014 IEEE International Conference on Robotics and Automation (ICRA). – Hong Kong, China, 2014. – P. 4015–4020.
4. Obstacle-Avoidance Path-Planning Algorithm for Autonomous Vehicles Based on B-Spline Algorithm / [P. Wang, J. Yang, Y. Zhang et al.] // World Electr. Veh. J. – 2022. – Vol. 13, №12:233. – P. 1–15.
5. Real-time 3D traffic scene understanding from moving vehicles / [L. Xu, Y. Ouyang, J. Ford et al.] // IEEE Transactions on Intelligent Transportation Systems. – 2018. – Vol. 19, № 9. – P. 2827–2838.
6. Mochurad L. Optimization of Regression Analysis by Conducting Parallel Calculations / L. Mochurad // COLINS-2021: 5th International Conference on Computational Linguistics and Intelligent Systems. – Kharkiv, Ukraine, 22–23 April, 2021. – P. 982–996.
7. Mochurad L. Canny Edge Detection Analysis Based on Parallel Algorithm, Constructed Complexity Scale and CUDA / L. Mochurad // Computing and Informatics. – 2022. – Vol. 41, № 4. – P. 957–980.
8. Li Sh. PyTorch Distributed: Experiences on Accelerating Data Parallel Training / [Shen Li, Yanli Zhao, Rohan Varma et al.] // Proceedings of the VLDB Endowment, 2020. – Vol. 13, № 12. – P. 3005–3018.
9. Petrović Đ. Traffic Accidents with Autonomous Vehicles: Type of Collisions, Manoeuvres and Errors of Conventional Vehicles' Drivers / Đ. Petrović, R. Mijailovic, D. Pešić // Transportation Research Procedia. – 2020. – Vol. 45. – P. 161–168.
10. Haris M. Navigating an Automated Driving Vehicle via the Early Fusion of Multi-Modality / M. Haris, A. Glowacz // Sensors. – 2022. – Vol. 22, № 4. – P. 1–18.
11. Hoffmann R. B. OpenMP as runtime for providing high-level stream parallelism on multi-cores / R. B. Hoffmann, J. Löff, D. Griebler et al. // J. Supercomput. – 2022. – Vol. 78. – P. 7655–7676.

DOI 10.15588/1607-3274-2023-4-11

12. Sierra-Canto X. Parallel Training of a Back-Propagation Neural Network Using CUDA / X. Sierra-Canto, F. Madera-Ramirez, V. Uc-Cetina// Ninth International Conference on Machine Learning and Applications. – Washington, DC, USA:IEEE, 2010. – P. 307–312.
13. Sewak M. Implement advanced deep learning models using Python / M. Sewak, Md. R. Karim, P. Pujari // Practical convolutional neural networks. – Birmingam-Mumbai : Packt Publishing, 2018. – 218 p.
14. Badrinarayanan V. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation / V. Badrinarayanan, A. Kendall, R. Cipolla // in IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2017. – Vol. 39, № 12. – P. 2481–2495.
15. Shijie J. Research on data augmentation for image classification based on convolution neural networks / J. Shijie, W. Ping, J. Peiyi and H. Siping // 2017 Chinese Automation Congress (CAC). – Jinan, China : IEEE, 2017. – P. 4165–4170.
16. Bjorck N. Understanding batch normalization / Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger // In Advances in Neural Information Processing Systems: 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada, 2018. – P. 7694–7705.
17. An Adaptive Kalman Filtering Approach to Sensing and Predicting Air Quality Index Values / J. Chen, K. Chen, C. Ding et al.] // in IEEE Access. – 2020. – Vol. 8. – P. 4265–4272.
18. Kang D.-J. Road lane segmentation using dynamic programming for active safety vehicles / Dong-Joong Kang and Mun-Ho Jung // Pattern Recognition Letters. – 2003. – Vol. 24, Issue 16. – P. 3177–3185.
19. Daigavane P. M. Road Lane Detection with Improved Canny Edges Using Ant Colony Optimization / P. M. Daigavane, P. R. Bajaj // 3rd International Conference on Emerging Trends in Engineering and Technology. – Goa, India, 2010. – P. 76–80.
20. Oliveira G. L. Efficient deep models for monocular road segmentation / G. L. Oliveira, W. Burgard, T. Brox // IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). – Daejeon, Korea (South), 2016. – P. 4885–4891.
21. Lane detection under artificial colored light in tunnels and on highways: an IoT-based framework for smart city infrastructure / [S. Ghanem, P. Kanungo, G. Panda et al.] // Complex Intell. Syst. – 2023. – Vol. 9, 2023. – P. 3601–3612.
22. Kortli Ya. Deep embedded hybrid CNN–LSTM network for lane detection on NVIDIA Jetson Xavier NX / [Yassin Kortli, Souhir Gabsi, Lew F. C. et al.], // Knowledge-Based Systems. – 2022. – Vol. 240. – P. 1–33.
23. A2-FPN: Attention Aggregation Based Feature Pyramid Network for Instance Segmentation / [Miao Hu, Yali Li, Lu Fang, Shengjin Wang] // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). – 2021. – P. 15343–15352.
24. Zapata M.A.D., Erkent Ö., Laugier Ch. YOLO-based Panoptic Segmentation Network / Manuel Alejandro Diaz Zapata, Özgür Erkent, Christian Laugier // COMPSAC 2021 – Intelligent and Resilient Computing for a Collaborative World 45th Anniversary Conference. – Madrid, Spain, Jul 2021. – P. 1–5.
25. Train here, drive there: ROS based end-to-end autonomous-driving pipeline validation in CARLA simulator using the NHTSA typology / [C. Gómez-Huélamo, J. Del Egido, L. M. Bergasa et al.] // Multimed Tools Appl. – 2022. – Vol. 81. – P. 4213–4240.
26. Kostrikov I. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels / I. Kostrikov, D. Yarats, R. Fergus // arXiv preprint arXiv:2004.13649. – 2020. – P. 1–22.
27. Li Sh. Pytorch distributed: Experiences on accelerating data parallel training / Shen Li et al. // arXiv preprint arXiv:2006.15704. – 2020. – P. 1–14.
28. Severino A. Autonomous Vehicles: An Analysis Both on Their Distinctiveness and the Potential Impact on Urban Transport Systems / [A. Severino, S. Curto, S. Barberi et al.] // Appl. Sci. – 2021. – Vol. 11, № 8:3604. – P. 1–17.

DOI 10.15588/1607-3274-2023-4-11

UDC 004.93

RCF-ST: RICHER CONVOLUTIONAL FEATURES NETWORK WITH STRUCTURAL TUNING FOR THE EDGE DETECTION ON NATURAL IMAGES

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 automating of the edge detection on natural images in intelligent systems is considered. The subject of the research is the deep learning convolutional neural networks for edge detection on natural images.

Objective. The objective of the research is to improve the edge detection performance of natural images by structural tuning the richer convolutional features network architecture.

Method. In general, the edge detection performance is influenced by a neural network architecture. To automate the design of the network structure in the paper a structural tuning of a neural network is applied. Computational costs of a structural tuning are incomparably less compared with neural architecture search, but a higher qualification of the researcher is required, and the resulting solution will be suboptimal. In this research it is successively applied first a destructive approach and then a constructive approach to structural tuning of the based architecture of the RCF neural network. The constructive approach starts with a simple architecture network. Hidden layers, nodes, and connections are added to expand the network. The destructive approach starts with a complex architecture network. Hidden layers, nodes, and connections are then deleted to contract the network. The structural tuning of the richer convolutional features network includes: (1) reducing the number of convolutional layers; (2) reducing the number of convolutions in convolutional layers; (3) removing at each stage the sigmoid activation function with subsequent calculation of the loss function; (4) addition of the batch normalization layers after convolutional layers; (5) including the ReLU activation functions after the added batch normalization layers. The obtained neural network is named RCF-ST. The initial color images were scaled to the specified size and then inputted in the neural network. The advisability of each of the proposed stages of network structural tuning was reseached by estimating the edge detection performance using the confusion matrix elements and Figure of Merit. The advisability of a structural tuning of the neural network as a whole was estimated by comparing it with methods known from the literature using the Optimal Dataset Scale and Optimal Image Scale.

Results. The proposed convolutional neural network has been implemented in software and researched for solving the problem of edge detection on natural images. The structural tuning technique may be used for informed design of the neural network architectures for other artificial intelligence problems.

Conclusions. The obtained RCF-ST network allows to improve the performance of edge detection on natural images. RCF-ST network is characterized by a significantly fewer parameters compared to the RCF network, which makes it possible to reduce the resource consumption of the network. Besides, RCF-ST network ensures the enhancing of the robustness of edge detection on texture background.

KEYWORDS: natural image, edge detection, convolutional network, richer convolutional features, structural tuning, batch normalization.

ABBREVIATIONS

CNN is a convolutional neural network;

RCF is a Richer convolutional features;

HED is a Holistically-nested edge detection;

LPCB is a Learning to predict crisp boundaries;

BDCN is a Bi-directional cascade network;

DexiNed is a Dense extreme inception network;

DSCD is a Deep structural contour detection;

PiDiNet is a Pixel difference network;

ReLU is a rectified linear unit;

RCF-ST is a Richer convolutional features with structural tuning;

BSDS500 is a Berkeley Segmentation Dataset with 500 images;

FOM is a Figure of Merit;

ODS is an Optimal Dataset Scale;

OIS is an Optimal Image Scale.

NOMENCLATURE

n is a number of rows of the image;

m is a number of columns of the image;

(x,y) are coordinates of the image pixel;

$\mathbf{I}(x,y)$ is a vector function representing an image by color components;

$I_R(x,y)$, $I_G(x,y)$, $I_B(x,y)$ are the functions of intensity of the red, green, blue color components respectively;

$struct_{RCF}$ is an architecture of the RCF network;

$param_{RCF}$ is a set of parameters of the RCF network;

W_{RCF} is a subset of RCF network layer weights;

B_{RCF} is a subset of RCF network bias values;

x_{ni} is a nth normalized output of the ith network layer;

γ_i is a compression of the x_{ni};

β_i is a shift of the x_{ni};

y_{ni} is a transformed with γ_i and β_i output of the network layer;

TP is apercentage of image background pixels that are correctly labeled as background;

TN is a percentage of image edge pixels that are correctly labeled as edge.

F_β is a F_β-score;

DOI 10.15588/1607-3274-2023-4-12

β is a F_β-score constant from 0 to infinity;
Pr is a precision;
Rc is a recall.

INTRODUCTION

The problem of edge detection on images remains relevant in the development of intelligence systems for a number of applications. These applications include technical and medical diagnostics, image search in databases and the Internet, face recognition, non-destructive testing, process control. The selection of the object edge detector for the implementing an intelligent system is determined primarily by the properties of the processed images, the noise level, as well as the requirements for the edge detection performance. So, in systems for monitoring the environment, transport and infrastructure, searching for images in databases and the Internet, and others, it becomes necessary to process natural images.

Natural images are characterized by a low level of noise. Objects on such images may contain texture areas or areas of smooth color change. When detecting the object edges on natural images, one should take into account not only color differences, but also the boundaries of texture areas. Then it is necessary to establish a correspondence between color differences and the boundaries of objects on natural images, ignoring the background texture and noisy pixels [1].

The object of research is the process of edge detection on natural images in intelligent systems.

In recent years, considering the problems of thick image edge contour, inaccurate positioning, and poor detection accuracy, a variety of edge detection methods based on deep learning CNN have been proposed. With the development of technology, the CNN edge detection accuracy has been increased. However, at the same time, the depth of the networks has been deepened, leading to problems such as a very large number of parameters, training difficulties, and model complexity [1].

For the effective use of a neural network, it is necessary to design its architecture and to train the weight coefficients. Network architectures are usually selected heuristically based on the experience of the developer. When image edge detecting, networks of too simple architecture are not able to adequately model the target dependence between the pixels of the original image and the edge map. Too complex architectures of neural networks imply an excessive number of free parameters, which in the learning process are tuned not only to restore the target dependence, but also noise [2]. One way to solve this problem is the structural tuning of CNN [3, 4].

The subject of the research is the structural tuning of the convolutional neural networks for edge detection on natural images.

When processing images of real scenes, the RCF network proved to be effective for edge detection [1]. However, the quality of the results of edge detection using this network is determined by the number of processing scales and by the network architecture. The latter implies the selection of such hyperparameters as the number and size of filter kernels in a layer, the adding and removing layers, including activation functions.

The aim of the research is to improve the edge detection performance of natural images by structural tuning the RCF deep learning network architecture.

1 PROBLEM STATEMENT

The color natural image is represented as $\mathbf{I}(x,y)=(I_R(x,y), I_G(x,y), I_B(x,y)\}$, where x=1, …, n; y=1, …, 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, 1]. To detect edges on the image, each pixel of the original image must be associated with the value of the target feature. There is a label of one of two classes, specifically, 0 for boundary pixels, 1 for pixels inside homogeneous areas. The values of the target feature for the natural image should be represented as a binary image which is the result of edge detection [5].

Let an RCF network RCF=$\{struct_{RCF}, param_{RCF}\}$ was preliminarily synthesized to detect the image edges. The set $struct_{RCF}$ includes layers of the synthesized network with layer hyperparameters such as the size of the convolution kernel and the number of convolutions. The set $param_{RCF}=\{W_{RCF}, B_{RCF}\}$[6].

The problem of structural tuning of the RCF network is as follows. It is necessary to make structural changes to the existing architecture of the RCF network $struct_{RCF}$. These changes should improve the image edge detection performance compared to the initial RCF network after training the parameters of the resulting network. At the same time, the number of parameters of the resulting network should not increase [2].

2 REVIEW OF THE LITERATURE

To solve the problem of object edge detection on natural images, the deep learning CNN have been widely used recently. In [1] such methods in terms of model structure, technical difficulties, method advantages, and backbone networks are classified into three types. These are codec-based CNN, network reconstruction-based methods, and multi-scale feature fusion-based CNN.

Edge detection methods based on codec were introduced, as they can accept input images of any size and produce output images of the same size [7–9]. Since CNNs reduce the size of an image after convolutions and pooling, their final output in fact does not correspond to every pixel in the original image. Fully convolutional networks are used to retain better low-level edge information, suppress non-edge pixels, and provide detailed edge location [7]. The encoder layers are produce feature maps with semantic information. The decoder layers are transform the low-resolution feature maps which outputted by the encoder back to the size of the input image by pixel classification [1].

Edge detection methods based on network reconstruction integrate various network modules based on deep learning [10–12]. Different modules show

DOI 10.15588/1607-3274-2023-4-12

different advantages for edge detection, so the combination of such modules through network reconstruction is an important way to improve the edge detection results [13, 14].

The edge detection methods based on multi-scale feature fusion combine features of different scales. The higher layer of the network has a larger perceptual field and a strong ability to characterize semantic information while the lower layer of the network has a smaller perceptual field but a strong ability to characterize geometric details. Then combining of the local and global information of the image improves the edge detection performance [1].

This paper is focused on the edge detection methods based on multi-scale feature fusion. The backbone networks of these methods are the HED [15] and RCF [16]. The LPCB [17], BDCN [18], DexiNed [19], DSCD [20], PiDiNet [21] and other networks are proposed based on the HED and RCF networks, as well as by combining with the architectures of other networks to improve the edge detection performance.

The HED algorithm is proposed in [15], where a fully convolutional network is used to resolve ambiguity in edge and object boundary detection. Deeply-supervised side replies were interpolated to initial image size and fused to obtain nested multi-scale features. Thus HED develops rich hierarchical representation automatically directed by deep supervision on side replies [15].

In [17] the HED network is improved to solve the problem of thick contour in edge detection. The obtained LPCB network is based on VGG16 network [22] and uses the fully convolutional network of bottom-up/top-down architecture [23]. Based on image similarity a new loss function is also proposed, which is very effective for classifying unbalanced data. The LPCB network resolves ambiguities in edge detection, and obtains accurate results without post-processing. Compared to the HED network, LPBC uses fewer parameters although the last network shows better edge detection performance.

Inspired by HED [15] and Xception [24] networks, in [19] the deep learning-based edge detector DexiNed is elaborated to generate thin edges without prior training or finetuning process. DexiNed can be regarded as two sub-networks: extremely dense initial network and up-sampling block. This network includes six encoders, and each of them outputs the corresponding feature for generating intermediate edge maps using the up-sampling block, which consists mainly of convolutional and deconvolutional layers. All edge maps generated by the up-sampling block are connected at the end of the network to produce the fusion edges.

In [16] the RCF network for accurate edge detection is designed as a fully convolutional network based on the VGG16 network [22], removing the fully connected layer and the fifth pooling layer. While RCF edge detection the network estimates multi-scale features of the image by convolutional layers which have different perceptual fields and pooling layers. Then fusing the layer level features, all the weight parameters are done by automatic learning. Thus RCF network bases on the pyramid architecture, and combines the underlying feature maps for edge detection [25].

In [18] the BDCN network is proposed to detect edges using multiscale information of images. The basic components of BDCN are ID Blocks. Each ID Block is learned by a bidirectional cascade structure, thus the output of two edge detections is passed separately to the shallow and high-level structures of the network. To enhance the features output from each layer, a Scale Enhancement Module is used. It consists of multiple parallel convolutions with different perceptual field [26], and finally outputting the result to multiple multi-scale feature fusion.

In [20] the proposed DSCD network uses a VGG16 encoder [22] to extract multi-scale and multi-level features. On top of the encoder a super-convolutional module is constructed to directly abstract the high-level features and avoid overfitting problem. The decoder is fused the high-level features and restored them to the original image size. A novel loss function based on the structural similarity of two images is proposed to minimize the distance between predicted and true values. The DSCD network better classifies the background texture and noisy pixels as compared with another codec networks, and generates clear and accurate image edges.

In [21] the elaborated PiDiNet integrates a novel pixel difference convolution into network convolutional layer. As a result this network can easily capture image gradient information conducive to edge detection, while retaining the powerful learning ability of deep CNN to extract information with semantic significance. Then the direct integration of the gradient estimation into the convolution operation results in the better robustness and edge detection accuracy.

As a result of the analysis of the literature, the following was observed. Methods of the first type have a similar encoder-decoder architecture, which has been effectively used to solve a number of applied problems. This architecture assumes a relatively small number of parameters compared to other convolutional networks. However, the inclusion of pooling layers reduces the image edge detection performance. Therefore, it is advisable to use methods of the first type when solving problems that do not require a high edge detection performance, for example, for localizing objects on images.

The methods of the second type are characterized by the use of additional modules that improve the edge detection performance after or together with the use of CNN. Difficulties arise in the development and configuration of these modules, as well as the combining of additional modules with the architecture of the basic CNN. However, with a rational choice of additional modules and architecture of the CNN, it is possible to achieve high image edge detection performance.

Methods of the third type implement the ideas of the two previous types of methods. A set of scale values is defined, which depends on the size, as well as the content

DOI 10.15588/1607-3274-2023-4-12

of the image. For each scale value, the boundaries of objects of a certain size are identified. These methods do not require additional modules to improve the quality of the contour. But there is a need to elaborate an approach to evaluate multi-scale features and to fuse the results of edge detection at different scales.

The analysis of edge detection methods based on multi-scale feature fusion showed that the directions for improving the existing basic HED and RCF architectures and their combination with other architectures are as follows. Firstly, it is the elaboration of the classifier with the best separation of pixel classes by changing the loss function, as in LPCB and DSCD. Secondly, this is the evaluation of features with the better class separation, since the result of edge detection is determined by the shape of pixel clusters and the presence of data outliers. Along the way, integration gradient estimation into the convolution identifying is offered, as in the PiDiNet network, or blocks that take into account information about the edges is added, as in BDCN and DexiNed. In this context in the paper it is proposed to use the structural tuning of the RCF network. This approach allows to select the features of natural images and a way of them fusion with the better separation of edge and background pixels.

3 MATERIALS AND METHODS

In general, the edge detection performance is influenced by a neural network architecture. A simple architecture network may not provide good performance owing to its limited information processing power. A network of complex architecture may have high implementation cost and some of its elements are redundant. At the last time neural architecture search is applied to automate the defining the network structure [27]. Although this technique yields an optimal solution, its computational cost is enormous. Therefore, a different technique is used in the paper. This is a structural tuning of a neural network. Computational costs of a structural tuning are incomparably less, but a higher qualification of the researcher is required, and the resulting solution will be suboptimal.

To tune the network structure, constructive and destructive approaches can be used [28]. The constructive approach starts with a simple architecture network. Hidden layers, nodes, and connections are added to expand the network. The destructive approach starts with a complex architecture network. Hidden layers, nodes, and connections are then deleted to contract the network [28].

In this research author successively applies first a destructive approach and then a constructive approach to structural tuning of the based architecture of the RCF neural network (Fig. 1).

Thus, as a structural tuning of the RCF network, the following is proposed: (1) reducing the number of convolutional layers; (2) reducing the number of convolutions in convolutional layers; (3) removing at each stage the sigmoid activation function with subsequent calculation of the loss function; (4) addition of the batch normalization layers after convolutional layers; (5) including the ReLU activation functions after the added batch normalization layers. Let's explain these steps in more detail.

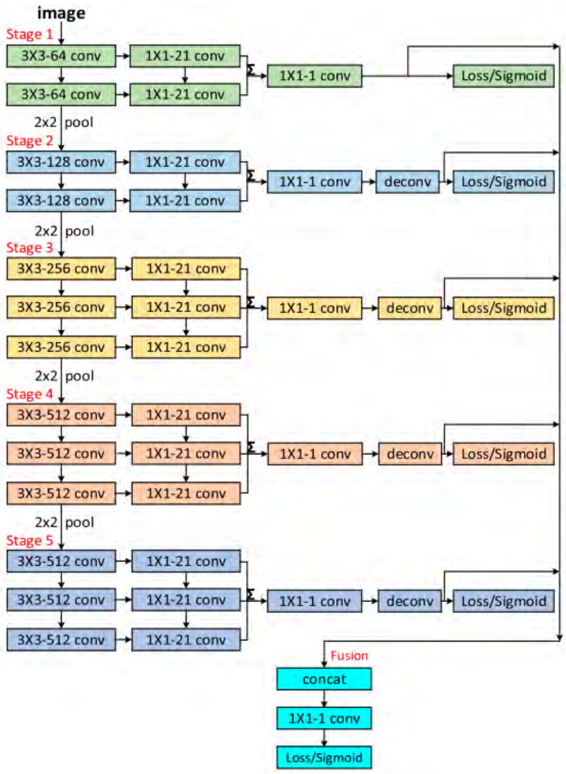


Figure 1 – The RCF architecture [16]

As a result of the destructive approach, the number of convolution layers and the number of convolutions in the remaining convolution layers were reduced (Fig. 2). Such operation is known as thinning of CNN [29].

Deep learning convolutional networks extract the image features in convolutional layers. For each such layer, the number of features to be evaluated is specified by the number of convolutions in the layer. Each feature is identified by a convolution kernel, as well as a kernel shift, and these same parameters determine the image scale on which this feature is extracted.

The redundant or poorly informative features in the resulting set reduce the rate of convergence of the network training, in particular, increases the variance of network parameter estimates. To increase the edge detection performance the noisy features can been discarded, as well as similar features. The last are processed as one feature with a large weight. The feature space dimension can be reduct by reducing the number of convolutions of CNN. This makes it possible to use a smaller training set, reduce training time, and reduce the network overfitting probability. The evaluation of features for edge detection on images influences on the separability of image classes, taking into account the fact that the number of edge pixels differs significantly from the number of background pixels. Therefore, the

DOI 10.15588/1607-3274-2023-4-12

structural tuning of the RCF network in this paper includes altering the number of convolutional layers and the number of convolutions in convolutional layers.

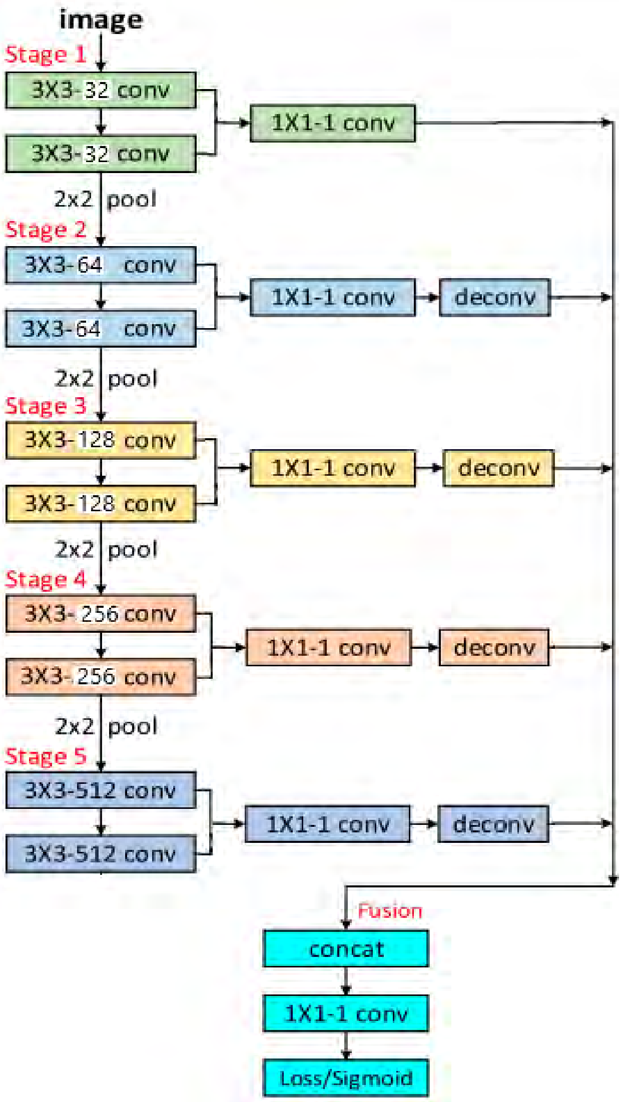


Figure 2 – The result of applying a destructive approach to the structural tuning of the RCF architecture

In addition, the layer containing the sigmoid activation function, followed by the calculation of the loss function, was removed from the architecture of the basic RCF neural network at each stage of processing (Fig. 2). This is due to the following. At each stage of edge detection by the basic RCF network edge probability map was formed as result of applying the sigmoid activation function. Then all obtaining probability edge maps were bilinear interpolated to the size of the original image. Further, for each stage, the value of the loss function was calculated taking into account the result of interpolation of the edge probability map and the ground-truth image. The values of the loss function at different stages were summed during network training.

Removing at each stage the layer containing the sigmoid activation function with subsequent calculation of the loss function avoids additional computational costs for determining the values of the loss function at different scales and introduces redundancy into the multiscale representation of the image. The latter contributes to an increase in the robustness of edge detection on natural images. In such images, it is often necessary to determine the intensity or color edges on the texture background. Natural images contain mostly statistical textures, which can be considered as noise and negatively affect the edge detection performance.

Further in the process of structural tuning of the RCF neural network, a constructive approach was used after applying the destructive approach. Namely, the layers of batch normalization and nonlinearity in the form of the ReLU activation function were added after the convolutional layers.

Batch normalization layer solves vanishing gradient problem. It is known that the error backpropagation algorithm converges faster if the input data is normalized (has zero mean and unit variance) [5, 30]. However, when a signal propagates through a neural network, its mean value and variance can change significantly. To avoid this, the standard normalization of the outputs of the convolutional layer is applied. Nevertheless, normalizing the output of a convolutional layer can change the representation of the data in the next layer. Therefore, two additional parameters γ_i and β_i are adjusted in the learning process along with the rest of the parameters and transform x_{ni} as $y_{ni}=\gamma_i x_{ni}+\beta_i$ [5].

The applying of batch normalization actually corresponds to edge contrasting which improves the edge detection performance. For CNNs, batch normalization reduces training time and reduces the chance of overfitting.

The ReLU activation function returns 0 for a negative argument, and in the case of a positive argument, returns the same. The applying of this function actually corresponds to thresholding in gradient edge detection methods. ReLU sharps object edges on an image because the advantage of this function over the sigmoid is the sparseness of activation (fewer neurons being activated).

The obtained neural network is named RCF-ST. It processes a three-channel image with a size of 320(480 pixels. Therefore, the initial color images were scaled to the specified size and then each image was inputted to the proposed neural network (Fig. 3). It is assumed that deconvolution and transposed convolution are the same operations. The architecture of RCF-ST network for the edge detection is shown in Table 1.

4 EXPERIMENTS

For experimental research of the results of each stage of the structural tuning of the neural network, the edge detection performance was evaluated for natural images from the BSDS500 dataset [31]. The dataset contains a total of 500 images, including 200 training images and 200 test images, and the remaining 100 validation images. The true values of the image edges are also presented on

DOI 10.15588/1607-3274-2023-4-12

ground-truth images which are binary images with contours selected by 5-8 experts (edge maps) [31]. The performance of edge detection was evaluated by comparing edges obtained in the RGB color space using the proposed CNN, with edges labeled by experts.

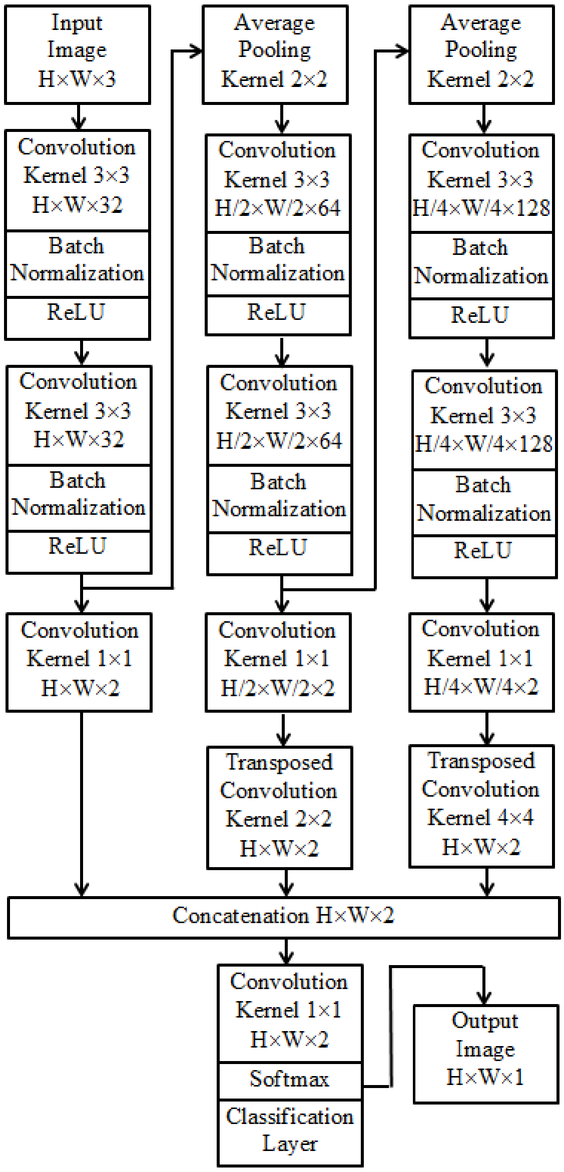


Figure 3 – The proposed RCF-ST network architecture for three stages of processing. It is assumed that deconvolution and transposed convolution are the same operations

The Adam method with an initial learning rate of 0.005 was used to train the RCF-ST network. A cross-entropy loss function was used, for which relative frequencies of the appearance of edge pixels and background pixels were taken into account [5].

To characterize the edge detection results the elements of confusion matrix *TP* and *TN* were used. In addition the *FOM* value [5] was estimated for the edge detection results. The *FOM* value is varied from 0 to 1 and normalized such that *FOM*=1 for a well detected edge.

For evaluation of edge detection results ODS and OIS are the widely used [1]. The ODS and OIS are defined based on F_β-score which is expressed as

$$F_\beta = (1+\beta^2)\, Pr\, Rc\, /\, (\beta^2 Pr + Rc),$$

where $Pr = TP/(TP+FP)$, $Rc = TP/(TP+FN)$. The degree of significance of precision *Pr* and recall *Rc* can be controlled by adjusting the value of β. ODS and OIS indicate different ways of setting the threshold β in this formula. ODS is equal to F_β-score if a fixed threshold β is selected and applied to all images so that the F_β-score on the whole dataset is maximized. OIS is estimated from F_β-score if a different threshold β is selected on each image that maximizes the F_β-score of that image [1].

The experiment was conducted in accordance with the stages of structural tuning of the neural network. The advisability of each of the proposed stages was researched by estimating the edge detection performance using the *TP*, *TN*, *FOM*. The advisability of a structural tuning of the neural network as a whole was estimated by comparing it with methods known from the literature using the ODS and OIS.

First of all, as part of the experiment, the advisability of addition of the batch normalization layers after convolutional layers, and including the ReLU activation functions after the added batch normalization layers is researched. For this the proposed RCF-ST network, and RCF network were used to detect the object edges on natural images [31]. A number of stages of the RCF-ST network, and RCF network is varied from 3 to 5.

Further, the values of the selected indexes of the edge detection were evaluated depending on the number of convolutions in convolutional layers.

At the next stage of the experiment, as an alternative to transposed convolution, bicubic interpolation of image feature maps at different scales was used. The interpolated feature maps (layers 6, 12, 19 from Table 1) were concatenated. Then 1x1 convolutional layer, Softmax activation function, and pixel classification layer (layers 21-24 from Table 1) were applied.

Then the edge detection performance of the proposed RCF-ST network, and methods known from the literature is compared using ODS and OIS.

At the last part of experiment a number of parameters and the processing time of the edge detection on BSDS500 images was estimated for networks with considered architectures.

5 RESULTS

The elements of confusion matrices and *FOM* values for the results of 3, 4, 5 processing stages with the proposed RCF-ST network, and RCF network is shown in Table 2. Values in this table were obtained by averaging the *FOM*, *TP*, and *TN* for the edge detection on BSDS500 natural images.

DOI 10.15588/1607-3274-2023-4-12

Table 1 – The proposed RCF-ST network architecture

Layer number	Type	Comment	Activations	Learnables
1	Image input	320×480×3 image with zero center normalization	320×480×3	–
2	Convolution	32 3×3×3 convolutions with stride [1 1] and same padding	320×480×32	Weights: 3×3×3×32 Bias: 1×1×32
3	Batch normalization and ReLU	Batch normalization with 32 channels and activation function	320×480×32	Offset: 1×1×32 Scale: 1×1×32
4	Convolution	32 3×3×32 convolutions with stride [1 1] and same padding	320×480×32	Weights: 3×3×32×32 Bias: 1×1×32
5	Batch normalization and ReLU	Batch normalization with 32 channels and activation function	320×480×32	Offset: 1×1×32 Scale: 1×1×32
6	Convolution	2 1×1×32 convolutions with stride [1 1] and same padding	320×480×2	Weights: 1×1×32×2 Bias: 1×1×2
7	Average pooling	2×2 average pooling with stride [2 2] and padding [0 0 0 0]	160×240×32	–
8	Convolution	64 3×3×32 convolutions with stride [1 1] and same padding	160×240×64	Weights: 3×3×32×64 Bias: 1×1×64
9	Batch normalization and ReLU	Batch normalization with 64 channels and activation function	160×240×64	Offset: 1×1×64 Scale: 1×1×64
10	Convolution	64 3×3×64 convolutions with stride [1 1] and same padding	160×240×64	Weights: 3×3×64×64 Bias: 1×1×64
11	Batch normalization and ReLU	Batch normalization with 64 channels and activation function	160×240×64	Offset: 1×1×64 Scale: 1×1×64
12	Convolution	2 1×1×64 convolutions with stride [1 1] and same padding	160×240×2	Weights: 1×1×64×2 Bias: 1×1×2
13	Transposed convolution	2 2×2×2 transposed convolutions with stride [2 2] and output cropping [0 0]	320×480×2	Weights: 2×2×2×2 Bias: 1×1×2
14	Average pooling	2×2 average pooling with stride [2 2] and padding [0 0 0 0]	80×120×64	–
15	Convolution	128 3×3×64 convolutions with stride [1 1] and same padding	80×120×128	Weights: 3×3×64×128 Bias: 1×1×128
16	Batch normalization and ReLU	Batch normalization with 128 channels and activation function	80×120×128	Offset: 1×1×128 Scale: 1×1×128
17	Convolution	128 3×3×128 convolutions with stride [1 1] and same padding	80×120×128	Weights: 3×3×128×128 Bias: 1×1×128
18	Batch normalization and ReLU	Batch normalization with 128 channels and activation function	80×120×128	Offset: 1×1×128 Scale: 1×1×128
19	Convolution	2 1×1×128 convolutions with stride [1 1] and same padding	80×120×2	Weights: 1×1×128×2 Bias: 1×1×2
20	Transposed convolution	2 4×4×2 transposed convolutions with stride [4 4] and output cropping [0 0]	320×480×2	Weights: 4×4×2×2 Bias: 1×1×2
21	Concatenation	Concatenate the resulting feature maps from the output of the layers 6, 13, 20	320×480×6	–
22	Convolution	2 1×1×6 convolutions with stride [1 1] and same padding	320×480×2	Weights: 1×1×6×2 Bias: 1×1×2
23	Softmax	Activation function	320×480×2	–
24	Pixel classification	Class weighted cross-entropy loss with classes “edge” and “background”	–	–

Table 2 – The values of *TP*, *TN*, and *FOM* for the results of edge detection with the RCF-ST network (with batch normalization), and RCF network (without batch normalization)

Stages number	*TP*	*TN*	*FOM*
With batch normalization			
3	97.374	92.304	0.492
4	99.368	91.621	0.512
5	98.691	91.087	0.508
Without batch normalization			
3	84.892	84.214	0.337
4	82.369	78.315	0.301
5	77.270	74.454	0.295

Fig. 4 shows the initial BSDS500 images, the ground-truth images, edge maps, obtained by RCF-ST network with transposed convolution, and edge maps, obtained by RCF-ST network with interpolation. The multi-scale representation of BSDS500 images obtained by RCF-ST network is shown on Fig. 5. It can be seen from the Fig. 4 that edge detection by the RCF-ST network with transposed convolution is characterized by high performance.

In Table 3 the results of edge detection by the RCF-ST network with three processing stages is shown. The number of convolutions in convolutional layers is varied. The transposed convolution is applied for up-sampling of feature maps of initial image on different scales.

DOI 10.15588/1607-3274-2023-4-12

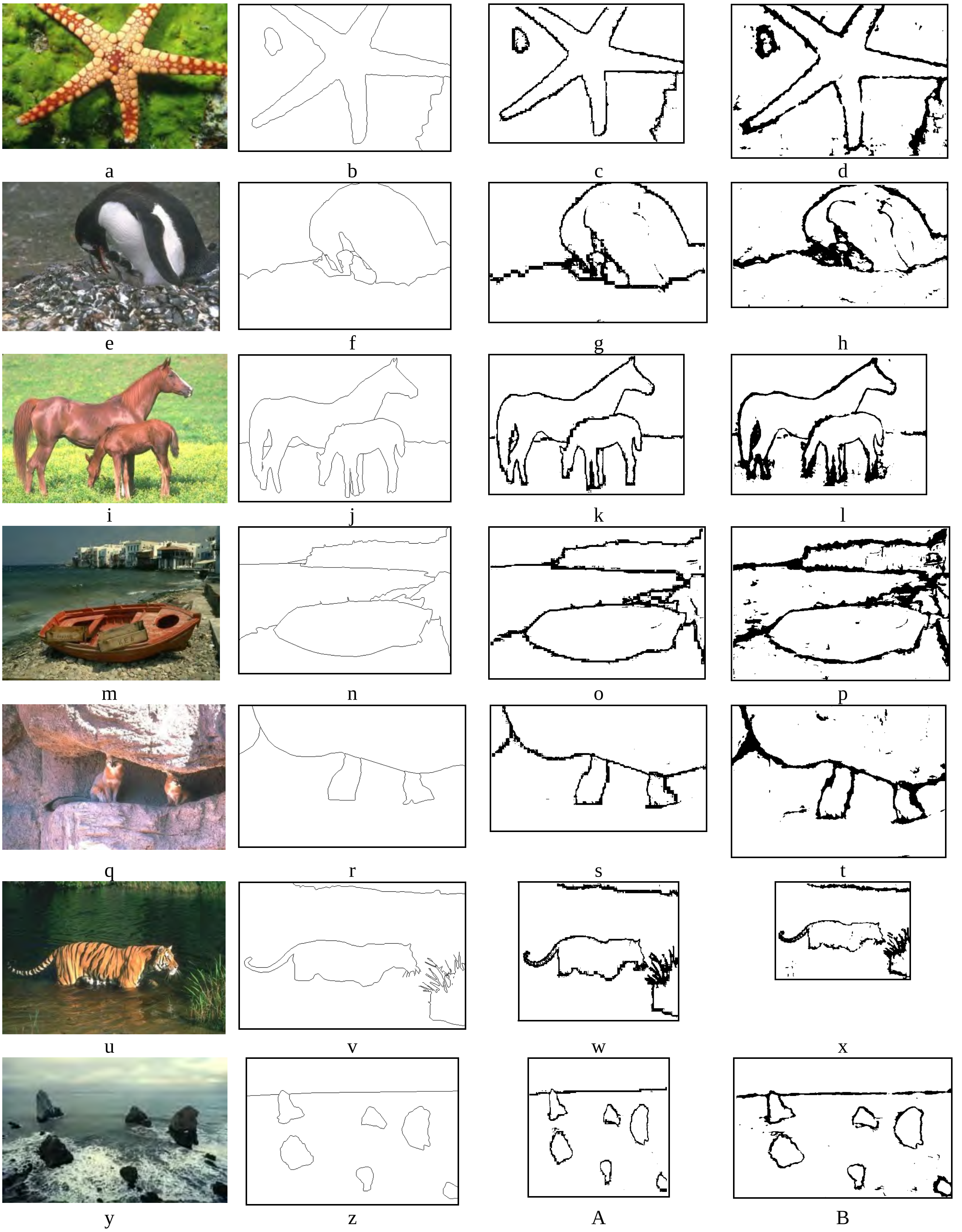


Figure 4 – The edge detection results by proposed RCF-ST network:
a, e, i, m, q, u, y – the initial BSDS500 images; b, f, j, n, r, v, z – the ground-truth images; c, g, k, o, s, w, A – edge maps, obtained by RCF-ST network with transposed convolution; d, h, l, p, t, x, B– edge maps, obtained by RCF-ST network with interpolation

DOI 10.15588/1607-3274-2023-4-12

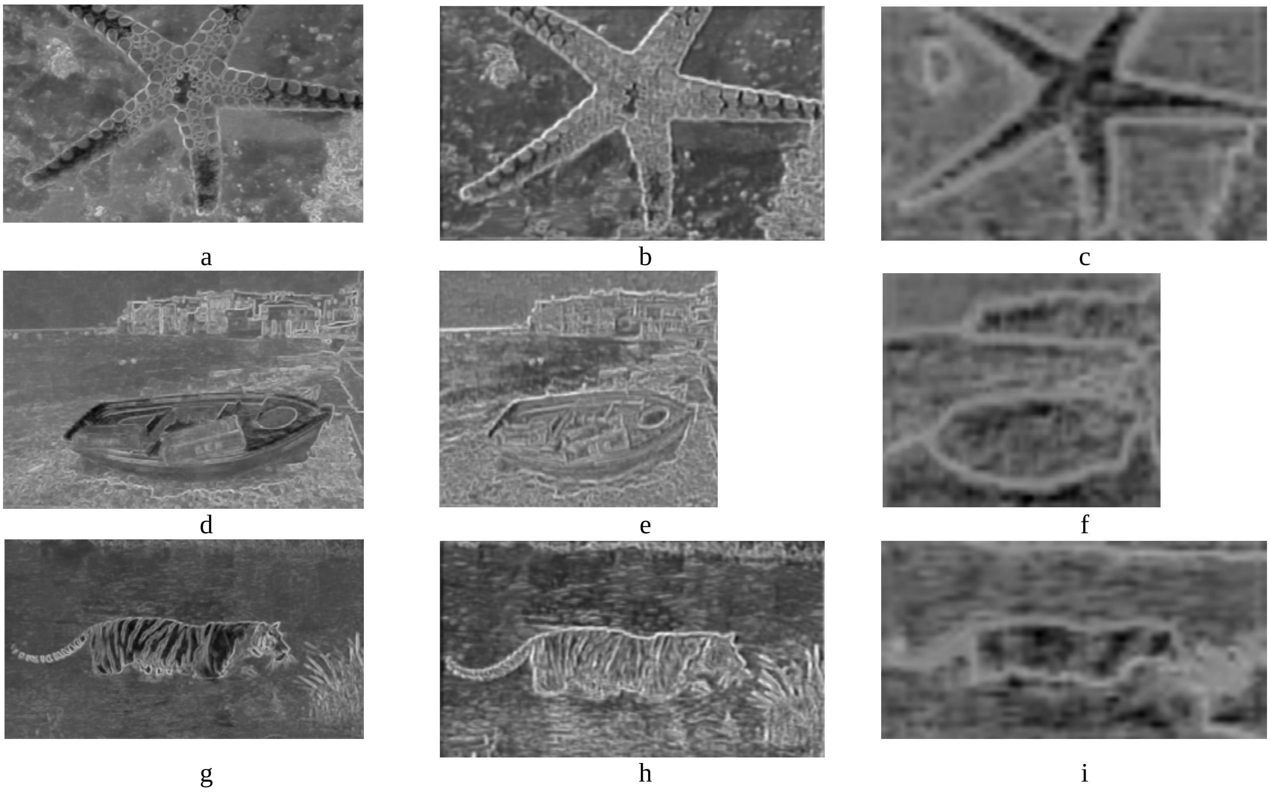


Figure 5 – The multi-scale representation of BSDS500 images from Fig. 4 obtained by RCF-ST network:
a, d, g – the fine scale images; b, e, h – the middle scale images; c, f, i – the coarse scale images

Table 3 – The results of edge detection by the RCF-ST network with the different number of convolutions in convolutional layers (three processing stages)

The number of convolutions in convolutional layers	*TP*	*TN*	*FOM*
8, 16, 32	96.423	86.705	0.343
16, 32, 64	98.062	91.219	0.445
32, 64, 128	97.374	92.304	0.492

The results of edge detection performance by the RCF-ST network with bicubic interpolation of image feature maps at different stages are presented in Table 4.

Table 4 – The values of *TP*, *TN*, and *FOM* for the results of edge detection with the RCF-ST network with interpolation

Stages number	*TP*	*TN*	*FOM*
With batch normalization			
3	96.383	87.292	0.344
4	95.149	87.026	0.377
5	95.926	86.299	0.369
Without batch normalization			
3	80.036	75.616	0.313
4	79.693	79.211	0.308
5	72.408	71.305	0.311

For comparison, the results of edge detection by the RCF network with similar architecture and bicubic interpolation of image feature maps at different stages is shown as well. The number of network processing stages is varied.

The edge detection performance of the BSDS500 images by the proposed RCF-ST network, and methods known from the literature are given in Table 5.

Table 5 – The ODS and OIS obtained from the BSDS500 images by the proposed RCF-ST network, and methods known from the literature [1, 21]

Reference, publication year, network name	ODS	OIS
[15], 2017, HED	0.788	0.808
[16], 2017, RCF	0.808	0.823
[17], 2018, LPCB	0.808	0.824
[18], 2022, BDCN	0.820	0.838
[19], 2020, DexiNed	0.831	0.845
[20], 2020, DSCD	0.826	0.857
[21], 2021, PiDiNet	0.807	0.823
Proposed RCF-ST network, 3 stages	0.823	0.853
Proposed RCF-ST network, 4 stages	0.887	0.894
Proposed RCF-ST network, 5 stages	0.862	0.872

For comparison, the edge detection performance of the Multicue dataset images by the methods known from the literature are given in Table 6.

Table 6 – The ODS and OIS obtained from Multicue dataset [32] images by the methods known from the literature [21]

Reference, publication year, network name	ODS	OIS
[15], 2017, HED	0.851	0.864
[16], 2017, RCF	0.857	0.862
[18], 2022, BDCN	0.891	0.898
[21], 2021, PiDiNet	0.858	0.863

DOI 10.15588/1607-3274-2023-4-12

6 DISCUSSIONS

Analysis of the indexes given in Table 2 showed that the using of batch normalization improves the edge detection performance. Specifically, *TP*, *TN*, and *FOM* have increased by 14–27%, 9–23%, 48–76% respectively. For reseached images more often background pixels were incorrectly assigned to the image edge.

Analysing Table 3 it should be noted the follow. If the number of convolutions in convolutional layers is decreased by two times then *FOM* are less by 9–32%. *TP* and *TN* mainly differed within the statistical error.

In addition, it is preferable to use average pooling than max pooling. The latter can enhance *TP* and *TN* by 0.5–1% depending on a number of processing stages.

The using of the interpolation instead of transposed convoluton of image feature maps at different processing stages is not advisability because the edge detection performance reduces. Specifically, *TP* is less by up to 4%, *TN* is less by 4–5%, *FOM* is less by up to 33%.

Analysis of the edge detection performance of the proposed RCF-ST network and the known methods [15–21] showed the following (Table 5). The ODS and OIS of the proposed RCF-ST network exceeds the known methods by 9–10% for BSDS500 images. The data given in Table 6 show that similar values of ODS and OIS are achievable by methods known from the literature, but on Multicue dataset [32].

A comparison of a number of parameters was made for the proposed RCF-ST network and RCF network with three processing stages. The basic RCF network with the architecture on Fig. 1 contains 1,758,600 parameters. The proposed RCF-ST network with the architecture in Table 1 contains 288,456 parameters. Thus, the number of parameters of the proposed RCF-ST network is 6 times less than the number of parameters of the basic RCF network, provided that the number of processing stages is equal.

A comparison of processing time was made for the edge detection on natural images by the proposed RCF-ST network with different number of processing stages. The researched natural images were cutted to a size of 320×480 pixels. Then the processing time of the RCF-ST network was calculated on average per image when training the network using the Adam method. It was 0.708–0.733; 0.921–0.982; 1.505–1.678 seconds per image when 3, 4, 5 processing stages are used correspondingly. The number of convolutions in convolutional layers was chosen as 32, 64, 128, 256, 512. The RCF network with similar architecture, that is, with the architecture as in Fig. 3, only without the batch normalization layers is also considered. It's average processing time for one image with 3, 4, 5 processing stages was 0.563–0.571; 0.640–0.653; 0.708–0.771 seconds correspondingly when trained by the Adam method. The research was performed using an Intel Core i5-7400 processor, 3 GHz CPU, 16GB memory, Windows 10 operating system, 64 bit. Thus, the proposed RCF-ST network requires on average 26–28, 44–50, 113–118 percents more time to process one image with 3, 4, 5 stages correspondingly than the RCF network. The number of training epochs of the RCF-ST network and RCF network for this edge detection problem is similar on average.

CONCLUSIONS

The actual scientific and applied problem of a structural tuning of a pre-synthesized neural network for edge detection on natural images has been solved.

The scientific novelty is the proposed technique of structural tuning of a deep learning neural network, which uses a sequentially destructive and constructive approach. According to the proposed technique, the network thinning and then removing at each stage the sigmoid activation function with subsequent calculation of the loss function were first performed as part of the destructive approach. Then, as part of a constructive approach, the batch normalization and ReLU layers are added after convolutional layers. As a result of applying this technique, the obtained RCF-ST network allows to improve the performance of edge detection on natural images. RCF-ST network is characterized by a significantly fewer parameters compared to the RCF network, which makes it possible to reduce the resource consumption of the network. Besides, RCF-ST network ensures the enhancing of the robustness of edge detection on texture background.

The practical significance of obtained results is that the software realizing the proposed RCF-ST network is developed, as well as experiments to research its edge detection performance are conducted. The experimental results allow to recommend the proposed RCF-ST for use in practice, as well as to determine effective conditions for the application of this network. The structural tuning technique may be used for informed design of the neural network architectures for other artificial intelligence problems.

Prospects for further research are to elaborate the postprocessing module which will thin and smooth the contours detected by the proposed RCF-ST network.

ACKNOWLEDGEMENTS

The author 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. Sun R., Lei T., Chen Q. et. al. Survey of image edge detection, *Frontiers of Signal Processing*, 2022, Vol. 2, article 826967. DOI: 10.3389/frsip.2022.826967
2. 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

DOI 10.15588/1607-3274-2023-4-12

3. D'souza R. N., Huang P. Y., Yeh F. C. Structural analysis and optimization of convolutional neural networks with a small sample size, *Scientific Reports*, 2020, № 10, pp. 1–13. DOI: 10.1038/s41598-020-57866-2
4. Nowakowski G., Dorogyy Y., Doroga-Ivaniuk O. Neural network structure optimization algorithm, *Journal of Automation, Mobile Robotics and Intelligent Systems*, 2018, № 12, pp. 5–13. DOI: 10.14313/JAMRIS_1-2018/1
5. Polyakova M. V. Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems, *Radio Electronics, Computer Science, Control*, 2023, № 1, pp. 51–61. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-5
6. Matychenko A. D., Polyakova M. V. The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space, *Herald of Advanced Information Technology*, 2023, Vol. 6, № 2, pp. 115–127. DOI: 10.15276/hait.06.2023.7
7. Yang J., Price B., Cohen S., Lee H., Yang M.-H. Object contour detection with a fully convolutional encoder-decoder network, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June 2016 : proceedings*. IEEE, 2016, pp. 193–202. DOI: 10.1109/cvpr.2016.28
8. Yu Z., Feng C., Liu M. Y., Ramalingam S. Casenet: Deep category-aware semantic edge detection, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings.* IEEE, 2017, pp. 5964–5973. DOI: 10.1109/cvpr.2017.191
9. Pu M., Huang Y., Guan Q., Ling H. RINDNet: Edge detection for discontinuity in reflectance, illumination, normal and depth, *Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021 : proceedings*. IEEE, 2021, pp. 6879–6888. DOI: 10.1109/iccv48922.2021.00680
10. Maninis K.-K., Pont-Tuset J., Arbeláez P., Van Gool L. Convolutional oriented boundaries, *Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)*. Springer, Cham, 2016, Vol. 9905, pp. 580–596. DOI: 10.1007/978-3-319-46448-0_35
11. Arbeláez P., Maire M., Fowlkes C., Malik J. Contour detection and hierarchical image segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010, Vol. 33, № 5, pp. 898–916. DOI: 10.1109/TPAMI.2010.161
12. Pont-Tuset J., Arbelaez P., Barron J., Marques F., Malik J. Multiscale combinatorial grouping for image segmentation and object proposal generation, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016, Vol. 39, № 1, pp. 128–140. DOI: 10.1109/TPAMI.2016.2537320
13. Xu D., Ouyang W., Alameda-Pineda X. et al. Learning deep structured multi-scale features using attention-gated CRFs for contour predict crisp boundaries, *Neural Information Processing Systems: 31st International Conference, NIPS'17, Long Beach, California, USA, 4–9 December, 2017 : proceedings*. Curran Associates Inc., 2017, pp. 3964–3973.
14. Mnih V., Heess N., Graves A., Kavukcuoglu K. Recurrent models of visual attention, *Neural Information Processing Systems: 27th International Conference, NIPS'14, Montreal, Canada, 8–13 December 2014 : proceedings*. MIT Press, 2014, Vol. 2, pp. 2204–2212.
15. Xie S., Tu Z. Holistically-nested edge detection, *International Journal of Computer Vision*, 2017, Vol. 125, № 5, pp. 1–16. DOI: 10.1007/s11263-017-1004-z
16. Liu Y., Cheng M. M., Hu X., Wang K., Bai X. Richer convolutional features for edge detection, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings*. IEEE, 2017, pp. 5872–5881. DOI: 10.1109/CVPR.2017.622
17. Deng R., Shen C., Liu S., Wang H., Liu X. Learning to predict crisp boundaries, *Computer Vision: 15th European Conference, ECCV, Munich, Germany, 8–14 September, 2018 : proceedings*. IEEE, 2018, part VI, pp. 562–578. DOI: 10.1007/978-3-030-01231-1_35
18. He J., Zhang S., Yang M., Chan Y., Huang T. BDCN: Bi-directional cascade network for all perceptual edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022, Vol. 44, № 1, pp. 100–113. DOI: 10.1109/TPAMI.2020.3007074
19. Poma X. S., Riba E., Sappa A. Dense extreme inception network: towards a robust CNN model for edge detection, *Applications of Computer Vision: IEEE/CVF Winter Conference, WACV, Snowmass Village, CO, USA, 1–5 March 2020 : proceedings*. IEEE, 2020, pp. 1912–1921. DOI: 10.1109/WACV45572.2020.9093290
20. Deng R., Liu S. Deep structural contour detection, *Multimedia: 28th ACM International Conference, MM'20, Virtual Event, Seattle, WA, USA, October 12–16, 2020 : proceedings*. ACM, 2020, pp. 304–312. DOI: 10.1145/3394171.3413750
21. Su Z., Liu W., Yu Z. et al. Pixel difference networks for efficient edge detection, *Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021: proceedings*. IEEE, 2021, pp. 5117–5127. DOI: 10.1109/ICCV48922.2021.00507
22. Simonyan K., Zisserman A. Very deep convolutional networks for large-scale image recognition, *Learning Representations: 3rd International Conference, ICLR 2015, San Diego, CA, USA, 7–9 May 2015 : proceedings* [Electronic resource]. Access mode: https://arxiv.org/pdf/1409.1556. DOI: 10.48550/arXiv. 1409.1556
23. Pinheiro P. O., Lin T.-Y., Collobert R., Dollár P. Learning to refine object segments, *Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds)*. Springer, Cham, 2016, Vol. 9905, pp. 75–91. DOI: 10.1007/978-3-319- 46448-0_5
24. Chollet F. Xception: Deep learning with depthwise separable convolutions, *Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings*. IEEE, 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195
25. Dai J., Li Y., He K., Sun J. R-fcn: Object detection via region-based fully convolutional networks, *Neural Information Processing Systems: 30th International Conference, NIPS'16, Barcelona, Spain, 4–8 December 2016 : proceedings*. Red Hook, NY, ACM, 2016, pp. 379–387.
26. Chen L. C., Papandreou G., Kokkinos I., Murphy K., Yuille A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected Crfs, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2017, Vol. 40, № 4, pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184
27. Elsken T., Metzen J. H., Hutter F. Neural architecture search: a survey, *Journal of Machine Learning Research*, 2019, Vol. 20, № 55, pp. 1–21.
28. Leung F. H. F., Lam H. K., Ling S. H., Tam P. K. S. Tuning of the structure and parameters of neural network using an improved genetic algorithm, *IEEE Transactions on Neural*

DOI 10.15588/1607-3274-2023-4-12

Networks, 2003, Vol. 14, № 1, pp. 79–88. DOI: 10.1109/TNN.2002.804317
29. Xie W., Nagrani A., Chung J. S., Zisserman A. Utterance-level aggregation for speaker recognition in the wild, *Acoustics, Speech and Signal Processing: IEEE International Conference, ICASSP, Brighton, Great Britain, 12–17 May 2019 : proceedings*. IEEE, 2019, pp. 5791–5795. DOI: 10.1109/ICASSP.2019.8683120
30. Polyakova M. V., Krylov V. N. Data normalization methods to improve the quality of classification in the breast cancer diagnostic system, *Applied Aspects of Information Technology*, 2022, Vol. 5, № 1, pp. 55–63. DOI: 10.15276/aait.05.2022.5
31. The Berkeley Segmentation Dataset and Benchmark Web site (2019) [Electronic Resource]. Access mode: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.
32. The multi-cue boundary detection dataset. Video collection [Electronic Resource]. Access mode: http://serre-lab.clps.brown.edu/resources-static/multicue-dataset.tar.bz2

Received 27.08.2023.
Accepted 02.11.2023.

УДК 004.93

RCF-ST: СТРУКТУРНЕ НАЛАШТУВАННЯ НЕЙРОННОЇ МЕРЕЖІ З НАСИЧЕНІШИМИ ЗГОРТКОВИМИ ОЗНАКАМИ ДЛЯ ВИДІЛЕННЯ КОНТУРІВ НА ЗОБРАЖЕННЯХ РЕАЛЬНИХ СЦЕН

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 автоматизації виділення контурів на зображеннях реальних сцен в інтелектуальних системах. Предметом дослідження є згорткові нейронні мережі глибокого навчання для виділення контурів на зображеннях реальних сцен. Метою дослідження є підвищення якості виділення контурів на зображеннях реальних сцен шляхом структурного налаштування архітектури нейронної мережі з насиченішими згортковими ознаками.

Метод. Для автоматизації проектування архітектури нейронної мережі, що впливає на якість виділення контурів зображень, в роботі застосовано структурне налаштування. Обчислювальні витрати на структурне налаштування незрівнянно менші порівняно з пошуком нейронної архітектури, але потрібна більш висока кваліфікація дослідника, і отримане рішення буде субоптимальним. У цьому дослідженні послідовно застосовано спочатку деструктивний, а потім конструктивний підхід до структурного налаштування архітектури базової нейронної мережі RCF. Згідно конструктивному підходу для розширення мережі простої архітектури додаються приховані шари, вузли та з'єднання. Деструктивний підхід з мережі складної архітектури видаляє приховані шари, вузли та з'єднання щоб спростити мережу. Структурне налаштування нейронної мережі RCF з насиченішими згортковими ознаками включає: (1) зменшення кількості згорткових шарів; (2) зменшення кількості згорток у згорткових шарах; (3) видалення на кожному етапі сигмоїдної функції активації з подальшим обчисленням функції втрат; (4) додавання шарів пакетної нормалізації після згорткових шарів; (5) додавання функції активації ReLU після шарів пакетної нормалізації. Отримана нейронна мережа RCF-ST потребує масштабування початкових кольорових зображень до заданого розміру перед поданням на вхід мережі. Доцільність кожного із запропонованих етапів структурного налаштування мережі досліджувано шляхом оцінки якості виділення контурів за допомогою елементів матриці помилок та критерія Претта. Доцільність структурного налаштування нейронної мережі в цілому оцінено шляхом її порівняння з відомими з літератури методами за допомогою Optimal Dataset Scale та Optimal Image Scale.

Результати. Запропоновану згорткову нейронну мережу програмно реалізовано та досліджено для розв'язання завдання виділення контурів на зображеннях реальних сцен. Запропоновані етапи структурного налаштування можна використовувати під час обґрунтованого проектування архітектури нейронної мережі для розв'язання інших завдань штучного інтелекту.

Висновки. Отримана мережа RCF-ST дозволяє підвищити якість виділення контурів на зображеннях. Мережа RCF-ST характеризується значно меншою кількістю параметрів у порівнянні з мережею RCF, що дозволяє знизити ресурсоспоживання мережі. Крім того, мережа RCF-ST забезпечує підвищення завадостійкості виділення контурів на фоні текстури.

КЛЮЧОВІ СЛОВА: зображення реальних сцен, виділення контурів, згорткова мережа, насиченіші згорткові ознаки, структурне налаштування, пакетна нормалізація.

ЛІТЕРАТУРА

1. Survey of image edge detection / [R. Sun, T. Lei, Q. Chen et. al.] // Frontiers of Signal Processing. – 2022. – Vol. 2. – Article 826967. DOI: 10.3389/frsip.2022.826967
2. 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
3. D'souza R. N. Structural analysis and optimization of convolutional neural networks with a small sample size / R. N. D'souza, P. Y. Huang, F. C. Yeh // Scientific Reports. – 2020. – № 10. – P. 1–13. DOI: 10.1038/s41598-020-57866-2
4. Nowakowski G. Neural network structure optimization algorithm / G. Nowakowski, Y. Dorogyy, O. Doroga-Ivaniuk // Journal of Automation, Mobile Robotics and Intelligent Systems. – 2018. – № 12. – P. 5–13. DOI: 10.14313/JAMRIS_1-2018/1
5. Polyakova M. V. Image segmentation with a convolutional neural network without pooling layers in dermatological disease diagnostics systems / M. V. Polyakova // Radio Electronics, Computer Science, Control. – 2023. – № 1. – P. 51–61. DOI: http://doi.org/ 10.15588/1607-3274-2023-1-5
6. Matychenko A. D. The structural tuning of the convolutional neural network for speaker identification in mel frequency cepstrum coefficients space / A. D. Matychenko, M. V. Polyakova // Herald of Advanced Information Technology. – 2023. – Vol. 6, № 2. – P. 115–127. DOI: 10.15276/hait.06.2023.7
7. Object contour detection with a fully convolutional encoder-decoder network / [J. Yang, B. Price, S. Cohen et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Las Vegas, NV, USA, 27–30 June 2016

DOI 10.15588/1607-3274-2023-4-12

: proceedings. – IEEE, 2016. – P. 193–202. DOI: 10.1109/cvpr.2016.28
8. Casenet: Deep category-aware semantic edge detection / [Z. Yu, C. Feng, M. Y. Liu, S. Ramalingam] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 5964–5973. DOI: 10.1109/cvpr.2017.191
9. RINDNet: Edge detection for discontinuity in reflectance, illumination, normal and depth / [M. Pu, Y. Huang, Q. Guan, H. Ling] // Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021 : proceedings. – IEEE, 2021. – P. 6879–6888. DOI: 10.1109/iccv48922.2021.00680
10. Convolutional oriented boundaries / [K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, L. Van Gool] // Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds). – Springer, Cham, 2016. – Vol. 9905. – P. 580–596. DOI: 10.1007/978-3-319-46448-0_35
11. Contour detection and hierarchical image segmentation/ [P. Arbeláez, M. Maire, C. Fowlkes, J. Malik] // IEEE Trans. Pattern Anal. Mach. Intell. – 2010. – Vol. 33, № 5. – P. 898–916. DOI: 10.1109/TPAMI.2010.161
12. Multiscale combinatorial grouping for image segmentation and object proposal generation / [J. Pont-Tuset, P. Arbelaez, J. Barron, F. Marques, J. Malik] // IEEE Trans. Pattern Anal. Mach. Intell. – 2016. – Vol. 39, № 1. – P. 128–140. DOI: 10.1109/TPAMI.2016.2537320
13. Learning deep structured multi-scale features using attention-gated CRFs for contour predict crisp boundaries / [D. Xu, W. Ouyang, X. Alameda-Pineda et al.] // Neural Information Processing Systems: 31st International Conference, NIPS'17, Long Beach, California, USA, 4–9 December, 2017 : proceedings. – Curran Associates Inc., 2017. – P. 3964–3973.
14. Recurrent models of visual attention / [V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu] // Neural Information Processing Systems: 27th International Conference, NIPS'14, Montreal, Canada, 8–13 December 2014 : proceedings. – MIT Press, 2014. – Vol. 2. – P. 2204–2212.
15. Xie S. Holistically-nested edge detection / S. Xie, Z. Tu // International Journal of Computer Vision. – 2017. – Vol. 125, № 5. – P. 1–16. DOI: 10.1007/s11263-017-1004-z
16. Richer convolutional features for edge detection / [Y. Liu, M. M. Cheng, X. Hu et al.] // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 5872–5881. DOI: 10.1109/CVPR.2017.622
17. Learning to predict crisp boundaries / [R. Deng, C. Shen, S. Liu et al.] // Computer Vision: 15th European Conference, ECCV, Munich, Germany, 8–14 September, 2018 : proceedings. – IEEE, 2018. – Part VI. – P. 562–578. DOI: 10.1007/978-3-030-01231-1_35
18. BDCN: Bi-directional cascade network for all perceptual edge detection / [J. He, S. Zhang, M. Yang et al.] // IEEE Trans. Pattern Anal. Mach. Intell. – 2022. – Vol. 44, № 1. – P. 100–113. DOI: 10.1109/TPAMI.2020.3007074
19. Poma X. S. Dense extreme inception network: towards a robust CNN model for edge detection / X. S. Poma, E. Riba, A. Sappa // Applications of Computer Vision: IEEE/CVF Winter Conference, WACV, Snowmass Village, CO, USA, 1–5 March 2020 : proceedings. – IEEE, 2020. – P. 1912–1921. DOI: 10.1109/WACV45572.2020.9093290
20. Deng R. Deep structural contour detection / R. Deng, S. Liu // Multimedia: 28th ACM International Conference, MM'20, Virtual Event, Seattle, WA, USA, October 12–16, 2020 : proceedings. – ACM, 2020. – P. 304–312. DOI: 10.1145/3394171.3413750
21. Pixel difference networks for efficient edge detection / [Z. Su, W. Liu, Z. Yu et al.] // Computer Vision: IEEE/CVF International Conference, ICCV, online, 11–17 October, 2021: proceedings. – IEEE, 2021. – P. 5117–5127. DOI: 10.1109/ICCV48922.2021.00507
22. Simonyan K. Very deep convolutional networks for large-scale image recognition / K. Simonyan, A. Zisserman // Learning Representations: 3rd International Conference, ICLR 2015, San Diego, CA, USA, 7–9 May 2015 : proceedings [Electronic resource]. – Access mode: https://arxiv.org/pdf/1409.1556. DOI: 10.48550/arXiv.1409.1556
23. Learning to refine object segments / [P. O. Pinheiro, T.-Y. Lin, R. Collobert, P. Dollár] // Computer Vision – ECCV 2016. Lecture Notes in Computer Science / Leibe, B., Matas, J., Sebe, N., Welling, M. (eds). – Springer, Cham, 2016. – Vol. 9905. – P. 75–91. DOI: 10.1007/978-3-319-46448-0_5
24. Chollet F. Xception: Deep learning with depthwise separable convolutions / F. Chollet // Computer Vision and Pattern Recognition: IEEE Conference, CVPR, Honolulu, HI, USA, 21–26 July 2017 : proceedings. – IEEE, 2017. – P. 1800–1807. DOI: 10.1109/CVPR.2017.195
25. R-fcn: Object detection via region-based fully convolutional networks / [J. Dai, Y. Li, K. He, J. Sun] // Neural Information Processing Systems: 30th International Conference, NIPS'16, Barcelona, Spain, 4–8 December 2016 : proceedings. – Red Hook, NY, ACM, 2016. – P. 379–387.
26. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected Crfs / [L. C. Chen, G. Papandreou, I. Kokkinos et al.] // IEEE Trans. Pattern Anal. Mach. Intell. – 2017. – Vol. 40, № 4. – P. 834–848. DOI: 10.1109/TPAMI.2017.2699184
27. Elsken T. Neural architecture search: a survey / T. Elsken, J. H. Metzen, F. Hutter // Journal of Machine Learning Research. – 2019. – Vol. 20, № 55. – P. 1–21.
28. Tuning of the structure and parameters of neural network using an improved genetic algorithm / [F. H. F. Leung, H. K. Lam, S. H. Ling, P. K. S. Tam] //IEEE Transactions on Neural Networks. – 2003. – Vol. 14, № 1. – P. 79–88. DOI: 10.1109/TNN.2002.804317
29. Utterance-level aggregation for speaker recognition in the wild / [W. Xie, A. Nagrani, J. S. Chung, A. Zisserman] // Acoustics, Speech and Signal Processing: IEEE International Conference, ICASSP, Brighton, Great Britain, 12–17 May 2019 : proceedings. – IEEE, 2019. – P. 5791–5795. DOI: 10.1109/ICASSP.2019.8683120
30. Polyakova M. V. Data normalization methods to improve the quality of classification in the breast cancer diagnostic system / M. V. Polyakova, V. N. Krylov // Applied Aspects of Information Technology. – 2022. – Vol. 5, № 1. – P. 55–63. DOI: 10.15276/aait.05.2022.5
31. The Berkeley Segmentation Dataset and Benchmark Web site (2019) [Electronic Resource]. – Access mode: https://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds.
32. The multi-cue boundary detection dataset. Video collection [Electronic Resource]. – Access mode: http://serre-lab.clps.brown.edu/resources-static/multicue-dataset.tar.bz2

DOI 10.15588/1607-3274-2023-4-12

ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

PROGRESSIVE INFORMATION TECHNOLOGIES

UDC 004.94 : 004.2

SYNTHESIS OF VHDL-MODEL OF A FINITE STATE MACHINE WITH DATAPATH OF TRANSITIONS

Barkalov A. A. – Dr. Sc., Professor, Professor of Institute of Computer Science and Electronics, University of Zielona Gora, Zielona Gora, Poland.

Titarenko L. A. – Dr. Sc., Professor, Professor of Institute of Computer Science and Electronics, University of Zielona Gora, Zielona Gora, Poland.

Babakov R. M. – Dr. Sc., Associate Professor, Professor of Department of Information Technologies, Vasyl Stus Donetsk National University, Vinnytsia, Ukraine.

ABSTRACT

Context. The problem of building a program model of a finite state machine with datapath of transitions using VHDL language is considered. The model synthesis process is identified with the synthesis of this type of finite state machine, since the built model can be used both for the analysis of the device's behavior and for the synthesis of its logic circuit in the FPGA basis. The object of the research is the automated synthesis of the logic circuit of the finite state machine with the datapath of transitions, based on the results of which numerical characteristics of the hardware expenses for the implementation of the state machine circuit can be obtained. This makes it possible to evaluate the effectiveness of using this structure of the finite state machine when implementing a given control algorithm.

Objective. Development and research of a VHDL model of a finite state machine with datapath of transitions for the analysis of the behavior of the state machine and the quantitative assessment of hardware expenses in its logic circuit.

Method. The research is based on the structural diagram of a finite state machine with datapath of transitions. The synthesis of individual blocks of the structure of the state machine is carried out according to a certain procedure by the given graph-scheme of the con-trol algorithm. It is proposed to present the result of the synthesis in the form of a VHDL description based on the fixed values of the states codes of the state machine. The process of synthesizing the datapath of transitions, the block of formation of codes of transitions operations and the block of formation of microoperations is demonstrated. VHDL description of that blocks is carried out in a synthesizable style, which allows synthesis of the logic circuit of the finite state machine based on FPGA with the help of modern CAD and obtaining numerical characteristics of the circuit, in particular, the value of hardware expenses. To analyze the correctness of the synthesized circuit, the process of developing the behavioral component of the VHDL model, the function of which is the generation of input signals of the finite state machine, is considered. The classical combination of the synthesizable and behavioral parts of the model allows presenting the results of the synthesis of a finite state machine with datapath of transitions as a separate project that can be used as a structural component of the designed digital system.

Results. Using the example of an abstract graph-scheme of the control algorithm, a VHDL model of a finite state machine with datapath of transitions was developed. With the help of CAD AMD Vivado, a synthesis of the developed model was carried out and behavioral modeling of the operation of the finite state machine circuit was carried out. The results of the circuit synthesis made it possible to obtain the value of hardware expenses when implementing the circuit in the FPGA basis. According to the results of behavioral modeling, time diagrams were obtained, which testify to the correctness of the implementation of the functions of transitions and outputs of the synthesized state machine.

Conclusions. In traditional VHDL models of finite state machines, the states do not contain specific codes and are identified using literals. This allows CAD to encode states at its own discretion. However, this approach is not suitable for describing a finite state machine with datapath of transitions. The transformation of states codes using a set of arithmetic and logic operations requires the use of fixed values of states codes, which determines the specifics of the VHDL model proposed in this paper. This and similar models can be used, in particular, in the study of the effectiveness of a finite state machine according to the criterion of hardware expenses in the device circuit.

KEYWORDS: finite state machine, datapath of transitions, VHDL model, hardware expenses, AMD Vivado CAD.

ABBREVIATIONS

CPLD is a complex programmable logic device;
FSM is a finite state machine;
DT is a datapath of transitions;
GSA is a graph-scheme of algorithm;
LUT is a look-up table;
TO – transitions operation.

NOMENCLATURE

A, *X*, *Y* – sets of FSM states, logical conditions and microoperations accordingly;

M, *L*, *N* – number of FSM states, logical conditions and microoperations accordingly;

R – bit depth of state code;

B – number of FSM transitions;

DOI 10.15588/1607-3274-2023-4-13

O – set of transitions operations;

I – number of transitions operations;

R_W – bit depth of code of transitions operation;

a_m, $K_1(a_m)$, $K_2(a_m)$ – current state and its scalar and vector codes;

a_s, $K_1(a_s)$, $K_2(a_s)$ – transition state and its scalar and vector codes;

X_h – logical conditions that ensure the transition h;

Y_h – microoperations formed during the transition h;

D_h – signals of code of transition state;

W_h – signals of code of transitions operations.

INTRODUCTION

Digital systems are widely used in human activity [1]. One of the central units of a digital systems is a control unit that coordinates the functioning of all system components [2, 3]. The control unit can be implemented in the form of a finite state machine (FSM), in which the control algorithm is implemented schematically [4, 5]. FSM can be implemented in the form of a Mealy FSM model or a Moore FSM model [2–5]. In comparison with other classes of control units, the FSM is characterized by maximum speed and maximum hardware expenses [2, 3]. Higher hardware expenses worsen such characteristics of the FSM circuit as cost, dimensions, energy consumption, reliability [6]. Therefore, the task of reducing hardware expenses in the finite state machine circuit is an important scientific and practical problem, forming a corresponding scientific direction [1–7].

One of the FSM types is a finite state machine with datapath of transitions (FSM with DT). Its structure includes a special datapath that converts states codes by a set of operations [8]. This approach allows, under certain conditions, to reduce hardware expenses in comparison with other FSM structures.

The design of the circuit of a digital device in the FPGA basis is carried out using specialized CAD based on the VHDL model of the device. At the moment, the problem of developing a VHDL model of the FSM with DT remains unresolved. This complicates the practical application of this class of finite state machines. This paper proposes a solution to the problem of building a VHDL model of an FSM with DT given by a graph-scheme (GSA) of control algorithm.

The object of the study is the automated synthesis of the logic circuit of a finite state machine with datapath of transitions in CAD AMD Vivado according to a VHDL model that corresponds to a given GSA.

The synthesis of a canonical finite state machine can be carried out in automatic mode using the XST tool built into CAD according to the VHDL model recommended by Xilinx [9]. In the case of FSM with DT, a VHDL model should be used, in the synthesis of which the capabilities of the XST tool are not used. One of the features of this model is the assignment of states codes of the FSM in the form of binary constants.

The subject of the study is a VHDL model of a finite state machine with datapath of transitions, which allows both the synthesis of the FSM circuit in the FPGA basis and the verification of the correctness of the functioning of the circuit by means of behavioral modeling in AMD Vivado CAD.

The purpose of the work is the development and research of the structure and methods of building a VHDL model of a finite state machine with datapath of transitions with the aim of systematizing approaches to the automated design of this class of finite state machines in the FPGA basis.

1 PROBLEM STATEMENT

Let us assume that a finite state machine with datapath of transitions is given by the graph-scheme of the algorithm G and is characterized by sets of states $A=\{a_1, ..., a_M\}$, input signals $X=\{x_1, ..., x_L\}$ and microoperations $Y=\{y_1, ..., y_N\}$. The design of the FSM logic circuit involves the implementation of the transition function $T=T(X, T)$ and the output function $Y=Y(X, T)$ in the FPGA element basis using AMD Vivado CAD (until 2023 – Xilinx Vivado CAD). The input data for design is the VHDL model of the designed device, which contains synthesized and behavioral parts and allows obtaining a quantitative value of hardware expenses for the implementation of the circuit of the state machine in a given element basis.

The work solves the problem of developing a VHDL model of an FSM with DT according to a given GSA and its investigation by means of AMD Vivado CAD.

2 REVIEW OF THE LITERATURE

In the modern theory of finite state machines, a wide range of methods for optimizing hardware expenses in the FSM circuit is known. For example, such methods are methods of structural decomposition [7], the essence of which consists in multiple transformation of logical signals, which leads to corresponding changes in the structural diagram of the FSM.

In this article, the method of operational transformation of states codes is considered as a method of hardware expenses optimization [8]. According to it, the conversion of states codes in the system of FSM transitions is carried out not by means of a system of canonical Boolean equations, but by means of a set of arithmetic and logical operations. Combinational circuits that implement these operations form the so-called datapath of transitions (DT). As a result, a structure of FSM with DT is formed, the synthesis of which is discussed in [10].

In paper [11], the justification of the effectiveness of FSM with DT in comparison with the canonical FSM structure according to the criterion of hardware expenses is presented. However, the canonical structure of FSM today has a rather theoretical value, while the practical implementation of FSM circuits is carried out with the help of appropriate CAD software, for example, AMD Vivado CAD. This is primarily due to the use of the FPGA element basis supported by CAD.

DOI 10.15588/1607-3274-2023-4-13

Since FSM is often included in designed digital systems, support for its synthesis is implemented at the AMD Vivado CAD level as part of the XST tool [9]. This tool supports several FSM synthesis methods aimed at optimizing various characteristics of the device circuit when implemented in the FPGA basis. Modeling the process of synthesizing the circuit of the state machine allows you to obtain the numerical values of the hardware expenses in the circuit of the device, expressed in the number of used LUT-elements.

The XST synthesis tool, built into the AMD Vivado CAD, is able, under certain conditions, to find code fragments in the VHDL model of the device that correspond to the description of the finite state machine (by state machine we mean a machine with undefined states codes). This process is called finite state machine extraction (FSM extraction). For the found state machine, the XST tool performs the following actions:

– states coding according to the chosen method;

– synthesis of the register circuit in accordance with the chosen method of states encoding;

– synthesis and optimization of the circuit for transition and output functions.

To ensure the possibility of automatic extraction of the state machine, in its VHDL description the following provisions should be observed:

1. The FSM states are specified in the form of a set of literals combined in an element of the enumerated type.

2. The memory register must be synchronous and have the ability to be reset to the initial state by a Reset signal.

3. Implementation of the transition and output function systems is realized using the case operator.

These requirements make it possible to specify an FSM in the form of a VHDL model using one, two or three processes [9, 12–14]. Regardless of how many processes uses the state machine, the XST tool is capable of extracting the state machine from the VHDL code and coding the states according to the chosen coding method.

The disadvantage of using the XST tool is that it is not possible to set specific values of states codes during the FSM synthesis. This makes it impossible to use optimization methods that are based on special coding of states. These methods also include the method of operational transformation of states codes. Therefore, the XST tool cannot be used for the synthesis of an FSM with DT circuit. As a result, the requirements for the VHDL model of the FSM given in [9, 12–14] cannot be directly applied to the FSM with DT and need to be adjusted.

3 MATERIALS AND METHODS

The structural diagram of an FSM with DT is shown in Fig. 1 and contains the following blocks [10].

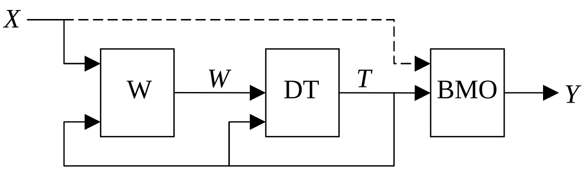


Figure 1 – Structural diagram of an FSM with DT

1. Block DT realizes the following function:

$$T = T\,(T, W), \tag{1}$$

that is, converts the current state code T to a transition state code using a transitions operation with the W code.

2. Block W realizes the following function:

$$W = W\,(T, X), \tag{2}$$

that is, it forms transitions operations codes that control the operations of the DT.

3. Block BMO realizes the function

$$Y = Y\,(X, T) \tag{3}$$

in the case of Mealy FSM or function

$$Y = Y\,(T) \tag{4}$$

in the case of a Moore FSM, that is, it provides the implementation of FSM output function. In fig. 1, the presence of a connection marked with a dashed line allows you to consider the structure as a Mealy FSM, the absence of a connection – as a Moore FSM.

The internal structure of the DT is shown in Fig. 2 [10].

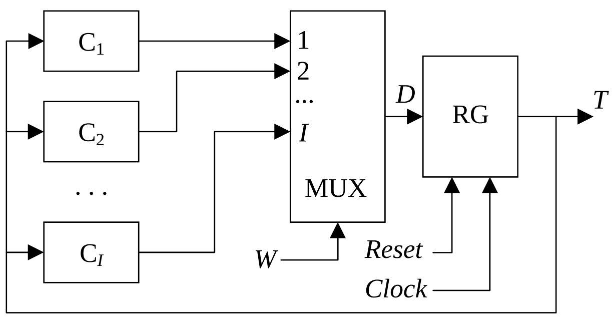


Figure 2 – Internal structure of DT

Blocks $C_1 – C_I$ correspond to combinational circuits implementing a set of transitions operations (TO) $O=\{O_1, ..., O_I\}$. In general, each TO can be arithmetic, logical or combined. When designing these blocks, if possible, each block should be optimized in order to increase performance and reduce hardware expenses.

The MUX block is an R-bit multiplexer with I-directions. Under the guidance of the W signals at the output of the multiplexer, the R-bit code D of the next FSM state is formed, which enters the input of the memory register RG.

The RG block is an R-bit synchronous register with the function of resetting to the initial state by the *Reset* signal. It should be noted that in the case of FSM with DT, the initial state does not necessarily have a zero code. This register performs the function of the memory of the datapath of transitions and the function of the memory register of the finite state machine.

Let FSM be given by GSA G (Fig. 3). This GSA is marked by states of Moore FSM and contains the set of states of the states $A=\{a_0, ..., a_{20}\}$ with cardinality M=21, the set of logical conditions $X=\{x_1, x_2, x_3\}$ with cardinality L=3, multiple set of microoperations (output signals)

DOI 10.15588/1607-3274-2023-4-13

$Y=\{y_1, ..., y_7\}$ with cardinality N=7 and B=29 FSM transitions. GSA has an abstract structure and content of operator vertices and is intended to demonstrate the process of building a VHDL model of an FSM with DT.

The main and most difficult stage of the synthesis of an FSM with DT is the so-called algebraic synthesis of FSM. In the process of algebraic synthesis, the following occurs [10]:

1. The FSM states are matched with unique codes from a certain set of states codes. In the case of GSA G, R=5 binary digits are enough to encode M=21 states.

2. FSM transitions correspond to certain transitions operations from a given set of TOs. The use of one TO for the implementation of several state machine transitions is permissible and contributes to the reduction of the total number of used TOs and, accordingly, to the reduction of hardware expenses in the FSM circuit. Those transitions that cannot be implemented by any of the specified TOs should be implemented in a canonical way using a system of Boolean equations.

We will carry out an algebraic synthesis for GSA G under the condition that the set of transitions operations is formed by the following ones: $O = \{O_1, O_2, O_3\}$:

$$O_1: D = T + 7_{10}; \quad (5)$$

$$O_2: D = T \,\&\, 01001_2; \quad (6)$$

$$O_3: D = T \oplus 00011_2. \quad (7)$$

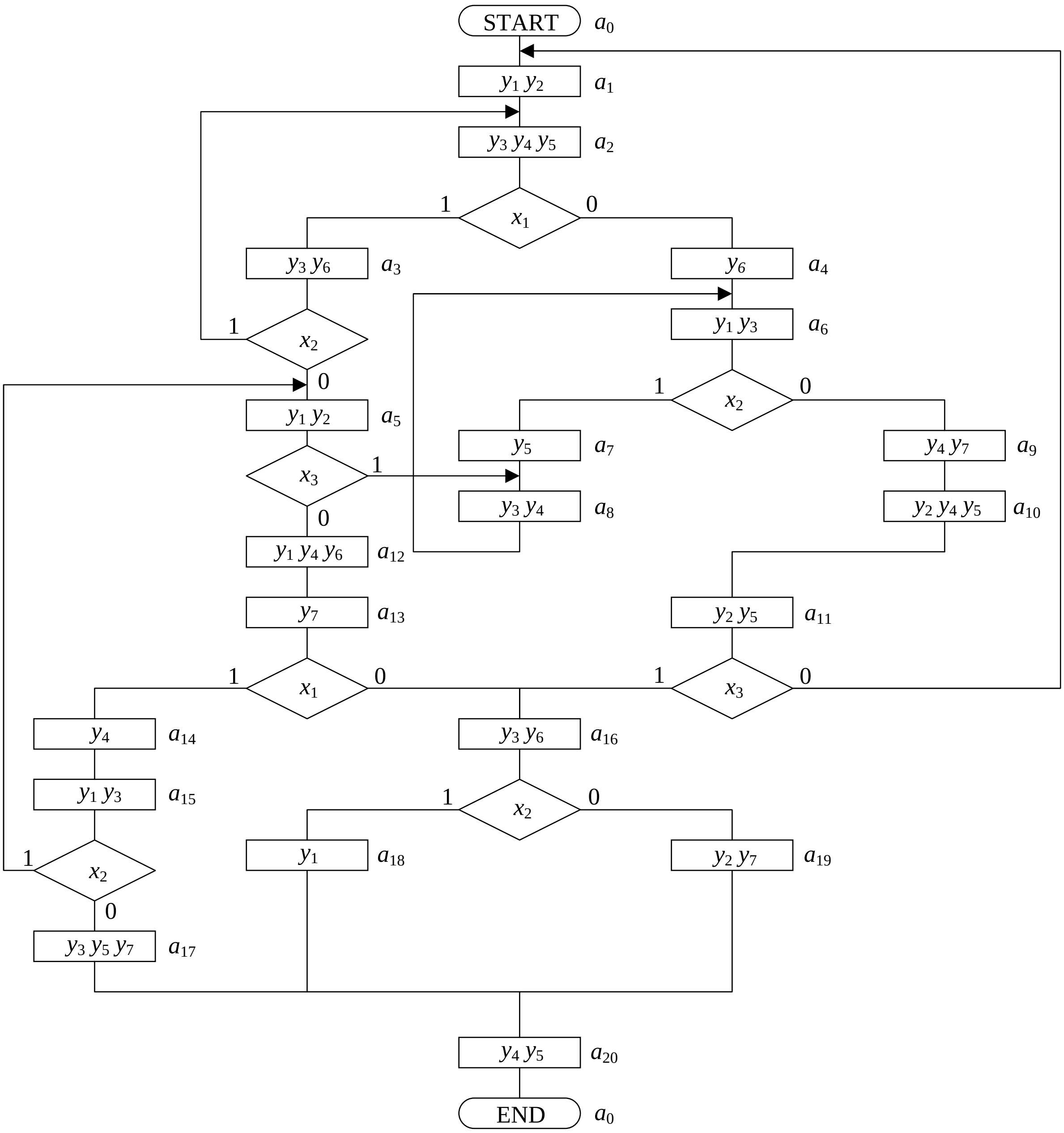


Figure 3 – Graph-scheme of algorithm G

DOI 10.15588/1607-3274-2023-4-13

In these expressions, T is the code of the current FSM state, D is the code of the state of the transition, which is formed at the output of the multiplexer and enters the RG (Fig. 2).

TO O_1 corresponds to the operation of adding the decimal constant 7 to the code of the current state. The code of the current state T is interpreted as an unsigned decimal number of 5 binary digits size. The operation is implemented on the basis of a 5-digit binary adder, in which the carry from the higher bit is discarded. This is equivalent to the operation "(T + 7) mod 32". For example, (25+10) mod 32 = 3.

TO O_2 is a bitwise logical conjunction operation on the binary value of the current state code T and the binary constant 01001.

TO O_3 is a bitwise logical operation XOR on the binary value of the current state code T and the binary constant 00011.

In general, some FSM transitions can be implemented in a canonical way without using the specified transitions operations. The circuit that implements all such transitions will act as a separate combinational circuit C_i as part of the DT (Fig. 2). In order for the multiplexer to be able to pass through the result of the operation of this circuit, we must consider it as a separate TO O_4, which has its own code. Formally, O_4 is some function Φ of the code of the current state of the automaton:

$$O_4 : T = \Phi\,(T). \quad (8)$$

Successful execution of algebraic synthesis gives us a formal solution of the algebraic synthesis problem [10]. In general, there may be several formal solutions. As an example, consider the formal solution shown in Fig. 4 (the method of obtaining it is not considered in this paper).

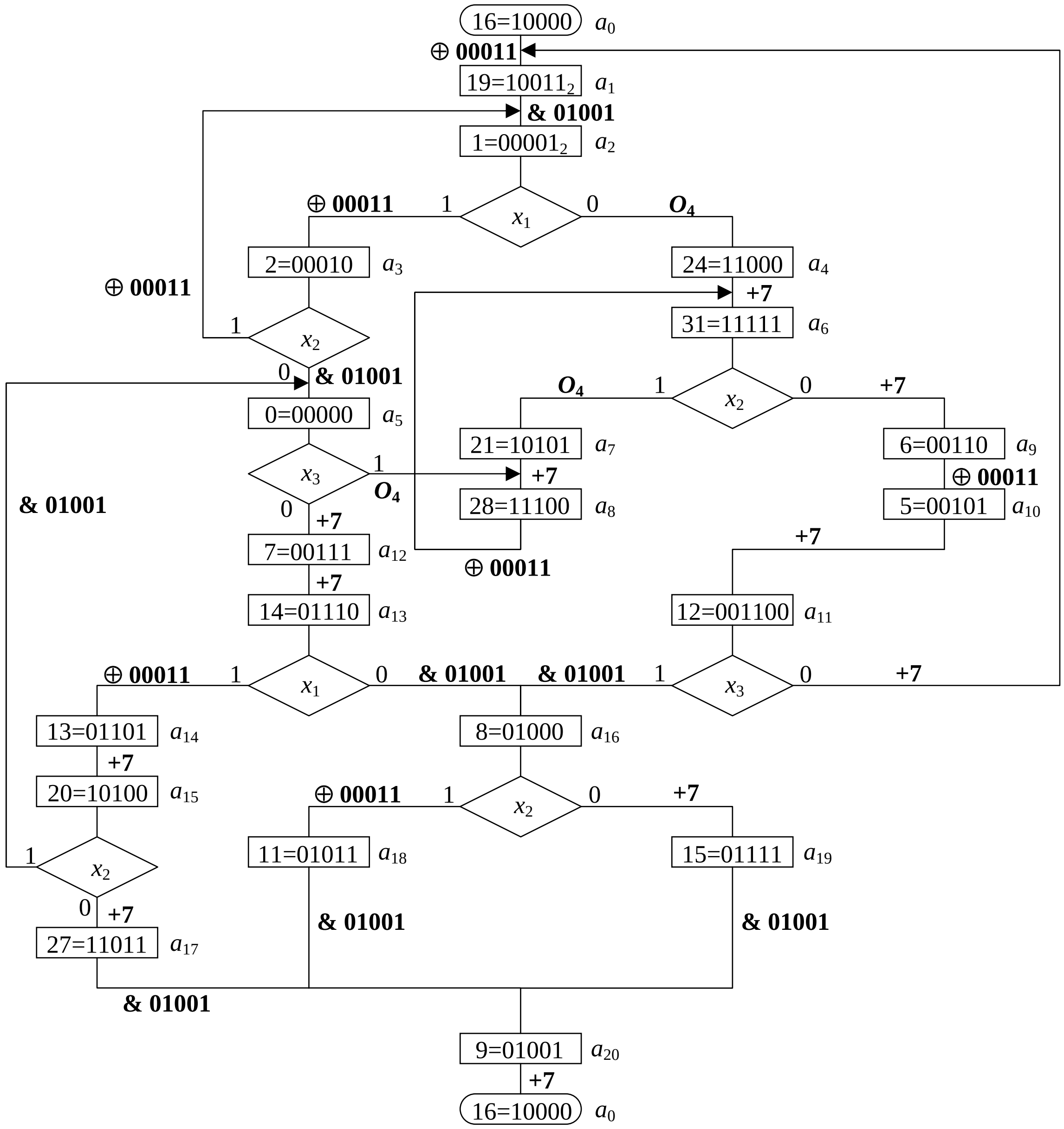


Figure 4 – Formal solution of problem of algebraic synthesis of an FSM with DT for GSA G (example)

DOI 10.15588/1607-3274-2023-4-13

Since the operational transformation of states codes does not affect the FSM output function, in Fig. 4 inside operational vertices, microoperations are not shown. Instead of them, selected decimal values of states codes and their binary equivalents are shown. Each FSM transition is marked by a transitions operation that is mapped to it. Operation O_1 is marked with "+7", O_2 – "& 01001", O_3 – "⊕ 00011" for clarity. Transitions implemented in the canonical way ($a_2 \rightarrow a_4$, $a_5 \rightarrow a_5$, $a_6 \rightarrow a_7$) are marked with the symbol O_4.

Let's explain the operational implementation of transitions in Fig. 4. Transition from state a_0 coded with $K(a_0)=16_{10}=10000_2$ to state a_1 coded with $K(a_1)=19_{10}=10011_2$ is carried out using O_3. Transition from state a_{16} coded with $K(a_{16})=8_{10}=01000_2$ to state a_{19} coded with $K(a_{19})=15_{10}=01111_2$ is carried out using O_1. Transition from state a_3 coded with $K(a_3)=2_{10}=00010_2$ to state a_5 coded with $K(a_5)=0_{10}=00000_2$ is carried out using O_2.

Let's encode operations of transitions $O_1 – O_4$ with unique binary codes of bit depth $R_W = \rceil \log_2 4 \lceil = 2$, formed by variables $W = \{w_1, w_2\}$. The result of coding is presented in Table 1.

Table 1 – Coding of transitions operations

O_i	$w_1\ w_2$
O_1	0 0
O_2	0 1
O_3	1 0
O_4	1 1

Let us present the result of algebraic synthesis in the form of an operational table of transitions (OTT) [10], which for our example has the form of a Table 2.

In the Table 4, each row corresponds to a separate FSM transition, the number of which is indicated in column h. The W_h column contains codes of transitions operations according to the table 1. In each cell of the W_h column, only those variables w that are equal to 1 in the code of the corresponding TO, are shown. For example, the transition h=5 is implemented using operation O_3 with binary code 10 (w_1=1, w_2=0), so in the row h=5 in the column W_h, only the variable w_1 is indicated, which is equal to 1 in binary code 10.

Let's proceed to the construction of the VHDL model of the FSM with DT, the OTT of which corresponds to the table. 2. We will present the model in the form of synthesizable and behavioral parts [5, 12, 14]. Consider the description of the synthesized part.

Table 2 – Operational table of transitions (GSA *G*)

a_m	$K_1(a_m)$	$K_2(a_m)$	a_s	$K_1(a_s)$	$K_2(a_s)$	X_h	W_h	Y_h	h
a_0	16	10000	a_1	19	10011	1	w_1	–	1
a_1	19	10011	a_2	1	00001	1	w_2	$y_1\ y_2$	2
a_2	1	00001	a_3	2	00010	x_1	w_1	$y_3\ y_4\ y_5$	3
			a_4	24	11000	$\bar{x}_1$	$w_1\ w_2$		4
a_3	2	00010	a_1	1	00001	x_2	w_1	$y_3\ y_6$	5
			a_5	0	00000	$\bar{x}_2$	w_2		6
a_4	24	11000	a_6	31	11111	1	–	y_6	7
a_5	0	00000	a_8	28	11100	x_3	$w_1\ w_2$	$y_1\ y_2$	8
			a_{12}	7	00111	$\bar{x}_3$	–		9
a_6	31	11111	a_7	21	10101	x_2	$w_1\ w_2$	$y_1\ y_3$	10
			a_9	6	00110	$\bar{x}_2$	–		11
a_7	21	10101	a_8	28	11100	1	–	y_5	12
a_8	28	11100	a_6	31	11111	1	w_1	$y_3\ y_4$	13
a_9	6	00110	a_{10}	5	00101	1	w_1	$y_4\ y_7$	14
a_{10}	5	00101	a_{11}	12	01100	1	–	$y_2\ y_4\ y_5$	15
a_{11}	12	01100	a_{16}	8	01000	x_3	w_2	$y_2\ y_5$	16
			a_1	19	10011	$\bar{x}_3$	–		17
a_{12}	7	00111	a_{13}	14	01110	1	–	$y_1\ y_4\ y_6$	18
a_{13}	14	01110	a_{14}	13	01101	x_1	w_1	y_7	19
			a_{16}	8	01000	$\bar{x}_1$	w_2		20
a_{14}	13	01101	a_{15}	20	10100	1	–	y_4	21
a_{15}	20	10100	a_5	0	00000	x_2	w_2	$y_1\ y_3$	22
			a_{17}	27	11011	$\bar{x}_2$	–		23
a_{16}	8	01000	a_{18}	11	01011	x_2	w_1	$y_3\ y_6$	24
			a_{19}	15	01111	$\bar{x}_2$	–		25
a_{17}	27	11011	a_{20}	9	01001	1	w_2	$y_3\ y_5\ y_7$	26
a_{18}	11	01011	a_{20}	9	01001	1	w_2	y_1	27
a_{19}	15	01111	a_{20}	9	01001	1	w_2	$y_2\ y_7$	28
a_{20}	9	01001	a_0	16	10000	1	–	$y_4\ y_5$	29

DOI 10.15588/1607-3274-2023-4-13

```
entity FSM is
    generic(R: integer := 5;                -- State code capacity
            Rw: integer := 2;               -- W code capacity
            L: integer := 3;                -- Number of input signals
            N: integer := 5);               -- Number of microoperations
    port (X: in std_logic_vector(1 to L);   -- Input signals
          Y: out std_logic_vector(1 to N);  -- Microoperations
          C: in std_logic;                  -- Clock
          Reset: in std_logic);             -- Reset
end FSM;
```

In the “generic” section, the setting constants that determine the bit depth of the signal buses are defined. The “port” section contains the bus of input signals *X*, the bus of output microoperations *Y*, the synchronization signal *Clock* and the *Reset* signal, by which the code of the initial state of the FSM is written into the memory register.

The architecture section contains a description of the internal FSM signals, as well as a description of the structural blocks in the view of processes in accordance with Fig. 1 and 2. The beginning of the description of the architecture block looks like next:

```
architecture FSM_A of FSM is

signal T, D: unsigned(1 to R);          -- State code and Next state code
signal Canonic : unsigned(1 to R);      -- Result of 'canonical' transitions
signal nT: unsigned(1 to R);            -- Negative values of State code
signal nX: std_logic_vector(1 to L);    -- Negative values of input signals
signal W: std_logic_vector(1 to Rw);    -- Code of datapath operation

begin

nT <= not T;
nX <= not X;
```

Here, the “Canonic” signal is the code of the next state, formed in a canonical way. Its use will be discussed later.

Below a process block describing the FSM memory register is shown. The register switches synchronously with the rising edge of the *Clock* signal.

```
process(C) -- Memory Register
begin
    if rising_edge(C) then
        if Reset = '1' then
            T <= "10000";
        else
            T <= D;
        end if;
    end if;
end process;
```

The peculiarity of this description is that by a *Reset* signal equal to one, the register is transferred to the initial state, the code of which, according to the results of algebraic synthesis (Table 2), is equal to 10000_2.

Let’s synthesize the block W, which forms the signals w_1, w_2 of the transitions operation code (Fig. 1). We implement these signals using canonical Boolean equations according to the table 2 and expression (2).

$$w_1 = a_0 \vee a_2 x_1 \vee a_3 x_2 \vee a_5 x_3 \vee a_6 x_2 \vee a_8 \vee a_9 \vee$$
$$\vee\, a_{13} x_1 \vee a_{16} x_2;$$
$$w_2 = a_1 \vee a_2 \bar{x}_1 \vee a_3 \bar{x}_2 \vee a_6 x_2 \vee a_{11} x_3 \vee a_{13} \bar{x}_1 \vee$$
$$\vee\, a_{15} x_2 \vee a_{17} \vee a_{18} \vee a_{19}.$$

We will use binary vectors $<T_1, ..., T_5>$ to represent the FSM states codes. Then, according to the coding of the states given in the Table 2, the Boolean equations for signals w_1, w_2 take the following form:

$$w_1 = T_1\bar{T}_2\bar{T}_3\bar{T}_4\bar{T}_5 \vee \bar{T}_1\bar{T}_2\bar{T}_3\bar{T}_4T_5x_1 \vee \bar{T}_1\bar{T}_2\bar{T}_3T_4\bar{T}_5x_2 \vee$$
$$\vee\, \bar{T}_1\bar{T}_2\bar{T}_3\bar{T}_4\bar{T}_5x_3 \vee T_1T_2T_3T_4T_5x_2 \vee T_1T_2T_3\bar{T}_4\bar{T}_5 \vee$$
$$\vee\, \bar{T}_1\bar{T}_2T_3T_4\bar{T}_5 \vee \bar{T}_1T_2T_3T_4\bar{T}_5x_1 \vee \bar{T}_1T_2\bar{T}_3\bar{T}_4\bar{T}_5x_2;$$
$$w_2 = T_1\bar{T}_2\bar{T}_3T_4T_5 \vee \bar{T}_1\bar{T}_2\bar{T}_3\bar{T}_4T_5\bar{x}_1 \vee \bar{T}_1\bar{T}_2\bar{T}_3T_4\bar{T}_5\bar{x}_2 \vee$$
$$\vee\, T_1T_2T_3T_4T_5x_2 \vee \bar{T}_1T_2T_3\bar{T}_4\bar{T}_5x_3 \vee \bar{T}_1T_2T_3T_4\bar{T}_5\bar{x}_1 \vee$$
$$\vee\, T_1\bar{T}_2T_3\bar{T}_4\bar{T}_5x_2 \vee T_1T_2\bar{T}_3T_4T_5 \vee \bar{T}_1T_2\bar{T}_3T_4T_5 \vee$$
$$\vee\, \bar{T}_1T_2T_3T_4T_5.$$

In general, these equations can be minimized in any convenient way. In this paper, we will not perform minimization and will immediately present block W in the form of the following VHDL process:

DOI 10.15588/1607-3274-2023-4-13

```
process (T, X, nT, nX) -- Block W
    begin
        W(1) <= (T(1) and nT(2) and nT(3) and nT(4) and nT(5)) or
                (nT(1) and nT(2) and nT(3) and nT(4) and T(5)) or
                (nT(1) and nT(2) and nT(3) and T(4) and nT(5) and X(2)) or
                (nT(1) and nT(2) and nT(3) and nT(4) and nT(5) and X(3)) or
                (T(1) and T(2) and T(3) and T(4) and T(5) and X(2)) or
                (T(1) and T(2) and T(3) and nT(4) and nT(5)) or
                (nT(1) and nT(2) and T(3) and T(4) and nT(5)) or
                (nT(1) and T(2) and T(3) and T(4) and nT(5) and X(1)) or
                (nT(1) and T(2) and nT(3) and nT(4) and nT(5) and X(2));

        W(2) <= (T(1) and nT(2) and nT(3) and T(4) and T(5)) or
                (nT(1) and nT(2) and nT(3) and nT(4) and T(5) and nX(1)) or
                (nT(1) and nT(2) and nT(3) and T(4) and nT(5) and nX(2)) or
                (nT(1) and nT(2) and nT(3) and nT(4) and nT(5) and X(3)) or
                (T(1) and T(2) and T(3) and T(4) and T(5) and X(2)) or
                (nT(1) and T(2) and T(3) and nT(4) and nT(5) and X(3)) or
                (nT(1) and T(2) and T(3) and T(4) and nT(5) and nX(1)) or
                (T(1) and nT(2) and T(3) and nT(4) and nT(5) and X(2)) or
                (T(1) and T(2) and nT(3) and T(4) and T(5)) or
                (nT(1) and T(2) and nT(3) and T(4) and T(5)) or
                (nT(1) and T(2) and T(3) and T(4) and T(5));
end process;
```

Please note that to ensure the correctness of the simulation, the process sensitivity list contains both direct and inverse values of the *X* and *T* signals.

Let's proceed to the synthesis of the datapath of transition. Let's clarify the structure of the DT shown in Fig. 2, according to the results of algebraic synthesis. The clarification of the structure consists in the fact that it contains four combinational circuits $C_1 – C_4$, which correspond to transitions operations $O_1 – O_4$, and the multiplexer is controlled by a two-bit binary code W=<w_1, w_2>.

As will be shown below, combinational circuits $C_1 – C_3$ have a trivial implementation using operators from the synthesizable subset of VHDL. However, block C_4 is non-standard, as it represents a canonical implementation of a certain part of an FSM transitions. In the general case, the code of the current state *T* and signals of logical conditions *X* are received at its inputs. So the refined structure of the OAP for GSA *G* is shown in Fig. 5.

Before developing the VHDL description of the DT, let's synthesize the C_4 block. For this purpose, we will use the technique discussed in [2, 3].

Transitions implemented in the canonical way, in Table 2, have numbers 4, 8 and 10. Let's do the following.

1. Let's agree to use the Boolean vector <D_1, ..., D_5> to encode the transition state code.

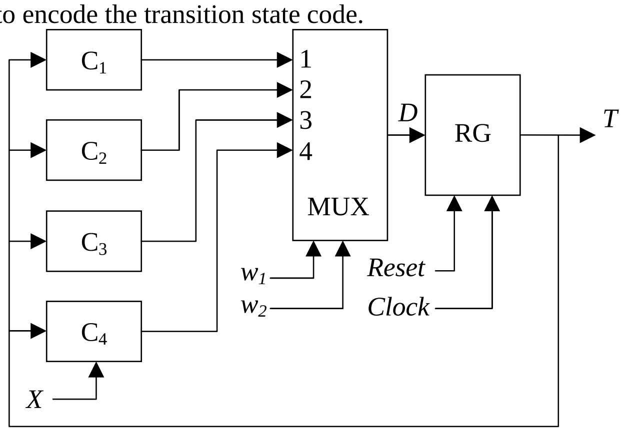


Figure 5 – Clarified structure of DT (GSA *G*)

2. Let's form a separate table of transitions from these lines, in which instead of the column W_h there is a column D_h. This column indicates those components of the vector <D_1, ..., D_5> which are equal to 1 in the binary code of the transition state $K_2(a_s)$ of this row of the table.

As a result, we will get a table of transitions implemented in the canonical way (Table 3).

Table 3 – Table of transitions implemented in the canonical way (GSA *G*)

a_m	$K_1(a_m)$	$K_2(a_m)$	a_s	$K_1(a_s)$	$K_2(a_s)$	X_h	D_h	Y_h	h
a_2	1	00001	a_4	24	11000	$\bar{x}_1$	$D_1 D_2$	y_3 y_4 y_5	1
a_5	0	00000	a_8	28	11100	x_3	$D_1 D_2 D_3$	y_1 y_2	2
a_6	31	11111	a_7	21	10101	x_2	$D_1 D_3 D_5$	y_1 y_3	3

Each of the three transitions presented in the Table 3, corresponds to the own term formed by the conjunction of signals T_1, ..., T_5 of the code of the current state (column $K_2(a_m)$) and the corresponding signal of the logical condition (column X_h). Let's form these terms:

$$Q_1 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 T_5 \bar{x}_1 ;$$

$$Q_2 = \bar{T}_1 \bar{T}_2 \bar{T}_3 \bar{T}_4 \bar{T}_5 x_3 ;$$

$$Q_3 = T_1 T_2 T_3 T_4 T_5 x_2 .$$

Now we can form the equation for signals $D_1 – D_5$ according to the contents of column D_h. Since the D_4 signal is not present in the D_h column, we will consider it always equal to 0.

$$D_1 = Q_1 \vee Q_2 \vee Q_3 ;$$

$$D_2 = Q_1 \vee Q_2 ;$$

$$D_3 = Q_2 \vee Q_3 ;$$

$$D_4 = 0 ;$$

DOI 10.15588/1607-3274-2023-4-13

$$D_5 = Q_3 .$$

The VHDL code for the combinational circuit C_4 is given below. As you can see, the vector <D_1, ..., D_5> in this model corresponds to the "Canonic" signal. Although this signal is declared as "unsigned (1 to R) ", each bit is generated separately. Also, pay attention to the presence in the process sensitivity list of both direct and inverse values of T and X signals.

```
        process (T, X, nT, nX) -- Canonical transitions
            variable Q1: std_logic;
            variable Q2: std_logic;
            variable Q3: std_logic;
        begin
            Q1 := nT(1) and nT(2) and nT(3) and nT(4) and T(5) and nX(1);
            Q2 := nT(1) and nT(2) and nT(3) and nT(4) and nT(5) and X(3);
            Q3 := T(1) and T(2) and T(3) and T(4) and T(5) and X(2);

            Canonic(1) <= Q1 or Q2 or Q3;
            Canonic(2) <= Q1 or Q2;
            Canonic(3) <= Q2 or Q3;
            Canonic(4) <= '0';
            Canonic(5) <= Q3;
        end process;
```

Although the combinational circuit C_4 is described as a separate process, structurally it is part of the DT block (Fig. 5). The description of the DT block in VHDL is as follows:

```
process (T, W, Canonic) -- Datapath
begin
    case W is
        when "00" =>                    -- O1
            D <= T + 7;
        when "01" =>                    -- O2
            D <= T and "01001";
        when "10" =>                    -- O3
            D <= T xor "00011";
        when "11" =>                    -- O4
            D <= Canonic;
        when others =>
            D <= "00000";
    end case;
end process;
```

The basis of this process is the "case" operator, which has four branches corresponding to operation codes of transitions O_1 – O_4. Operations O_1 – O_4 are implemented with the help of "+", "and" and "xor" operators, which are included in the synthesized subset of VHDL and can work directly with the "unsigned" data type. In the case of O_4, to the output bus D the result obtained from the output of the combinational circuit C_4 (input signal "Canonic") is passed through.

The following fragment of the VHDL code describes the BMO block that forms FSM output signals according to (4). The method of describing this block is not fundamental and can be implemented both with the help of the "case" operator and by setting a system of canonical Boolean equations by analogy with the considered blocks W and C_4. Also, this block can be synthesized using various methods of output function optimization [2–4, 7].

```
process (T, nT)
begin
    case T is
        when "10011" => Y <= "1100000";
        when "00001" => Y <= "0011100";
        when "00010" => Y <= "0010010";
        when "11000" => Y <= "0000010";
        when "00000" => Y <= "1100000";
        when "11111" => Y <= "1010000";
        when "10101" => Y <= "0000100";
        when "11100" => Y <= "0011000";
        when "00110" => Y <= "0001001";
        when "00101" => Y <= "0101100";
        when "01100" => Y <= "0100100";
        when "00111" => Y <= "1001010";
        when "01110" => Y <= "0000001";
        when "01101" => Y <= "0001000";
        when "10100" => Y <= "1010000";
        when "01000" => Y <= "0010010";
        when "11011" => Y <= "0010101";
        when "01011" => Y <= "1000000";
        when "01111" => Y <= "0100001";
        when "01001" => Y <= "0001100";
        when others =>  Y <= "0000000";
    end case;
end process;
```

The fragments of the VHDL code considered above form a synthesizable part of the VHDL model of the FSM with DT. For the correct functioning of the model, its first lines should be lines connecting the necessary libraries:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;
```

As the last line, you should add the architecture block completion statement:

```
end FSM_A;
```

To check the correctness of the considered VHDL model, it is necessary to develop its behavioral part. The function of the behavioral part in our case is the generation of external signals and their supply to the FSM inputs. Time intervals of signals generation in this case are of no fundamental importance, since behavioral modeling will take place without reference to the physical characteristics of the device.

The behavioral part can be described by the following VHDL code fragment:

DOI 10.15588/1607-3274-2023-4-13

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity Model is
    generic(R: integer := 5;                        -- State code capacity
            Rw: integer := 2;                       -- W code capacity
            N: integer := 7;                        -- Number of microoperations
            L: integer := 3);                       -- Number of logical conditions
    port (Y: out std_logic_vector(1 to N));         -- Microoperations
end Model;

architecture Model_A of Model is
signal C: std_logic;                                -- Clock
signal Reset: std_logic;                            -- Reset
signal X: std_logic_vector(1 to L);                 -- Logical conditions

component FSM is
    generic(R: integer := 5;                        -- State code capacity
            Rw: integer := 2;                       -- W code capacity
            L: integer := 3;                        -- Number of logical conditions
            N: integer := 7);                       -- Number of microoperations
    port (X: in std_logic_vector(1 to L);           -- Input signals
          Y: out std_logic_vector(1 to N);          -- Microoperations
          C: in std_logic;                          -- Clock
          Reset: in std_logic);                     -- Reset
end component FSM;

begin
        process                                                         -- Clock
        begin
           C <= '0'; wait for 80 ns;
           C <= '1'; wait for 20 ns;
        end process;

        Reset <= '0' after 0 ns, '1' after 10 ns, '0' after 90 ns;      -- Reset

        process                                                         -- X1
        begin
           X(1) <= '1'; wait for 17 ns; X(1) <= '0'; wait for 37 ns;
        end process;

        process                                                         -- X2
        begin
           X(2) <= '1'; wait for 43 ns; X(2) <= '0'; wait for 36 ns;
        end process;

        process                                                         -- X3
        begin
           X(3) <= '1'; wait for 38 ns; X(3) <= '0'; wait for 17 ns;
        end process;

        L1: component FSM
            port map (X, Y, C, Reset);

end Model_A;
```

The behavioral part has the following features:

1. Microoperations formed by the FSM are displayed on the output port *Y*.

2. The *Clock* signal has an interval of 100 ns. The *Reset* signal is generated once at the start of the device's functioning.

3. Signals *X* are formed in separate processes, which makes them independent of each other. The intervals of the upper and lower levels are random and can have any values.

4 EXPERIMENTS

For the developed VHDL model of FSM with DT, the authors conducted experimental research with the help of CAD AMD Vivado version 2023.1 (Vivado ML Standard Edition, free version). The research sets two goals:

1. Checking the correctness of the work of the synthesized FSM circuit using behavioral modeling.

2. Checking the possibility of synthesis of MPA logic circuit in FPGA basis.

Achieving the first goal will allow us to consider the proposed approach to building a VHDL model of an FSM with DT correct. Achieving the second goal will confirm the possibility of using the developed model to evaluate

DOI 10.15588/1607-3274-2023-4-13

the effectiveness of FSM with DT according to the criterion of hardware expenses [11].

5 RESULTS

Behavioral modeling of the developed model of FSM with DT for GSA *G* was performed in AMD Vivado CAD using standard modeling parameters. A fragment of the timing diagram of the state machine is shown in Fig. 6. Signals *T* and *D* are in unsigned decimal format, other signals are in binary format. Three markers are set on the diagram, which allow to analyze important moments of time in functioning of the FSM. Let's consider them.

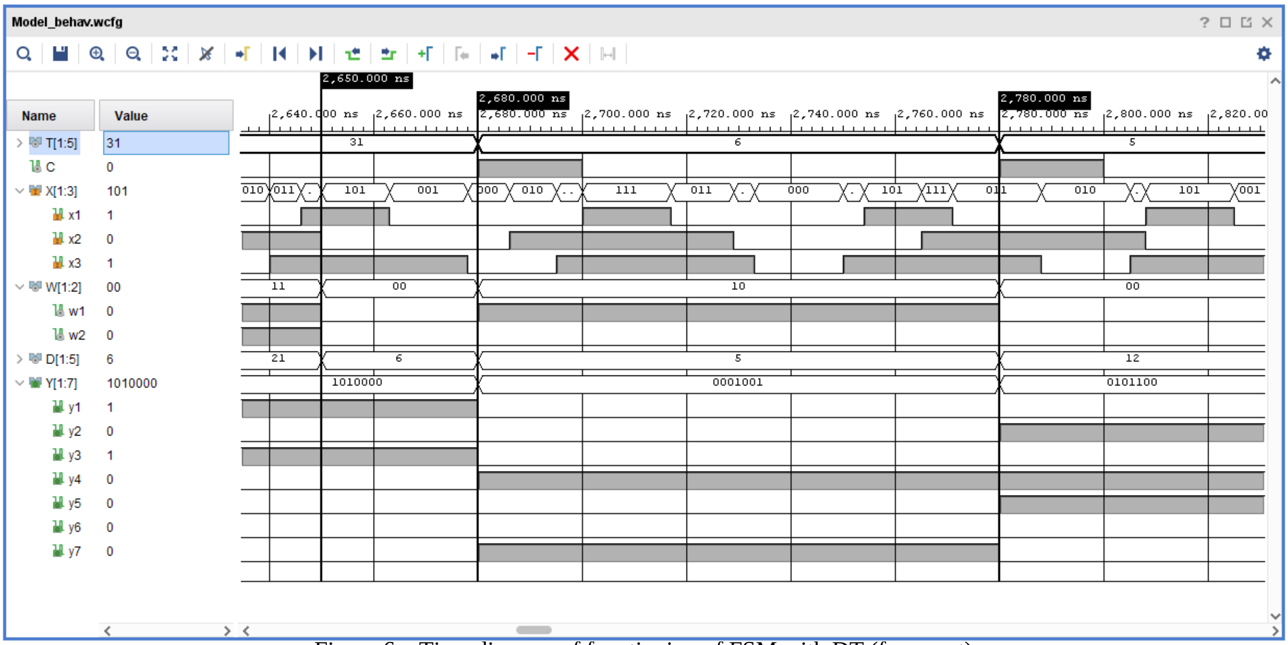


Figure 6 – Time diagram of functioning of FSM with DT (fragment)

Until the moment t_1=2650 *ns*, the FSM is in the state with code $T=31_{10}=11111_2$ (state a_6) and forms microoperations y_1, y_3. At time t_1, signal x_2 becomes equal to 0, which sets the values of signals w_1 and w_2 into zero values. This results to executing of operation O_1:

$$D = 31_{10} + 7_{10} = (38 \bmod 32)_{10} = 6_{10} = 00110_2.$$

This value is formed on bus *D* and is the state code of the transition of the FSM in the next cycle of functioning.

At the moment t_2=2680 *ns*, the rising edge of the synchronization signal *C* arrives. Following this signal, the value $D=6_{10}=00110_2$ is loaded into the memory register and appears on the bus *T*. Thus, the FSM correctly transitioned from state a_6 with code 31_{10} to state a_9 with code 6_{10}. Also, with a change in *T*, there is a change in the output signals: microoperations y_4, y_7 are formed on the *Y* bus. This coincides with Fig. 3 and confirms the correct functioning of the BMO circuit.

At the moment t_3=2780 *ns*, the FSM after the rising edge of signal *C* passes from the state with code $T=6_{10}=00110_2$ (state a_9) to the state with code $D=5_{10}=00101_2$ (state a_{10}). The transformation of the state code proceeds using the operation O_3: $00110_2 \oplus 00011_2 = 00101_2$. Since this transition is unconditional, the operation code $W(O_3)$=<10> is formed on the *W* bus immediately after the FSM transition to the state $T=00110_2$ (starting from the moment t_2) and does not change when the values of signals $x_1 - x_3$ change. Also, microoperations y_2, y_4, y_5 are formed at time t_4. This corresponds to state a_{10} in Fig. 3 and to transition *h*=15 in table 2.

Thus, it can be concluded that the developed VHDL model of FSM with DT is correct and corresponds to the given graph-scheme of algorithm *G*.

Let's check the possibility of synthesizing the developed VHDL model in the FPGA basis. Experiments have shown that stages of synthesis and implementation in the FPGA chip xc7a12tcpg238-1 occur without errors. As a result of the synthesis, the numerical values of the hardware expenses for the implementation of the synthesizable part of the VHDL model of the FSM with DT were obtained, consisting of 16 LUT elements and 5 triggers. Thus, the developed VHDL model can be used to evaluate the efficiency of the FSM circuit according to the criterion of hardware expenses.

6 DISCUSSION

The method of operational transformation of states codes, which is the base of the structure of FSM with DT, provides for special coding of states of the FSM. Values of states codes, selected transitions operations and their mapping to FSM transitions form a full picture, which is called a formal solution to the problem of algebraic synthesis of an FSM with DT. On the one hand, the special coding of states makes it impossible to use the finite state machine synthesis tools built into the XST AMD Vivado CAD [9]. On the other hand, knowing the specific values of states codes allows you to apply your own optimization methods aimed at optimizing hardware expenses in various structural blocks of an FSM with DT.

DOI 10.15588/1607-3274-2023-4-13

Thus, the application of structure of FSM with DT requires the development of its own VHDL model, which differs from the models recommended by AMD Vivado CAD developers. The example considered in this paper demonstrates the following features of the approach to building such a model:

1. In the FSM with DT it is allow the use of any arithmetic and logical operations. They can provide a different interpretation to the states codes of the FSM – scalar (numeric) or vector (binary). Accordingly, the VHDL model must also support different ways of interpreting the states codes. In the considered example, the “unsigned” data type is used to represent states codes, which allows performing both arithmetic (scalar) and logical (vector) operations on states codes. If necessary, it will allow the use of other scalar types available in the VHDL language libraries. For example, to work with signed numbers, the “integer” data type can be used. In cases where the set of TOs contains only vector operations, it is sufficient to use only vector data type (such as “std_logic_vector”) for states codes.

2. In the considered example, the combinational circuit C_4 implements FSM transitions, which are implemented in a canonical way according to the system of Boolean equations. The necessity of using such a circuit is due only to the results of the algebraic synthesis carried out by the authors for a given GSA *G*. In general, situations are possible when all FSM transitions are implemented using a given set of TOs. In this case, there will be no need for a circuit like C_4.

3. The number of combinational circuit reflects the number of different transitions operations. In our case, it is a number of lines of the “case” operator, which corresponds to the multiplexer in Fig. 2. In general, the smaller number of combinational circuit in the datapath (that is, the smaller number of different TOs) corresponds to decrease in hardware expenses in the FSM with DT circuit.

4. Algebraic synthesis of FSM with DT demands the formation of set of transitions operations. Such a formation can occur both at the beginning of algebraic synthesis and during of its execution. When forming the set of OT, it is necessary to ensure that these operations can be implemented with the help of operators from the synthesizable subset of the VHDL language. If there is a need to implement a non-standard operation (scalar or vector), you can use an approach similar to the development of the C_4 circuit in the above example.

5. FSM with DT belongs to the class of FSM with “hardware” logic. It should be understood that, in general, any change in the input data (the GSA, the set of TOs, the bit depth of states codes, etc.) requires a complete resynthesis of the FSM and its VHDL model. This applies to the circuits of any custom digital devices.

CONCLUSIONS

The paper proposes a solution to the scientific problem of developing a VHDL model of a finite state machine with datapath of transitions. The correctness of the model was checked in AMD Vivado CAD.

The scientific novelty of the work lies in the fact that all stages of development of a VHDL model are demonstrated on a specific example, which allows you to understand its peculiarities and differences from typical models of finite state machines. The main feature of the proposed model is that it is not focused on the use of finite state machine synthesis tools built into CAD and can be used in CADs of different FPGA manufacturers.

The practical use of the obtained results is possible in the development of methods of synthesis, optimization and evaluation of the efficiency of a finite state machines with datapath of transitions, as well as other structures and methods aimed at optimizing the characteristics of an FSM circuit.

Prospects for further research consist in solving a range of scientific and practical problems related to the development, implementation and evaluation of the effectiveness of the structures and methods of synthesis of finite state machines with optimized hardware expenses.

ACKNOWLEDGEMENTS

The paper is supported by the state budget scientific research project of Vasyl’ Stus Donetsk National University “Methods, algorithms and tools of computer-aided design of control units of computing systems” (state registration number 0122U200085).

REFERENCES

1. Bailliul J., Samad T. Encyclopedia of Systems and Control. Springer, London, UK, 2015, 1554 p.
2. Sklyarov V., Sklyarova I., Barkalov A., Titarenko L. Synthesis and Optimization of FPGA-Based Systems; Volume 294 of Lecture Notes in Electrical Engineering. Springer, Berlin, Germany, 2014, 432 p.
3. Baranov S. Logic and System Design of Digital Systems. Tallin, TUTPress, 2008, 267 p.
4. Micheli G. D. Synthesis and Optimization of Digital Circuits. McGraw-Hill, Cambridge, MA, USA, 1994, 579 p.
5. Minns P., Elliot I. FSM-Based Digital Design Using Verilog HDL. JohnWiley and Sons, Hoboken, NJ, USA, 2008, 408 p.
6. Grout I. Digital Systems Design with FPGAs and CPLDs. Elsevier Science, Amsterdam, The Netherlands, 2011, 784 p.
7. Baranov S. Logic Synthesis for Control Automata. Dordrecht, Kluwer Academic Publishers, 1994, 312 p.
8. Barkalov A. A., Babakov R. M. Operational formation of state codes in microprogram automata, *Cybernetics and Systems Analysis*, 2011, Volume 47 (2), pp. 193–197.
9. Xilinx. XST UserGuide. V.11.3. Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/xst.pdf (accessed on 12 April 2023).
10. Barkalov A. A., Titarenko L. A., Babakov R. M. Synthesis of Finite State Machine with Datapath of Transitions According to the Operational Table of Transitions, *Radio Electronics, Computer Science, Control*, 2022, Volume 3 (62), pp. 109–119.
11. Barkalov A. A., Babakov R. M. Determining the Area of Efficient Application of a Microprogrammed Finite-State Machine with Datapath of Transitions, *Cybernetics and Systems Analysis*, 2019, Volume 54 (3), pp. 366–375.
12. Czerwinski R., Kania D. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices. Berlin, Springer, 2013, 172 p.
13. Mano M. Digital design (4th Edition). New Jersey, Prentice Hall, 2006, 624 p.
14. Zwolinski M. Digital System Design with VHDL. Boston, Addison-Wesley Longman Publishing Co., Inc. 2000, 416 p.

Received 04.10.2023.
Accepted 30.11.2023.

DOI 10.15588/1607-3274-2023-4-13

УДК 004.94 : 004.2

СИНТЕЗ VHDL-МОДЕЛІ МІКРОПРОГРАМНОГО АВТОМАТА З ОПЕРАЦІЙНИМ АВТОМАТОМ ПЕРЕХОДІВ

Баркалов О. О. – д-р техн. наук, професор, професор Інституту комп'ютерних наук та електроніки університету Зеленогурського, м. Зельона Гура, Польща.

Тітаренко Л. О. – д-р техн. наук, професор, професор Інституту комп'ютерних наук та електроніки університету Зеленогурського, м. Зельона Гура, Польща.

Бабаков Р. М. – д-р техн. наук, доцент, професор кафедри інформаційних технологій Донецького національного університету імені Василя Стуса, м. Вінниця, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто задачу побудови програмної моделі мікропрограмного автомата з операційним автоматом переходів мовою VHDL. Процес синтезу моделі ототожнюється із синтезом даного типу автомата, оскільки побудована модель може бути використана як для аналізу поведінки пристрою, так і для синтезу логічної схеми в базисі FPGA. Об'єктом дослідження є автоматизований синтез логічної схеми мікропрограмного автомата з операційним автоматом переходів, за результатами якого можуть бути отримані чисельні характеристики апаратурних витрат на реалізацію схеми автомата. Це дозволяє оцінити ефективність використання даної структури мікропрограмного автомата при реалізації заданого алгоритму керування.

Мета. Розробка і дослідження VHDL-моделі мікропрограмного автомата з операційним автоматом переходів для аналізу поведінки автомата та кількісної оцінки апаратурних витрат в його логічній схемі.

Метод. В основу дослідження покладено структурну схему мікропрограмного автомата з операційним автоматом переходів. Синтез окремих блоків структури автомата здійснюється за певною процедурою відповідно до заданої граф-схеми алгоритму керування. Результат синтезу запропоновано представляти у вигляді VHDL-опису, що оснований на фіксованих значеннях кодів станів автомата. Продемонстрований процес синтезу операційного автомата переходів, блоку формування кодів операцій переходів та блоку формування мікрооперацій. VHDL-опис даних блоків здійснюється у синтезованому стилі, що дозволяє провести синтез логічної схеми автомата в базисі FPGA за допомогою сучасних САПР та отримати числові характеристики схеми, зокрема значення апаратурних витрат. Для аналізу коректності роботи синтезованої схеми розглянуто процес розробки поведінкової складової VHDL-моделі, функцією якої є генерація вхідних сигналів автомата. Класичне поєднання синтезованої та поведінкової частин моделі дозволяє представити результати синтезу мікропрограмного автомата з операційним автоматом переходів як окремий проєкт, що може бути використаний в якості структурної складової проєктованої цифрової системи.

Результати. На прикладі абстрактної граф-схеми алгоритму керування розроблено VHDL-модель мікропрограмного автомата з операційним автоматом переходів. За допомогою САПР AMD Vivado проведено синтез розробленої моделі та проведене поведінкове моделювання роботи схеми автомата. Результати синтезу схеми дозволили отримати значення апаратурних витрат при реалізації схеми в базисі FPGA. За результатами поведінкового моделювання отримані диаграми часу, які свідчать про коректність реалізації функцій переходів та виходів синтезованого автомата.

Висновки. У традиційних VHDL-моделях кінцевих автоматів стани не містять конкретних кодів і ідентифікуються за допомогою літералів. Це дозволяє САПР проводити кодування станів на власний розсуд. Однак такий підхід не підходить для опису мікропрограмного автомата з операційним автоматом переходів. Перетворення кодів станів за допомогою множини арифметико-логічних операцій вимагає використання фіксованих значень кодів станів, що визначає специфіку VHDL-моделі, запропонованої в даній роботі. Дана і подібні моделі можуть бути використані, зокрема, при дослідженні ефективності мікропрграмного автомата за критерієм апаратурних витрат в схемі пристрою.

КЛЮЧОВІ СЛОВА: мікропрограмний автомат, операційний автомат переходів, VHDL-модель, апаратурні витрати, САПР AMD Vivado.

ЛІТЕРАТУРА

1. Bailliul J. Encyclopedia of Systems and Control / J. Bailliul, T. Samad. – Springer : London, UK, 2015. – 1554 p.
2. Sklyarov V. Synthesis and Optimization of FPGA-Based Systems; Volume 294 of Lecture Notes in Electrical Engineering / V. Sklyarov, I. Sklyarova, A. Barkalov, L. Titarenko. – Springer: Berlin, Germany, 2014. – 432 p.
3. Baranov S. Logic and System Design of Digital Systems / S. Baranov. – Tallin : TUTPress, 2008. – 267 p.
4. Micheli G. D. Synthesis and Optimization of Digital Circuits / G. D. Micheli. – McGraw-Hill : Cambridge, MA, USA, 1994. – 579 p.
5. Minns, P. FSM-Based Digital Design Using Verilog HDL / P. Minns, I. Elliot. – JohnWiley and Sons : Hoboken, NJ, USA, 2008. – 408 p.
6. Grout, I. Digital Systems Design with FPGAs and CPLDs / I. Grout. – Elsevier Science : Amsterdam, The Netherlands, 2011. – 784 p.
7. Baranov, S. Logic Synthesis for Control Automata / S. Baranov. – Dordrecht : Kluwer Academic Publishers, 1994. – 312 p.
8. Barkalov A.A. Operational formation of state codes in microprogram automata / A. A. Barkalov, R. M. Babakov // Cybernetics and Systems Analysis. – 2011. – Volume 47 (2). – P. 193–197.
9. Xilinx. XST UserGuide. V.11.3. Available online: https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/xst.pdf (accessed on 12 April 2023).
10. Barkalov A. A. Synthesis of Finite State Machine with Datapath of Transitions According to the Operational Table of Transitions / A. A. Barkalov, L. A. Titarenko, R. M. Babakov // Radio Electronics, Computer Science, Control. – 2022. – Volume 3 (62). – P. 109–119.
11. Barkalov A.A. Determining the Area of Efficient Application of a Microprogrammed Finite-State Machine with Datapath of Transitions / A. A. Barkalov, R. M. Babakov // Cybernetics and Systems Analysis. – 2019. – Volume 54 (3). – P. 366–375.
12. Czerwinski R. Finite State Machine Logic Synthesis for Complex Programmable Logic Devices / R. Czerwinski, D. Kania. – Berlin : Springer, 2013. – 172 p.
13. Mano M. Digital design (4th Edition) / M. Mano. – New Jersey : Prentice Hall, 2006. – 624 p.
14. Zwolinski M. Digital System Design with VHDL / M. Zwolinski. – Boston : Addison-Wesley Longman Publishing Co., Inc, 2000. – 416 p.

DOI 10.15588/1607-3274-2023-4-13

UDC 004.912

TECHNOLOGY FOR AUTOMATED CONSTRUCTION OF DOMAIN DICTIONARIES WITH SPECIAL PROCESSING OF SHORT DOCUMENTS

Kungurtsev O. B. – PhD, Professor, Professor of the Software Engineering Department, Odessa Polytechnic National University, Odessa, Ukraine.

Mileiko I. I. – Student of the Software Engineering Department, Odessa Polytechnic National University, Odessa, Ukraine.

Novikova N. O. – PhD, Associate Professor of the Department of Technical Cybernetics and Information Technologies named after professor R.V. Merct, Odessa National Maritime University, Odessa, Ukraine.

ABSTRACT

Context. The task of automating the construction of domain dictionaries in the process of implementing software projects based on the analysis of documents, taking into account their size and presentation form.

Objective. The goal of the work is to improve the quality of the dictionary based on the use of new technology, including special processing of short documents.

Method. A model of a short document is proposed, which presents it in the form of three parts: header, content and final. The header and final parts usually contain information not related to the subject area. Therefore, a method for extracting content based on the use of many keywords has been proposed. The size of a short document (its content) does not allow determining the frequency characteristics of words and, therefore, identifying multi-word terms, the share of which reaches 50% of all terms. To make it possible to identify terms in short documents, a method for their clustering is proposed, based on the selection of nouns and the calculation of their frequency characteristics. The resulting clusters are treated as ordinary documents, since their size allows for the selection of multi-word terms. To highlight terms, it is proposed to select sequences of words containing nouns in the text. Analysis of the frequency of repetition of such sequences allows us to identify multi-word terms. To determine the interpretation of terms, a previously developed method of automated search for interpretations in dictionaries was used.

Results. Based on the proposed model and methods, software was created to build a domain dictionary and a number of experiments were conducted to confirm the effectiveness of the developed solutions.

Conclusions. The experiments carried out confirmed the performance of the proposed software and allow us to recommend it for use in practice for creating dictionaries of the subject area of various information systems. Prospects for further research may include the construction of corporate search systems based on dictionaries of terms and document clustering.

KEYWORDS: domain dictionary, information system, term, clustering, information technology, short document.

ABBREVIATIONS

DD is domain dictionary;
IS is information system;
FCA is Formal Concept Analysis;
LDA is Latent Dirichlet Allocation;
OS is organizational system.

NOMENCLATURE

addr is a location of the document;
Cd_i is a i -th cluster of the content of short documents;
d_j is a document included in the cluster;
Ds is a set of short document;
ds_i is a document;
$Ks_{j,p}$ is a coefficient of proximity of the document d_j with the central document of the cluster Cd_p;
kw_i is a tuple of the set of keywords;
mKw is a lists (sets) of keywords;
Mt_i is a set of one-word terms of the document d_i;
Nc is a number of the last words of the document;
Ncd is a number of words in the document;
Ncorp is a number of documents in the corpus;
Nh is a number of the first words of the document;
nm_i is a number of different terms in the document;
nm_q is a number of occurrences of the term t_q in the document d_i;
ns is a quantity of documents;
nw_i is a size of the document in words;
Nww is a number of “erroneous” words;
Qs is a quality of selection;
Qsa is a quality of separating the content for the corpus of documents;
*r*1 is an index of the first word of the content of the document;
*r*2 is an index of the last word of the content of the document;
Tc_i is a concatenation of the texts of the documents included in it;
$text_i$ is a text of the document;
t_q is a term represented by a noun.

INTRODUCTION

The DD is one of the first artifacts that is created in the process of designing an IS. DD allows the Developer and the Customer to determine a common language of communication [1]. With its help, requirements for IS are formulated, user interfaces are created, instructions are written [2]. Such dictionary is recommended to be used in various subject areas [3]. The creation of DD primarily involves the definition of terms. The simplest way to highlight terms is to study the texts of documents [4] that represent the subject area of IS. Manual text analysis is a very time-consuming process that requires special knowledge in the field of linguistics. Therefore, in recent

DOI 10.15588/1607-3274-2023-4-14

years, more and more attention has been paid to the automated selection of terms and their interpretation [5].

The condition for a reasonable selection of a term is its repeated occurrence in the text of the analyzed document. Only in this case it is possible to distinguish stable phrases. The smaller the document size, the lower the probability of correct selection of terms in it.

Fig. 1 shows the existing technology for building DD, which does not provide for special processing of short documents.

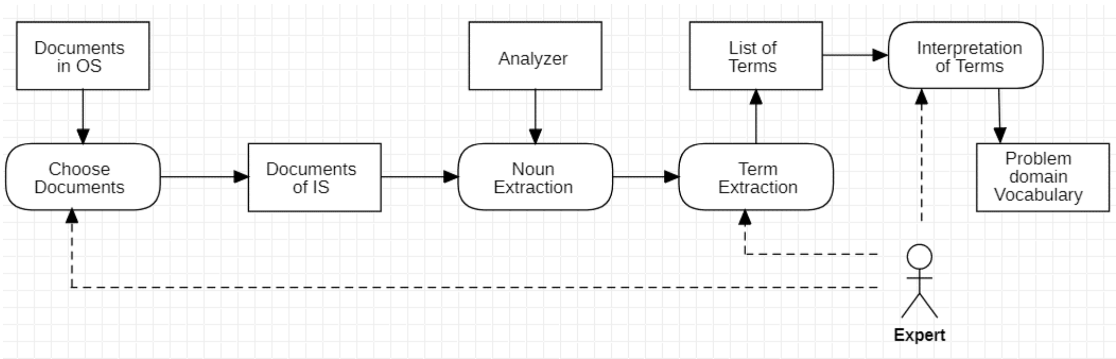


Figure 1 – The existing technology for creating DD

A feature of the corpus of documents used to build DD is many short documents (letters, orders, instructions, etc.) related to different topics, in the structure of which a significant proportion is occupied by the heading and closing parts with terms that do not correspond to the topic of the document. The concept of a short document does not have a clear quantitative definition in terms of isolating terms from its text. Short documents often contain formal header and tail sections that need to be excluded from analysis. To build DD, it is necessary to develop a separate methodology for processing short documents.

1 PROBLEM STATEMENT

Suppose that there are many documents used to develop an automated information system –

$D = Do \cup Ds$, where Do – is a regular document, and Ds – is a short. Since it is impossible to correctly extract a term from a short document, it is necessary to perform a number of transformations on the set Ds:

$Ds \Rightarrow Ds'$, where Ds' – the set of short documents without header and tail parts;

$Ds' \Rightarrow CDs$, where CDs – the set clusters of short documents.

A sufficiently large cluster text can be considered as a regular document. Then $D' = Do \cup CDs$ – a virtual set of documents from which one can get a set of terms – *Term*, that define subject area: $D' \Rightarrow Term$.

2 LITERATURE REVIEW

In [6], the problem of the low frequency of keywords in short documents was noted. An algorithm is proposed that uses the domain ontology to calculate the semantic distance between short documents. The question of ontology formation remained open. In articles [7,8] was proposed a method that uses thematic models that make short text less sparse and more thematically oriented for classification. These methods are difficult to apply to DD, which is created at the first stage of IS design. The task of analyzing short texts is relevant for Web applications such as social networks, forums, and blogs. To solve the problem, an extension of terms, also known as an extension of documents, was proposed [9], based on the classification of texts and their semantic analysis. The paper does not specify the way of obtaining the initial information, but the proposals deserve attention. In [10] a system that can perform automatic summation of several documents using semantic text analysis, clustering, based on the representation of a document as a set of triplets (subject, verb, object) is proposed. The disadvantage of this solution is the rather complicated text analysis and system configuration for a specific user. In [11], dynamic clustering is proposed, which allows you to track the time-varying distribution of topics across documents and words across topics. From the point of view of the formation of DD, the method is difficult to apply due to insufficient time and the number of texts for training. In article [12] is used the concept of weighted similarity scores of terms in Formal Concept Analysis and was explored its effectiveness for expanding terms. It is shown that the weighted measures of similarity of terms, when choosing the appropriate weight value, give a good result. The material is of practical interest and will be partially implemented in this work.

Analysis of the effectiveness of using two approaches to expand terms: weighted measures of similarity, studied in FCA, and a number of measures of correlation, often used in data mining was carried in [13]. It has been empirically shown that the cosine correlation measure is superior to other methods for short documents. The paper [14] describes an experiment comparing short document classifiers based on two methods: Latent Dirichlet Allocation (LDA) and Formal Concept Analysis (FCA). It has been shown that FCA leads to a much higher degree of correlation between terms than LDA, and initially gives a lower classification accuracy. The disadvantage of

DOI 10.15588/1607-3274-2023-4-14

the considered methods is a long and laborious learning process.

In [15], a preliminary clustering method is proposed, which allows one to limit oneself to only a corpus of documents representing the subject area of the designed IS. However, in this work, as well as in [16, 17], there is no clear definition of a short document.

Short documents used to build DD are significantly different from short texts on the Internet [18]. The workflow of most organizational structures is dominated by formalized documents [19] with heading and closing parts. The need to highlight the meaningful part of a short document was noted in [15], but a clear algorithm for solving this problem is not presented.

3 MATERIALS AND METHODS

Short document model. In accordance with [15], we will assume that the corpus of all documents under study is presented in the form

$$D = \{d_i\},\ \ i = 1, n. \tag{1}$$

Let's extend the previously used model to represent a single document. Since the documents based on which the DD is built can be located on different computers, disks and directories in the Customer's organization, it is necessary to introduce the concept of an address for a document. To highlight the content of the document, you should limit the search area of the heading and final parts of the document. Thus, document can be represented by a tuple

$$D_i = \langle addr, text_i, r1, r2, nw_i, Mt_i \rangle, \tag{2}$$

$$Mt_i = \{\langle t_q, nm_q \rangle\},\ \ q = 1, nm_i. \tag{3}$$

To apply the model, it was necessary to clarify the concept of a "short document". As a result of the experiments (see the Results section), it was proposed to consider a document up to 1400 words as short. Thus, the set of short documents can be represented as

$$Ds = \{d_i \mid d_i \in D \wedge d_i.nw_i \le 1400\},\ i = 1, ns. \tag{4}$$

A method for highlighting the content of a short document. To highlight the meaningful part, it is necessary to have signs of the heading and closing parts. Such signs are certain "keywords" of these parts of the document. It should be noted that the list of all formalized documents for a certain state has hundreds of names. It is undesirable to use such a list in the algorithm for highlighting the content of a document, since the interpretation of keywords from one subject area may not coincide with their interpretation in another. Therefore, it makes sense to compose sets of keywords for the heading and closing parts in relation to the subject area (perhaps in the narrow sense of the word).

In general, within the framework of one project of *IS*, several lists of *mKw* (sets) of keywords can be created:

$$mKw = \{kw_i\}\ ,\ \ i = 1, q, \tag{5}$$

where $kw_i = \langle mKwHead, mKwEnd \rangle$ is a tuple of the set of keywords in the header part of documents; $mKwHead_i = \{ w_{i,j} \}, j = 1, qh_i$ and the corresponding set of keywords of the final part of the documents; $mKwEnd = \{ w_{i,j} \}, j = 1, qe_i$.

For example, for the personnel department of a university, the set *mKw* will look like:

$$mKw = \{\{labor\ contract, order\}, \{signature, date\},\ldots,\{\}\ \}.$$

Thus, a short document can be presented in the form

$$Ds_i = \langle mWhead, mWcontent, mWend \rangle, \tag{6}$$

where the heading is *mWhead* , represented by an ordered set of words

$$mWhead = \{w_1, w_2, \ldots, w_i, \ldots, w_p\},$$

while $\exists\ w_i \mid w_i \in mKwHead \wedge w_i \in mWhead$;
and the final part *mWend* , represented by an ordered set of words

$mWend = \{w_r, w_{r+1}, \ldots, w_j, \ldots, w_z\}$, at the same time .$\exists\ w_j \mid w_j \in mKwEnd \wedge w_j \in mWEnd.$

For example, the following heading keywords can be distinguished from personnel documentation: "Agreement", "Order", "Card", "Order", "Time sheet", "Statement", "Act", "Schedule", "Note" and the following the words of the final part: "Signature"/"Signatures", "Acquainted", "Approve".

To highlight the content part, you need to determine its first and last words. To do this, it is proposed to determine the possible boundaries of the heading and closing parts by searching for terms from *mKw*. Terms from *mKwHead* are searched from the beginning to the end of the document, and terms from *mKwEnd* are searched from the end to the beginning of the document. It is proposed to limit the search area of the heading and final parts of the document to reduce the probability of errors in the case when the document does not belong to the category of formalized ones. In addition, limiting the search area reduces the time for document analysis.

It is not possible to analytically determine the boundaries of the header and footer search for many different documents, so experimental studies were carried out on two sets of documents of various formats. 74 documents in Russian and 68 documents in English from the trade organization's workflow were processed. The number of words for highlighting the heading part *Nh* and the closing part *Nc* was set equal to 5, 15, 25, 30, 35 and 50. The quality of highlighting the content part was assessed by an expert depending on the number of "extra" and "missing" words in the content part. Under the quality

DOI 10.15588/1607-3274-2023-4-14

of selection for a separate document, it is proposed to understand *Qs*, calculated by the formula:

$$Qs = \frac{100 \times (Ncd - Nww)}{Ncd}. \quad (7)$$

Quality of separating the content for the corpus of documents for certain values of *Nh* and *Nc* by the formula

$$Qsa = \left| \frac{\sum_{i=1}^{Ncorp} Qs_i}{Ncorp} \right|. \quad (8)$$

It has been experimentally shown that the best value for *Nh* is 35, and for *Nc* – 25 (see the Results section). Let us formulate the steps of the method.

– Find among the first *Nh* words of the document the word $w_i \mid w_i \in mWhead$. If the word is found, then the index of the first word of the substantive part $StartInd = i+1$ is, otherwise $StartInd = 0$.

– Find the word among the last *Nc* words of the document $w_j \mid w_j \in mWend$. If the word is found, then the index of the last word of the substantive part $EndInd = j - 1$ is, otherwise $EndInd = Ncd$.

– Crop the document at the edges – before the index *fInd* and after the index *lInd* .

– Find how many characters need to be further indented after the beginning of the truncated document to remove lines that have less than five words or less than 50 characters.

– Crop the document according to the received data.

Clustering short documents. In accordance with a previous study (Fig. 2), in short documents the average frequency of repetition of nouns is low, which does not allow to qualitatively distinguish verbose terms. Therefore, it is proposed to define terms not within a single document, but within a cluster of short documents. For this purpose, the preliminary clustering method was used [15]. The method allows you to calculate the proximity coefficient $K_{i,j} = K_{1i,j} + \gamma * K_{2i,j}$ of documents d_i and d_j based on the relative number of matching nouns (component $K_{1i,j}$) and the frequency of matching nouns (component $\gamma * K_{2i,j}$). Optimization of the composition of clusters is ensured by adjusting their composition depending on the proximity of the document to the cluster core.

Let us represent the clustering process in the form $Ds \xrightarrow{clustering} \{Cd_i\}$.

The practical use of the short document clustering method [13] showed that after the completion of clustering based on kernels, several clusters Cd_i are formed that contain only one document, that is

$$Cd_i = \{d_j\}, j = 1. \quad (9)$$

Therefore, an additional stage is introduced into the method, at which for each document *dj* a cluster Cd_p is found to which it can be attached.

$$d_j \xrightarrow{add} Cd_p.$$

In this case, the next condition is fulfilled

$$Ks_{j,p} = \max_{q=1,n;q \neq j} Ks_{j,q}.$$

Extracting terms from a cluster

Multiword terms make up a significant part of all terms. The work [20] shows that 29.13% of the terms from the Internet request contain three or more words. In the documents of organizations, terms containing two and three words make up about 50% of all terms. This determines the need to extract multiword terms from short documents.

Let us represent some cluster Cd_i in the form

$$Cd_i = \{d_j\}, j = 1, n_j.$$

Let's form the text of the cluster Cd_i.

$Tc_i = \cup_1^{n_j} d_j$, as a concatenation of the texts of the documents included in it.

Let's represent the cluster text as a sequence of elements

$$Tc_i = e_1 e_2 \ldots e_k \ldots e_{q-1} e_q.$$

The text element can be a word or a punctuation mark.

Let us denote the text element corresponding to the noun as eN, and the text of the cluster as

$$Tc_i = e_1 e_2 \ldots e_{k-2}\, e_{k-1}\, eN_k e_{k+1}\, e_{k+2}\, eN_{k+3}\, e_{k+4} \ldots e_{q-1} e_q.$$

To highlight multiword terms in each sentence of the text, sequences of words containing nouns are selected. Let there be some sequence of words, which is bounded on the left and right by punctuation marks:

$$S = e_{k-2}\, e_{k-1}\, eN_k e_{k+1}\, e_{k+2}\, eN_{k+3}\, e_{k+4}.$$

If we take a noun eN_k as a base, standing at the beginning or end of a term, the following sequences of words, which can be terms will be selected from *S*: $e_{k-2}\, e_{k-1}\, eN_k$, $e_{k-1}\, eN_k$, eN_k , $eN_k\, e_{k+1}$, $eN_k\, e_{k+1}\, e_{k+2}$, $eN_k\, e_{k+1}\, e_{k+2}\, eN_{k+3}$, $eN_k\, e_{k+1}\, e_{k+2}\, eN_{k+3}\, e_{k+4}$ At the same time, the term cannot begin and end with a union, a preposition and a numeral, and these parts of speech are not considered "important". In this work, it was decided to limit the length of the term to three "important" words.

Definition of interpretations of the term.

Defining definitions for terms is a long and laborious process [21], which it is desirable to automate [22] . Since such a task is beyond the scope of this study, it is proposed to use a ready-made solution [23], which provides a detailed analysis of a dictionary entry, automated removal of redundant definitions, and a simple procedure for expanding the dictionary bank (software product DictionaryOfInterpretations).

Technology for creating DD with separate processing of short documents. Given the large number of short documents in the corpus of documents to be analyzed for the construction of DDs, it is proposed to introduce additional procedures for processing short documents (Fig. 2).

DOI 10.15588/1607-3274-2023-4-14

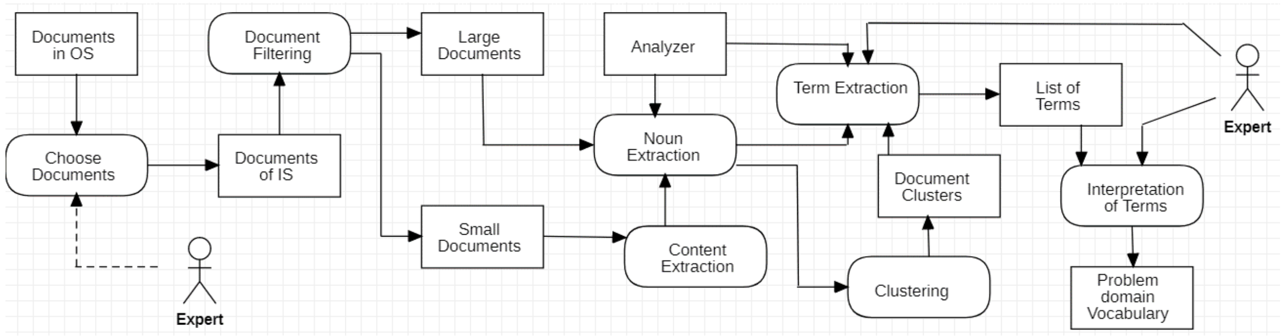


Figure 2 – New technology for building DD

According to Figure 2, the procedure for building an DD is as follows:

– The expert (representative of the customer) selects documents of the organizational system (*OS*) that are of interest to the designed *IS*.

– As a result of filtering, short documents are selected from the entire corpus of documents.

– From short documents, their substantive part stands out.

– Using the analyzer, nouns are highlighted in texts and the number of their occurrences in documents is counted.

– For short documents, clustering is performed, because of which clusters are formed according to the principle of belonging to one topic.

– From documents (large) and clusters of documents (short), terms (generally multiword) are distinguished. The expert analyzes and edits the list of terms in terms of their belonging to the subject area of the projected IS.

– Based on the received list of terms, the user himself or with the help of an external system performs the interpretation of the terms.

4 EXPERIMENTS

Development of a software product. To implement the proposed technology, the software product TerEx was developed , the general class diagram of which is shown in Fig. 3.

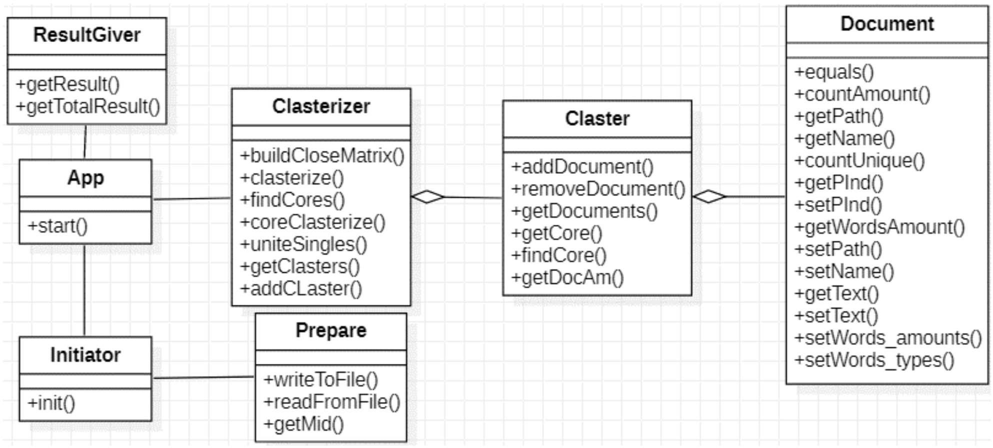


Figure 3 – Class diagram of the TerEx system

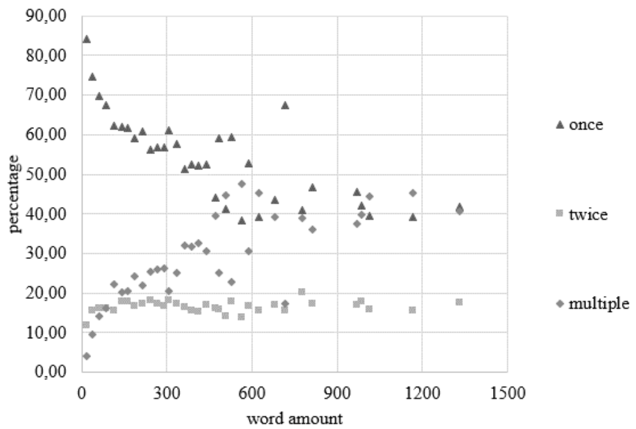


Figure 4 – Graph of the dependence of the frequency of repetition of nouns on the length of the document

DOI 10.15588/1607-3274-2023-4-14

TerEx allows you to perform the following actions on documents:

– highlight individual words, conduct their morphological analysis, count the number of occurrences in texts;

– highlight the content of the document;

– perform clustering of documents;

– highlight single-word and multi-word terms from the text.

5 RESULTS

Definition of a short document. We studied a corpus of 381 documents in Ukrainian, English and Russian, containing from 15 to 1332 nouns. The results of the experiment are presented in the form of a graph (Fig. 4), based on which it was concluded that documents up to 1300–1400 words in size should be considered short. The experiment showed that the subject area and the language of the documents do not have a noticeable effect on the results obtained.

Determining the quality of highlighting the content of a short document.

To determine the dependence of the quality of highlighting the content of the document on the values of *Nh* and *Nc*, 74 short documents were analyzed. The results are shown in Table 1. The best quality is achieved with a limit of 35 words from the beginning of the document and 25 words from the end of the document.

Table 1 – Dependence of the quality of highlighting the meaningful part on the values of *Nh and Nc*

Nh	*Qs*	*Nc*	*Qs*
35	97.14	25	95.64
30	96.16	35	95.41
25	96.06	30	94.99
15	89.16	15	89.79
5	84.16	5	82.98
50	82.37	50	824

After setting these values, the average correctness of extracting the meaningful part of 96.39% was obtained.

Determining the quality of clustering. When requesting clustering, a new folder "ClusterizationResultFolder" is created in the directory selected for displaying the result, in which the file "_ totalTerms . txt ", containing a general list of terms from all clusters, as well as new folders – Cluster _1 ... Cluster _ N , where N is the resulting number of clusters is created. An example of the contents of the "ClusterizationResultFolder" directory and the structure of the files before they are processed is shown in Figure 5. In the folder of each cluster is the file "_ terms.txt", containing a list of terms in this cluster, as well as copies of all documents included in this cluster.

104 documents were subject to analysis. Five clusters were found. Analysis of the result showed that the documents in the selected clusters are really close to each other in terms of the nouns they contain. No errors were found.

Determining the quality of highlighting multi-word terms. A comparative analysis of the results of the work of the TerEx product and the well-known online service for text analysis SketchEngine [24] was carried out. A corpus of 100 documents was studied. The results of the experiment are presented in Table 2.

Defective terms are those that contain extraneous characters, cut words, prepositions, or are not words at all.

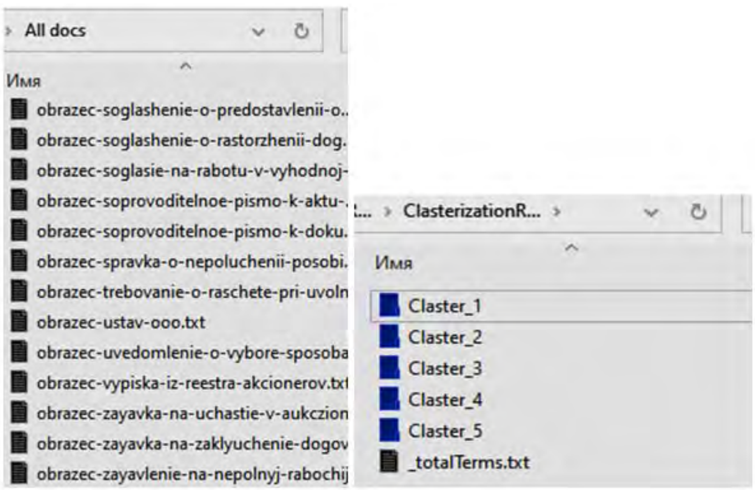


Figure 5 – File structure before (left) and after (right) processing

DOI 10.15588/1607-3274-2023-4-14

Table 2 – Comparative analysis of the quality of term selection

	TerEx	SketchEngine
Number of highlighted terms	11119	15315
Number of defective terms	585	2095
Error Percentage	5.26%	13.68%
Elapsed time	4 minutes 45 seconds	5 minutes 12 seconds
File limit	No limits	100 files
License	Is free	Paid subscription
Distribution of documents by clusters	Performed	Not supported

6 DISCUSSION

The experiments carried out made it possible to quantify a short document in terms of specific requirements for its analysis. The conditions for the implementation of the method of highlighting the content of a short document were identified, and its effectiveness was shown (96.39%). The implemented clustering of short documents can be of independent importance, for example, in information retrieval tasks. However, in this study, it allowed to significantly improve the quality of multiword selection, which was confirmed by the results of comparative tests of TerEx and Sketch Engine (Table 2).

CONCLUSIONS

An information technology for constructing a dictionary of the subject area has been developed, which provides for special processing of short documents.

The scientific novelty of the research lies in:

– improvement of the mathematical model of a short document, by introducing indexing of the beginning and end of the content, as well as the address of the location of the document, which made it possible to further formalize operations for its processing;

– obtaining an experimentally substantiated definition of a short document, which allows you to sort documents quickly and efficiently for further processing.

– development of a method for highlighting the content of a short document that implements the exclusion from the document of the heading and ending parts that contain terms not related to the subject of the document, which made it possible to further improve the quality of *DD*;

– improvement of the method of preliminary clustering of short documents by introducing an additional stage of merging clusters containing 1 document, which made it possible to increase the frequency of terms in clusters

The practical value of the work lies in combining the model and methods into a single technology for creating *DD*.

The conducted experiments confirmed the effectiveness of the theoretical results of the work.

The practical implementation of models and methods can be used to create *DD* in various subject areas.

REFERENCES

1. Larman K. Primenenie UML 2.0 i shablonov proektirovanija. Prakticheskoe rukovodstvo. 3-e izdanie. Moscow, Izdatel'skij dom "Vil'jams", 2013, 736 p. [in Russian].
2. Bourgeois D., Mortati J., Wang S., et al. Information Systems for Business and Beyond. Information systems, their use in business, and the larger impact they are having on our world [Electronic resource]. Access mode: https://opentextbook.site/exports/ ISBB-2019.pdf
3. Artamonov A., Kshnyakov D., Danilova V. et al. Methodology for the Development of Dictionaries for Automated Classification System, *8th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA 2017). Procedia Computer Science Volume:123*. Moscow, Russia, 1–6 August 2017, pp. 57–62. doi:10.1016/j.procs.2018.01.010
4. Dalglish S. L., Khalid H., McMahon S. A. Document analysis in health policy research: the READ approach, *Health Policy and Planning*, 2020, Vol. 35, Issue 10, pp. 1424–1431.
5. Cheng Y. Huang Y. Research and development of domain dictionary construction system, *Proceedings of the International Conference on Web Intelligence, August* 2017, pp. 1162–1165. https://doi.org/10.1145/3106426.3109046
6. Liang S., Yilmaz E., Kanoulas E. Dynamic Clustering of Streaming Short Documents, *International Conference on Knowledge Discovery and Data Mining, August 2016*, pp. 995–1004.
7. Wang Y., Yang S. Outlier detection from massive short documents using domain ontology. *International Conference on Intelligent Computing and Intelligent Systems, 29–31 Oct. 2010*, Xiamen, China. DOI: 10.1109/ICICISYS.2010.5658426
8. Shi T., Kang K., Choo J. et al. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations, *WWW '18: Proceedings of The Web Conference 2018, April 2018*. Lyon, France. DOI: 10.1145/3178876.3186009
9. Hafeez R., Khan S., Abbas M. et al. Topic based Summarization of Multiple Documents using Semantic Analysis and Clustering, *International Conference on Smart; 8–10 Oct. 2018*. Islamabad, Pakistan. DOI: 10.1109/HONET.2018.8551325
10. Vo D-T., Ock C-Y. Learning to classify short text from scientific documents using topic models with various types of knowledge, *Expert Systems with Applications: An International Journal*, 2015, V. 42, Issue 3, pp. 1684–1698. https://doi.org/10.1016/j.eswa.2014.09.031

DOI 10.15588/1607-3274-2023-4-14

11. Liang S., Yilmaz E., Kanoulas E. Dynamic Clustering of Streaming Short Documents, *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, US, 13 August 2016*, pp. 995–1004. https://doi.org/10.1145/2939672.2939748
12. Seki H., Toriyama S. On Term Similarity Measures for Short Text Classification, *11th International Workshop on Computational Intelligence and Applications (IWCIA), 9–10 Nov.2019*, pp. 53–58, DOI: 10.1109/IWCIA47330.2019.8955045.
13. Seki H., Toriyama S. Using term similarity measures for classifying short document data, *International Journal of Computational Intelligence Studies*, Vol. 10, Issue 2–3, https://doi.org/10.1504/IJCISTUDIES.2021.115430
14. Rogers N., Longo L. A. Comparison on the Classification of Short-text Documents Using Latent Dirichlet Allocation and Formal Concept Analysis, *25th Irish Conference on Artificial Intelligence and Cognitive Science, AICS 2017*. Dublin, Ireland, 7–8 December 2017, V. 2086, pp. 50–62.
15. Kungurtsev O., Zinovatna S., Potochniak I. et al. Development of Methods for Pre-clustering and Virtual Merging of Short Documents for Building Domain Dictionaries, *Eastern-european Journal of Enterprise Technologies*, 2020, Vol. 5, № 2 (107), pp. 39–47. http://doi.org/10.15587/1729-4061.2020.215190
16. Kungurtsev O., Zinovatnaya S., Potochniak Ia. et al. Development of information technology of term extraction from documents in natural language, *Eastern-European Journal of Enterprise Technologies*, 2018, V. 6, №2 (96), pp. 44–51. doi: https://doi.org/10.15587/1729-4061.2018.147978
17. García R. G., Beltrán B., Vilariño D. et al. Comparison of Clustering Algorithms in Text Clustering Tasks, *Computación y Sistemas*, 2021, Vol. 24, № 2. https://doi.org/10.13053/cys-24-2-3369
18. Shevchenko A. Organizacija elektronnogo dokumentoobigu na pidpryjemstvi [Electronic resource]. Access mode: https://uteka.ua/ua/publication/commerce-12-dokumentooborot-2-organizaciya-elektronnogo-dokumentooborota-na-predpriyatii [in Ukrainian].
19. Typova instrukcija z dilovodstva v ministerstvah, inshyh central’nyh ta miscevyh organah vykonavchoi’ vlady [Electronic resource]. Access mode: https://borispol-rada.gov.ua/item/39961-typova-instruktsiya-z-dilovodstva-v-ministerstvakh-inshykh-tsentralnykh-ta-mistsevykh-orhanakh-vykonavchoi-vlady.html [in Ukrainian].
20. Lions K. Long-Tail Keywords: What They Are & How to Use Them for SEO. [Electronic resource]. Access mode: https://www.semrush.com/blog/how-to-choose-long-tail-keywords/
21. Borysova, N. V., Kanyshheva O. V., Kanyshheva O. V. The formation of problem domain dictionary, *Eastern-European Journal of Enterprise Technologies*, 2013. Vol. 5, №3(65), pp. 16–19. https://doi.org/10.15587/1729-4061.2013.18
22. Rahoo L. A., Unar M. A. Design and Development of an Automated Library Management System for Mehran University Library, Jamshoro, *Control Theory and Informatics*, 2016, № 6(1), pp. 1–6.
23. Kungurtsev O., Novikova N., Kozhushan M. Automation of Serching for Terms in the Explanatory Dictionary, *Proceedings of Odessa Polytechnic University*, 2020, № 3(62), pp. 91–100. DOI: 10.15276/opu.3.62.2020.11
24. Sketch Engine. [Electronic resource]. Access mode: https://www.sketchengine.eu/

Received 14.09.2023.
Accepted 06.11.2023.

УДК 004.912

ТЕХНОЛОГІЯ АВТОМАТИЗОВАНОГО ПОБУДУВАННЯ СЛОВНИКІВ ПРЕДМЕТНОЇ ГАЛУЗІ ЗІ СПЕЦІАЛЬНОЮ ОБРОБКОЮ КОРОТКИХ ДОКУМЕНТІВ

Кунгурцев О. Б. – канд. техн. наук, професор кафедри Інженерії програмного забезпечення Національного університету «Одеська політехніка», м. Одеса, Україна.

Милейко І. І. – студентка кафедри Інженерії програмного забезпечення Національного університету «Одеська політехніка», м. Одеса, Україна.

Новікова Н. О. – канд. техн. наук, доцент кафедри Технічна кібернетика й інформаційні технології ім. професора Р. В. Меркта Одеського національного морського університету, м. Одеса, Україна.

АНОТАЦІЯ

Актуальність. Розглянуто завдання автоматизації побудови словників предметної галузі у процесі виконання програмних проектів на основі аналізу документів з урахуванням їх розміру та форми подання.

Мета роботи – підвищення якості словника на основі застосування нової технології, що включає спеціальну обробку коротких документів.

Метод. Пропонується модель короткого документа, яка представляє його у вигляді трьох частин: заголовної, змістовної та заключної. У заголовній і заключній частинах зазвичай міститься інформація, що не має відношення до предметної області. Тому запропоновано метод виділення змістовної частини, заснований на використанні множини ключових слів. Розмір короткого документа (його змістовної частини) не дозволяє визначити частотні характеристики слів і виявити багатослівні терміни, частка яких сягає 50% від усіх термінів. Для забезпечення можливості виділення термінів у коротких документах запропоновано метод їх кластеризації, заснований на виділенні іменників та обчисленні їх частотних характеристик. Утворені кластери розглядаються як звичайні документи, оскільки їхній розмір дозволяє виділяти багатослівні терміни. Для виділення термінів запропоновано виділяти в тексті послідовності слів, що містять іменники. Аналіз частот повторення таких послідовностей дозволяє визначити багатослівні терміни. Для визначення тлумачення термінів використано раніше розроблений метод автоматизованого пошуку тлумачень у словниках.

Результати. На основі запропонованої моделі та методів створено програмне забезпечення для побудови словника предметної галузі та проведено низку експериментів, що підтверджують ефективність розроблених рішень.

Висновки. Проведені експерименти підтвердили працездатність запропонованого програмного забезпечення та дозволяють рекомендувати його до використання на практиці для створення словників предметної галузі різних

DOI 10.15588/1607-3274-2023-4-14

інформаційних систем. Перспективи подальших досліджень можуть включати побудову корпоративних пошукових систем на основі словників термінів та кластеризації документів.

КЛЮЧОВІ СЛОВА: словник предметної галузі, інформаційна система, термін, кластеризація, інформаційна технологія, короткий документ.

ЛІТЕРАТУРА

1. Larman K. Primenenie UML 2.0 i shablonov proektirovanija. Prakticheskoe rukovodstvo. 3-e izdanie / K. Larman. – M. : Izdatel'skij dom "Vil'jams", 2013. – 736 p. [in Russian].
2. Information Systems for Business and Beyond. Information systems, their use in business, and the larger impact they are having on our world [Electronic resource] / [D. Bourgeois, J. Mortati, S. Wang, et al.] – Access mode: https://opentextbook.site/exports/ ISBB-2019.pdf
3. Methodology for the Development of Dictionaries for Automated Classification System / [A. Artamonov, D. Kshnyakov, V. Danilova et al.] // 8th Annual International Conference on Biologically Inspired Cognitive Architectures (BICA 2017). Procedia Computer Science Volume:123. Moscow, Russia, 1–6 August 2017. – P. 57–62. doi:10.1016/j.procs.2018.01.010
4. Dalglish S. L. Document analysis in health policy research: the READ approach / S. L. Dalglish, H. Khalid, S. A. McMahon // Health Policy and Planning. – 2020. – V. 35, Issue 10. – P. 1424–1431.
5. Cheng Y. Research and development of domain dictionary construction system / Y. Cheng, Y. Huang // Proceedings of the International Conference on Web Intelligence, August 2017. – P.1162–1165. https://doi.org/10.1145/3106426.3109046
6. Liang S. Dynamic Clustering of Streaming Short Documents / S. Liang, E. Yilmaz, E. Kanoulas // International Conference on Knowledge Discovery and Data Mining. August 2016. – P. 995–1004.
7. Wang Y. Outlier detection from massive short documents using domain ontology / Y. Wang, S. Yang // International Conference on Intelligent Computing and Intelligent Systems, 29–31 Oct. 2010, Xiamen, China. doi: 10.1109/ICICISYS.2010.5658426
8. Short-Text Topic Modeling via Non-negative Matrix Factorization Enriched with Local Word-Context Correlations. / [T. Shi, K. Kang, J. Choo et al.] // WWW '18: Proceedings of The Web Conference 2018, April 2018, Lyon, France. doi: 10.1145/3178876.3186009
9. Topic based Summarization of Multiple Documents using Semantic Analysis and Clustering / [R. Hafeez, S. Khan, M. Abbas et al] // International Conference on Smart; 8–10 Oct. 2018, Islamabad, Pakistan. doi: 10.1109/HONET.2018.8551325
10. Vo D-T. Learning to classify short text from scientific documents using topic models with various types of knowledge / D-T. Vo, C-Y. Ock // Expert Systems with Applications: An International Journal. – 2015. – V. 42, Issue 3. – P. 1684–1698. https://doi.org/10.1016/j.eswa.2014.09.031
11. Liang S. Dynamic Clustering of Streaming Short Documents / S. Liang, E. Yilmaz, E. Kanoulas // 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, US, 13 August 2016. – P. 995–1004. https://doi.org/10.1145/2939672.2939748
12. Seki H. On Term Similarity Measures for Short Text Classification / H. Seki, S. Toriyama // 11th International Workshop on Computational Intelligence and Applications (IWCIA), 9–10 Nov. 2019. – P. 53–58. DOI: 10.1109/IWCIA47330.2019.8955045.
13. Seki H. Using term similarity measures for classifying short document data / H. Seki, S. Toriyama // International Journal of Computational Intelligence Studies. – Vol. 10, Issue 2–3, https://doi.org/10.1504/IJCISTUDIES.2021.115430
14. Rogers N. Comparison on the Classification of Short-text Documents Using Latent Dirichlet Allocation and Formal Concept Analysis / N. Rogers, L. A. Longo // 25th Irish Conference on Artificial Intelligence and Cognitive Science, AICS 2017, Dublin, Ireland, 7–8 December 2017. – V. 2086. – P. 50–62.
15. Development of Methods for Pre-clustering and Virtual Merging of Short Documents for Building Domain Dictionaries / [O. Kungurtsev, S. Zinovatna, I. Potochniak et al.] // Eastern-european Journal of Enterprise Technologies. – 2020. – Vol. 5, № 2 (107). – P. 39–47. http://doi.org/10.15587/1729-4061.2020.215190
16. Development of information technology of term extraction from documents in natural language / [O. Kungurtsev, S. Zinovatnaya, Ia. Potochniak et al.] // Eastern-European Journal of Enterprise Technologies. – 2018. – V. 6, №2 (96), – P. 44–51. doi: https://doi.org/10.15587/1729-4061.2018.147978
17. Comparison of Clustering Algorithms in Text Clustering Tasks. / [R. G. García, B. Beltrán, D. Vilariño et al.] // Computación y Sistemas. – 2021. – Vol. 24, № 2. https://doi.org/10.13053/cys-24-2-3369
18. Шевченко А. Організація електронного документообігу на підприємстві [Electronic resource] / А. Шевченко. – Access mode: https://uteka.ua/ua/publication/commerce-12-dokumentooborot-2-organizaciya-elektronnogo-dokumentooborota-na-predpriyatii
19. Типова інструкція з діловодства в міністерствах, інших центральних та місцевих органах виконавчої влади [Electronic resource]. – Access mode: https://borispol-rada.gov.ua/item/39961-typova-instruktsiya-z-dilovodstva-v-ministerstvakh-inshykh-tsentralnykh-ta-mistsevykh-orhanakh-vykonavchoi-vlady.html
20. Lions K. Long-Tail Keywords: What They Are & How to Use Them for SEO. [Electronic resource] / K. Lions. – Access mode: https://www.semrush.com/blog/how-to-choose-long-tail-keywords/
21. Borysova, N. V. The formation of problem domain dictionary / N. V. Borysova, O. V. Kanyshheva, O. V. Kanyshheva // Eastern-European Journal of Enterprise Technologies. – 2013. – Vol. 5, №3(65). – P. 16–19. https://doi.org/10.15587/1729-4061.2013.18
22. Rahoo L. A. Design and Development of an Automated Library Management System for Mehran University Library, Jamshoro. / L. A. Rahoo, M. A. Unar // Control Theory and Informatics. – 2016. – № 6(1). – P. 1–6.
23. Kungurtsev O. Automation of Serching for Terms in the Explanatory Dictionary / O. Kungurtsev, N. Novikova, M. Kozhushan // Proceedings of Odessa Polytechnic University. – 2020. – № 3(62). – P. 91–100. DOI: 10.15276/opu.3.62.2020.11
24. Sketch Engine. [Electronic resource]. – Access mode: https://www.sketchengine.eu/

DOI 10.15588/1607-3274-2023-4-14

УДК 004.9

ТЕХНОЛОГІЯ ПОРОДЖЕННЯ ПРОДОВЖЕННЯ ПІСЕНЬ НА ОСНОВІ СТРАТЕГІЙ ГЕНЕРАЦІЇ ТЕСТУ, TEXTMINING І МОВНОЇ МОДЕЛІ T5

Медяков О. О. – магістр кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

Висоцька В. А. – канд. техн. наук, доцент, доцент кафедри «Інформаційні системи та мережі», Національний університет «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Наперед навчені великі мовні моделі, на сьогодні – це локомотив розвитку не лише NLP, а й систем глибинного навчання загалом. Моделі трансформери здатні розв'язувати фактично усі задачі, що наразі існують, за умови виконання певних вимог та практик їх навчання. В свою чергу, слова, речення та тексти є базовим і найважливішим способом комунікації між інтелектуально розвиненими істотами. Звичайно, мовлення і тексти використовуються для донесення певних емоції, подій тощо. Один з основних напрямків використання мови для опису пережитих емоцій – це пісні з текстами. Проте, часто через необхідність збереження рими та римування, розмірності віршових рядків, структури пісні тощо, артистам приходиться використовувати повторення рядків у текстах. Крім того, процес написання текстів може бути тривалим.

Метою дослідження є розробка інформаційної технології генерації продовження текстів пісень на основі моделі машинного навчання T5 з (SA, specific author) та без (NSA, non-specific author) врахування стилю автора.

Метод. Для процесу генерації важливим є питання вибору стратегії декодування. Проте, замість того, щоб надати перевагу конкретній стратегії, у системи буде підтримка множини стратегій. Зокрема такі 8 стратегій: Contrastive search, Top-p sampling, Top-k sampling, Multinomial sampling, Beam search, Diverse beam search, Greedy search, та Beam-search multinomial sampling.

Результати. Розроблено модель машинного навчання для генерації продовження текстів пісень з допомогою великих мовних моделей, зокрема моделі T5, для прискорення, доповнення та підвищення гнучкості процесу написання текстів до пісень.

Висновки. Створена модель показує відмінні результати генерації продовження текстів пісень на тестових даних. Аналіз вихідних даних показав, що модель NSA має менш деградаційні результати, а для моделі SA необхідно збалансовувати кількість тексту для кожного автора. Обраховано кілька текстових метрик як BLEU, RougeL та RougeN для кількісного порівняння результатів моделей та стратегій генерування. Значення метрики BLEU є найрізноманітнішим, і його значення значно змінюється, залежно від стратегії. При цьому Rouge метрики мають меншу варіативність, менший розмах значень. Для порівняння використано 8 різних методик декодування для генерації тексту, що підтримуються бібліотеку transformers. З усіх результатів порівняння текстів видно, що метрично найкращим методом генерації текстів пісень є beam пошук та його варіації, зокрема променевий семплінг. Contrastive search зазвичай перевершував звичайний жадібний підхід. Методи top-p та top-k не мають однозначної переваги один на одним, і в різних ситуаціях давали різні результати.

КЛЮЧОВІ СЛОВА: генерація тексту, мовна модель T5, Transformers, стиль автора, Contrastive search, Top-p sampling, Top-k sampling, Multinomial sampling, Beam search, Diverse beam search, Greedy search, та Beam-search multinomial sampling.

АБРЕВІАТУРА

ІС – інтелектуальна система;
ІТ – інформаційна технологія;
ШІ – штучний інтелект;
ПО – предметна область;
BLEU – Bilingual Evaluation Understudy;
CNN – Convolutional Neural Network;
LCS – Longest Common Subsequence;
ML – machine learning;
NLP – Natural Language Processing;
NSA – Non-Specific Author;
RNN – Recurrent Neural Network;
SA – Specific Author;
T5 – Text-To-Text Transfer Transformer.

НОМЕНКЛАТУРА

T – довжина рядка, та кількість кроків генерації;

t – поточний крок;

w_t – слово (гіпотеза), отримане на кроці t;

$w_{<t}$ – підланцюжок від початкового (першого) до t–1 слова (гіпотези);

$w_{k:t}$ – підланцюжок від i до t слова (гіпотези) включно, при цьому $t,k \in [1, T]$;

V – словник, тобто множина можливих слів;

W_0 – спеціальний початкове слово (токен, гіпотеза) ланцюжка;

EOS – символ завершення ланцюжка (з англ. end of sequence), де W_0, $EOS \in V$;

n – гіперпараметр як кількість (number of beams), розмір (beam size) або ширина (beam width) променю;

C_t – множина всіх можливих новий стрічок на основі попередньої множини гіпотез W_{t-1};

α – штраф;

s – функція, косинусоїдна подібність векторів;

h_{wi} – векторне представлення обраховане моделлю для конкатенованого рядка $w_{<t}\circ w_i$.

ВСТУП

Стратегія генерації тексту або метод декодування тексту в мовних моделях – це алгоритм вибору наступного слова послідовності із результатів мовних моделей, зважаючи на попередні слова [1–2]. Існування різних підходів до декодування зумовлене різноманітністю завдань, призначень та архітектури моделей, а також розвитком методологій в NLP [3–4].

DOI 10.15588/1607-3274-2023-4-15

Метою дослідження є розробка інформаційної технології генерації продовження текстів пісень моделлю Т5 з та без врахування стилю автора. Відповідно до мети, основними задачами роботи є наступні:

– створення навчальної та тестової вибірки даних, їх опрацювання, стандартизація та підготовка;

– fine-tuning двох моделей типу Т5 для задачі генерації рядків пісень;

– проектування ІС для використання цих моделей користувачами;

– експериментальна апробація натренованих моделей ML;

– аналіз та обговорення отриманих даних як результату експериментальної апробації.

Об'єктом дослідження є процес штучної генерації текстів моделями-трансформерами типу Т5. Предмет дослідження є процес донавчання мовних моделей генерації для створення продовжень текстів пісень (з та без врахування стиля певного автора).

Створення такого проєкту матиме кілька ефектів:

– Науково-технічний та соціальний, що виражатиметься у створенні нового набору даних (спеціальної форми для конкретної задачі, проте відкритий для модифікацій), публікації донавчених моделей;

– Потенційно прискорить та збільшить гнучкість процесу створення тексту для пісень, оскільки ІС буде здатна створювати кілька варіантів стрічок продовження наявних рядків;

– Потенційно прискорить експериментування з різними текстами для створення кінцевої композиції.

1 ПОСТАНОВКА ПРОБЛЕМИ

Необхідно розробити ІТ генерації продовження текстів пісень з допомогою великих мовних моделей, зокрема моделі Т5, для прискорення, доповнення та підвищення гнучкості процесу написання текстів до пісень з/без врахування стиля певного автора. Використання достатньо потужних та сучасних мовних моделей, зокрема Т5, забезпечить якість виконання завдання, за умови створення якісного набору тренувальних даних, дотримання правил донавчання моделей та вибору правильної стратегії генерування тексту. Підхід до завдання як чистого текст-до-тексту, що підтримує Т5 модель [2, 5], дозволить використовувати одну модель для генерації текстів у стилі багатьох авторів, оскільки зашифрувати автора пісні можна в самому вхідному тексті. Потенційно, такий підхід дозволяє створити одну модель для двох підзадач (генерування з та без задання стилю автора), проте з метою порівняння можливостей відтворення текстів авторів прийнято рішення побудови двох окремих моделей, що буде навчено за однакових умов.

Для процесу генерації важливим є питання вибору стратегії декодування. Проте, замість того, щоб надати перевагу конкретній стратегії, у ІС буде підтримка множини стратегій, зокрема:

1. Жадібний пошук (Greedy search);
2. Променевий пошук (Beam search);
3. Урізноманітнений променевий пошук (Diverse beam search) [6];
4. Поліноміальна вибірка (Multinomial sampling);
5. Променево-пошукова вибірка (Beam-search multinomial sampling);
6. Топ-k вибірка (Top-k sampling);
7. Топ-p вибірка (Top-p sampling);
8. Контрастивний пошук (Contrastive search).

Методики 1,2,3 і 8 є детермінованими, а 4–7 – стохастичними. Кожна з цих методик є авторегресійною [2], що можна проінтерпретувати як припущення, що розподіл генерованого ланцюжка спрощується до добутку умовних ймовірностей наступних слів. Тобто:

$$P(w_{1:T}|W_0)=\prod^{T}_{t=1}P(w_t|w_{<t},W_0).$$

У всіх випадках визначення T відбувається в процесі генерації, і рівний такому t, при якому $w_t = EOS$. Загалом, для порівняння, у роботі використано 8 різних методик декодування для генерації тексту, що підтримуються бібліотеку transformers. Для безпосередньої розробки, а також навчання моделі, підготовки даних та інших кроків процесу імплементації ІС обрано мову Python та множу необхідних бібліотек. Найважливіша бібліотека – transformers, дозволяє завантажити, донавчити та зберегти Т5 модель та відповідний токенизатор. Модель є об'єктом з TensorFlow [13].

2 ОГЛЯД ЛІТЕРАТУРИ

В науковому світі прийнято експериментувати з різними стратегіями генерації тексту, підбираючи та вивчаючи кращу для тієї чи іншої задачі. У даній роботі розглядається дві мовні моделі-трансформери Т5 [2, 5], донавчені для задачі генерації продовження стрічок текстів пісень. Ці моделі розв'язують одну й туж задачу, проте з різним підходом. Метою даної роботи є порівняння якості відтворення генерований текстів пісень за спеціальними текстовими метриками, для двох моделей, при використанні різних стратегій генерації.

Перша модель, яка надалі називатиметься NSA, навчена для завдання продовження пісень без будь-яких умов, а друга модель – SA, розв'язує завдання генерації при вказані стилю автора. Для другої моделі при навчанні додатковим параметром додано хто є автором того чи іншого набору рядків та їх продовження. При генерації встановлено певні додаткові обмеження та параметри, зокрема обмеження довжини стрічки – від 8 до 128 токенів, температура функції sotfmax – 0,98, штраф повторювання [3] – 0,98. Всі інші налаштування є специфічними для кожної зі стратегій.

Варто зауважити, що надалі у роботі ототожнюватиметься поняття рядка, стрічки та ланцюжка як послідовності елементів, зокрема слів. При цьому поняття слова, як елементу ланцюжка,

DOI 10.15588/1607-3274-2023-4-15

може бути замінене на токен або гіпотезу або кандидата. Ця заміна можлива, оскільки в роботі токеном вважається індекс слова у словнику.

При аналізі аналогів до ІС, що розробляється у цій роботі, необхідно враховувати як близький так і далеких аналогів, що розв'язують ту саму задачу, так і аналоги мовних моделей, які можна використати для поставленої мети. Основними непрямими аналогами для ІС порівняння стратегій генерації – є інші проєкти, що дозволяють створювати продовження текстів пісень. Наприклад:

– MuseNet від OpenAI: MuseNet є музичною мовною моделлю, яка може генерувати продовження музичних композицій.

– Jukedeck: Jukedeck є платформою, яка використовує ШІ для автоматичної генерації музики. Вона здатна створювати продовження музичних композицій з урахуванням стилю, настрою та інших параметрів.

– Magenta Project від Google: Magenta є відкритим джерелом для дослідження творчості, музики та ML. Вона включає різні моделі та інструменти для генерації музики, включаючи генерацію тексту продовження пісень.

– Amper Music: Amper Music є платформою, яка використовує ШІ для автоматичної генерації музики для відео та інших медійних проектів. Вона може генерувати продовження музичних композицій з врахуванням заданих параметрів.

Проте, концептуально вищим рівень аналогів, що необхідно розглянути є вид нейронних мереж та моделей, що можна використати для NLP задач, та зокрема для генерації тексту. Можна виділити таку множину близьких конкурентів-моделей, що можна використати для генерації тексту: CNN, RNN або Transformers.

CNN є найменш адаптованими до задачі генерації тексту (хоча вони мають місце для задач класифікації тексту тощо), що досить очевидно, зважаючи на їх неможливість роботи з потоковими даними та природою згортки.

RNN – це перша якісна альтернатива, оскільки такий вид нейронних мереж дозволяє розв'язувати задачу послідовного генерування текстів пісень. Проте, існує кілька аспектів, у яких рекурентні мережі програють трансформерам [1]:

– Рекурентні мережі не здатні зберігати довгострокові зв'язки у тексті, що дуже важливо для генерації зв'язних текстів пісень, при цьому через природу роботи механізму уваги, трансформери здатні це робити;

– Процес навчання рекурентних мереж обмежений, оскільки вони страждають від проблеми згасання градієнтів, а також їх роботу неможливо паралелізувати, що не є проблемою для трансформерів;

– Складний процес додавання додаткових параметрів вхідного тексту, зокрема, бажаний стиль тексту.

Проте, моделі Transformers мають і свої недоліки, що можуть вплинути на роботу ІС, наприклад:

– Потреба надвеликих об'ємів даних для якісної роботи, проте цю проблему можна розв'язати використанням трансферного навчання, тобто використати наперед навчені модель і довчити для конкретного завдання (у даному випадку – генерації текстів пісень);

– Деякі моделі мають тенденцію до генерування некреативного тексту, зокрема перша версія T5 [7];

– Якість роботи мережі залежить від якості даних та організації процесу донавчання;

– Потреба великого об'єму обчислювального ресурсу, що має вплив на екологію планети [8].

Звичайно, вплив більшості недоліків можливо мінімізувати відповідними діями та виконання практик донавчання Transformers. Використання трансферного навчання та Transformers, зокрема моделі T5, надає ІС множину переваг, а саме [12]:

– Контекстне генерування, тобто врахування контексту вхідних рядків тексту для генерації вихідних;

– Простота реалізації умовного генерування, із вказанням бажаного стилю, та безпосередня можливість вивчення властивостей авторства;

– Можливість переходу на більш якісні або новіші версії конкретних архітектур Transformers.

Враховуючи наданий аналіз, саме моделі-Transformers з використанням трансферного навчання є найбільш оптимальним варіантом досягнення поставленої мети.

3 МАТЕРІАЛИ ТА МЕТОДИ

При розгляді кожної зі стратегій як прикладу генерації з її використанням, на вхід мережі типу NSA подано кілька стрічок тексту пісні, що мережа раніше не бачила і відповідно декодовано. Вхідна стрічка для прикладу, авторства Т. Свіфт та А. Дезнера (фрагмент з "long story short" [15]):

"Past me
I wanna tell you not to get lost in these petty things
your nemeses
will defeat themselves before you get the chance to swing
and he's passing by
rare as the glimmer of a comet in the sky
and he feels like home
if the shoe fits, walk in it everywhere you go
and I fell from the pedestal
right down the rabbit hole"

1. Greedy search. Найпростіший у реалізації метод декодування, жадібний алгоритм пошуку, генерує наступним словом те, що має найбільшу ймовірність появи для кожного кроку t, тобто:

$$w_t = \text{argmax}_w P(w|w_{<t}).$$

Очевидно, що простота реалізації не гарантує якісні результати. Жадібний підхід страждає від

DOI 10.15588/1607-3274-2023-4-15

повторюваності, несумісності та деградації тексту [18], проте основним його недоліком є пропуски слів з високою ймовірністю, приховані за словом з низькою ймовірністю. Приклад декодованого рядка з використанням цього алгоритму:

```
i'm a liar, a liar, a liar
i'm a liar, a liar, a liar
i'm a liar, a liar, a liar
```

Видно, що жадібне декодування створює повторювальний текст, проте варто зауважити, що мала різноманітність тексту не супроводжується граматичними помилками.

2. Beam search. Ідея стратегії полягає у збережені кожного кроку n найбільш ймовірних гіпотез (тобто слів), та обрання кінцевим результатом той ланцюжок, що має найбільшу ймовірність загалом. Таким чином променевий пошук розв'язує головну проблему жадібного підходу. На Рис. 1 подано приклад роботи променевого пошуку при n=2.

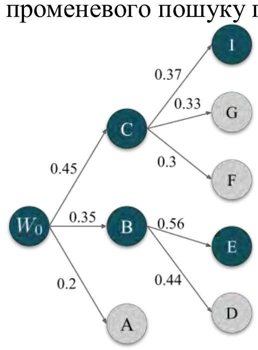


Рисунок 1 – Приклад побудови ланцюжка променевим пошуком при n=2

На кожному кроці t алгоритму виконується дві основі операції – формування множини всіх можливих новий стрічок C_t на основі попередньої множини гіпотез W_{t-1}, та обрання з неї нової множини гіпотез – n ланцюжків з найбільшим значенням функції правдоподібності.

$$C_t = \{w_{<t} \circ w_t \mid w_{<t} \in W_{t-1}, w_t \in V\}. \quad (1)$$

$$W_t = \underset{w_{<t+1} \in C_t,\ |W_t|=n}{\operatorname{argmax}} P(w_{<t+1} \mid W_0). \quad (2)$$

Відповідно, враховуючи формули (1)–(2), вихідна стрічка описується залежністю з формули (3):

$$w_{1:T} = \underset{w'_{<T+1} \in C_T}{\operatorname{argmax}} P(w'_{<T+1} \mid W_0). \quad (3)$$

Такий пошук завжди генерує стрічку з вищою загальною ймовірністю, ніж жадібний підхід, проте не має гарантії, що отриманий ланцюжок є найбільш ймовірним результатом. Очевидні недоліки такого підходу як вища обчислювальна складність та повторюваність у результатних ланцюжках. Крім того алгоритм теж має тенденцію до генерування повторюваних стрічок та формування «слабо дивних» (англ. less surprising) текстів, у порівнянні з людським текстом [17]. Для прикладу, при n=4, та враховуючи змінену температуру й введення штрафу повторів, при декодуванні променевим пошуком створено таке продовження тексту пісні:

```
and he's a monster
oh, i'm gonna get you out of it
and you'll be able to see me through the cracks
but if the shoe fits, walk in it everywhere you go
you fell from the pedestal
right down the rabbit hole
```

Створений текст не містить повторів, через наявність штрафу, порте більшість згенерованого тексту повторює частину вхідної стрічки. Крім того існують різні методи покращення результатів генерації з променевим пошуком, такі як N-грамні штрафи [19] та інші [20]. Менш очевидні недоліки випливають з [21], де показано, що beam підхід найкраще підходить для генерації текстів, де довжина вихідного тексту легко передбачувана (як при перекладі), проте менш якісно працює для задач відкритої генерації, як та, що розглядається у роботі.

3. Diverse beam search [6]. Цей підхід є спробою вдосконалити результати класичного променевого пошуку, шляхом побудови більш різноманітних наборів променів. Відповідно до оригінальної статті [6], результати такого підходу значно кращі за звичайний beam пошук. Крім кількості променів, цей алгоритм має додатковий параметр – кількість груп променів (number of beam groups). Групи вибираються так, щоб вони були досить чіткими порівняно з іншими, а в кожній групі використовується звичайний пошук за променем [6]. Проте, як видно з результатної стрічки нижче, цей алгоритм видає значно гірші результати за променевий пошук, що найймовірніше викликано процесом донавчання (fint-tuning) моделі, а не властивостями самого алгоритму:

```
and he's a monster
he's a monster
he's a monster
he's a monster
he's a monster
he's a monster
```

Отримана стрічка зациклена на 6-грамі he's a monster, де 6 токен є спеціальним роздільником <sep>, і тому його опущено.

4. Multinomial sampling. Проблема з класичними детерміністичними алгоритмами це тренд до створення повторюваних ланцюжків. Одне з простих рішень – це додавання випадковості до процесу вибору наступного слова. Базовий спосіб досягнення цього – це проста вибірка слова з розподілу над всіма елементами словника – формула (4).

$$w_t \sim P(w \mid w_{<t}). \quad (4)$$

DOI 10.15588/1607-3274-2023-4-15

По-перше, це призводить до того, що усі слова з ненульовою ймовірністю можуть потрапити до ланцюжка, що потенційно зменшує повторюваність в результаті. По-друге, оскільки процес вибору не є детерміністичним, то модель, при однаковій вхідній стрічці, генеруватиме різні результати при кожному проході. На жаль цей метод має важливий недолік – тенденція генерувати нісенітниці або несумні тексти [17]. Один з способів нівелювати ймовірність створення несумісних N-грам – це зменшення «температури» функції softmax, тобто збільшити значення правдоподібності слів з високою ймовірністю та зменшення для слів з низькою ймовірністю. При «температурі» 1 – отримані оцінки впевненості з softmax є класичними, а чим вона нижче (від 0 до 1) тим «гостріше» виглядатиме розподіл токенів. Приклад одного з варіантів тексту, декодованого за допомогою семплінгу:

```
and it's mad as hell
don't know why you're dying
those who loved me will never love you back
i'm richer, more powerful
i'm all i want from life
i want you to be free
```

Текст маж достатньо високу різноманітність, не містить повторів з вхідної стрічки, згенеровані рядки граматичні та синтаксично коректні, проте загальна комбінація стрічок не є цілком зв'язаною. Інший приклад згенерованого тексту за цим алгоритмом:

```
not about winning
you can wreck me
dj schwer, who deserves my blood
reach me, skinny dipping, dj fit
goddamn
you're right there, man
you're so good, you know you're got a killer soon
last call out for me
as you grow increasingly understand
this is a metamorphosis
and i am a master of disguise
don't be afraid to step
```

При однакових вхідних параметрах та налаштуваннях ця стрічка є менш сумісною за попередню, адже частина словосполучень не мають змісту. Наприклад, три 2-грами reach me, skinny dipping, dj fit належать до одного рядка, і були відокремлені спеціальний сепаруючим токеном, проте вони абсолютно без сенсу, і не відносяться до вхідного тексту.

5. Beam-search multinomial sampling. Комбінація стохастичного процесу вибору та детерміністичного променевого пошуку. При такій стратегії кожного кроку t з $P(w \mid w_{<t})$ незалежним чином вибирається n слів. Використовуючи цю стратегію, один з отриманих результатів має такий вигляд:

```
and he's passing by
rare as the glimmer of a comet
and he feels like home
if the shoe fits, walk in it everywhere you go
and he feels like home
if the shoe fits, walk in it everywhere you go
and he feels like home
```

Отриманий ланцюжок повністю складається з слів вхідної стрічки, що, аналогічно до попередньої стратегії, ймовірно викликано процесом навчання.

6. Top-*k* sampling [16]. Досить потужна модифікація стохастичного підходу вперше запропонована у [16]. Ідея полягає в тому, що з розподілу залишають лише топ k токенів з найбільшими ймовірностями. Далі обчислюється нова функція розподілу відповідно до нової множини токенів, і вже з цього нового розподілу проводять вибірку слова. Нехай множина слів, що обирається кожного кроку t позначається як $V^t_{top\text{-}k}$, тоді:

$$V^t_{top\text{-}k} = \underset{w \in V,\, |V^t_{top\text{-}k}|=k}{\operatorname{argmax}} P(w_t \mid w_{<t}). \tag{5}$$

$$w_t \sim P_{V^t_{top\text{-}k}}(w \mid w_{<t}),\ w_t \in V^t_{top\text{-}k}. \tag{6}$$

Така фільтрація кандидатів частково нівелює ймовірність появи дивних чи несумісних комбінацій слів, проте головний недолік – це відсутність адаптивності розміру множини кандидатів до їх розподілу. Без такої адаптивності можливі два негативних наслідки. Перший – при дуже «гострому» оригінальному розподілі, коли після перерозподу будуть утворюватися несумісні рядки. Другий – відсутність різноманітності генерованого тексту пісень, що виникатиме при рівномірності ймовірностей відібраних k токенів. Очевидно, що при k=1 алгоритм поводитиметься як жадібний, а при k=0 або k=card(V) – як звичайний семплінг. Приклади отриманих стрічок, з використанням цього алгоритму при k=15 та зменшеній температурі softmax функції:

```
the glimmer's a shard of glass
and the color's a diamond
the fire's burning bright
but he's not a monster
he's more like a monster
i'm in love, love's a sliver of glass
and his touch is sweet as honey
he's like a prince, a prince, a prince
i love you
```

Тексту властива слабка повторюваність, відсутність римування, проте текст граматично коректний. Проте, при зменшені k до 5, згенерована стрічка швидко деградує:

DOI 10.15588/1607-3274-2023-4-15

```
i'm just stumbling, stumbling, stumbling, stumbling
just stumbling, stumbling
stumbling, stumbling
stumbling, stumbling, stumbling
stumbling, stumbling
```

7. Top-p sampling [17]. Для корегування недоліків top-k підходу у [17] запропонували ідею відбору не просто найймовірніших k токенів, а стільки токенів, щоб їх кумулятивна ймовірність була не менше за p. Аналогічно до top-k, після фільтрації токенів їх функція розподілу перерозподіляється, після чого відбувається вибирання. Якщо позначити множину кандидатів токенів на кожному кроці t як $V^t_{top\text{-}p}$, то наступне слово вибиратиметься з розподілу відповідно до формули (7).

$$w_t \sim P_{V^t_{top\text{-}p}}(w \mid w_{<t}),\ w_t \in V^t_{top\text{-}p}. \qquad (7)$$

При p=1 алгоритм перетворюється у класичний семплінг, а при p=0 – у жадібний алгоритм. Приклад стрічки продовження тексту пісні згенерований при p=0,96:

```
but the light comes back to you
like a sailor flying through the sky
and the wind blows through the trees
you can't miss him, baby
he's just a little bit of magic
you're just a tiny bit away
and you're just a little bit farther
and you're just a little bit further
you're just a tiny bit further
i've been doing my
```

Згенерований текст і справді виглядає як продовження тексту пісні, хоча й процес генерування було перервано на незавершеній фразі. Текст граматично коректний, хоча його сумісність та зміст важко оцінити як якісні.

8. Contrastive search [18]. Щоб побороти проблему генерації виродженого тексту (англ. degenerate solutions), тобто повторювального та/чи несумісного тексту, що виникає при використанні вже розглянутих детерміністичних та стохастичних алгоритмів, автори [18] запропонували зіставне рішення – контрастивний метод декодування. Оригінальна робота авторів пропонує не лише алгоритм декодування, а і заміну класичному методу максимізації функції правдоподібності при навчання трансформерів для генерації чи в загальному – моделюванні мов. Проте, враховуючи контекст цієї роботи, та використання звичайного алгоритму навчання для двох моделей, що розглядаються, цю заміну не буде розглянуто. Важливо зауважити, що не використання цього алгоритму впливає є на можливості зіставного декодування. Ідея алгоритму генерування наступного слова складається з двох ключових аспектів [18]:

– гіпотеза повинна бути обрана з множини найймовірніших (як при top-k);

– згенерований результат має бути досить дискримінаційним щодо попереднього контексту.

Математично, алгоритм на кроці t описується як:

$$w_t = \underset{w \in V^t_{top\text{-}k}}{\operatorname{argmax}} \{(1-\alpha)\times p(w|w_{<t}) - \alpha\times(\max\{s(h_w,h_{wj})|1\le j\le t-1\})\}.$$

Формулу можна розбити на дві частини, відповідно до двох раніше описаних аспектів. Перша – обчислює впевненість моделі (тобто вихід з softmax), а друга (так званий штраф виродження) – вимірює, наскільки дискримінаційний кандидат w щодо попереднього контексту $w_{<t}$. Оскільки цей алгоритм є детерміністичним, і враховуючи тенденцію інших подібних алгоритмів створювати повторювані рядки, то низьке значення штрафу α автоматично породжувати однакові рядки, зміст яких залежатиме від параметру k.

Наприклад, при α = 0,5, модель генерує такі рядки:

k = 3
```
i'm the one who got away
i'm the one who got away
i'm the one who got away
i'm the one who got away
i'm the one who got away
i'm the one who got away
i'm the one who got away
```

k = 10
```
hey, look at me
i'm in the back of the car
and he's like a wildfire
i'm on the other side
i'm the one who got away
i'm the one who got away
i'm the one who got away
i'm the one who got away
```

k = 20
```
now i'm in the back of the car
waiting for you to come back
i'm like a second-hand hero
no, i'm not a bad guy
i'm just a bad guy
i'm just a bad guy
i'm just a bad guy
```

k = 50
```
now i'm in the back of the car
waiting for you to come undone
i'm making my way home
no one's gonna do it like me
i'm an old codger
i'm the anti-hero
i'm the anti-hero
i'm the anti-hero
```

При збільшенні кандидатів, збільшується і різноманітність перших рядків, проте процес генерації збігається до повторювання. Проте, при збільшенні штрафу, наприклад, до 0,95, виникає протилежна ситуація – при збільшенні кількості кандидатів алгоритм вибиратиме найбільш різноманітний токен, що призводить до нісенітниць. Тому необхідно знайти оптимальну комбінацію двох параметрів. Як видно з прикладу нижче, при невеликому значенні k модель генерує сумісний текст, навіть вставляючи вигуки, як частини тексту пісень. При k=13 у ланцюжку з’являється непов’язані комбінації слів, що не мають змістовне навантаження.

DOI 10.15588/1607-3274-2023-4-15

$k = 3$

```
i fell from a pedestal (woo, woah)
right into the rabbit hole (ooops, woah, aww)
and now a day has come that's just for me
```

$k = 5$

```
i know that the end is near (yo, nope),
but it feels like the first step
```

k=13:

```
now i need an air freshened life
get the hang of what life in our neck of delinque, lenor mcallist
```

Питання відкритої генерації продовження пісень важко оцінювати з точки зору змісту, рими, римування, розмірності віршів тощо. Проте, можливо порівняти та оцінити наскільки близький текст, що генерує мережа, до того, що очікувався. Для цього необхідно використовувати спеціальні текстові метрики. У даній роботі пропонується розглянути три метрики: BLEU [22], RougeL та RougeN [23].

1. BLEU-метрика, створена для оцінки машинного перекладу [22], проте вона користується достатньою популярністю і серед інших задач обробки природної мови. Алгоритм обчислення метрики ґрунтується на порівнянні N-грам в оригінальному та згенерованому тексті. Для обчислення метрики обчислюється точність для кожної довжини N-грам (відношення співпадінь до кількості N-грам цієї довжини). Далі обчислюється середнє геометричне точності, вводяться спеціальні штрафи, та обчислюється кінцева оцінка в межах [0, 1], де більше значення відповідає за кращий результат.

2. RougeL зі сімейства метрик Rouge [23], призначене для оцінки підсумовувань текстів, проте, аналогічно до BLEU адаптоване і для інших задач. Приставка L походить від LCS, тобто найдовшої спільної підстрічки. Ідея алгоритму полягає в обчисленні LCS, далі обчислення класичних метрик класифікації – точність (precision) та повнота (recall), як відношення довжини LCS до довжини згенерованого та очікуваного тексту відповідно. Після цього обчислюється F1 оцінка точності та повноти. Саме F1 значення використовуватиметься для порівняння результатів різних мереж.

3. RougeN. Аналогічно до попередньої метрики, RougeN є частиною сімейства Rouge метрик. На відміну від RougeL, використовує N-грамний підхід. При обчисленні метрики обраховується перекриття n-грамами (де n – заданий параметр) згенерованої стрічки до реальної, і далі аналогічно до RougeL, обчислюється precision, recall та F1 [23].

У роботі використано офіційну імплементацію цих метрик бібліотекою KerasNLP [24–25].

4 ЕКСПЕРИМЕНТИ

Для побудови необхідного ПЗ необхідно чітко визначити основні об’єкти ПО порівняння стратегій генерації тексту. До таких об’єктів можна віднести:

– Алгоритми декодування: дослідження стратегій декодування тексту вимагає розуміння різних алгоритмів, які використовуються для згенерування послідовності слів. Це можуть бути як чисто детерміністичні, стохастичні алгоритми чи їх комбінації. Важливо визначити переваги, недоліки та вплив цих алгоритмів на згенерований текст;

– Мовні моделі: дослідження в області, що розглядається в роботі, вимагає розуміння різних мовних моделей, зокрема тих, які використовуються для генерації тексту. У цій роботі цей аспект ПО обмежено до мовних моделей типу T5, процесу їх навчання, text-to-text підходу до NLP задач;

– Метрики оцінки: Для порівняння стратегій декодування необхідно використовувати певні метрики, що визначають якість згенерованого тексту. Це можуть бути метрики, які оцінюють подібність створених, декодованих стрічок з тими, що очікувались як результат генерації тексту, зокрема тексту продовження пісні.

Похідні об’єкти, що мають більший рівень абстракції, проте є не менш важливі для обговорення:

– Проблеми з надмірною генерацією – один з найважливіших аспектів аналізу є розгляд проблем, пов’язаних з деградованою генерацією тексту, тобто наявність повторення, неконкретності, несумісності або незв’язності. Розуміння цих проблем і розробка методів для їх вирішення є важливим кроком у вдосконаленні стратегій декодування;

– Адаптація до конкретних завдань: Враховуючи різноманітність завдань, для яких використовуються мовні моделі, важливо вивчати, які стратегії декодування найкраще працюють у конкретних випадках. Деякі завдання можуть вимагати швидкості, інші – більшої точності або креативності в генерації тексту. У даній роботі визначення конкретна задача, що розглядається – процес генерації тексту продовження пісні.

Для коректної побудови ІС та її специфікації необхідно визначити основні дефініції, терміни та об’єкти ПО. Проте ширина сфери генерації штучних текстів є достатньо потужною, тому необхідно визначити межі ПО. На рис. 2 продемонстровано діаграму прецедентів пропонованої ІС. Роль цієї діаграми показати які функціональні вимоги повинна мати системи, щоб мати змоги виявити межі ПО. З діаграми видно, що концептуально найвищим прецедентом ІС є опрацювання вхідного запиту користувача на генерацію продовження тексту пісень. Враховуючи описані прецеденти, а також особливості розробки ІС на базі великих мовних моделей, можна виділити три важливих аспекти ПО:

– Створення навчального датасету – процес опрацювання та підготовки текстових даних;

– Донавчання (fine-tuning) мовної моделі – процес навчання наперед навченої моделі для виконання конкретного завдання;

– Генерація тексту пісні – процес використання моделі та декодування її результатів.

Кожен з цих аспектів містить певні терміни та процеси та варіанти, якими необхідно оперувати для досягнення поставленої мети.

DOI 10.15588/1607-3274-2023-4-15

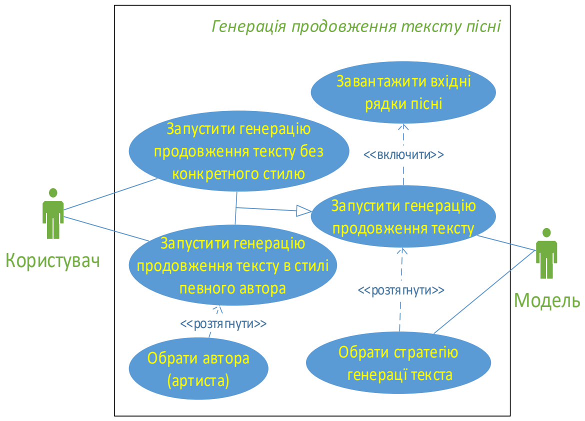


Рисунок 2 – Діаграма прецедентів системи генерації продовження текстів пісень

Створення навчального датасету є найважливішим кроком для ІС, що використовують ШІ, оскільки якість даних є необхідною умовою для побудови моделі, що відповідатиме поставленим вимогам.

З цієї частини ПО необхідно виділити, що означає генерація продовження тексту пісні. Це питання постає, оскільки означити це завдання можна кількома способами. Наприклад, класичне завдання мовних моделей – це відновлення пропущених (замаскованих) слів чи N-грам у тексті, так зване моделювання мови. За таким ж принципом можна означати процес генерації продовження пісень, тобто маскуючи закінчення рядків або випадкові n-грами з тексту пісень, і навчати модель їх відновлювати. Такий підхід вимагає від потенційних користувачів надавати на вхід не тільки текст, а місця де їм необхідно доповнити текст. Натомість у цій роботі пропонується інший підхід – постстрічкове доповнення тексту пісні. Замість того аби продовжувати кожну стрічку окремо, модель буде намагатися відтворити кілька наступних стрічок. Для цього у текст буде додано спеціальний токен розділення рядків, і модель буде намагатися генерувати його в тому місці, де закінчуватиметься рядок. Другий аспект ПО – це процес fine-tuning мовної моделі. Основна мета процесу – покращити результати моделі та забезпечити більш точні та відповідні генерації тексту продовження пісень. Цей аспект включає три кроки: створення специфікованих даних (описаний раніше), вибір архітектури та переналаштування моделі.

Для контексту даної роботи, важливо відмітити другий крок – вибір архітектури. Цей процес є складним, оскільки існує багато різних моделей з різними розмірами, шаровою структурою та кількістю параметрів. При виборі архітектури треба враховувати особливості задачі, наявні обчислювальні можливості, наявні ресурси пам'яті та враховувати вплив процесу навчання моделей на навколишнє середовище. Останній аспект, який необхідно розглянути – це безпосередній процес генерації тексту з мовною моделлю. Для опису цього процесу необхідно ввести кілька означень.

– Токенизація – це розбиття стрічок на окремі компоненти (токени), якими можуть бути слова, символи чи фрагменти слів. При цьому кодування при токенизації – це процес відображення токенів у їх числове представлення.

– Декодування токенів – процес відображення індексів у відповідні частини мови (слова, символи тощо).

– Стратегія генерації або стратегія декодування [3] – це алгоритм вибору наступного слова, залежно від попередніх. Вибір стратегії впливає на безпосереднє використання моделі при генерації, має суттєвий вплив на кінцевий результат, сумісність та якість згенерованого тексту.

Весь процес генерації тексту продовження пісень, в межах сформованого контексту можна подати у вигляді послідовності операцій, оформлених у діаграму діяльності з рис. 3. Після введення основних термінів та підпроцесів, можливо сформувати опис головного процесу ІС – опрацювання запиту на генерацію тексту продовження пісні. На рис. 4 виведено відповідну діаграму діяльності. Об'єкти, що використовуються на рис. 4 є абстракціями, що не потребують специфікації. Опис обраний методів та засобів розробки ІС можна розділити на дві частини, перша – засоби для розв'язання задачі генерації, друга – технології розробки програмного забезпечення.

Обраною архітектурою моделі-трансформера є T5 [2]. Особливість цієї моделі – її повнота (енкодер – декодер модель), гнучкість та використання текст-до-текст підходу. Такий підхід дозволяє уніфікувати одну модель для кількох задач, наприклад генерацію тексту пісень в стилі різних авторів, шляхом додавання так званого taskspecific префіксу.

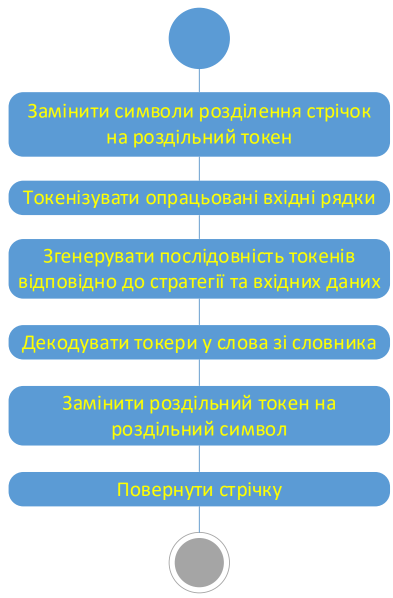


Рисунок 3 – Діаграма діяльності процесу генерації тексту продовження пісні

DOI 10.15588/1607-3274-2023-4-15

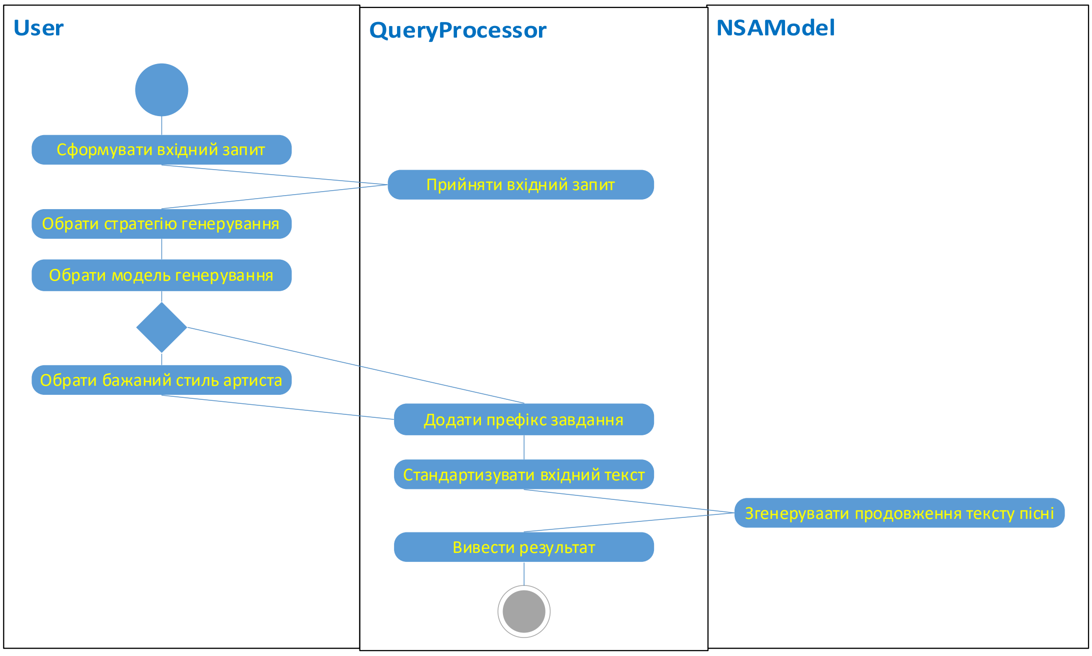


Рисунок 4 – Діаграма діяльності процесу опрацювання запиту на генерацію тексту продовження пісні

Інші переваги моделі:

– Контекстне генерування, тобто врахування контексту вхідного тексту для генерації вихідного;

– Направленість генерації, тобто використання спеціальних токенів та масок;

– Спектр різних розмірів, що дозволяє підібрати модель згідно обчислювальних обмежень та потреб.

Відповідний токенизатор та детокенизатор для цієї моделі працює на sentecepiece [9], що імплементує модель subword units токенизації. Використання такого токенизатора має низку переваг, наприклад, він не залежить від мови, легкий та дуже швидкий [9]. Відповідно до авторів цього токенизатора, він є кращою альтернативою для задач генерації ніж subword-nmt [10] чи WordPiece [11]. На рис. 5 відображено схему взаємодії основних, додаткових та допоміжних об’єктів ПО, у вигляді діаграми класів.

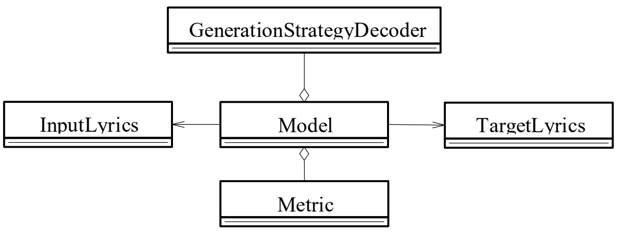


Рисунок 5 – Діаграма класів взаємодії об’єктів

5 РЕЗУЛЬТАТИ

Для використання конретного алгоритму декодування, необхідно розглянути їх імплементації в бібліотеках, що використано для побудови двох мовних моделей, що були раніше описані. Такої бібліотекою є transformers. Відповідно, для використання жадібного алгоритму, при використанні методу генерації тексту, необхідно вказати таку комбінацію параметрів:

```
model.generate(**processed_text, do_sample=False,
**gen_params)
```

Для beam search та променево-пошукова вибірки відповдно:

```
model.generate(**processed_text, do_sample=False,
num_beams=num_beams, **gen_params)
model.generate(**processed_text, do_sample=True,
num_beams=num_beams, **gen_params)
```

Для урізноманітненого променевого пошуку:

```
model.generate(**processed_text, do_sample=False,
num_beams=num_beams,
num_beam_groups=num_beam_groups, **gen_params)
```

Для звичайного семплінгу відповідний метод має вигляд:

```
model.generate(**processed_text, do_sample=True,
num_beams=1, top_k=0, **gen_params)
```

Стохастичні методи top-k та top-p мають такі комбінації параметрів:

```
model.generate(**processed_text, do_sample=True,
num_beams=1, top_k=top_k, **gen_params)
model.generate(**processed_text, do_sample=True,
num_beams=1, top_k=top_k, top_p = top_p,
**gen_params)
```

А контрастивний пошук, в свою чергу, таку:

```
model.generate(**processed_text,
penalty_alpha=penalty_alpha, top_k=top_k,
**gen_params)
```


DOI 10.15588/1607-3274-2023-4-15

Для проведення експериментів та порівняння результатів необхідно мати відповідний інструментарій (стратегії декодування та метрики) та вхідні дані. Такими даними у цій роботі є три підмножини тестових даних, які використовувалися для валідації результатів навчання SA та NSA моделей. Кожна з підмножини є текстами пісень одного артиста.

Для підготовки даних необхідно для початку їх отримати. Для цього використано базу даних сервісу Genius [14] станом на літо 2021 рік у вигляді json файлу. Для створення даних відібрано 10 різних артистів, після чого відібрано тексти їх пісень. Загалом отримано 626 унікальних пісень. Кожен текст пісні – це безпосередньо стрічки слів та довідникова інформація (частини структури пісні або хто з співавторів виконує конкретний текст). Перший крок опрацювання тексту – очистка від довідникової інформації та розбиття тексту на стрічки.

Після підготовки текстових даних необхідно розвити стрічки на початок та продовження, тобто на вхідні та вихідні дані. Перед опрацюванням тексту треба визначити скільки пар вхід-вихід можна утворити з кожної пісні. Для цього проведено дослідницький статистичний аналіз для кількості стрічок у піснях, а також кількості слів та унікальних слів у піснях. На Рис. 6 відображено гістограми, boxplot та статистика кожної з характеристик.

Використовуючи дані з Рис. 6 розроблено процес розбиття стрічок на вхідні вихідні. Цей процес повинен враховувати кілька вимог, зокрема: пара вхід-вихід має бути послідовними слова пісні, довжина вхідний та вихідних стрічок повинна варіюватися від екземпляра до екземпляра, кількість вхідних та вхідних стрічок мають бути не менше встановленого мінімуму.

Для об'єднання стрічок вхідної та вихідної групи використано спеціальний токен <sep>. Цей токен також додано до токенизатора. В результаті розбиття кожної пісні на кілька пар вхідних-вихідних стрічок отримано 1874 навчальних екземплярів та 465 тестових. Для ілюстрації прикладу процесу перетворення тексту пісні на один екземпляр датасету на Рис. 7 виведено відповідне зображення.

Проведено донавчання двох мовних моделей NSA та SA для задачі генерації продовження тексту пісень. Для обох моделей як базову обрано t5-base. Ця версія T5 містить 223 мільйони параметрів. Як раніше описано, у роботі розроблено дві моделі. Перша вирішує задачу генерації проводження пісень, а друга – генерацію в стилі певного автора. Всього є 10 авторів, що відповідає 10 артистам, чиї пісні відібрано для навчання. Для першої моделі, яка надалі називатиметься NSA, task specific префікс виглядає так "continue lyrics:". Відповідно, перед початком донавчання моделі, усі вхідні тексти пісень доповнюються цим префіксом. Відповідно для другої моделі, що матиме назву SA, такий префікс залежить від автора тексту, проте загальна форма така "continue lyrics as [певний автор]". Налаштування гіперпараметрів для моделей є однаковими:

– Кількість епох: 3;
– Розмір партії даних: 2 записи;
– Оптимізатор: Adam з швидкістю навчання $5 \cdot 10^{-5}$.

Метрики навчання – значення функції втрат, перплексивність та точність топ-3. Остання метрика визначає точність передбачення необхідного токена, враховуючи три найймовірніші результати. Результати навчання для кожної з моделей оформлені у вигляді графіків кривих навчання, для NSA моделі – рис. 8, а для SA – рис. 9.

Аналіз кривих навчання вказує на те, що модель NSA матиме менш деградаційні результати, а для моделі SA необхідно збалансовувати кількість тексту для кожного автора. Після розробки та навчання моделей, можливо застосовувати моделі для генерації тексту. У якості контрольних прикладів розглянуто два варіанти, перший – розгляд безпосередньо згенерованого тексту для конкретного вхідного запиту, а другий – кількісна оцінка якості моделі для множини вхідних запитів. Загалом, для порівняння, у роботі використано 8 різних методик декодування для генерації тексту, що підтримуються бібліотеку transformers. Незважаючи на те, що існує можливість вибору конкретної стратегії декодування, у контрольну приклади розглянуто всі 8. Для обох моделей використано однакову вхідну стрічку, однакові штрафи за повторюваність, зменшено так звану температуру функції softmax, та однакові налаштування параметрів стратегій генерування.

Для різнобічності оцінки можливостей мовних моделей, бажано мати вхідні дані різної розмірності. Для цього, обрано три артисти, чиї тексти займають найбільшу, найменшу та середню кількість від усієї навчальної та тестової вибірок, зокрема, тексти:

– артиста-1 – 25% навчальної та тестової вибірок;
– артиста-2 – 9% навчальної та тестової вибірок;
– артиста-3 – 3% навчальної та тестової вибірок.

Така різноманітність дозволить дослідити вплив кількості стрічок артиста, що мовна модель бачила в процесі fine-tuning, на значення метрик для обох моделей. Емпірично, знання про відсоток даних артиста від усіх має грати роль лише для SA моделі, де цей артист вказується, при цьому NSA модель нічого не знає про авторство тексту, що необхідно продовжувати. Порядок подання результатів оцінки значень метрик є таким: першим подано результати продовження текстів на SA моделі, тобто з вказанням бажано стилю (автора пісні), другим подано результати тих самих даних при використанні NSA моделі. Налаштування стратегій генерування, що зафіксовані для усіх процесів декодування:

– Променевий пошук (та похідні методи): кількість променів = 4;
– Diverse-beam: кількість груп променів = 4;
– Top-k: k = 5;
– Top-p: p = 0,96;

DOI 10.15588/1607-3274-2023-4-15

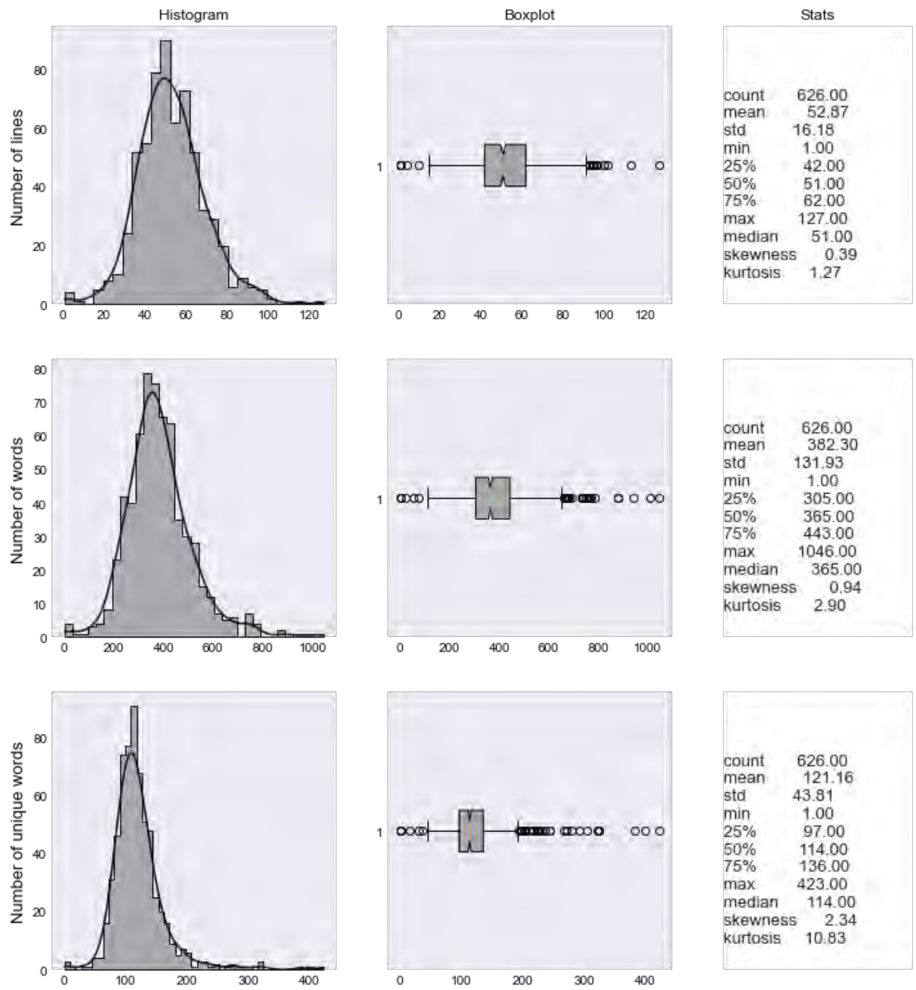


Рисунок 6 – Статистичний аналіз кількості стрічок, слів та унікальних слів у відібраних піснях

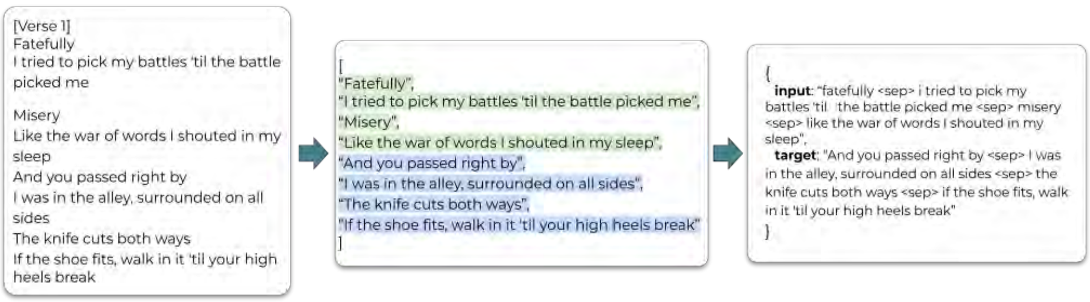


Рисунок 7 – Приклад опрацювання тексту пісні (чиста, розбиття та стрічки, формування пари вхід-вихід)

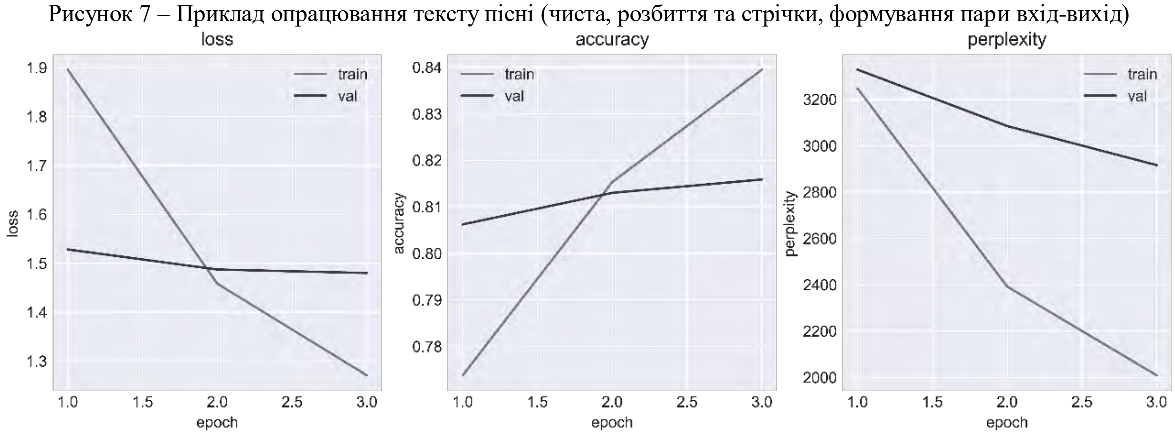


Рисунок 8 – Криві навчання моделі NSA

DOI 10.15588/1607-3274-2023-4-15

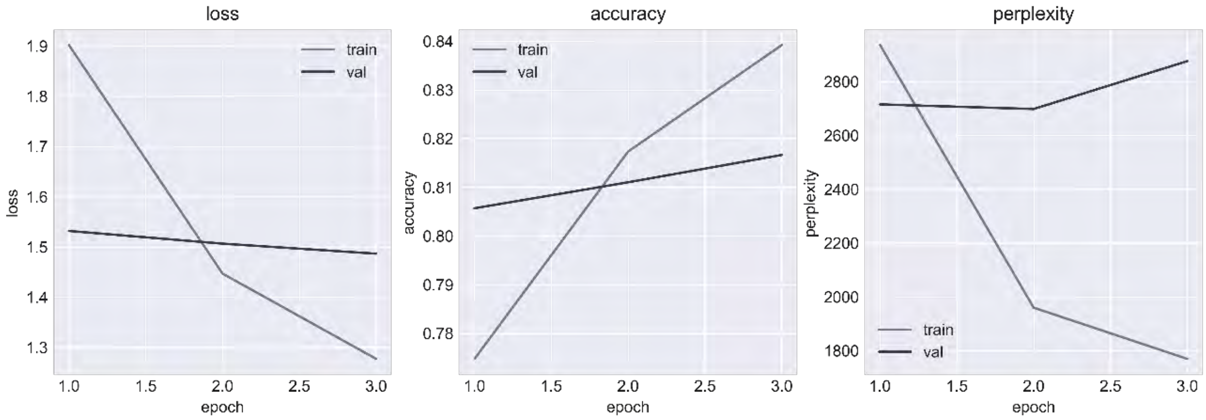


Рисунок 9 – Криві навчання моделі SA

– Констрастивний пошук: k = 3, штраф виродження = 0,9;

Для метрик RougeN порядок n-грами рівний 2, а для метрики BLEU від 1 до 4.

Результати NSA моделі виведено на рис. 10–11, а з SA – рис. 12–13.

Для другого контрольного прикладу відібрано множину вхідних-вихідних пар стрічок, та обраховано кілька текстових метрик, для кількісного порівняння результатів моделей та стратегій генерування. Такими метриками обрано BLEU, RougeL та RougeN. На Рис. 12 виведено значення метрик для надвеликої вибірки з тестових даних для моделі NSA, оскільки її результати є кращими, за відповідні з SA (рис. 13). Усі результати з рис. 8 та 9 не містять повторюваності тексту, проте частина з текстів є граматично некоректною, частина має не логічні поєднання слів. Променевий пошук, top-p та урізноманітнений променевий пошук містять аномально довгі рядки.

Рисунок 10 – Результати генерування стратегіями greedy пошук, beam пошук, top-p та top-k

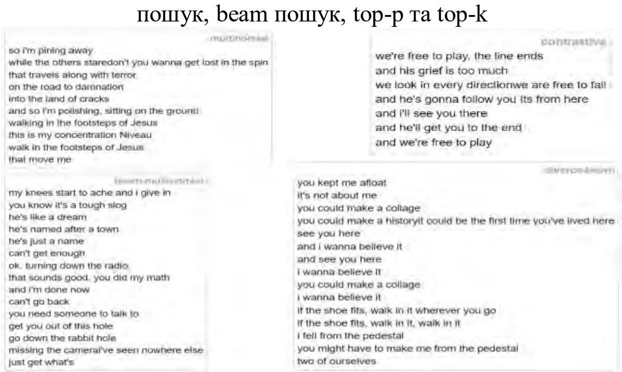


Рисунок 11 – Результати генерування стратегіями вибірки, променевого семплінгу, diverce-beam та контрастивним пошуками

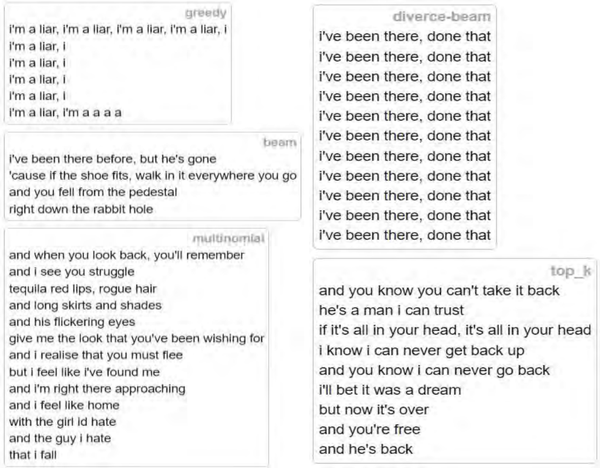


Рисунок 12 – Результати генерування 5 стратегіями з SA моделі

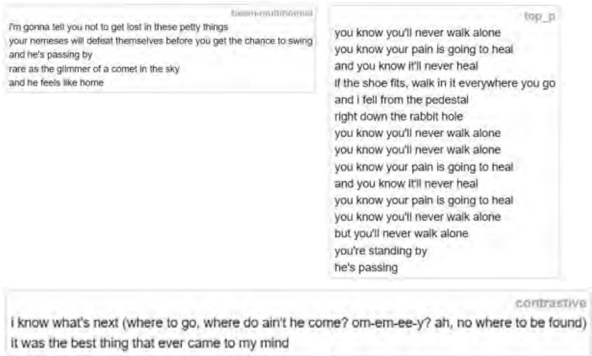


Рисунок 13 – Результати генерування 3 стратегіями з SA моделі

Результати з SA моделі мають видимі ознаки деградації тексту, повторюваність та несумність. Ймовірно, ці проблеми є ознаками перегляду навчального процесу.

6 ОБГОВОРЕННЯ

На рис. 14 для Артиста-1 виведено результати метрик з використанням моделі, що не містить інформації про авторство текстів. Якщо розглядати значення метрик окремо, то BLEU значення діаметрально змінилися, наприклад, двічі зменшилося значення для семплінгу, майже у два рази зменшилося значення для контрастивного методу, проте удвічі збільшилося для diverse-beam пошуку, майже утричі зросли для top-*k*.

DOI 10.15588/1607-3274-2023-4-15

Перш за все, найкращий результат розділяють променевий пошук та променевий семплінг. Результати при жадібному алгоритмі не змінилися. З рис. 14 видно, наскільки зміна стратегії змінює значення метрики. В даному випадку, стратегії променевого семплінгу має найкращі результати.

Значення метрик текстової подібності з моделі SA (рис. 15) стосуються автора з найбільшою кількістю даних на SA моделі. Найкращий результат, відповідно до усіх трьох метрик, досягнуто при променевому пошуку. Найгірший результат залежить від обраної метрики. Значення BLEU є найрізноманітнішим, і його значення значно змінюється, залежно від стратегії. При цьому Rouge метрики мають меншу варіативність, менший розмах значень.

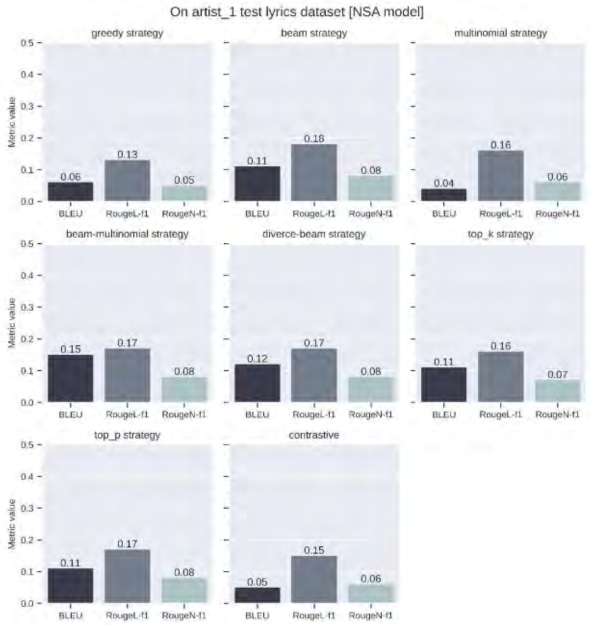


Рисунок 14 – Значення метрик текстової подібності з моделі NSA на даних артиста-1

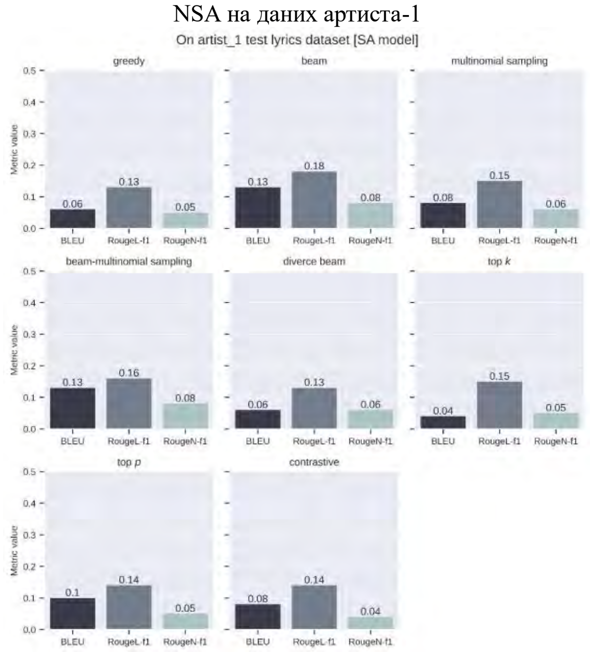


Рисунок 15 – Значення метрик текстової подібності з моделі SA на даних артиста-1

Значення обох Rouge метрик зросли або не змінилися у всіх випадках. Такі результати можу вказувати на те, що навчання на незалежних від авторства текстах, у загальному краще розв'язують задачу генерації, за умови що використано дані, що займають значну частину навчальної вибірки. Для перевірки такої гіпотези необхідно розглянути результати для інших авторів.

Результати для другого автора при вказані його стилю незначно відрізняються від аналогічних для Артиста-1 (рис. 16–17). Найкращі значення знову при використані beam алгоритму. Порівняння результатів Артиста-2:

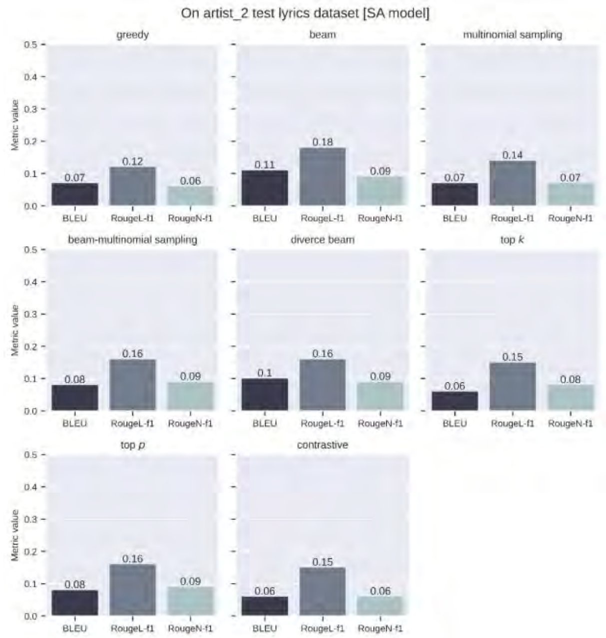


Рисунок 16 – Значення метрик для різних стратегій генерації на даних артиста-2 (модель SA)

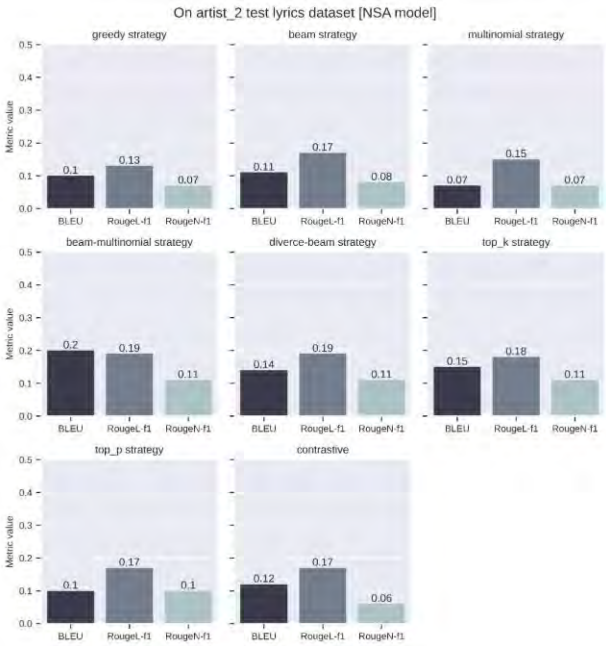


Рисунок 17 – Значення метрик для різних стратегій генерації на даних артиста-2 (модель NSA)

DOI 10.15588/1607-3274-2023-4-15

Рис. 17 демонструє результатні значення метрик для артиста-2 при використанні NSA моделі. Найцікавіший факт цього результату – найбільші значення BLEU метрики серед усіх результатів, що розглядаються у роботі. Значення з рис. 17 є більшими або рівними відповідним з рис. 16. Найймовірніше, що такі аномальні кращі результати для автора, чиї тексти не займають більшість датасету, пов'язані із загальністю та простотою цих текстів. Оскільки NSA модель будує узагальнення серед усіх текстів пісень, що їй надано при навчальному процесі, то ймовірно, що модель вибудували певне усереднення або шаблони для текстів. Через це, тексти пісень, що не є складними, або мають багато повторень або використовують загальні шаблонні фрази є найлегшими для відтворення моделлю. Така властивість моделі не вказує на якість текстів, що вона генерує, а більше на необхідність перегляду процесу її навчання [7].

Порівняння результатів Артиста-3:

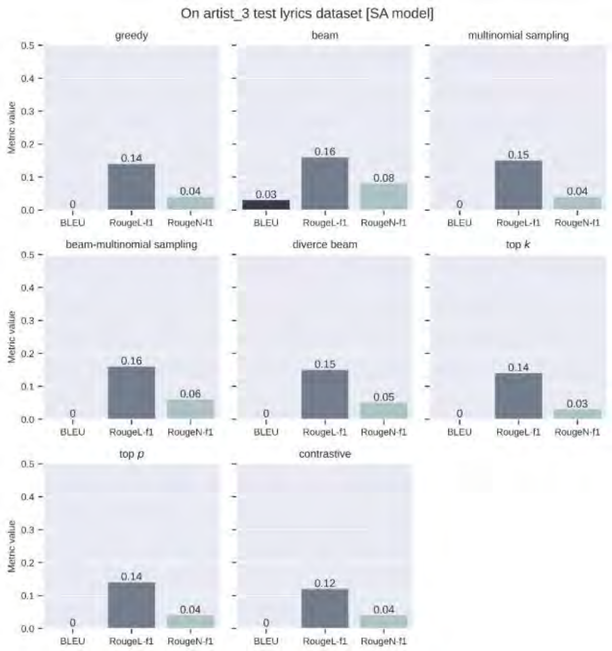


Рисунок 18 – Значення метрик для різних стратегій генерації на даних артиста-3 (модель SA)

З рис. 18, на якому виведено результати метрик для текстів автора з найменшою кількістю екземплярів у датасетах, видно суттєве зниження значень метрики, зокрема для BLEU.

Лише променевий пошук має ненульове значення цієї метрики. При цьому, Rouge метрики показують незначне падіння. Такі результати пов'язані як з кількістю тестових даних, так і з природою текстів (їх унікальністю і різноманітністю). Останні результати, що розглянуто в роботі, продемонстровані на рис. 19, показують, що узагальнене вміння генерувати тексти моделлю NSA є кращим (по значенням метрик) за аналогічне в SA моделі. Знову, лише променевий погук отримав ненульове значення для BLEU. Аналогічно до результатів інших авторів, модель NSA отримала кращі результати за SA. Такі результати можуть вказувати на кілька важливих аспектів, по-перше, необхідно розширювати набір даних, збільшуючи його різноманітність, покращуючи якість. При цьому, збільшення кількості разів оновлення параметрів моделі можуть призвести до того, що модель SA зможе краще запам'ятовувати та відтворювати тексти конкретних авторів.

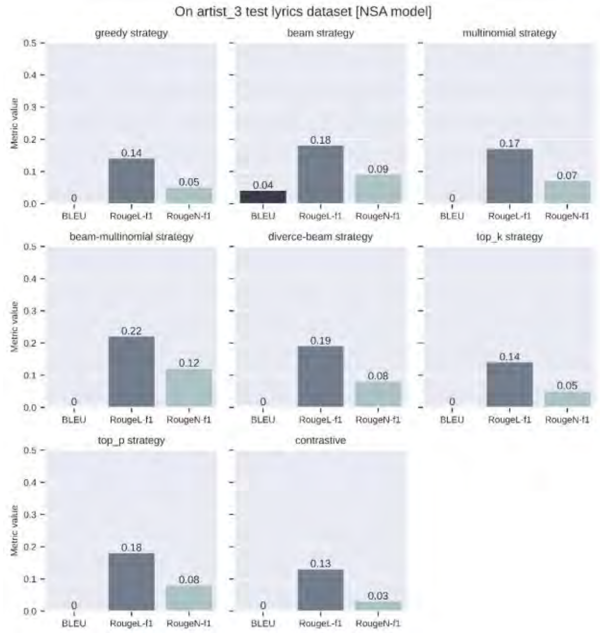


Рисунок 19 – Значення метрик для різних стратегій генерації на даних артиста-1 (модель NSA)

З усіх результатів порівняння текстів видно, що метрично найкращим методом генерації текстів пісень є beam пошук та його варіації, зокрема променевий семплінг. Контрастивний пошук зазвичай перевершував звичайний жадібний підхід. Методи top-*p* та top-*k* не мають однозначної переваги один на одним, і в різних ситуаціях давали різні результати.

Враховуючи результати, отримані при проведенні експериментів, прийнято рішення додатково дослідити кілька алгоритмів декодування на залежність значень метрик від параметрів цих алгоритмів. Для прикладу обрано дві стратегії, одну стохастичну – променевий семплінг, та одну детерміністичну – контрастивний пошук. Проблема дослідження променевого пошуку полягає у природній випадковості процесу генерації, що може суттєво впливати на отримані результати генерації. Щоб нівелювати цю проблему при проведені даного алгоритму зафіксовано random seed. Досліджуваними параметрами обрано кількість променів та температуру функції softmax. Метрика оцінювання – BLEU, множина тестового датасету – від Артиста 2.

В результаті повторної генерації та оцінювання, побудовано теплову діаграму залежності значення BLEU метрики від комбінації двох визначених параметрів для тестування NSA, виведену на рис. 20.

DOI 10.15588/1607-3274-2023-4-15

Як видно з рис. 20, збільшення температури, а відповідно згладжування розподілу (тобто створення умов для вибору менш ймовірних слів) погіршує результат, адже зростає різномастість тексту. Аналогічно, при збільшення кількості променів зменшується і значення метрики, проте цей результат може змінитися від наступного процесу генерації.

Аналогічне дослідження проведено для контрастивного пошуку. Проте його детерміністичність дозволяє уникнути проблеми різних значень при повторних генераціях. Параметрами для зіставного пошуку обрано ті, що описані в Розділі 4, а саме: k та α штрафу. Відповідні значення обчислень виведені на рис. 21.

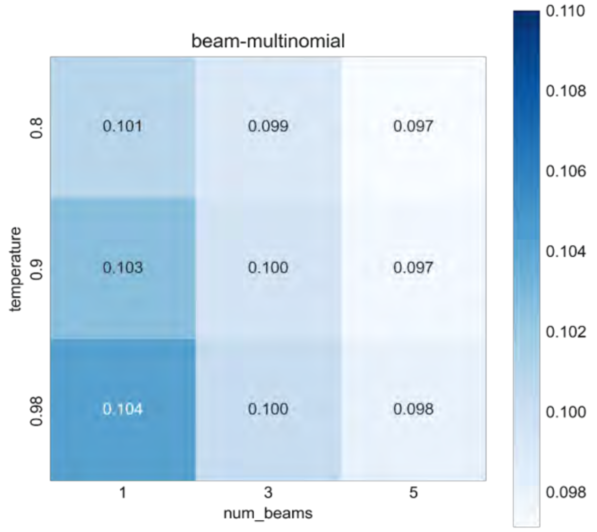


Рисунок 20 – Теплова карта BLEU метрики залежно від комбінації параметрів променевого семплінгу

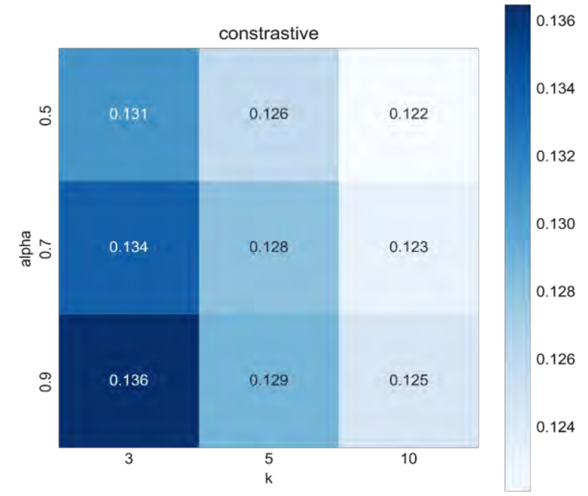


Рисунок 21 – Теплова карта BLEU метрики залежно від комбінації параметрів контрактивного пошуку

З рис. 21 видно, що зменшення штрафу призводить до зменшення значення метрики, що може бути наслідком генерування повторювальних рядків. При цьому збільшення кількості елементів, що враховуються в алгоритмі декодування, теж призводить до спаду значення метрики. Така тенденція зменшення може бути викликана ростом різноманітності та дискримінативності згенерованих стрічок, що збільшує різницю між бажаним результатом. В загальному випадку, не існує кращої стратегії декодування, а її вибір залежить від конкретних умов донавчання, наявного набору даних, можливостей моделі, задачі, та поставлених цілей процесу генерації.

В контексті розробленої ІС дві характеристики є найбільш важливими для аналізу. Перша – оцінка кількості викидів та впливу на екологію планети через використання та навчання моделей на апаратних прискорювачах. Для оцінки цієї характеристики можна використовувати ідею з [8].

Так, на момент виконання роботи, враховуючи експерименти та використання моделей, навчання двох T5 моделей на прискорювачі T4 в середовищі Google Colab, усереднено викинуто майже 0,55 кг карбон діоксиду (вуглекислого газу). Друга характеристика напряму пов’язана з першою – це час виконання генерації та декодування тексту з використанням різних стратегій. Для аналізу цього часу, при виконання генерації прикладів, зафіксовано час обчислення та декодування тексту. Звичайно, що цей час залежить від розміру отриманої стрічки, інших процесів ІС тощо. Тому, перед візуалізацією, отриманий час нормалізовано за максимальним значенням. На рис. 22 виведено умовні витрати часу для генерації результатів моделлю NSA, а на рис. 23 – моделлю SA на малих за обсягом вхідних даних.

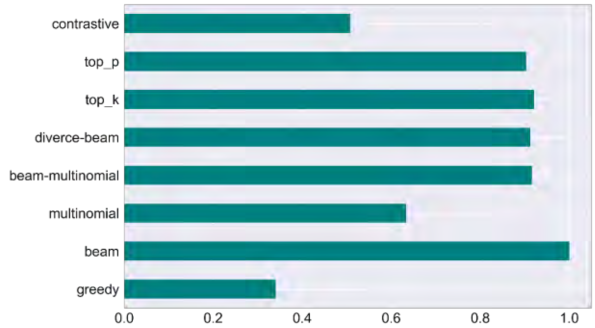


Рисунок 22 – Нормалізований час виконання генерації моделлю NSA

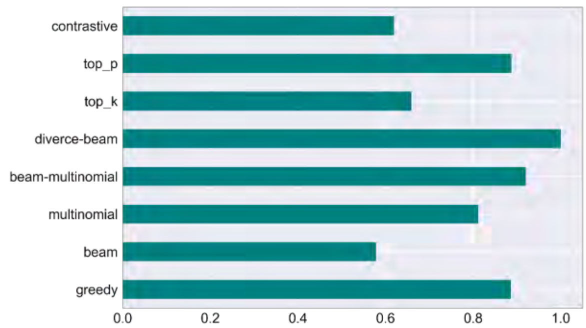


Рисунок 23 – Нормалізований час виконання генерації моделлю SA

DOI 10.15588/1607-3274-2023-4-15

В обох випадках найдовшими за часом виконання є варіації променевого пошуку, а контрастивний пошук – один з найкоротших.

ВИСНОВКИ

При формулюванні контексту даної роботи визначено основну його мету – створення ІС генерації продовження текстів пісень з та баз врахування стилю автора. У результаті виконання цієї роботи цю мету досягнуто, оскільки:

– Проведено системний аналіз ПО. Виявлено основні поняття, процеси та необхідні терміни, що дозволяють створити та реалізувати описану ІС;

– Виконано задачі створення текстового набору даних пар вхідні-вихідні стрічки пісень;

– Проведено fine-tuning двох моделей-трансформерів архітектури T5 для розв'язання задачі штучної генерації тексту проводження пісні з та без врахування стилю автора;

– Проаналізовано вплив процесу навчання на навколишнє середовище, оцінено час виконання процесу генерації, залежно від стратегії декодування;

– Наведено приклади генерації продовження пісень, проаналізовано отриманий текст;

– Кількісно оцінено якість генерації текстів створеними моделями з допомогою метрик порівняння стрічок.

В результаті стаття описує ІТ генерації продовження текстів пісень з допомогою великих мовних моделей, зокрема моделі T5, для прискорення, доповнення та підвищення гнучкості процесу написання текстів до пісень з/без врахування стиля певного автора. Для створення даних відібрано 10 різних артистів, після чого відібрано тексти їх пісень. Загалом отримано 626 унікальних пісень. В результаті розбиття кожної пісні на кілька пар вхідних-вихідних стрічок отримано 1874 навчальних екземплярів та 465 тестових. Проведено донавчання двох мовних моделей NSA та SA для задачі генерації продовження тексту пісень. Для обох моделей як базову обрано t5-base. Ця версія T5 містить 223 мільйони параметрів. Аналіз вихідних даних показав, що модель NSA має менш деградаційні результати, а для моделі SA необхідно збалансовувати кількість тексту для кожного автора. Обраховано кілька текстових метрик як BLEU, RougeL та RougeN для кількісного порівняння результатів моделей та стратегій генерування. Значення метрики BLEU є найрізноманітнішим, і його значення значно змінюється, залежно від стратегії. При цьому Rouge метрики мають меншу варіативність, менший розмах значень. Загалом, для порівняння, у роботі використано 8 різних методик декодування для генерації тексту, що підтримуються бібліотеку transformers, зокрема Contrastive search, Top-p sampling, Top-k sampling, Multinomial sampling, Beam search, Diverse beam search, Greedy search, та Beam-search multinomial sampling. З усіх результатів порівняння текстів видно, що метрично найкращим методом генерації текстів пісень є beam пошук та його варіації, зокрема променевий семплінг. Контрастивний пошук зазвичай перевершував звичайний жадібний підхід. Методи top-p та top-k не мають однозначної переваги один на одним, і в різних ситуаціях давали різні результати.

DOI 10.15588/1607-3274-2023-4-15

Логічно, що утворена ІС має шляхи розвитку та вдосконалення. Основними перспектива для покращення ІС є такі, що стосуються даних навчання, моделей трансформерів та безпосередньо процесу розробки. Тому можна виділити такі напрямки майбутніх досліджень:

– Використання більш нових чи потужних моделей ніж класична T5;

– Розширення кількості та різноманітності навчальних даних. Заміна або поєднання кількох видів задачі генерації текстів пісень;

– Розширення групи мов, що підтримуються ІС;

– Включення інших структурних елементів пісні (ритм, акорди, розмірність віршового рядка) або музичний супровід відповідної пісні як вхідну інформацію для генерації текстів;

– Вдосконалення процесу навчання моделей, наприклад застосуючи контрастивний процес навчання або refinement, тобто залучення справжніх авторів для оцінки якості та покращення результатів моделей.

Основні перспективи проведення майбутніх досліджень, для поглиблення та вдосконалення аналізу стратегій генерації текстів пісень є:

– Виявлення, створення та оцінка нових чи інших методів декодування тексту для генерації тексту, що може включати і комбінування вже розглянутих алгоритмів;

– Перехід до інших класів мовних моделей-трансформерів (BERT, GPT, Bart, T5v1.1 тощо), та порівняльний аналіз стратегій для цих моделей;

– Огляд або розробка інших метрик, що можуть включати порівняння семантичного чи синонімічного наповнення тексту, риму, чи збереження розміру віршованих текстів.

ЛІТЕРАТУРА

1. Attention Is All You Need / [A. Vaswani, N. Shazeer, N. Parmar et al.] // arXiv. – Access mode: https://arxiv.org/abs/1706.03762
2. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer / [C. Raffel, N. Shazeer, A. Roberts et al.] // arXiv. – Access mode: https://arxiv.org/abs/1910.10683
3. Hugging Face community. Text generation strategies. – Access mode: https://huggingface.co/docs/transformers/v4.29.0/en/generation_strategies
4. von Platen P. How to generate text: using different decoding methods for language generation with Transformers / P. von Platen. – Access mode: https://huggingface.co/blog/how-to-generate
5. Hugging Face community. T5. – Access mode: https://huggingface.co/docs/transformers/model_doc/t5

6. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models / [A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju et al.] // arXiv. – Access mode: https://arxiv.org/abs/1610.02424
7. Hugging Face community. T5v1.1. – Access mode: https://huggingface.co/docs/transformers/model_doc/t5v1.1
8. Quantifying the Carbon Emissions of Machine Learning / [A. Lacoste, A. Luccioni, V. Schmidt, T. Dandres] // arXiv. – Access mode: https://arxiv.org/abs/1910.09700
9. Google. SentencePiece. – Access mode: https://github.com/google/sentencepiece
10. Sennrich R. Subword Neural Machine Translation / R. Sennrich. – Access mode: https://github.com/rsennrich/subword-nmt
11. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation / [Y. Wu, M. Schuster, Z. Chen et al.] // arXiv. – Access mode: https://arxiv.org/abs/1609.08144
12. Hugging Face community. Transformers. State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX. – Access mode: https://huggingface.co/docs/transformers/index
13. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems / [M. Abadi, A. Agarwal, P. Barham et al.] // arXiv. – Access mode: https://arxiv.org/abs/1603.04467
14. Shah D. Song Lyrics Dataset / D. Shah // Kaggle. – Access mode: https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset
15. Swift T. Long story short / T. Swift, A. Dessner // Genius. Taylor Swift Music. – Access mode: https://genius.com/Taylor-swift-long-story-short-lyrics
16. Fan A. Hierarchical Neural Story Generation / A. Fan, M. Lewis, Y. Dauphin // Association for Computational Linguistics : 56th Annual Meeting, Melbourne, Australia, July 2018 : proceedings. – Melbourne: ACL, 2018. – P. 889–898. DOI: 10.18653/v1/p18-1082
17. Chiang T.-R. Relating Neural Text Degeneration to Exposure Bias / T.-R. Chiang, Y.-N. Chen // Analyzing and Interpreting Neural Networks for NLP : the Fourth BlackboxNLP Workshop, Punta Cana, Dominican Republic, November 2021 : proceedings. – Punta Cana: ACL, 2021. – P. 228–239. DOI: 10.18653/v1/2021.blackboxnlp-1.16
18. A Contrastive Framework for Neural Text Generation / [Y. Su, T. Lan, Y. Wang et al.] // arXiv. – Access mode: https://arxiv.org/abs/2202.06417
19. Paulus R. A Deep Reinforced Model for Abstractive Summarization / R. Paulus, C. Xiong, R. Socher // arXiv. – Access mode: https://arxiv.org/abs/1705.04304
20. OpenNMT: Open-Source Toolkit for Neural Machine Translation / [G. Klein, Y. Kim, Y. Deng et al.] // System Demonstrations : Association for Computational Linguistics, Vancouver, Canada, July 2017 : proceedings. – Vancouver: ACL, 2017. – P. 67–72. DOI: 10.18653/v1/p17-4012
21. Murray K. Correcting Length Bias in Neural Machine Translation / K. Murray, D. Chiang // arXiv. – Access mode: https://arxiv.org/abs/1808.10006
22. Mathur N. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics / N. Mathur, T. Baldwin, T. Cohn // Association for Computational Linguistics : 58th Annual Meeting, Online, July 2020 : proceedings. – Online: ACL, 2020. – P. 4984–4997. DOI: 10.18653/v1/2020.acl-main.448
23. Lin C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries / C.-Y. Lin // Text Summarization Branches Out : Association for Computational Linguistics, Barcelona, Spain, July 2004 : proceedings. – Barcelona: ACL, 2004. – P. 74–81. Access mode: https://aclanthology.org/W04-1013
24. KerasNLP. Access mode: https://keras.io/keras_nlp/
25. Prokipchuk O. Ukrainian Language Tweets Analysis Technology for Public Opinion Dynamics Change Prediction Based on Machine Learning / O. Prokipchuk, V. Vysotska // Radio Electronics, Computer Science, Control. – 2023. – No. 2 (65). – P. 103–116. DOI: 10.15588/1607-3274-2023-2-11

Стаття надійшла до редакції 15.09.2023.
Після доробки 25.10.2023.

UDC 004.9

SONGS CONTINUATION GENERATION TECHNOLOGY BASED ON TEST GENERATION STRATEGIES, TEXTMINING AND LANGUAGE MODEL T5

Mediakov O. – Post-graduate student of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

Vysotska V. – PhD, Associate Professor of Information Systems and Networks Department, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. Pre-trained large language models are currently the driving force behind the development of not only NLP, but also deep learning systems in general. Model transformers are able to solve virtually all problems that currently exist, provided that certain requirements and training practices are met. In turn, words, sentences and texts are the basic and most important way of communication between intellectually developed beings. Of course, speech and texts are used to convey certain emotions, events, etc. One of the main ways of using language to describe experienced emotions is songs with lyrics. However, often due to the need to preserve rhyme and rhyming, the dimensions of verse lines, song structure, etc., artists have to use repetition of lines in the lyrics. In addition, the process of writing texts can be long.

Objective of the study is to develop information technology for generating the continuation of song texts based on the T5 machine learning model with (SA, specific author) and without (NSA, non-specific author) consideration of the author's style.

Method. Choosing a decoding strategy is important for the generation process. However, instead of favoring a particular strategy, the system will support multiple strategies. In particular, the following 8 strategies: Contrastive search, Top-p sampling, Top-k sampling, Multinomial sampling, Beam search, Diverse beam search, Greedy search, and Beam-search multinomial sampling.

Results. A machine learning model was developed to generate the continuation of song lyrics using large language models, in particular the T5 model, to accelerate, complement and increase the flexibility of the songwriting process.

DOI 10.15588/1607-3274-2023-4-15

Conclusions. The created model shows excellent results of generating the continuation of song texts on test data. Analysis of the raw data showed that the NSA model has less degrading results, while the SA model needs to balance the amount of text for each author. Several text metrics such as BLEU, RougeL and RougeN are calculated to quantitatively compare the results of the models and generation strategies. The value of the BLEU metric is the most variable, and its value varies significantly depending on the strategy. At the same time, Rouge metrics have less variability, a smaller range of values. For comparison, 8 different decoding methods for text generation, supported by the transformers library, were used. From all the results of the text comparison, it is clear that the metrically best method of song text generation is beam search and its variations, in particular beam sampling. Contrastive search usually outperformed the conventional greedy approach. The top-p and top-k methods are not clearly superior to each other, and in different situations gave different results.

KEYWORDS: text generation, T5 language model, Transformers, author's style, Contrastive search, Top-p sampling, Top-k sampling, Multinomial sampling, Beam search, Diverse beam search, Greedy search, and Beam-search multinomial sampling.

REFERENCES

1. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention Is All You Need, *arXiv.* Access mode: https://arxiv.org/abs/1706.03762
2. Raffel C., Shazeer N., Roberts A., Lee K., Narang S., Matena M., Zhou Y., Li W., Liu P. J. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, *arXiv.* Access mode: https://arxiv.org/abs/1910.10683
3. Hugging Face community. Text generation strategies. Access mode: https://huggingface.co/docs/transformers/v4.29.0/en/generation_strategies
4. von Platen P. How to generate text: using different decoding methods for language generation with Transformers. Access mode: https://huggingface.co/blog/how-to-generate
5. Hugging Face community. T5. Access mode: https://huggingface.co/docs/transformers/model_doc/t5
6. Vijayakumar A. K., Cogswell M., Selvaraju R. R., Sun Q., Lee S., Crandall D., Batra D. Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models, *arXiv.* Access mode: https://arxiv.org/abs/1610.02424
7. Hugging Face community. T5v1.1. Access mode: https://huggingface.co/docs/transformers/model_doc/t5v1.1
8. Lacoste A., Luccioni A., Schmidt V., Dandres T. Quantifying the Carbon Emissions of Machine Learning, *arXiv*. Access mode: https://arxiv.org/abs/1910.09700
9. Google. SentencePiece. Access mode: https://github.com/google/sentencepiece
10. Sennrich R. Subword Neural Machine Translation. Access mode: https://github.com/rsennrich/subword-nmt
11. Wu Y., Schuster M., Chen Z., Le Q. V., Norouzi M., Macherey W., Krikun M., Cao Y., Gao Q., Macherey K., Klingner J., Shah A., Johnson M., Liu X., Kaiser Ł., Gouws S., Kato Y., Kudo T., Kazawa H., Stevens K., Kurian G., Patil N., Wang W. , Young C., Smith J., Riesa J., Rudnick A., Vinyals O. , Corrado G., Hughes M., Dean J. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, *arXiv.* Access mode: https://arxiv.org/abs/1609.08144
12. Hugging Face community. Transformers. State-of-the-art Machine Learning for PyTorch, TensorFlow, and JAX. Access mode: https://huggingface.co/docs/transformers/index
13. Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G. S., Davis A. , Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mane D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viegas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, *arXiv.* Access mode: https://arxiv.org/abs/1603.04467
14. Shah D. Song Lyrics Dataset, *Kaggle*. Access mode: https://www.kaggle.com/datasets/deepshah16/song-lyrics-dataset
15. Swift T., Dessner A. Long story short, *Genius. Taylor Swift Music*. Access mode: https://genius.com/Taylor-swift-long-story-short-lyrics
16. Fan A., Lewis M., Dauphin Y. Hierarchical Neural Story Generation, *Association for Computational Linguistics : 56th Annual Meeting, Melbourne, Australia, July 2018 : proceedings*. Melbourne, ACL, 2018, pp. 889–898. DOI: 10.18653/v1/p18-1082
17. Chiang T.-R., Chen Y.-N. Relating Neural Text Degeneration to Exposure Bias, *Analyzing and Interpreting Neural Networks for NLP : the Fourth BlackboxNLP Workshop, Punta Cana, Dominican Republic, November 2021 : proceedings.* Punta Cana, ACL, 2021, pp. 228–239. DOI: 10.18653/v1/2021.blackboxnlp-1.16
18. Su Y., Lan T., Wang Y., Yogatama D., Kong L., Collier N. A Contrastive Framework for Neural Text Generation, *arXiv*. Access mode: https://arxiv.org/abs/2202.06417
19. Paulus R., Xiong C., Socher R. A Deep Reinforced Model for Abstractive Summarization, *arXiv.* Access mode: https://arxiv.org/abs/1705.04304
20. Klein G., Kim Y., Deng Y., Senellart J., Rush A. OpenNMT: Open-Source Toolkit for Neural Machine Translation, *System Demonstrations : Association for Computational Linguistics, Vancouver, Canada, July 2017 : proceedings.* Vancouver, ACL, 2017, pp. 67–72. DOI: 10.18653/v1/p17-4012
21. Murray K., Chiang D. Correcting Length Bias in Neural Machine Translation, *arXiv.* Access mode: https://arxiv.org/abs/1808.10006
22. Mathur N., Baldwin T., Cohn T. Tangled up in BLEU: Reevaluating the Evaluation of Automatic Machine Translation Evaluation Metrics, *Association for Computational Linguistics : 58th Annual Meeting, Online, July 2020 : proceedings*. Online, ACL, 2020, pp. 4984–4997. DOI: 10.18653/v1/2020.acl-main.448
23. Lin C.-Y. ROUGE: A Package for Automatic Evaluation of Summaries, *Text Summarization Branches Out : Association for Computational Linguistics, Barcelona, Spain, July 2004 : proceedings*. Barcelona, ACL, 2004, pp. 74–81. Access mode: https://aclanthology.org/W04-1013
24. KerasNLP. Access mode: https://keras.io/keras_nlp/
25. Prokipchuk O., Vysotska V. Ukrainian Language Tweets Analysis Technology for Public Opinion Dynamics Change Prediction Based on Machine Learning, *Radio Electronics, Computer Science, Control*, 2023, No. 2(63), pp. 103–116. DOI: 10.15588/1607-3274-2023-2-11

DOI 10.15588/1607-3274-2023-4-15

UDC 550.34.01

DETECTION OF A SEISMIC SIGNAL BY A THREE-COMPONENT SEISMIC STATION AND DETERMINATION OF THE SEISMIC EVENT CENTER

Vakaliuk T. A. – Dr. Sc., Professor, Professor at the Department of software engineering, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine.

Pilkevych I. A. – Dr. Sc., Professor, Professor at the Department of Computer Information Technologies, Korolov Zhytomyr Military Institute, Zhytomyr, Ukraine.

Hordiienko Y. O. – PhD, Head of the Research Department of the Scientific Center, Korolov Zhytomyr Military Institute, Zhytomyr, Ukraine.

Loboda V. V. – Researcher Officer of the Scientific Center, Korolov Zhytomyr Military Institute, Zhytomyr, Ukraine.

Saliy A. O. – Course Officer of Retraining and Advanced Training Courses, Korolov Zhytomyr Military Institute, Zhytomyr, Ukraine.

ABSTRACT

Context. The work is devoted on the development of theoretical foundations aimed at automating process of determining location at the seismic event center.

Objective. The purpose of the work is to develop a method for determining the center of a seismic event based on the use of the features of the angular characteristics of the constituent volume waves of a seismic signal obtained with the help of a three-component seismic station. The proposed method will reduce the time it takes to provide users with preliminary information about the fact of a seismic event and its parameters.

Method. The method of automatic detection focal point is based on features of orthogonality in the angular characteristics volume waves registered sample of three-component seismic recordings from a certain direction. Implementation of the proposed approaches makes it possible to reduce the processing time of the seismic record with appropriate reliability compared to processing in manual mode.An example application of the proposed method (algorithm) for processing a seismic signal in the Vrancea zone on 27.10.2004 with magnitude M=5.7 is considered.

Results. The proposed approach to processing the measured data of a separate seismic three-component seismic station using a polarization analysis device allows detecting the arrival of a seismic signal, identifying the main components of a seismic signal, and estimating the location of the epicenter of a seismic event.Experimental research on the use of the proposed algorithm for determining the location of the epicenter of a seismic event showed that the time of establishing an emergency event within the borders of Ukraine was reduced five times (from 15 to 3 minutes), and the detection error was 37 km.

Conclusions. The formed basis and proposed approach to detecting a seismic signal, identifying its components and determining a seismic event focal point based on results of processing a three-component seismic record are effective. Proposed method (algorithm) should be used to automate process of seismic signal detection by a three-component seismic station and to determine the seismic event center.

KEYWORDS: Earthquake, seismic waves, seismic signal, signal detection, three-component seismic station, seismic monitoring, automatically controlled monitoring system, source of emergency.

ABBREVIATIONS

CPS is a Civil Protection System;
CSE is center of the seismic event;
DMNS is a disaster monitoring system;
ES is emergency situations;
MCSM is a Main Center for Special Monitoring;
OP is a observation points;
SGS is a seismic grouping system;
SON is a seismic observation network;
SSA is a State Space Agency;
STA/LTA Short Time Average to Long Time Average;
TCSS is a three-component seismic stations;
UN is a United Nations.

NOMENCLATURE

T is a sample duration for which signal estimation is performed;

η is a signal-to-noise value;

$\{n_i, e_i, z_i\}$ – a actual coordinates of soil particle displacement;

G is a linearity coefficient;

b is a smallest semi-axis of ellipsoid;

c is a largest semi-axis of ellipsoid;

M is a covariance matrix;

G_i is a current value of the linearity coefficient;

P-waves is a seismic wave that travels through the Earth in a longitudinal manner;

S-waves is another type of seismic wave that travel through the Earth;

L_r Rayleigh is surface waves that have both longitudinal and transverse motion;

L_q Love is surface waves that only have transverse motion;

Ω_i value of angle between the position of the main semi-axis of ellipsoid for the obtained sample and direction of *P*-wave arrival;

DOI 10.15588/1607-3274-2023-4-16

$\{x_p, y_p, z_p\}$ – coordinates of a unit vector position for the main axis in *P*-wave;

$\{x_i, y_i, z_i\}$ – coordinates of a unit vector position the main semi-axis by the current sample.

INTRODUCTION

The direct result of human activity is man-made ES, which arise as a result of disruptions in normal living conditions and human activity at facilities or territories. According to UN, more than 70 thousand people die annually in Ukraine as a result of emergencies and significant material losses are incurred by the state [1].

Earthquakes are one of the most dangerous natural phenomena that can occur in Ukraine. Depending on causes and location, earthquakes are divided into tectonic, volcanic, landslide, and seaquakes [2]. Earthquake centers are located almost 60 km deep, and sometimes up to 700 km deep. Many earthquakes are accompanied by large human casualties. Number of victims depends on suddenness of natural disaster, strength, area of damage and population density in the area of the earthquake.

In Ukraine, seismically active zones are located in the southwest and south: Zakarpattia, Vrancha, Crimean-Black Sea and South Azov zones. It is known [2] that 290 thousand square kilometers of the territory of our country with a population 15 million people are located in earthquake zones. In order to organize and ensure the protection against consequences of man-made disasters, the CPS was created in Ukraine [3]. Main tasks of civil protection are as follows [4]: prevention and implementation measures to reduce damages and losses in case of industrial and natural disasters; prompt notification on occurrence or threat of disasters, timely and reliable information about the current situation and measures taken to prevent disasters and overcome their consequences; organization of protection against disasters, provision emergency psychological, medical and other assistance to victims.

In order to fulfill tasks assigned to the civil protection system, it is necessary timely receive information about emergencies and their consequences. For this purpose, according to the Civil Protection Code of Ukraine, a DMNS is created. In Ukraine, the disaster monitoring system is based on collecting and analyzing information coming from various means of control and surveillance about conditions of relevant hazardous facilities or territories. At the same time, the task of enhancing SES capabilities by expanding monitoring methods is becoming urgent.

One information segment within the DMNS is the MCSM of SSA, which, through the National Seismic Observation System and the Governmental Information and Analytical System for Disasters, provides information to the CPS on the seismic situation in Ukraine and neighboring countries [5]. The SON of MCSM includes TCSS, a SGS, which was included in the International Seismic Monitoring Network as PS-45 station, and a National Data Center (Fig. 1).

Figure 1 – Network of seismic observation MCSM SSA of Ukraine

The object of study is the process for monitoring seismic situation and determining location of a seismic event center.

The subject of study is method automatic determination of seismic event center.

Currently, processing of seismic observation data is carried out in manual and automatic modes. However, decisions on seismic event parameters and possible consequences are made by operational duty officer on duty of MCSM based on the results analysis of information on seismic signal parameters (time arrival of main types a seismic waves, their amplitude and period) received from each observation point [5]. The temporary loss the Crimean peninsula led to territorial limitations of the GCSC seismic observation network. This necessitated the development algorithms for identifying nature a seismic source based upon results of seismic data processing in automatic mode from individual observation points.

The aim of the work is to develop a methodology for determining location of the seismic event center based on data analysis of a three-component seismic station, which is part of the MCSM SON.

1 PROBLEM STATEMENT

Currently, the MCSM has implemented methods of seismic measurement data processing that allow providing users with preliminary information about the parameters of a seismic event and possible consequences within 15 minutes, and the final information within 40 minutes from the time of event [6]. This period is due to data processing based on algorithms for "manual" processing by seismic station operators. Within the framework established by national and international programs, the MCSM is modernizing seismic observation equipment, transmission and processing of measurement data (transition to digital information processing). This will allow us to move to a qualitatively new level of seismic monitoring.

These circumstances require solving an important scientific task, essentially developing a method for determining a center of a seismic event based on determining angular characteristics of components of TCSS seismic signal volume waves.

DOI 10.15588/1607-3274-2023-4-16

Formulate a mathematical statement of the problem. Assume a given distance between a post and a monitoring object, as well as angular characteristics of first seismic signal arrival at ground surface. Then the problem of determining a seismic event center by a three-component seismic station is to calculate coordinates of a seismic source. Specifically, azimuth at the source relative from the observation point and distance between our observation point and source of seismic event. A polarization analysis apparatus is used to solve the problem.

2 REVIEW OF THE LITERATURE

In [7], a method is considered of data polarization analysis, which determines angular characteristics a seismic wave arrives to the Earth's surface in certain seismic survey areas identified as a result of preliminary detection. Polarization analysis units used to process TCSS measurement data are also considered. This method of automatic seismic event detection is based on neural networks. In general, the method is reliable and capable of correctly identifying phase types, while ensuring the accuracy and precision needed to assess them. The disadvantage of using this method is that it is difficult to apply.

In [8], another approach to using polarization analysis devices is considered, which consists in increasing a signal-to-noise ratio of seismic vibrations from a certain direction (polarization filtering). This approach allows detecting seismic signals from sources with cells located along the propagation path (beam), which is characterized by azimuth α and angle of incidence on Earth's surface γ, but does not determine the exact position within the beam.

In [9], automatic methods for detecting seismic events are considered, which allow for quick data analysis. The simplest automatic methods for processing seismic data that have become popular in international and national data centers around the world are STA/LTA [10]. These methods use a criterion for exceeding amplitude thresholds when detecting seismic signals based on TCSS observations. The disadvantage with these methods is that their application leads to an increase in the number of observations in the seismic monitoring system and, accordingly, the number of individual OP.

Since MCSM seismic station network has limited territorial coverage, especially after temporary loss of Sevastopol and Yevpatoriya stations due to occupation of Crimea by the Russian Federation, there is a need to develop methodological principles for solving seismic monitoring problems by individual stations with TCSS [11]. In addition, this need is caused by problems of ensuring functional stability of seismic observation network as a whole, especially in cases when seismic data from several stations cannot be used – equipment failure, communication interruption, maintenance.

Thus, it is important to develop an automated method for determining a focus of a seismic event that could lead to an emergency situation involving man-made and natural hazards using a single OP.

3 MATERIALS AND METHODS

Solving seismic monitoring tasks at a single observation point consists of the following stages: detection a seismic signal, identification of seismic record components (determination of seismic wave types), location a seismic event center, and estimation seismic source parameters. In case of single-position observations, the last three stages are solved if the problem of determining main components of the seismic record is confidently solved. Therefore, when analyzing the existing methods of seismic data detection and processing, the main attention will be paid to this problem solving capability. Another criterion should be the simplicity of the software and algorithmic implementation of processing methods, which in turn will ensure a possibility of processing the measurement data in real time.

Most of the implemented approaches to detecting seismic signals based on TCSS observations use the criterion of exceeding the threshold in amplitude, which are quite effective at an energy signal-to-noise ratio of at least 3. At present, the STA/LTA detector, whose definition is given in [7], is widely used for preliminary detection of seismic signals based on TCSS observations

$$STA_i = \frac{1}{T}\sum_{j=i}^{i+T}\sqrt{n_j^2 + e_j^2 + z_j^2}, \qquad (1)$$

$$LTA_i = \frac{1}{5T}\sum_{j=i-5T}^{i}\sqrt{n_j^2 + e_j^2 + z_j^2}, \qquad (2)$$

$$\eta_i = \frac{STA_i}{LTA_i}. \qquad (3)$$

This approach requires a relatively small amount of computation, which is a significant argument for its use in real-time measurement data processing systems [7]. However, use of this approach leads to a limitation of the magnitude sensitivity of the seismic station. In addition, this method doesn't allow to accurately determine the components of a seismic signal, since arrival of a next seismic wave occurs against background of tail part of the previous one.

Figure 2 shows results of processing seismic signals from earthquakes with centers in the Vrancea seismically active zone (Romanian part of the Carpathians) using the STA/LTA detection method.

As can be seen from this example, the application of the amplitude-based detection method doesn't always allow to determine the seismic signal components. For the second signal after first arrival, threshold is exceeded five times, since the arrival of next seismic wave occurs against a background of tail part of the previous one (Figure 2 b).

DOI 10.15588/1607-3274-2023-4-16

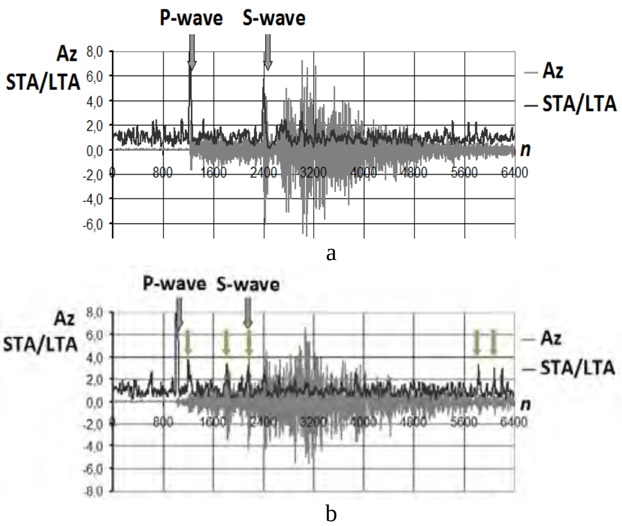


Figure 2 – Results of processing seismic signals from earthquakes with centers in the Vrancea zone, using the STA/LTA detection method: a – 27.9.2004, M=4.6; b – 27.10.2004, M=5.7

As can be seen from this example, the application of the amplitude-based detection method doesn't always allow to determine the seismic signal components. For the second signal after first arrival, threshold is exceeded five times, since the arrival of next seismic wave occurs against a background of tail part of the previous one (Figure 2 b).

Importance of information.

When solving problems of detecting seismic signal components based on the results of processing measurement data of a three-component seismic recording, it is necessary to apply additional criteria (features).

4 EXPERIMENTS

One of features of the seismic signal and its components in TCSS recording is polarization properties [7, 9]. Recordings of seismic waves from explosions, earthquakes, and other sources are characterized by linear polarization of oscillations, while noise is the result a superposition of waves arriving on the station from different sources and having a low level of linearity of polarization. These differences between signals and noise can be detected by polarization analysis of oscillations. They are especially useful in detecting signals with a frequency close to noise frequency, where frequency filtering is ineffective. Advantage of using PAA is that its results, in addition to the time of seismic signal arrival, make it possible to determine main components of seismic record and their angular characteristics (azimuth and angle of exit to day surface), which in turn is related to location of seismic event center relative to OP [8].

Figure 3 shows the nature of soil particle displacement for main types a seismic waves typical for seismic events with foci in local and regional zones.

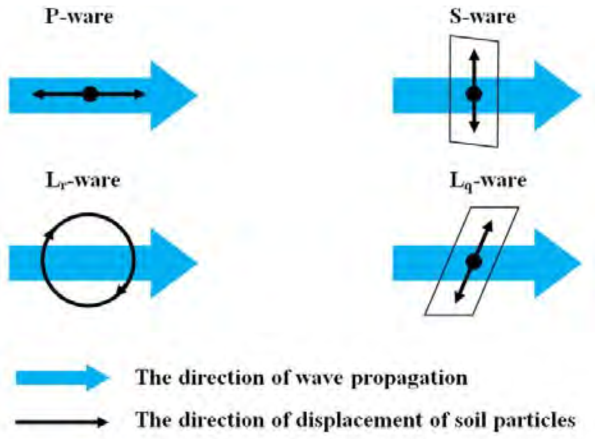


Figure 3 – The nature of soil displacement for different types of seismic waves

According to the nature of soil movement during seismic wave propagation, they are divided into volumetric – longitudinal *P*, transverse *S*; and surface – L_r and L_q waves. For *P*-waves, compression and liquefaction occurs in direction of wave propagation. For *S*-waves, soil displacement occurs perpendicular to direction of wave propagation. Along with bulk waves, surface waves propagate. These waves are of two types: L_r and L_q. In a Rayleigh wave, soil particles move in a vertical plane oriented along wave propagation, and their trajectories form ellipses. In the Lova wave, particles move in a horizontal plane across wave propagation direction [12].

Figure 4 presents a three-component record of seismic signal from earthquake with a center in Vrancea seismic zone (1.05.2011, M=4.8) registered by TCSS installed at OP Vorsovka (Zhytomyr region) and shows horizontal components of soil particle displacement for background and seismic signal components. A similar pattern of ground vibrations is typical for the Chornobyl NPP location.

Taking into account specific features of soil particle displacement for each main type of seismic waves in an event with centers in regional areas, angle characteristics of such waves will be related to the position of the seismic source relative to the OP as follows [8]:

– *P*-wave – since oscillation of particles in soil occurs along seismic wave propagation direction, the calculated angular characteristics coincide with position of seismic event center relative to the OP ($\alpha_P=\alpha_{sec}$, $\gamma_P=\gamma_{sec}$);

– *S*-wave – due to soil oscillation occurs perpendicular to direction of wave propagation at this phase, angular characteristics are calculated (α_S, γ_S) will be different from direction to seismic event center relative to OP at 90°;

– *Lr*-wave – a superficial wave with elliptical polarization focused perpendicular to propagation direction, so its calculated azimuth will be different from input azimuth one on 90°,that is $\alpha_{Lr} \perp \alpha_P$;

– *Lq* – wave is a surface wave with an elliptical polarization in direction of propagation, so the calculated azimuth will coincide with actual azimuth to seismic source ($\alpha_{Lq}=\alpha_P$) [3].

DOI 10.15588/1607-3274-2023-4-16

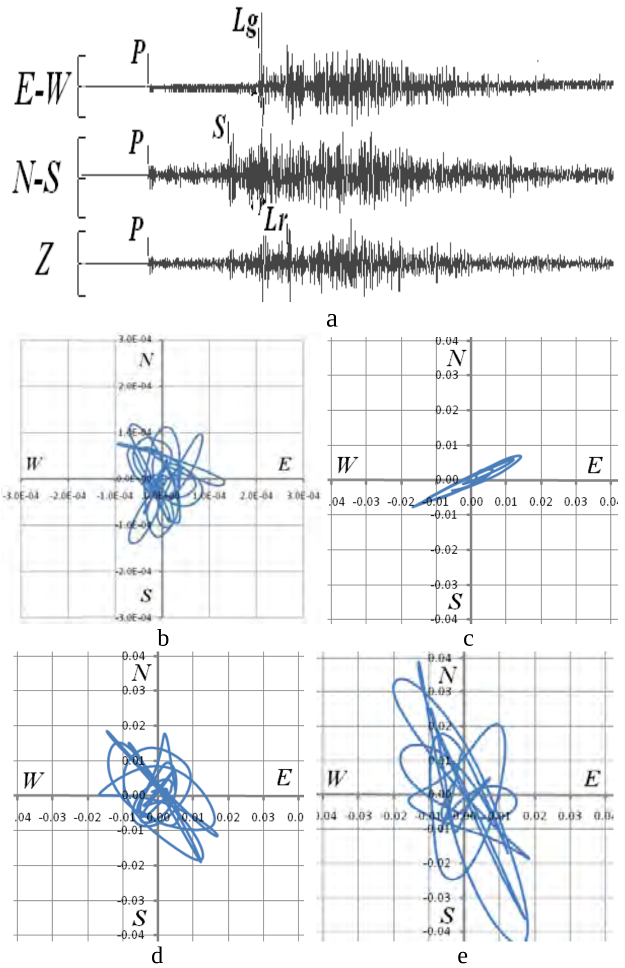


Figure 4 – Seismic signal from an earthquake with a focus in Vrancea area (a), horizontal displacements of soil particles for background area (b) and seismic signal components *P*-wave (c), *S*-wave (d) and L_g-wave (e)

At the same time, for seismic events with foci in local and regional zones, it is advisable to limit detection and identification to only bulk waves, since for a group of surface waves, due to relatively small distances, differences in arrival times of surface waves between each other and *S*-wave are insignificant, which leads to a complex wave pattern.

Taking into account the peculiarities of displacement particles in soil during passage by bulk waves, detection S-wave seismic signal can be realized as a search for seismic signal recording area for which conditions of orthogonal angular characteristics are met $(\alpha_P, \gamma_P) \perp (\alpha_S, \gamma_S)$.

Polarization analysis of a three-component seismic record. Trajectories of soil particles during seismic wave propagation have a shape of elongated ellipses, and in case of noise, they are close to circular. Polarization analysis of a seismic wave recording allows us to numerically estimate how closely oscillations describe a shape that corresponds to an ellipsoid. For this purpose, a three-component recording is used to $\{x_i, y_i, z_i\}$ covariance matrix M is calculated [13].

$$M = \begin{Vmatrix} \text{cov}(x,x) & \text{cov}(x,y) & \text{cov}(x,z) \\ \text{cov}(y,x) & \text{cov}(y,y) & \text{cov}(y,z) \\ \text{cov}(z,x) & \text{cov}(z,y) & \text{cov}(z,z) \end{Vmatrix}. \qquad (4)$$

A quadratic shape (ellipsoid) defined by this matrix is reduced to major axes. Major axis of ellipsoid characterizes orientation in space of full seismic wave displacement vector by angles – azimuth α and angle of incidence γ. At the same time, azimuth of the *P*-wave arrival coincides with azimuth of seismic event center relative to the observation point. Linearity coefficient G $(0<G<1)$ adopted for the implementation a three-component recording is defined as [11]:

$$G = 1 - \frac{b}{c}. \qquad (5)$$

For determining a degree of linearity in a three-component seismic record, sequential polarization filtering, approximation the trajectory path of the soil particles by an ellipsoid, etc. [11, 13, 14]. Despite different methodological principles, main goal of known approaches to polarization analysis is determining how close oscillations within a registered sample of a three-component seismic record are to a certain direction.

5 RESULTS

Feature of angular characteristics orthogonality of volume waves can be used as a basis for the algorithm of seismic event focal point detection. Based on above, algorithm for determining seismic event focal point by using angular characteristics of seismic signal volume wave components based on results of TCSS measurement data processing will include the following stages:

Stage 1 – detection of seismic signal arrival (*P*-wave);

Stage 2 – estimation of angular characteristics of the seismic signal arrival (*P*-wave) – α_P and γ_P;

Stage 3 – finding the seismic signal recording areas with angular characteristics that correspond to the S-wave – fulfillment of the condition $(\alpha_P, \gamma_P) \perp (\alpha_S, \gamma_S)$;

Stage 4 – determination coordinates of the seismic source center.

Let us consider application this approach (algorithm) for processing seismic signal from earthquake with center in Vrancea zone on 27.10.2004, M=5.7 (Figure 5).

At first stage, seismic signal arrival is detected by amplitude criterion using STA/LTA detector (Figure 5 a). Detector parameters: T = 40 samples, which at a sampling rate of 40 Hz corresponds to 1s of seismic recording. A decision on presence signal is made if the STA/LTA threshold is exceeded STA/LTA≥3 (Figure 5 b).

DOI 10.15588/1607-3274-2023-4-16

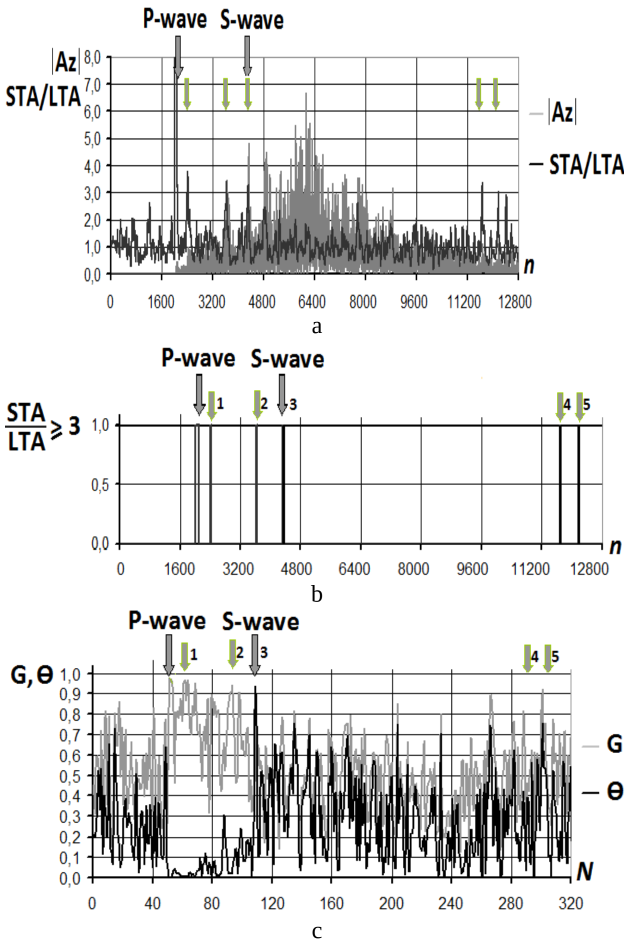


Figure 5 – Results of processing a seismic signal from an earthquake with a center in the Vrancea zone (27.10.2004, M=5.7) by a proposed method: a – envelope of a seismic record; b – STA/LTA detector response; c – values of linearity coefficient G and solving function θ

At stage two, polarization analysis apparatus (4) is used for first seismic signal arrival, which results in value of linearity degree G_P, angular position of major axis ellipsoid, which in turn determines azimuth of seismic wave arrival α_P by angle of exit to day surface γ_P. Azimuth of *P*-wave arrival coincides with azimuth to seismic event center relative to observation point.

At a third stage, we search for a recording section for which a condition is fulfilled – angular position for the main semi-axis ellipsoid for a sample obtained is orthogonal to position of the *P*-wave $(\alpha_P, \gamma_P) \perp (\alpha_i, \gamma_i)$. Determination from the seismic signal component corresponding to *S*-wave is performed by searching for a maximum value of solving function (Figure 5 c):

$$\theta_i = G_i \cdot \sqrt{1 - \Omega_i^2}\ ; \qquad (6)$$

$$\Omega_i = \left| x_P \cdot x_i + y_P \cdot y_i + z_P \cdot z_i \right| . \qquad (7)$$

Coordinates a unit vector position main axis are related to azimuth α and angle exit γ seismic wave and is defined as [9]:

$$x = \cos(\gamma) \cdot \cos(\alpha);$$
$$y = \cos(\gamma) \cdot \sin(\alpha); \qquad (8)$$
$$z = \sin(\gamma).$$

In the fourth stage, coordinates for a seismic event focal point are determined by applying a direct geodetic problem, which is to determine the coordinates a seismic source (λ, φ) based on information about coordinates of an observation point (λ_{ps}, φ_{ps}), azimuth to source relative an observation point and distance between an observation point and a seismic event source.

Based on results from processing by proposed method, following results were obtained:

– as a result of using STA/LTA detector (Figure 5 b), first seismic signal arrival falls at 52 seconds from beginning of recording. After first signal arrival, detector is triggered at 61, 52, 108, 292 and 304 seconds.

– as a result of applying APA for first arrival, following values were obtained: G_P=0.99, α_P=205°, γ_P=44°;

– maximum decisive function θ falls at 108 seconds relative that of seismic;

– distance between CSE and OP is determined by seismic wave hodograph.

Figure 6 shows results for calculating CSE location using the proposed method and results of processing measurement data from the Ukrainian State Space Agency seismic observation network. Error in determining for the location a CSE when using our pro-posed method and based on results of processing the SON MCSM data is Δ=37 km.

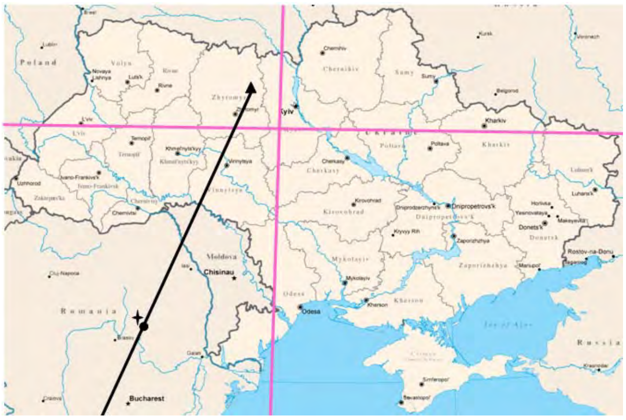


Figure 6 – Results of calculating location of OP using proposed method and based on results from pro-cessing of measurement data of SON MCSM

As a result, a proposed approach to processing the measurement data of a separate three-component seismic station using a polarization analysis apparatus allows detecting seismic signal arrival, identifying the main components of a seismic signal, and estimating location of a seismic event center.

DOI 10.15588/1607-3274-2023-4-16

6 DISCUSSION

Although there is a general global trend in building and developing seismic observation networks, results obtained in this work prove possible to solve almost entire list of seismic monitoring tasks in the regional area by a separate TCSS – detection of a seismic signal, determination of the main components of a seismic record, and location of the seismic event center.

However, this is important in context with the territorial limitations of Ukraine's seismic observation network of the SON MCSM, especially after temporary loss of "Sevastopol" and "Yevpatoriya" observation points due to occupation of the Crimean Peninsula.

Methodological principles for detecting main types of seismic waves are based on polarization properties as well as volume waves and orthogonality oscillations in soil particles. Determination a seismic event center is based on kinetic and dynamic properties of main types a seismic waves and their connection between position a seismic event center relative to OP.

Determined relationships and results obtained on their basis can be considered as basis for approaches on continuous remote monitoring of potential sources of emergency events by seismic means

CONCLUSIONS

Prospects of seismic monitoring, within framework of fulfillment tasks for information support by seismic event center on seismic situation, are development and improvement of methodological and algorithmic means for complex processing seismic measurement data in order to promptly provide conclusions about possibility and fact a dangerous event. In this work, we analyze components of seismic signals from events with foci in the regional zone.

The scientific novelty of the study obtained is to establish a connection between angular characteristics between main seismic signal components for events with foci in the regional zone and position of a seismic event focal point relative to the observation point. Basic principles are formed and an approach is proposed for detecting and identifying seismic signal components in automatic mode, depending on the features and angular characteristics of seismic signal components, and determining seismic event focal point.

The practical orientation of the study is intended for use as basis of developed methodological principles for automatic processing of measurement information at output from a TCSS. Implementation of proposed approaches allows to reduce processing time of seismic data compared to manual processing by operator.

1. In order to automate seismic detection process of hazard factors related to man-made and natural disasters, it is advisable to use separate three-component stations that are part of the seismic observation network of the Main Center for Special Control of Ukraine.

2. At present, energy method is used to detect seismic signals based on the results of observations by three-component stations, which requires a small amount related to computation, which is an essential argument for its use in real-time measurement data processing systems. However, using this approach leads to a limitation in magnitude sensitivity a seismic station.

3. When solving the problems of detecting seismic signal components based on the results processing measurement data from a three-component seismic recording, it is proposed using a polarization apparatus, which will allow taking into account the peculiarities and angular characteristics a seismic signal's volume wave components.

4. Experimental studies using the proposed method for determining the location center of a seismic event showed that the time for establishing an emergency event within Ukraine was reduced from 15 to 3 minutes, while the determination error was 37 km. This indicates that our method is effective in determining location of the event center by a three-component seismic station.

Prospects for further research is to develop algorithms for identifying nature a seismic source based on results from seismic data processing in automatic mode in order develop an information system for detecting hazards of man-made and natural disasters by seismic means.

ACKNOWLEDGEMENTS

We thank the management of our educational institutions for the opportunity to conduct research.

REFERENCES

1. Vasyutinska K., Barbashev S., Smyk S. Analysis of Urbanization Impact on the Dynamics of Emergencies and Risks for Population in Ukraine, *Proceedings of Odessa Polytechnic University*, 2018, Issue 2(55), pp. 88–96. DOI: 10.15276/opu.2.55.2018.10.
2. Dorussen H., Madokoro D. &Tago A. Civil protection: Enhancing resilience through collaboration,*In EU-Japan Security Cooperation: Trends and Prospects*, 2018, 1st ed., pp. 127–144. DOI: https://doi.org/10.4324/9780429456114.
3. Vakaliuk T. A., Pilkevych I. A., Hordiienko Yu. O. et al. Application of Polarization-Time Model Seismic Signal for Remote Monitoring of Potential Sources Emer-gencies by Three-Component Seismic Station, *The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS 2023), Ukraine, Zaporizhzhia, 3 May 2023 : seminar materials*. Aachen, Germany : CEUR Workshop Proceedings, 2023, Vol. 3392, pp. 52–64. DOI: https://doi.org/10.327 82/cmis/3392-5.
4. Code of Civil Protection of Ukraine on October 2, 2012 № 5403-VI, *The Official Bulletin of the Verkhovna Rada of Ukraine (English Version)*, 2013, Nos. 34–35, Article 458. DOI: https://zakon.rada.gov.ua/laws/show/en/5403-17/print.
5. Liashchuk O., Kariahin Y., Kolesnykov L. Andrushchenko Y., Tolchonov I. and Poikhalo A. The Infrasound Network of Ukraine, *EGU General Assembly*, 2020, Online, 4–8 May 2020, EGU2020-3895. DOI: https://doi.org/10.5194/egusphere-egu2020-3895.
6. Aiello G. Techniques and methods of seismic data processing in active volcanic areas: some applications to multichannel seismic profiles (Gulf of Naples, Southern Tyrrhenian sea, Italy), *Journal of Geography and Cartogram*, Online, March 2019. DOI: 10.24294/jgc.v0i0.1101.
7. Gentili S., Michelini A. Automatic picking of P and S phases using a neural tree, *Journal of Seismology*, 2006,

DOI 10.15588/1607-3274-2023-4-16

Vol. 3392, pp. 39–63. DOI: https://doi:10.1007/s10950-006-2296-6.

8. Hordiienko Y., Tiutiunyk V., Chernogor L.,Kalugin V. Features of Creating an Automatically Controlled System of Detecting and Identifying the Seismic Signal Bulk Waves from High Potential Events of Technogenic and Natural Origin, *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, Kharkiv, Ukraine, 2021, pp. 267–272. DOI: 10.1109/PICST54195.2021.9772159.
9. Lewis W., Vigh D., Popovici A. et al. Deep learning prior models from seismic images for full-waveform inversion, *87th Annual International Meeting, USA, Houston, 28 September2017 : expanded abstracts,* SEG Library [Elektronyi resurs], pp. 1512–1517. DOI: https://doi:10.1190/segam2017-17627643.1.
10. Tiutiunyk V. V., Chernogor L. F., Kalugin V. D. et al. Information and technical method of monitoring and forecasting the seismic danger of the local territory of the earth, *Information processing systems*, 2020, 2/161, pp. 99–112. DOI: doi:10.30748/soi.2020.161.12.
11. Chernogor L.F. Possible Generation of Quasi-Periodic Magnetic Precursors of Earthquakes,*Geomagnetism and Aeronomy*, 2019, Vol. 59, No 3, pp. 374–382. DOI: 10.1134/S001679321903006X.
12. Reading A., Mao W., Gubbins D. Polarization filtering for automatic picking of seismic data and improved converted wave detection, *Geophysical Journal International*, 2001, 1(147), pp. 227–234. DOI: https://doi.org/10.1046/j.1365-246X.2001.00501.x.
13. Shafapourtehrany M.,Batur M., Shabani F., Pradhan B., Kalantar B., Özener, H. A Comprehensive Review of Geospatial Technology Applications in Earthquake Preparedness, Emergency Management, and Damage Assessment, *Remote Sensing*, 2023, Vol. 15, No. 7, P. 1939. DOI: https://doi.org/10.3390/rs15071939.
14. Lorido-Botran T., Bhatti M., Impala E. Towards an optimal policy for efficient resource management at the edge, *Journal of Edge Computing* [Online], 2022, 1(1), pp. 43–54. DOI: https://doi.org/10.55056/jec.572.

Received 18.09.2023.
Accepted 17.11.2023.

УДК 550.34.01

ВИЯВЛЕННЯ СЕЙСМІЧНОГО СИГНАЛУ ТРИКОМПОНЕНТНОЮ СЕЙСМІЧНОЮ СТАНЦІЄЮ ТА ВИЗНАЧЕННЯ ОСЕРЕДКУ СЕЙСМІЧНОЇ ПОДІЇ

Вакалюк Т. А. – д-р пед. наук, професор, професор кафедри інженерії програмного забезпечення Держаний університет «Житомирська політехніка», Житомир, Україна.

Пількевич І. А. – д-р техн. наук, професор, професор кафедри комп’ютерних інформаційних технологій Житомирського військового інституту імені С. П. Корольова, Житомир, Україна.

Гордієнко Ю. О. – канд. техн. наук, начальник науково-дослідного відділу наукового центру, Житомирський військовий інститут імені С. П. Корольова, Житомир, Україна.

Лобода В. В. – науковий співробітник наукового центру Житомирського військового інституту імені С. П. Корольова, м. Житомир, Україна.

Салій А. О. – курсовий офіцер курсів перепідготовки та підвищення кваліфікації, Житомирський військовий інститут імені С. П. Корольова, Житомир, Україна.

АНОТАЦІЯ

Актуальність. Робота присвячена розвитку теоретичних засад, спрямованих на автоматизацію процесу визначення місцезнаходження осередку сейсмічної події.

Мета роботи – розробка методу визначення осередку сейсмічної події на основі використання особливостей кутових характеристик складових об’ємних хвиль сейсмічного сигналу, отриманого за допомогою трикомпонентної сейсмічної станції. Запропонований метод призведе до зменшення часу надання попередньої інформації користувачам про факт сейсмічної події та її параметри.

Метод. В основу методу автоматичного визначення осередку сейсмічної події покладено особливості ортогональності кутових характеристик об’ємних хвиль зареєстрованої вибірки трикомпонентного сейсмічного запису з певного напрямку. Реалізація запропонованих підходів дозволяє з відповідною достовірністю зменшити час обробки сейсмічного запису у порівнянні з обробкою у ручному режимі. В роботі розглянутий приклад застосування запропонованого методу (алгоритму) для обробки сейсмічного сигналу у зоні Вранча 27.10.2004 р. з магнітудою М=5,7.

Результати. Запропонований підхід щодо обробки вимірюваних даних окремої сейсмічної трикомпонентної сейсмічної станції з використанням апарату поляризаційного аналізу, дозволяє здійснювати виявлення вступу сейсмічного сигналу, проводити ідентифікацію основних складових сейсмічного сигналу та проводити оцінку місцеположення осередку сейсмічної події. Експериментальні дослідження по використанню запропонованого алгоритму визначення місцезнаходження осередку сейсмічної події показав, що час встановлення надзвичайної події в межах України скорочено в п’ять разів (з 15 до 3 хвилин), а помилка визначення становить 37 км.

Висновки. Сформовані основи та запропонований підхід щодо виявлення сейсмічного сигналу, ідентифікацію його складових та визначення осередку сейсмічної події за результатами обробки трикомпонентного сейсмічного запису є ефективними. Запропонований метод (алгоритм) доцільно застосовувати при автоматизації процесу виявлення сейсмічного сигналу трикомпонентною сейсмічною станцією та визначення осередку сейсмічної події.

КЛЮЧОВІ СЛОВА: землетрус, сейсмічні хвилі, сейсмічний сигнал, виявлення сигналу, трикомпонентна сейсмічна станція, сейсмічний моніторинг, автоматизована система моніторингу, джерело надзвичайних подій.

DOI 10.15588/1607-3274-2023-4-16

ЛІТЕРАТУРА

1. Vasyutinska K. Analysis of Urbanization Impact on the Dynamics of Emergencies and Risks for Population in Ukraine / K. Vasyutinska, S. Barbashev, S. Smyk // Proceedings of Odessa Polytechnic University, 2018. – Issue 2(55). – P. 88–96. DOI: 10.15276/opu.2.55.2018.10.
2. Dorussen H. Civil protection: Enhancing resilience through collaboration / H. Dorussen, D. Madokoro, A. Tago // EU-Japan Security Cooperation: Trends and Prospects, 2018. – 1st ed. – P. 127–144. DOI: https://doi.org/10.4324/9780429456114.
3. Application of Polarization-Time Model Seismic Signal for Remote Monitoring of Potential Sources Emer-gencies by Three-Component Seismic Station / [T. A. Vakaliuk, I. A. Pilkevych, Yu. O. Hordiienko et al.] // The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS 2023), Ukraine, Zaporizhzhia, 3 May 2023. – CEUR Workshop Proceedings, 2023. – Vol. 3392. – P. 52–64. DOI: https://doi.org/10.327 82/cmis/3392-5.
4. Code of Civil Protection of Ukraine on October 2, 2012 № 5403-VI,The Official Bulletin of the Verkhovna Rada of Ukraine (English Version), 2013, Nos. 34–35, Article 458. DOI: https://zakon.rada.gov.ua/laws/show/en/5403-17/print.
5. The Infrasound Network of Ukraine / [O. Liashchuk, Y. Kariahin, L. Kolesnykov et al.] // EGU General Assembly, 2020. – 4–8 May 2020. – EGU2020-389. DOI: https://doi.org/10.5194/egusphere-egu2020-3895.
6. Aiello G. Techniques and methods of seismic data processing in active volcanic areas: some applications to multichannel seismic profiles (Gulf of Naples, Southern Tyrrhenian sea, Italy) / G. Aiello // Journal of Geography and Cartogram, 2019. DOI: 10.24294/jgc.v0i0.1101.
7. Gentili S. Automatic picking of P and S phases using a neural tree / S. Gentili, A. Michelini // Journal of Seismology, 2006/ – Vol. 3392. – P. 39–63. – DOI: https://doi:10.1007/s10950-006-2296-6.
8. Features of Creating an Automatically Controlled System of Detecting and Identifying the Seismic Signal Bulk Waves from High Potential Events of Technogenic and Natural Origin / [Y. Hordiienko, V.Tiutiunyk, L. Chernogor et al.] // IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), Kharkiv, Ukraine, 2021. – P. 267–272. DOI: 10.1109/PICST54195.2021.9772159.
9. Deep learning prior models from seismic images for full-waveform inversion / [W. Lewis, D. Vigh, A. Popovici et al] // 87th Annual International Meeting, USA, Houston, 28 September 2017 : expanded abstracts. – SEG Library, 2021. – P. 1512–1517. DOI: https://doi:10.1190/segam2017-17627643.1.
10. Інформаційно-технічний метод моніторингу та прогнозування рівня сейсмічної небезпеки локальної території Земної кулі / [В. В. Тютюник, Л. Ф. Чорногор, В. Д. Калугін та ін.] // Системи обробки інформації. – 2020. – № 2(161). – С. 99–113. DOI: https://doi.org/10.30748/soi.2020.161.12.
11. Chernogor L. F. Possible Generation of Quasi-Periodic Magnetic Precursors of Earthquakes / L. F. Chernogor //Geomagnetism and Aeronomy, 2019. – Vol. 59, No. 3. – P. 374–382. DOI: 10.1134/S001679321903006X.
12. Reading A. Polarization filtering for automatic picking of seismic data and improved converted wave detection / A.Reading, W. Mao, D. Gubbins //Geophysical Journal International, 2001. – Vol. 1(147). – P. 227–234. DOI: https://doi.org/10.1046/j.1365-246X.2001.00501.x.
13. Comprehensive Review of Geospatial Technology Applications in Earthquake Preparedness, Emergency Management, and Damage Assessment / [M. Shafapourtehrany, M. Batur, F. Shabani et al.] // Remote Sensing, 2023. – Vol. 15, No. 7. – P. 1939. DOI: https://doi.org/10.3390/rs15071939.
14. Lorido-Botran T. Towards an optimal policy for efficient resource management at the edge / T. Lorido-Botran, M. Bhatti, E. Impala // Journal of Edge Computing. – 2022. – Vol. 1(1). – P. 43–54. DOI: https://doi.org/10.55056/jec.572.

DOI 10.15588/1607-3274-2023-4-16

UDC 004.82:343.9

DEVELOPMENT OF APPLIED ONTOLOGY FOR THE ANALYSIS OF DIGITAL CRIMINAL CRIME

Vlasenko L. O. – PhD, Associate Professor of the Department of Software Engineering and Cyber Security, State University of Trade and Economics, Kyiv, Ukraine.

Lutska N. M. – Dr. Sc., Professor of the Department of Automation and Computer Technologies of Control Systems, National University of Food Technologies, Kyiv, Ukraine.

Zaiets N. A. – Dr. Sc., Professor of the Department of Department of Automation and Robotic Systems, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine.

Savchenko T. V. – PhD, Associate Professor of the Department of Software Engineering and Cyber Security, State University of Trade and Economics, Kyiv, Ukraine.

Rudenskiy A. A. – Senior lecturer of the Department of Automation and Robotic Systems, National University of Life and Environmental Sciences of Ukraine, Kyiv, Ukraine.

ABSTRACT

Context. A feature of the modern digital world is that crime is often committed thanks to the latest computer technologies, and the work of law enforcement agencies faces a number of complex challenges in the digital environment. The development of information technology and Internet communications creates new opportunities for criminals who use digital traces and evidence to commit crimes, which complicates the process of identifying and tracking them.

Objective. Development of an applied ontology for a system for analyzing a digital criminal offense, which will effectively analyze, process and interpret a large amount of digital data. It will help to cope with the complex task of processing digital data, and will also help automate the process of discovering new knowledge.

Methods. To build an ontological model as a means of reflecting knowledge about digital crime, information was collected on existing international and domestic classifications. The needs and requirements that must be satisfied by the developed ontology were also analyzed. The creation of an ontological model that reflects the basic concepts, relationships in the field of digital criminal offense was carried out in accordance with a recognized ontological analysis of a specialized subject area.

Results. An applied ontology contains the definition of entities, properties, classes, subclasses, etc., as well as the creation of semantic relationships between them. At the center of the semantics is the Digital Crime class, the problem area of which is complemented by the interrelated classes Intruder, Digital evidence, Types of Crime, and Criminal liability. Such characteristics as motive, type of crime, method of commission, classification signs of digital traces and types of crime, as well as other individual information were assigned to the attributes of the corresponding classes. An ontological model was implemented in OWL using the Protégé software tool. A feature of the implementation of the applied ontology was the creation of SWRL rules for automatically filling in additional links between a class instance. Manual and automatic verification of the ontology showed the integrity, consistency, a high degree of correctness and adequacy of the model. The bugs found were usually related to technical aspects and semantic inconsistencies, which were corrected during further development iterations.

Conclusions. The research confirmed the effectiveness of the developed applied ontology for the analysis of digital criminality, providing more accurate and faster results compared to traditional approaches.

KEYWORDS: ontology, digital forensic, digital crime, digital evidence.

ABBREVIATIONS

CCU is a Criminal Code of Ukraine;
CS is a Computer System;
DSS is Decision Support Subsystem.

NOMENCLATURE

O^{ao} is an extended Applied ontology;
C is a set of domain classes;
A is a set of attributes that describe class objects;
R is a set of relationships between concepts;
T is a set of standard attribute value types;
F is a set of restrictions for the values of concepts and attribute relations (rules and axioms);
D is a set of instances of classes;
R^a is the associative relation (Object Property);
R^h is the heredity relation “SubDataPropertyOf”;
R^{ca} is the relation class-data (Data Property);
R^{cd} is the relation class-individual “has individual”.

INTRODUCTION

With the advent of computer, microprocessor technology, the Internet, etc. A new sphere and new tools for committing crimes have appeared in the world – digital crimes. Modern digital crimes are so diverse that they are committed against both private individuals and government agencies. Their negative consequences for the lives of people, the functioning of organizations, governments are often quite strong. Until recently, such offenses were not regulated by law and therefore active work was carried out in this direction. Now most states of the world have in their criminal code a section dedicated to liability in the event of a certain type of digital offense. But the legal and policy aspects of responding to digital crime cannot be permanent and must constantly change and improve.

Experts note that in the past few years there has been an increase in the number of digital offenses. Digital criminals are learning, using new approaches and tech-

DOI 10.15588/1607-3274-2023-4-17

nologies, and therefore the methods of identifying and disclosing them are becoming more complicated. According to the official statistics of the Office of the Prosecutor General of Ukraine [1] (Fig. 1), the number of digital crimes in Ukraine in 2022 increased by almost 64% compared to 2019. As for the current 2023, the number of crimes committed in the first half of the year exceeds the number for the entire 2019. Therefore, it can be assumed that in 2023 the total number will be the highest over the past 5 years. This can be explained by several additional factors.

First of all, there was a significant intensification of cybercrimes in the framework of the Russian-Ukrainian war, such as unauthorized interference in the operation of information, electronic communication, information and communication systems, electronic communication networks, etc. The emergence, development and active implementation of artificial intelligence in various areas of human activity and life can become the next challenge for protection against digital crimes. That is why it is necessary to stimulate the development of modern tools that could help and facilitate the work of cyber police, information security specialists [2–4].

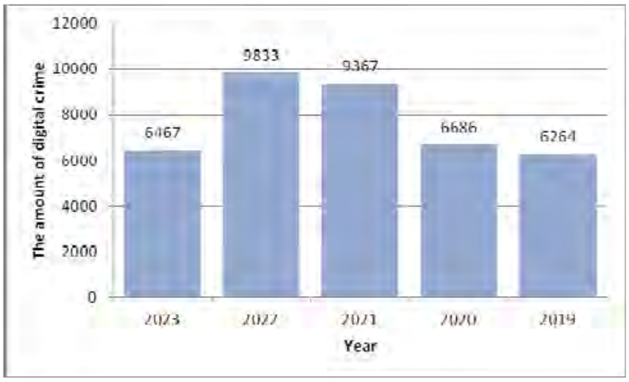


Figure 1 – Statistics of digital crimes in Ukraine for the period 2019–2023

The use of a generally acceptable classification scheme in this area would contribute to the improvement of legislation, the development of various quick countermeasures, deepening cooperation, etc. A number of problems in creating a subsystem for supporting the analysis of a digital criminal offense is due to the fact that the existing classifications of types and kinds of digital crimes, motives and threats to commit them, as a rule, are incomplete, fragmentary, use different terminology to define the same object, are developed to solve specific task. This makes them incompatible with each other.

Automating the process of digital crime analysis is very important, but complicated by working with unstructured forensic data, as it comes from different sources. Another problem is the lack of semantics for the concept of investigating a digital crime and determining punishment for it.

The authors propose a variant of the ontology, which includes, in addition to the type and kind of crime, accounting for digital traces, methods for committing a crime, and also determining the type of punishment in accordance with the current legislation of Ukraine. This work may be of interest to those involved in digital forensics, cybercrime, information security.

The object of study is the process of combining knowledge and data of digital forensics, cyber defense, Ukrainian legislation, aimed at identifying the commission of a digital crime and the type of criminal punishment for it.

The subject of study is an applied domain ontology for a digital crime analysis system.

The purpose of the work is to build an applied ontology to combine heterogeneous digital crime data obtained from different sources and systems, automate data processing, analyze large data sets and develop decision options for the decision support subsystem.

1 PROBLEM STATEMENT

The purpose of a decision support system (DSS) for digital crime analysis based on applied ontology is the ability to provide information to investigators, cyberpoliticians and other users about possible digital crimes and criminal liability options for committing them. This can significantly improve the organization of work on the analysis of criminal digital crime, reduce the time to make more informed decisions based on the array of available data. The developed subsystem can be used as an auxiliary tool in the work of professional workers and as a tool to increase the awareness of citizens.

The applied ontology is described by a tuple (1):

$$O^{ao} = < C, A, R, T, F, D >. \quad (1)$$

The ontology (1) for the analysis of criminal digital crime should include only those elements that will be used.

In particular, the set of relations (1) consists of relations: associative, subordinate, “is – a”, “class – data”, and is represented by a tuple (2):

$$R = < R^a, R^h, R^{ca}, R^{cd} >. \quad (2)$$

The applied ontology, which is part of the DSS, is used to solve the following tasks:

– structured representation of the subject area;

– introduction of a clear classification of terminology specific to the subject area;

– creation of a knowledge base for decision support for the industry;

– improvement of processing, search and filtering of heterogeneous information;

– identification of additional (implicit) links between concepts based on data semantics;

– analysis of the trend of changes taking place in the field of digital crime.

2 REVIEW OF THE LITERATURE

The creation of ontologies for digital crimes is overwhelmingly reduced to cybercrime ontologies. Ontologies of criminal offenses are also often created, where cyber-

DOI 10.15588/1607-3274-2023-4-17

crime is one of the subtypes. In particular, the classification of cybercrime may have a semi-formal approach to the development of a taxonomy of cybercrime [2]. The use of design science (DS) as a paradigm for solving organizational problems makes it possible to take into account the emergence and evaluation of innovative artifacts in the ontology of cybercrime classification [5].

The active development of artificial intelligence in recent years has brought the creation of ontological systems of digital forensics to a new technical and philosophical level. This type of ontologies [6] makes it possible to trace the entire chain of a digital crime, identify anomalies in the investigation process, and automate the processing of digital evidence traces [7]. The semantic ontology of digital evidence allows an investigator to quickly discover what artifacts may be available on a device before the time-consuming process of investigating digital devices begins, preventing the creation of data that has no practical value for the investigation.

A separate capacitive process in digital forensics is the analysis of the results of a forensic medical examination [8]. The use of semantic web technologies, in particular, ontologies, can greatly facilitate the work with them for the investigator when analyzing digital evidence [9] using RDF [10]. Modern ontologies support the specification of a web service. Creating a convenient and friendly graphical interface allows the investigator to receive a forensic examination report online based on requests, ready to submit it to the court [11].

Separately, it should be noted the development of ontologies in order to analyze the content received from Android smartphones [10]. The ontology may provide for the organization of evidence retrieved from mobile devices. Thus, a network of interconnected material is formed, in which it improves the process of data analysis and search for relevant evidence for the investigation [12].

In [13], an ontology of the subject area of cyberforensics for criminal investigation was built in accordance with the categories of cybercrime, laws, evidence, and information criminals. This ontology does not contain an application layer.

It should be noted that in the Ukrainian conceptual environment digital crime is a broader concept and cybercrime is its subspecies. While in the vast majority of foreign sources they are used by the authors as identical concepts. Also, most of the developed ontologies are of a general nature and do not take into account the specifics of the legislation of a particular country.

3 MATERIALS AND METHODS

According to Ukrainian legislation, criminal liability for digital crimes depends on several factors. Important among the factors are the type of digital offense committed, the extent of damage and harm caused, or the specific type of crime, motivation, as well as the personal characteristics of the person who committed this crime (for example, the circumstances of the commission, preliminary criminal actions and cooperation with law enforcement, special features, etc.). Thus, it is possible to identify a set of basic concepts that are connected by different types of connections (Fig. 2). Five concepts of which are nonempty classes: Digital Crime, Types of Crime, Criminal liability, Digital evidence, Intruder.

The developed applied ontology of the decision support subsystem for the analysis of a digital criminal offense is based on the processing and generalization of the analysis of research works of Ukrainian and foreign scientists [14–20].

Any crime starts with motive and intruder. Digital crime is no exception. The most common **Motives**, as a property of a certain crime, include the following: enrichment, terrorism, espionage, military and economic espionage, targeting national information infrastructure, revenge, hate (national origin, gender, race), greed. **Intruders**, in turn, were divided according to the type of crime they commit into: spy, terrorist, corporate raiders, professional criminals, hacker and others.

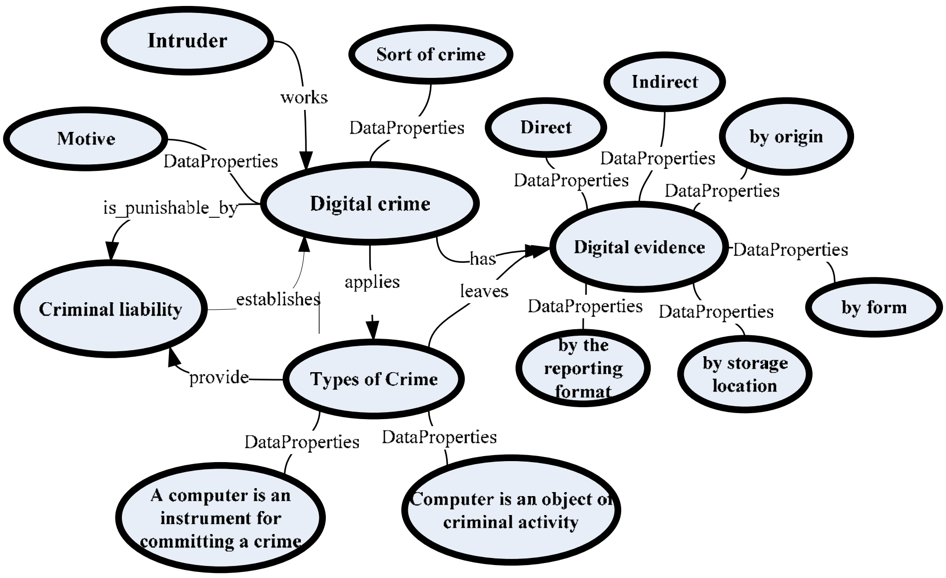


Figure 2 – Conceptual Model of Applied Ontology for Digital Crime Analysis System

DOI 10.15588/1607-3274-2023-4-17

Each individual digital crime belongs to a specific **Sort of Crimes**. There are various Sort of Crimes: Against Individuals – the most common are spamming and related threats, e-mail spoofing, cyber defamation, cyber stalking, cyber harassments, libel and false information, phishing; Against Property – internet time theft, credit card frauds, intellectual property crimes, identity theft, misuse of devices; Against Organizations – different types of attacks, computer-related offences, unauthorized accessing of computer, economic espionage, denial of service, Industrial espionage, computer contamination, e-mail bombing, copyright-related offences; Against Society – pornography, illegal gambling and online games, glorification of violence, hate speech, forgery, religious offence, racism, Web Jacking; Against Government – cyber terrorism, hacking, military espionage, accessing confidential information, cyber warfare (crimes are considered an attack on that national sovereignty).

In the world and in Ukraine, digital crimes have a certain typification. **Types of Crime** were divided into two groups for convenience: A computer is an instrument for committing a crime – Content violations, Unauthorized modification of data, software, Improper use of Communications; Computer is an object of criminal activity – unauthorised access, malicious code, interruption of services, Theft or misuse of services.

Each crime has **Digital evidence**, which are carriers of a certain set of information and provide an evidence base in identifying the offender, proving his guilt and issuing an appropriate sentence. The peculiarities of digital evidences are their heterogeneity, for the frequent short period of their existence, they can be forged or destroyed. Also, digital evidences can be located in various hardware and software, in particular, on a computer hard drive, flash drive, local network devices, websites, social networks, emails, etc. In order to take into account, the existing diversity as a basis for the Digital evidence class [14], the chosen classification was developed in (Fig. 2).

In Ukraine, **Criminal liability** for committing a digital crime is regulated by the following legal documents: the Constitution of Ukraine, the Criminal Code of Ukraine (CCU) [21], the laws of Ukraine: "On Information", "On the Protection of Information in Information and Telecommunication Systems", "On the Basics National Security" "On the Basic Principles of Ensuring Cybersecurity of Ukraine", "On Amendments to the Criminal Code of Ukraine to Improve the Efficiency of Combating Cybercrime in Martial Law". The article presents fragments of this class, containing separate articles corresponding to the crimes considered in the examples as a possible form of punishment (Fig. 3).

Thus, we get a set of classes:

$$C = \{\text{Digital\_Crime, Intruder, Digital\_evidence, Types\_of\_Crime, Criminal\_liability}\} \quad (3)$$

and a set of connections between the individuals of these classes:

$$R^a = \{\text{has, works, applies, is\_punishable\_by, establishes, leaves, provides}\} \quad (4)$$

Figure 3 shows an extended fragment of the conceptual model, which includes a part that describes the classification of crimes based on their motive with reference to the punishment provided for by the current legislation of Ukraine. The model also shows relationships between concepts such as Data Properties Rcd, Object Properties Ra, and class-individual Rcd, as well as the values of these attributes (set T).

Taking into account the peculiarities of the subject area, when creating the ontology, it was taken into account that some of the selected classes should be defined and the filling process is provided by a group of specialists. It includes developers, industry experts, future users (selectively if necessary). In the future, users cannot independently make changes and additions to the classes: Types of Crime and Criminal liability. For instances of the Digital Crime, Intruder and Evidence classes, individual positions are filled in by users. in the Table. 1 shows the corresponding characteristics of the classes.

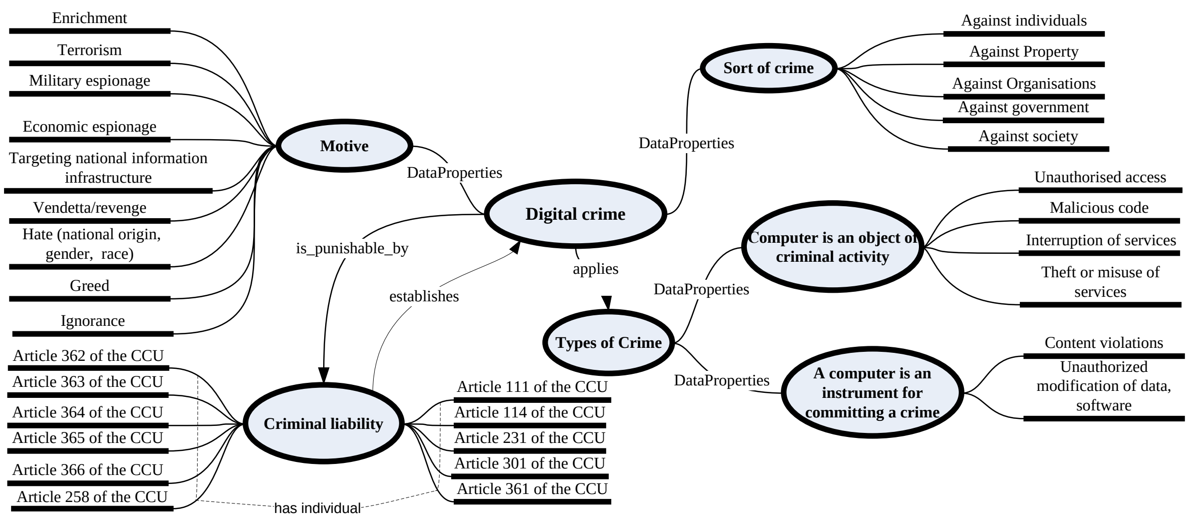


Figure 3 – Extended fragment of the conceptual model with attributes for the Digital Crime and Type of Crime classes and instances of the Criminal liability class

DOI 10.15588/1607-3274-2023-4-

Table 1 – Class characteristics

Class	Individuals	User changes	Data Property	Object Property
Digital Crime	+	+	+	+
Criminal liability	+	–	–	–
Evidence	+	+	+	+
Intruder	+	+	+	+
Types of Crime	+	–	+	+

In order to conduct a deeper and more automated data analysis, a formalized expression of logical conditions and relationships based on SWRL rules was used. Their advantage is the ability to define complex logical connections and relationships between objects, finding new knowledge based on existing ones and establishing new relationships between elements, improving the quality of search in the ontology. Also, SWRL rules can be used to passively check the consistency of data in the ontology, identify contradictions and inconsistencies.

The developed rules are divided into the corresponding categories of Types of Crime for structure. The ontology contains 206 rules developed by SWRL. Table 2 shows some of them. The rules presented relate to the section on cyberterrorism. Thus, new links are formed between user and non-user class instances: applies, is_punishable_by, establishes, leaves.

The developed system is designed in such a way that, if necessary, it can be expanded and supplemented with additional concepts, attributes, relationships, etc.

Table 2 – Example of rules developed for applied ontology (*F*)

Rule Number	SWRL Rule	Rule category
KT1	Evidence(?x) ^byForm(?x,?y) –> leaves(Ciberterorrism,?x)	Ciberterorrism
KT2	Evidence(?x) ^ UnauthorizedAccessDevices(?x,?y) –> leaves(Ciberterorrism,?x)	Ciberterorrism
KT3	Evidence(?x) ^ byOrigin(?x, "InformationGeneratedByCSBasedOnDataInput") –> leaves(Ciberterorrism,?x)	Ciberterorrism
KT4	Evidence(?x) ^ ByTheReportingFormat(?x, "HumanReadableInformation") –> leaves(Ciberterorrism,?x)	Ciberterorrism
KT5	Evidence(?x) ^ ByStorageLocation (?x, "InformationStoredInCS") –> leaves(Ciberterorrism,?x)	Ciberterorrism
KT6	Digital_crime(?x) ^Motive(?x, "Terrorism") ^ Sort_of_Crime(?x, "AgainstSociety") ^ has(?x,?y) ^ leaves (Ciberterorrism,?y) -> applies(?x, Ciberterorrism)	Ciberterorrism
KT7	Digital_crime(?x) ^Motive(?x, "Terrorism") ^ Sort_of_Crime(?x, "AgainstGoverment") ^ has(?x,?y) ^ leaves(Ciberterorrism,?y) -> applies(?x, Ciberterorrism)	Ciberterorrism
KT8	Digital_crime(?x) ^Motive(?x, "Enrichment") ^ Sort_of_Crime(?x, "AgainstSociety") ^ has(?x,?y) ^ leaves (Ciberterorrism,?y) -> applies(?x, Ciberterorrism)	Ciberterorrism
KT9	Digital_crime(?x) ^Motive(?x, "Enrichment") ^ Sort_of_Crime(?x, "AgainstGoverment") ^ has(?x,?y) ^ leaves (Ciberterorrism,?y) -> applies(?x, Ciberterorrism)	Ciberterorrism
KT10	Intruder(?x) ^ byType(?x, "Terrorist") ^ works(?x,?y) ^ applies(?y, Ciberterorrism) -> is_punishable_by(?y, Article_258_CCU)	Ciberterorrism
KT11	Intruder(?x) ^ byType(?x, "Terrorist") ^ works(?x,?y) ^ applies(?y, Ciberterorrism) -> is_punishable_by(?y Article_361_CCU)	Ciberterorrism
KT12	Intruder(?x) ^ byType(?x, "Hackers") ^ works(?x,?y) ^ applies(?y, Ciberterorrism) -> is_punishable_by(?y, Article_258_CCU)	Ciberterorrism
KT13	Intruder(?x) ^ byType(?x, "Hackers") ^ works(?x,?y) ^ applies(?y, Ciberterorrism) -> is_punishable_by(?y, Article_361_CCU)	Ciberterorrism

4 EXPERIMENTS

To implement the developed ontological model, the semantic language OWL and the free Protege software were chosen. These tools were favored due to OWL's powerful expressiveness, its versatility and integration with other tools (such as search and visualization), its large developer community, and ease of development and deployment.

Figure 4 shows the VOWL-graph of the developed ontology. You can see the set of standard types of attribute values *T* from (1), as well as the Data Property hierarchy introduced for convenience – the set R^h. Figure 5 shows the set of Object Property Ra associations and an example of the constraints imposed on the is_punishable_by relation.

Figure 6, a shows a fragment of individuals of the Type of Crime class and their attributes. Note that these classes are filled with knowledge engineers. In Figure 6, b is presented respectively in the Data Property.

Figure 7 shows an example of the SWRL rules that were coded in the SWRL Protégé5.0 tab and the results and their explanations obtained after putting these rules into the ontology. In particular, in the Edit window that opens, you can see a rule that forms a new is_punishable_by relationship between objects of the Digital Crime and Criminal liability class through works applies custom relationships. Note that the last link is also formed based on the SWRL rules.

DOI 10.15588/1607-3274-2023-4-

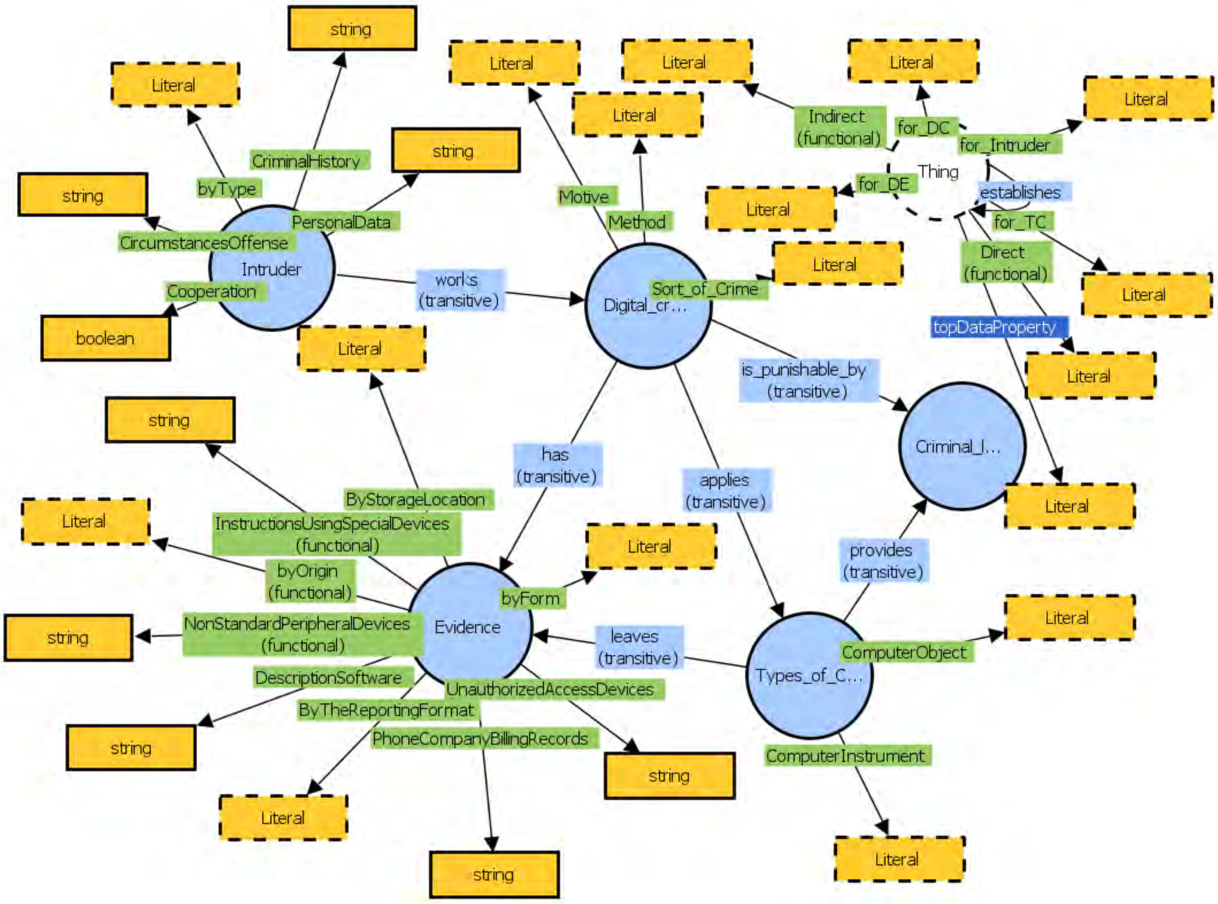


Figure 4 – VOWL-graph of the developed ontology

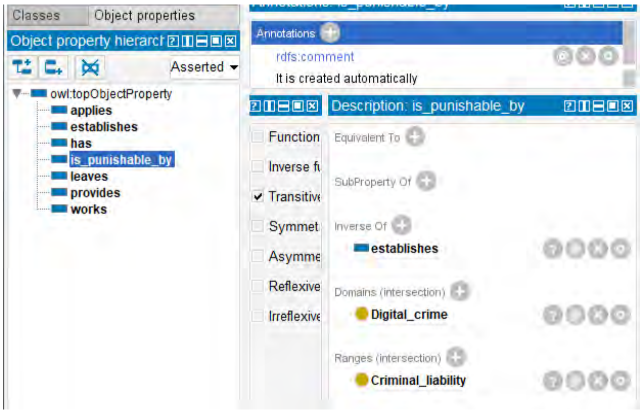


Figure 5 – Object Property

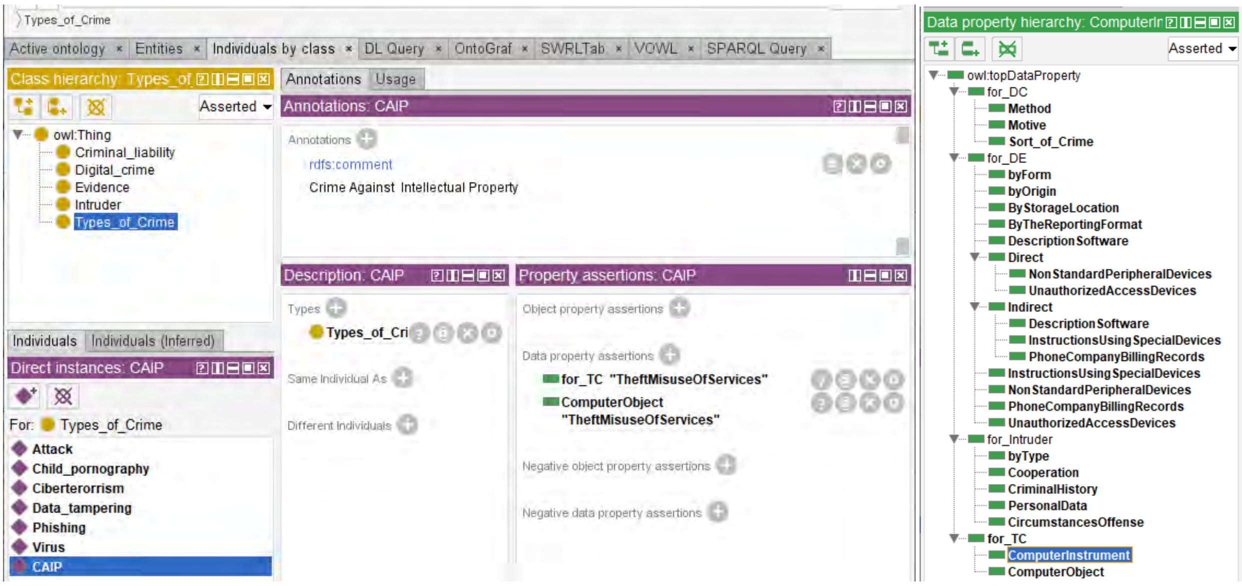
5 RESULTS

To check the correctness of the developed ontology, Reasoner machines were used. The purpose of Reasoner is to classify, reproduce the class hierarchy embedded in the ontological model, work with instances, determine their belonging to classes in accordance with logical rules and axioms, check consistency, and show inconsistencies. Reasoner is based on the concept of Open World Reasoning.

Protégé 5.5.0 has the ability to work with three Reasoners: Fact++, HermiT 1.4.3.456 and Pellet. The main difference between them is the algorithms for building links, data formats and the ontological modeling language that Reasoner supports.

To check the correctness of the ontology, a series of experiments were carried out. Each series of experiments corresponded to a certain type of digital crime. Individuals were created for the corresponding classes, for each of them a unique set of attributes was set.

DOI 10.15588/1607-3274-2023-4-17



a) b)

Figure 6 – Hierarchies of classes and individuals of the class Type of Crime (a) and Data Property (b) in Protégé

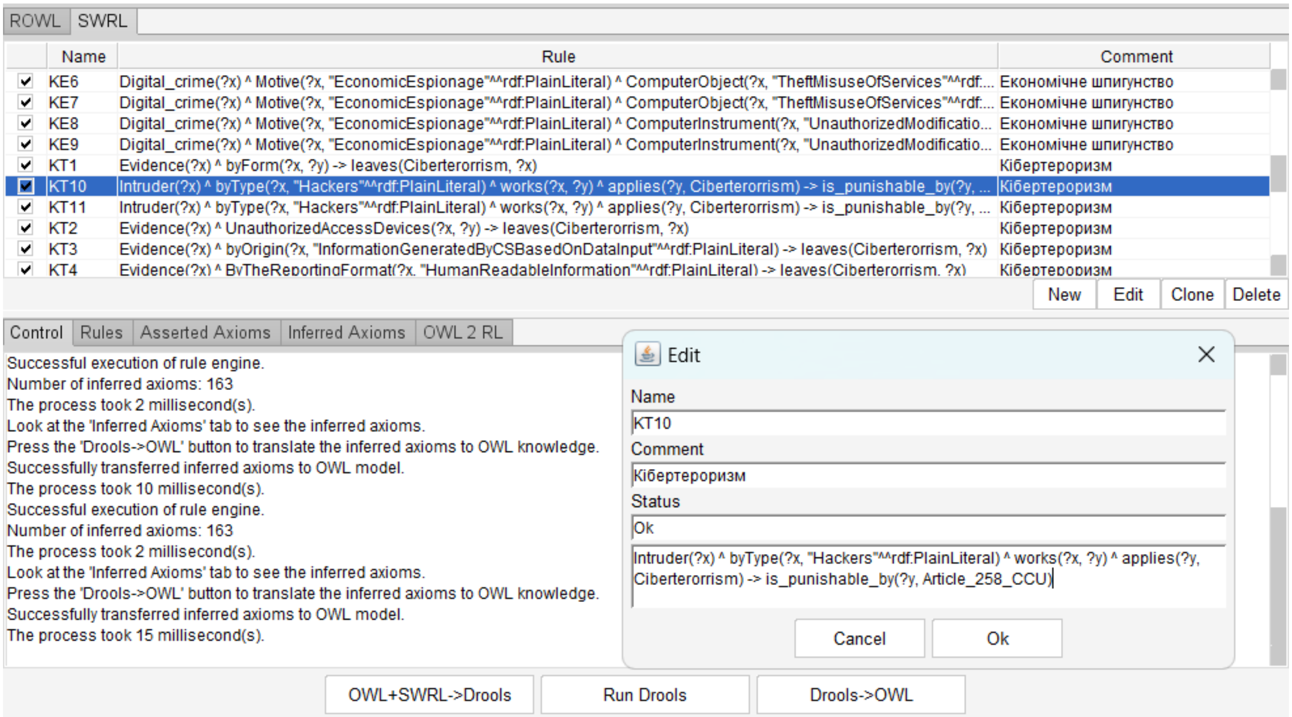


Figure 7 – Example SWRL rules encoded into an ontology

Figure 8 shows the result of Reasoner's work on a series of experiments on the digital crime of cyberterrorism for an individual IndDC97 of the Digital Crime class. For this digital crime, the corresponding instances of the Data Properties set were set, digital traces were created – individuals DE97, DE97a. The digital crime IndDC97 is related to the Types of Crime of the Ciberterorrism individual through logical rules (Table 2) and the relation applies. Criminal liability is also defined in accordance with Articles 361 and 258 of the Criminal Code of Ukraine by the connection is punishable by.

Figure 9 illustrates the result of the Reasoner check for digital crime IndDC97. For the Ciberterorrism individual of the Types of Crime class, its digital traces DE97, DE97a were matched by logical rules and the relation leaves.

The digital crime ontograph IndDC97 demonstrates connections between class individuals (Fig. 10), which are formed automatically based on the introduced SWRL rules (Table 2). The new connections can be seen on the left in Figure 10.

DOI 10.15588/1607-3274-2023-4-17

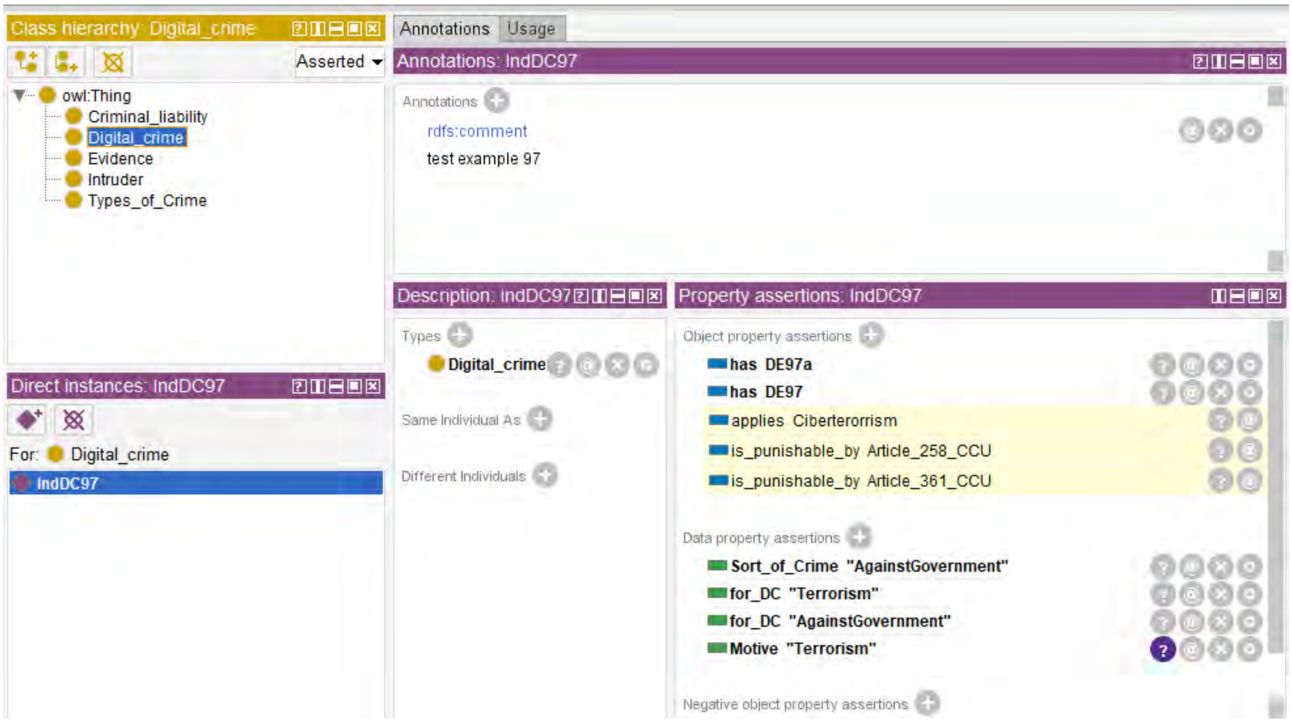


Figure 8 – Reasoner example for the Ciberterorrism individual of the Types of Crime class

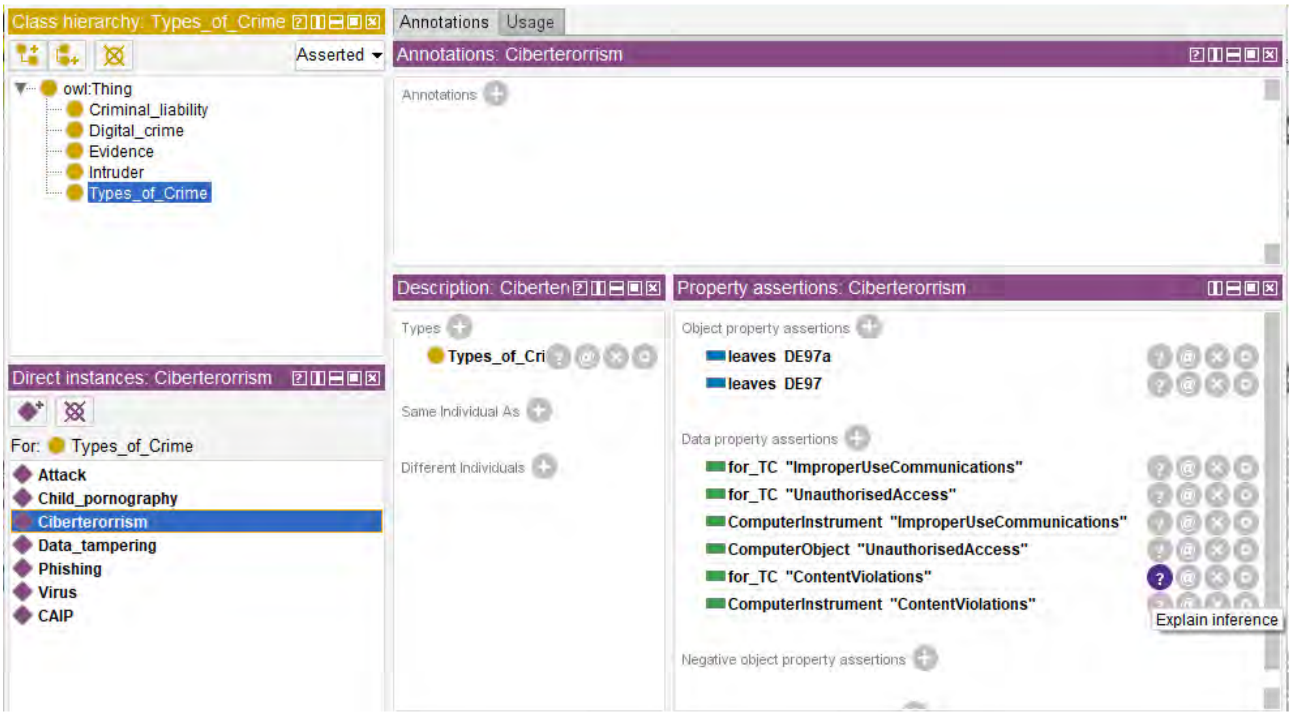


Figure 9 – Reasoner example for the Ciberterorrism individual of the Types of Crime class

6 DISCUSSION

Validation of an applied ontology depends on specific needs and requirements. For small ontologies, manual verification may be sufficient. For large ontologies, semi-automated or automatic approaches are usually produced – in particular, testing. The developed ontology was checked manually and automatically.

98 test instances of the Digital Crime class and the corresponding objects of the Intruder and Evidence classes were developed. Syntax errors were found during a manual check. No semantic errors were found.

The developed ontology was tested using the online resource Ontology testing Themis [22]. The test results (Table 3) showed that the developed ontology was successfully tested, is adequate, accurate, valid and does not contain contradictions.

DOI 10.15588/1607-3274-2023-4-17

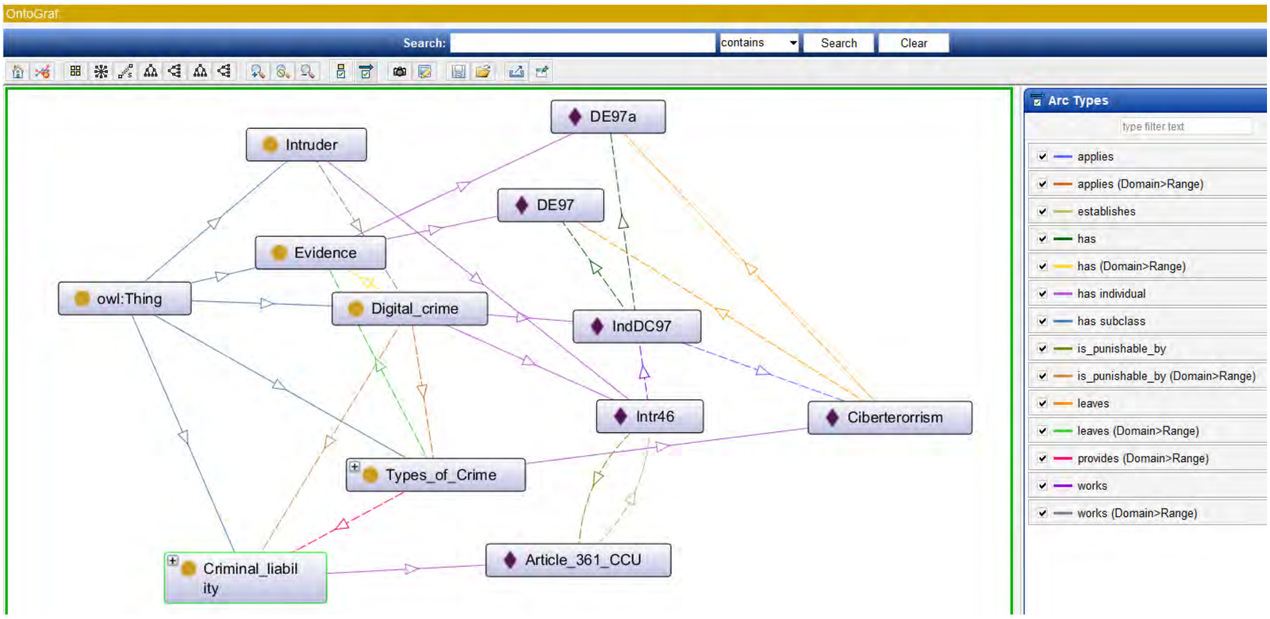


Figure 10 – Ontograph for Digital Crime IndDC97

Table 3 – Test results by Ontology testing Themis

Metric	Result
Percentage of covered requirement	97.83%
Percentage of requirements with terms that are undefined in the ontology	2.17%
Percentage of requirements that lead to conflict	0
Percentage of tested terms	83.06%

The practical value of an ontology lies in its use as a basis for a decision support system. Due to the use of the OWL language, such an application can be WEB-oriented [23]. At the same time, in addition to the semantic database and knowledge, the user can form different queries to the ontology. For example, in Fig. Figure 11 shows the DL-query of the following template: "Find instances of the Evidence class that have feedback from has with an instance of the Digital crime class, which in turn has feedback from works with an instance of the Intruder class, and also has a corresponding byForm attribute value".

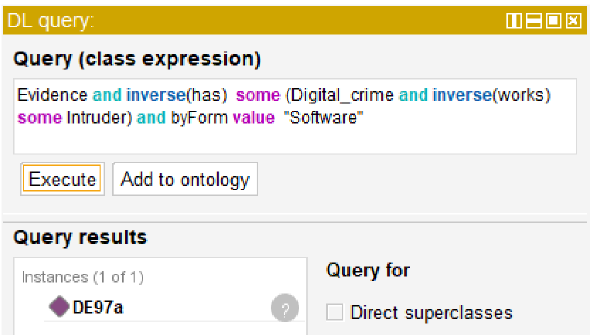


Figure 11 – DL-query example

The proposed ontological model does not accompany the process of solving a crime, since such models already exist in sufficient numbers. It aims to collect and analyze already solved crimes and obtain information on automatically classified crimes, as well as to analyze their common features, patterns and trends.

CONCLUSIONS

The proposed model can be used to improve the analysis, organization and interpretation of data related to digital criminal activity. The developed ontology helps to understand and classify crimes, make connections between different aspects of digital crimes and promote effective interaction between users.

The scientific novelty. The paper presents a modern approach to modeling digital crime. The developed ontology allows obtaining new knowledge, approaches or tools in the field of digital crime analysis, improves cooperation between various organizations, exchanges information and jointly analyzes data.

The practical significance. The developed ontology can be used for intelligent data analysis of digital crimes, collecting this data, classifying, grouping, combining digital traces, types of crimes and criminals, as well as determining the possible punishment. The ontology does not contain closed and confidential information, therefore it is publicly available. The use of the OWL language makes it compatible and integrated with the vast majority of modern applications. It can also be used in the educational process in the relevant specialties, in particular 125 "Cybersecurity and Information Protection".

Prospects for further research. It is planned to expand the developed ontology with additional classes and attributes, in particular, classes that will contain data about the injured party and the result of the harm caused to them.

REFERENCES

1. Pro zareyestrovani kryminalni pravoporushennya ta rezultaty yikh dosudovoho rozsliduvannya (2023) In: ofis holovnoho prokurora. https://gp.gov.ua/ua/posts/pro-zareyestrovani-kriminalni-pravoporushennya-ta-rezultati-yih-dosudovogo-rozsliduvannya-2. Accessed 20 Aug 2023
2. Barn R., Barn B. An ontological representation of a taxonomy for cybercrime, *Twenty-Fourth European Conference on Information Systems (ECIS)*. İstanbul, Turkey, 2016. Access mode: https://core.ac.uk/download/pdf/42490758.pdf

DOI 10.15588/1607-3274-2023-4-17

3. Vlasenko L., Lutska N., Zaiets N., Korobiichuk I., Hrybkov S. Core Ontology for Describing Production Equipment According to Intelligent Production, *Applied System Innovation*, 2022, Vol. 5, Issue 5, pp. 98–111. DOI: 10.3390/asi5050098
4. Vlasenko L. O., Lutska N. M., Zaiets N. A., Shyshak A. V., Savchuk O. V. Domain ontology development for condition monitoring system of industrial control equipment and devices, *Radio Electronics, Computer Science, Control*, 2022, Vol. 1, pp. 157–166. DOI: 10.15588/1607-3274-2022-1-16
5. Donalds C., Osei-Bryson K. M. Toward a cybercrime classification ontology: A knowledge-based approach, *Computers in Human Behavior*, 2019, Vol. 92, pp. 403–418. DOI: 10.1016/j.chb.2018.11.039
6. Sikos L. F. AI in digital forensics: Ontology engineering for cybercrime investigations. *Wiley Interdisciplinary Reviews: Forensic Science*, 2020, Vol. 3, Issue 3, pp. 1–11. DOI: 10.1002/wfs2.1394
7. Karanasios S., Thakker D., Lau L., Allen D., Dimitrova V., Norman A. Making sense of digital traces: An activity theory driven ontological approach, *Journal of the American Society for Information Science and Technology*, 2013, Vol. 64, Issue 12, pp. 2452–2467. DOI: 10.1002/asi.22935
8. Brady O., Overill R., Keppens J. Addressing the increasing volume and variety of digital evidence using an ontology. *2014 IEEE joint intelligence and security informatics conference*. IEEE, 2014. pp. 176–183. DOI: 10.1109/JISIC.2014.34
9. Michel M. C., Carvalho M., Crawford H., Esterline A. C. Cyber identity: Salient trait ontology and computational framework to aid in solving cybercrime, *2018 17th IEEE International Conference on Trust, Security and Privacy In Computing and Communications/12th IEEE International Conference on Big Data Science And Engineering (TrustCom/BigDataSE)*, IEEE, 2018, pp. 1242–1249. DOI: 10.1109/TrustCom/BigDataSE.2018.00171
10. Alzaabi M., Jones A., Martin T. A. An Ontology-Based Forensic Analysis Tool. *Proceedings of the 2013 Annual ADFSL Conference on Digital Forensics, Security and Law*, 2013, pp. 121–136. Access mode: https://commons.erau.edu/adfsl/2013/wednesday/5
11. Akremi A., Sriti M.-F., Sallay H., Rouached M., Ontology-based smart sound digital forensics analysis for web services, *International Journal of Web Services Research*, 2019, Vol. 16, № 1, 70–92. DOI: 10.4018/IJWSR.2019010104
12. Alzaabi M., Ontology-Based Forensic Analysis of Mobile Devices, *Proceedings of the 20th IEEE International Conference on Electronics, Circuits and Systems*, IEEE, 2013, pp. 64–68. DOI: 10.1109/ICECS.2013.6815346
13. Park H., Cho S., Kwon H. Cyber Forensics Ontology for Cyber Criminal Investigation. *Forensics in Telecommunications, Information and Multimedia, LNICST*, 2009, Vol. 8, pp. 160–165. DOI: 10.1007/978-3-642-02312-5_18
14. Naidyon Ya. Ponyattya ta klasyfikatsiya virtualnykh slidiv kiberzlochyniv, *Pidpryyemnytstvo, hospodarstvo i pravo*, 2019, 5, pp. 304–307. DOI: 10.32849/2663-5313/2019.5.56
15. Lysenko S. M., et al. Rezilientnist kompiuternykh system v umovakh kiberzahroz: Ontolohiia ta taksonomiia, Radioelektronni i kompiuterni systemy, 2020, No. 1, pp. 17–28. DOI: 10.32620/reks.2020.1.02
16. Maskun M., Achmad A., Naswar N., Assidiq H., Syafira A., Napang M., Hendrapati M. Qualifying Cyber Crime as a Crime of Aggression in International Law, *Cybercrime under International Law*, 2020, – Vol. 13, № 2, pp. 397–418. DOI: 10.14330/jeail.2020.13.2.08
17. Uma M., Padmavathi G. A survey on various cyber attacks and their classification, *Int. J. Netw. Secur.*, 2013, Vol. 15, №5, pp. 390–396. DOI:10.6633/IJNS.201309.15(5).09
18. Chandra A., Snowe M. J. A taxonomy of cybercrime: Theory and design, *International Journal of Accounting Information Systems*, 2020, Vol. 38, pp. 100467. DOI: 10.1016/j.accinf.2020.100467
19. Harmandeep S. B., Kumar G. Cybercrimes: A proposed taxonomy and challenges, *Journal of Computer Networks and Communications*, 2018, Vol. 11. DOI: 10.1155/2018/1798659
20. Goni O. Cyber Crime and Its Classification, *Int. J. of Electronics Engineering and Applications*, 2022, Vol. 10, № 1, pp. 1–17, DOI: 10.30696/IJEEA.X.I.2021.01-17In
21. Kryminalnyi kodeks Ukrainy (2001) In: Verkhovna Rada Ukrainy. Zakonodavstvo Ukrainy. https://zakon.rada.gov.ua/laws/show/2341-14. Accessed 20 Aug 2023
22. Themis. Ontology testing. https://themis.linkeddata.es/index.html. Accessed 20 Aug 2023
23. Lutska N., Vlasenko L., Ladanyuk A., Zaiets N., Korobiichuk I. Ontological Support System of Managerial Decision-Making of Production Tasks for a Food Enterprise, *Machinery & Energetics*, 2022, Vol. 13, № 3, pp. 53–61. DOI: 10.31548/machenergy.13(3).2022.53-61

Received 30.08.2023.
Accepted 15.11.2023.

УДК 004.82:343.9

РОЗРОБКА ПРИКЛАДНОЇ ОНТОЛОГІЇ ДЛЯ АНАЛІЗУ ЦИФРОВОГО КРИМІНАЛЬНОГО ЗЛОЧИНУ

Власенко Л. О. – канд. техн. наук, доцент, доцент кафедри інженерії програмного забезпечення та кібербезпеки, Державний торговельно-економічний університет, Київ, Україна.

Луцька Н. М. – д-р техн. наук, професор, професор кафедри автоматизації та комп'ютерних технологій систем управління, Національний університет харчових технологій, Київ, Україна.

Заєць Н. А. – д-р техн. наук, професор, професор кафедри автоматики и робототехнічних систем ім. академіка І. І. Мартиненка, Національний університет біоресурсів и природокористування України, Київ, Україна.

Савченко Т. В. – канд. техн. наук, доцент, доцент кафедри інженерії програмного забезпечення та кібербезпеки, Державний торговельно-економічний університет, Київ, Україна.

Руденський А. А. – ст. викл кафедри автоматики и робототехнічних систем ім. академіка І. І. Мартиненка, Національний університет біоресурсів и природокористування України, Київ, Україна.

АНОТАЦІЯ

Актуальність. Особливістю сучасного цифрового світу є те, що злочинність вже нерідко вчиняється завдяки новітнім комп'ютерним технологіям, а робота правоохоронних органів стикається з низкою складних викликів у цифровому середовищі.. Розвиток інформаційних технологій та інтернет-комунікацій створює нові можливості для злочинців, які використовують цифрові сліди та докази для здійснення злочинів, що ускладнює процес їх виявлення та відслідковування.

Мета. Розробка прикладної онтології для системи аналізу цифрового кримінального злочину, яка дозволить ефективно аналізувати, обробляти та інтерпретувати велику кількість цифрових даних. Вона допоможе впоратися зі складним завданням обробки цифрових даних, а також сприятиме автоматизації процесу виявлення нових знань.

DOI 10.15588/1607-3274-2023-4-17

Метод. Для побудови онтологічної моделі як засобу відображення знань про цифровий злочин було зібрано інформацію про існуючі міжнародну та вітчизняну класифікації, а також проаналізовані потреби та вимоги, які мають бути задоволені розробленою онтологією. Створення онтологічної моделі, яка відображає основні поняття, взаємозв'язки у сфері цифрового кримінального злочину, було здійснено відповідно з визнаним онтологічним аналізом спеціалізованої предметної області.

Результати. Прикладна онтологія містить означення сутностей, властивостей, класів, підкласів тощо, а також створення семантичних відношень між ними. В центрі семантики знаходиться клас цифровий злочин (Digital Crime), проблемну область якого доповнюють взаємозв'язані класи злочинець (Intruder), цифровий слід (Digital evidence), тип злочину (Types of Crime) та кримінальна відповідальність (Criminal liability). Такі характеристики, як мотив, вид злочину, метод скоєння, класифікаційні ознаки цифрових слідів та типів злочину, а також інша індивідуальна інформація були віднесені до атрибутів відповідних класів. Реалізована онтологічна модель на мові OWL програмним засобом Protégé. Особливістю реалізації прикладної онтології було створення SWRL-правил для автоматичного заповнення додаткових зв'язків між екземплярами класу. Ручна та автоматична перевірка онтології показала цілісність, узгодженість, високу ступінь коректності та адекватності моделі. Виявлені помилки були, як правило, пов'язані з технічними аспектами та семантичними неузгодженостями, які були виправлені під час подальших ітерацій розробки.

Висновки. Дослідження підтвердило ефективність розробленої прикладної онтології для аналізу цифрового кримінального злочину, забезпечуючи більш точні та швидкі результати порівняно з традиційними підходами.

КЛЮЧОВІ СЛОВА: онтологія, цифрова криміналістика, цифровий злочин, цифрові сліди.

ЛІТЕРАТУРА

1. Про зареєстровані кримінальні правопорушення та результати їх досудового розслідування 2023. Офіс генерального прокурора. – Режим доступу: https://gp.gov.ua/ua/posts/pro-zareyestrovani-kriminalni-pravoporushennya-ta-rezultati-yih-dosudovogo-rozsliduvannya-2
2. Barn R. An ontological representation of a taxonomy for cybercrime / R. Barn, B. Barn // Twenty-Fourth European Conference on Information Systems (ECIS): proceedings. – İstanbul, Turkey, 2016. – Access mode: https://core.ac.uk/download/pdf/42490758.pdf
3. Core Ontology for Describing Production Equipment According to Intelligent Production / [L. Vlasenko, N. Lutska, N. Zaiets et al.] // Applied System Innovation. – 2022. – Vol. 5, Issue 5. – P. 98–111. DOI: 10.3390/asi5050098
4. Domain ontology development for condition monitoring system of industrial control equipment and devices / [L. O. Vlasenko, N. M. Lutska, N. A. Zaiets et al.] // Radio Electronics, Computer Science, Control. – 2022. – Vol. 1. – P. 157–166. DOI: 10.15588/1607-3274-2022-1-16
5. Donalds C. Toward a cybercrime classification ontology: A knowledge-based approach / C. Donalds, K. M. Osei-Bryson // Computers in Human Behavior. – 2019. – Vol. 92. – P. 403–418. DOI: 10.1016/j.chb.2018.11.039
6. Sikos L. F. AI in digital forensics: Ontology engineering for cybercrime investigations / L. F. Sikos // Wiley Interdisciplinary Reviews: Forensic Science. 2020. – Vol. 3, Issue 3. – P. 1–11. DOI: 10.1002/wfs2.1394
7. Making sense of digital traces: An activity theory driven ontological approach / [S. Karanasios, D. Thakker, L. Lau et al.] // Journal of the American Society for Information Science and Technology. – 2013. – Vol. 64, Issue 12. – P. 2452–2467. DOI: 10.1002/asi.22935
8. Brady O. Addressing the increasing volume and variety of digital evidence using an ontology / O. Brady, R. Overill, J. Keppens // 2014 IEEE joint intelligence and security informatics conference. IEEE, 2014. – P. 176–183. DOI: 10.1109/JISIC.2014.34
9. Cyber identity: Salient trait ontology and computational framework to aid in solving cybercrime / [M. C. Michel, M. Carvalho, H. Crawford et al.] // 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. 12th IEEE International Conference on Big Data Science and Engineering. IEEE, 2018. – P. 1242–1249. DOI: 10.1109/TrustCom/BigDataSE.2018.00171
10. Alzaabi M. An Ontology-Based Forensic Analysis Tool / M. Alzaabi, A. Jones, T. A. Martin // Proceedings of the 2013 Annual ADFSL Conference on Digital Forensics, Security and Law, 2013. – P. 121–136. – Access mode: https://commons.erau.edu/adfsl/2013/wednesday/5
11. Ontology-based smart sound digital forensics analysis for web services / [A. Akremi, M.-F. Sriti, H. Sallay, M. Rouached] // International Journal of Web Services Research. – 2019. – Vol. 16, № 1. – P. 70–92. DOI: 10.4018/IJWSR.2019010104
12. Alzaabi M. Ontology-Based Forensic Analysis of Mobile Devices / M. Alzaabi // Proceedings of the 20th IEEE International Conference on Electronics, Circuits and Systems. IEEE, 2013. – P. 64–68. DOI: 10.1109/ICECS.2013.6815346
13. Park H. Cyber Forensics Ontology for Cyber Criminal Investigation / H. Park, S. Cho, H. Kwon // Forensics in Telecommunications, Information and Multimedia, LNICST, 2009. – Vol. 8. – P. 160–165. DOI: 10.1007/978-3-642-02312-5_18
14. Найдьон Я. Поняття та класифікація віртуальних слідів кіберзлочинів / Я. Найдьон // Підприємництво, господарство і право. – 2019. – № 5. – С. 304–307. DOI: 10.32849/2663-5313/2019.5.56
15. Резильєнтність комп'ютерних систем в умовах кіберзагроз: Онтологія та таксономії / [С. М. Лисенко та ін.] // Радіоелектронні і комп'ютерні системи. – 2020. – № 1 – С. 17–28. DOI: 10.32620/reks.2020.1.02
16. Qualifying Cyber Crime as a Crime of Aggression in International Law / [M. Maskun, A. Achmad, N. Naswar et al.] // Cybercrime under International Law. – 2020. – Vol. 13, № 2. – P. 397–418. DOI: 10.14330/jeail.2020.13.2.08
17. Uma M. A survey on various cyber attacks and their classification / M. Uma, G. Padmavathi // Int. J. Netw. Secur. – 2013. – Vol. 15, № 5. – P. 390–396. DOI:10.6633/IJNS.201309.15(5).09
18. Chandra A. A taxonomy of cybercrime: Theory and design / A. Chandra, M. J. Snowe // International Journal of Accounting Information Systems. – 2020. – Vol. 38. – P. 100467. DOI: 10.1016/j.accinf.2020.100467
19. Harmandeep S. B. Cybercrimes: A proposed taxonomy and challenges / S. B. Harmandeep, G. Kumar // Journal of Computer Networks and Communications. – 2018. – Vol. 11. DOI: 10.1155/2018/1798659
20. Goni O. Cyber Crime and Its Classification / O. Goni // Int. J. of Electronics Engineering and Applications. – 2022. – Vol. 10, № 1. – P. 01–17, DOI: 10.30696/IJEEA.X.I.2021.01-17
21. Кримінальний кодекс України. Верховна Рада України. Законодавство України. – Access mode: https://zakon.rada.gov.ua/laws/show/2341-14
22. Themis. Ontology testing. – Access mode: https://themis.linkeddata.es/index.html
23. Ontological Support System of Managerial Decision-Making of Production Tasks for a Food Enterprise / [N. Lutska, L. Vlasenko, A. Ladanyuk et al.] // Machinery & Energetics. – 2022. – Vol. 13, №3. – P. 53–61. DOI: 10.31548/machenergy.13(3).2022.53-61

DOI 10.15588/1607-3274-2023-4-17

p-ISSN 1607-3274 Радіоелектроніка, інформатика, управління. 2023. № 4
e-ISSN 2313-688X Radio Electronics, Computer Science, Control. 2023. № 4

УПРАВЛІННЯ У ТЕХНІЧНИХ СИСТЕМАХ

CONTROL IN TECHNICAL SYSTEMS

UDC 62.505:629.524

THE FREQUENCY METHOD FOR OPTIMAL IDENTIFICATION OF CLOSE-LOOP SYSTEM ELEMENTS

Osadchyi 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.

Zozulya V. A. – PhD, Associate Professor of the Department of Digital Economy and System Analysis, State University of Trade and Economics, Kyiv, Ukraine.

Kalich V. M. – PhD, Professor of the Department of Automation of Production Processes, Central Ukrainian National Technical University, Kropyvnytskyi, Ukraine.

Timoshenko A. S. – Senior Lecturer of the Department of Aeronautics, Meteorology and Air Traffic Management, Flight Academy of the National Aviation University, Kropyvnytskyi, Ukraine.

ABSTRACT

Context. The article is devoted to overcoming the contradictions between the assumptions adopted in known methods of closed-loop control system identification and the design and conditions of its operation. The article presents a new method of identifying the transfer functions matrix of a two-level closed-loop control system element, which functions under the conditions of multidimensional stationary centered random influences.

Objective. The purpose of the study, the results of which are presented in this paper, is to extend the indirect identification method to the case of estimating one of the two-level closed-loop control system elements' dynamics model based on passive experiment data.

Method. To solve the optimal identification problem, a variational method for minimizing the quality functional on the class of fractional-rational transfer function matrices was used.

Results. As a result of the research, the identification problem formulation was formalized, the rules for obtaining experimental information about the input and output signals were determined, the rules for identifying the transfer functions matrix of a two-level closed-loop control system element, which minimizes the sum of the variances of identification errors in the frequency domain, and the verification of these rules was carried out.

Conclusions. Justified rules allow to correctly determine transfer functions matrices of the closed-loop systems selected element when fulfilling the defined list of conditions. The closed-loop systems control paths signals analysis proves the possibility of the effect of changing these signals statistical means, even under conditions of only centered stationary input influences actions on the system. Based on this, the further development of research can be aimed at overcoming such effects.

KEYWORDS: Identification, transfer function matrix, spectral density, error variance, quality functional.

NOMENCLATURE

M_1 is a matrix of dimension $m\times n$, the elements of which are polynomials from the differentiation operator ρ;

m is a number of signals at the output of the local control system;

N_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;

N_+ is a matrix of fractional rational functions whose poles are located in the left half-plane of the complex plane;

N_- is a matrix of fractional rational functions with poles in the right half-plane;

n is a number of inputs of the local system;

$O_{m\times n}$ is a zero matrix of size $m\times n$;

P_1 is a matrix of dimension $m\times m$;

R is an additionally defined weight matrix;

r is a vector of programme signals;

S'_{rr} is a transposed spectral density matrix of the vector r;

$S'_{x_n x_n}$ is a transposed spectral density matrix of the vector x_n;

$S'_{\zeta x}$ is a transposed matrix of mutual spectral densities between the generalised input vector ζ and the vector x_n;

$S'_{x\zeta}$ is a transposed matrix of mutual spectral densities between the vectors x_n and ζ;

DOI 10.15588/1607-3274-2023-4-18

$S'_{\delta\delta}$ is a transposed matrix of spectral densities of uncorrelated white noise of single intensity;

$S'_{\varphi\varphi}$ is a transposed spectral density matrix of measurement noise;

$S'_{\varphi_0\varphi_0}$ is a transposed spectral density matrix of the dummy noise vector φ_0;

$S'_{\psi\psi}$ is a transposed matrices of spectral densities of disturbances;

$S'_{\zeta\zeta}$ is a transposed spectral density matrix of the generalised input vector;

u_1 is a vector of input signals of the local system;

u_n is a mismatch vector;

W_n is a matrix of transfer functions that determines the relationship between the operator's reactions to changes in the components of the misalignment vector u_n;

W_p is a matrix of transfer functions that determines the relationship between the operator's actions to prevent and probe pulses δ;

x_1 is a vector of signals at the output of the local control system;

x_n is a vector of control signals;

y is a vector of master feedback signals;

Φ is a block matrix of transfer functions of size $n\times(n+m)$;

δ is a vector of sensing pulses;

ε_x is the vector of identification errors;

φ is a vector of measurement noise;

φ_0 is a dummy measurement noise vector;

ρ is a vector of additional signals;

ψ is a vector of disturbances with m components;

ζ is a generalised vector of input influences.

INTRODUCTION

According to the definition given in the well-known article [1], one of the central problems in systems theory is the problem of identification. According to L. Zadeh, this problem is "determination, on the basis of observation of input and output, of a system within a specified class of systems to which the system under test is equivalent; determination of the initial or terminal state of the system under test". If we limit ourselves to considering the works [2, 3] devoted to the determining automatic control systems elements dynamics models, it is obvious that the whole set of such studies is divided into two parts. The first part is, for example, works [3–5], which are devoted to determining open-loop systems and their elements dynamics models. The second part, for example works [2, 6–8], combines studies aimed at solving the closed-loop control system elements identification problem. Despite the large number of papers devoted to solving the latter problem, the search for new methods and means of determining the dynamics models of closed-loop control system elements is still relevant.

This relevance is due to the existence of contradictions between the assumptions made when formulating the identification method and the design and operating conditions of a closed-loop control system. In the context of the fourth industrial revolution, there is a rapid increase in the diversity of control objects and, accordingly, systems. Therefore, the requirements of practice require bringing the identification procedures into line with the conditions of the closed-loop control system design tasks.

The object of study in this paper is a two-level closed-loop control system.

The subject of study is identification of a transfer functions matrix of a two-level closed-loop control sytem element.

The purpose of the work is to substantiate the rules for estimating two-level closed-loop control system's one of the elements dynamics model based on passive experiment data.

1 PROBLEM STATEMENT

As a rule, modern control systems have physical subsystems with many inputs and outputs as objects. These subsystems operate under the influence of vector stochastic useful signals, measurement noise, and interference. For example, the flight control system of an unmanned aerial vehicle or aircraft. Thus, of all considered. identification methods [1–8], only the indirect and joint methods remain. They allow identifying a multidimensional closed-loop control system if its structure can be represented as shown in Fig. 1, and the sensors have low inertia and low intensity of measurement noise.

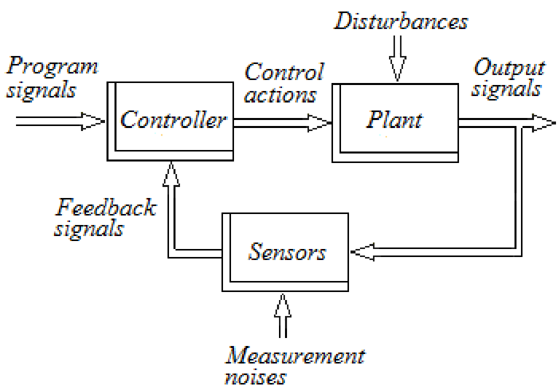


Figure 1 – Architecture of a closed-loop (local) control system

However, the development of the principles of controlling such objects has led to the emergence of closed-loop systems that have two control loops (Fig. 2) [4, 16].

The external loop is used to generate a vector of control signals that are transmitted to the local control system. This vector is generated by the master controller as a result of comparing the vectors of program signals and master feedback signals. The master feedback signals differ slightly from the controlled variables due to the influence of measurement noise and the inertial properties of the master sensors. The relationship between the controlled variables and the signals at the output of the control object is characterised by a kinematic link. If the control object is a moving vehicle, this link solves the inverse kinematics problem [17].

DOI 10.15588/1607-3274-2023-4-18

Thus, the use of the known indirect and joint methods of identification to solve the problem of determining the system (Fig. 2) elements dynamics models requires modification of these methods.

We will assume that, due to preliminary experiments and the use of known identification methods, a linearized model of the dynamics of the local control system has been determined and presented as a system of ordinary differential equations with constant coefficients of the form

$$P_1 x_1 = M_1 u_1 + \psi , \qquad (1)$$

where P_1 is a dimension matrix $m \times m$, the elements of which are polynomials from the differentiation operator

$$p = \frac{d}{dt} ,$$

m is the number of signals at the output of the local control system; x_1 is a vector of signals at the output of the local control system (Output signals); M_1 is a matrix of dimension $m \times n$, the elements of which are polynomials from the differentiation operator p; n is the number of inputs of the local system; u_1 is a vector of input signals of the local system; ψ is a vector of disturbances with m components. In this case, the architecture of a two-level closed-loop control system is transformed into a block diagram (Fig. 3). As you can see, this diagram has two parts.

The first part combines the main controller and the communication system with the main sensors (Fig. 3). Three vectors act on the inputs of the main controller (Fig. 3): program signals r, main feedback signals y, and additional signals ρ. The origin and effect of additional signals depend on the purpose and design of the control system. At the outputs of the master controller, a vector of control signals x_n is formed. This vector is simultaneously the local control system input signals vector u_1. Relationship between these vectors is characterised by two transfer function matrices W_n and W_p. For example,

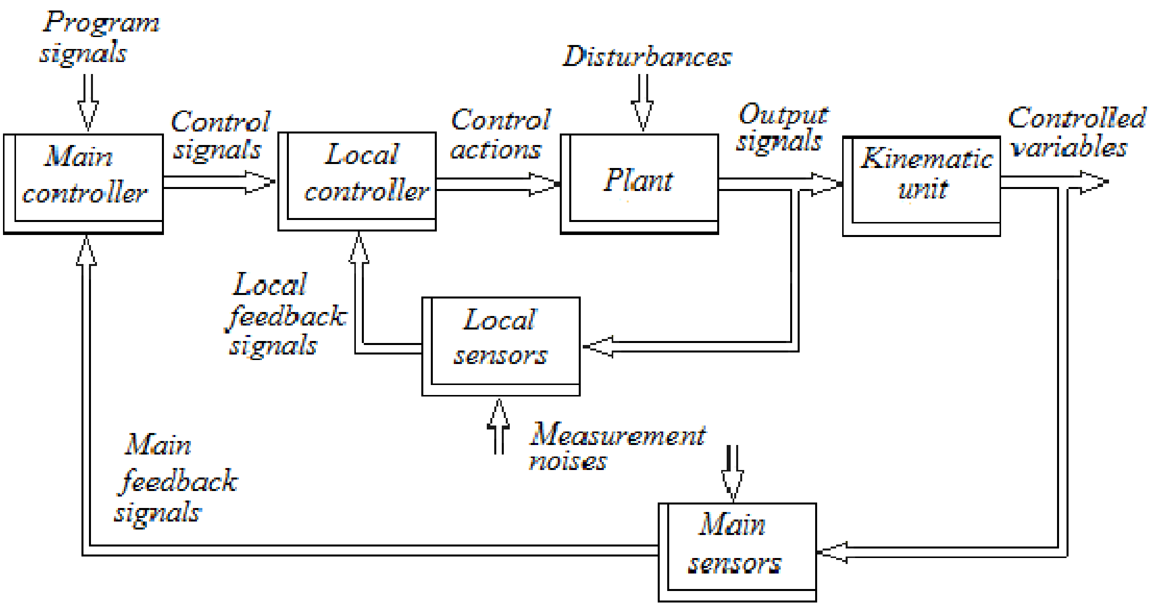


Figure 2 – Architecture of a two-level closed-loop control system

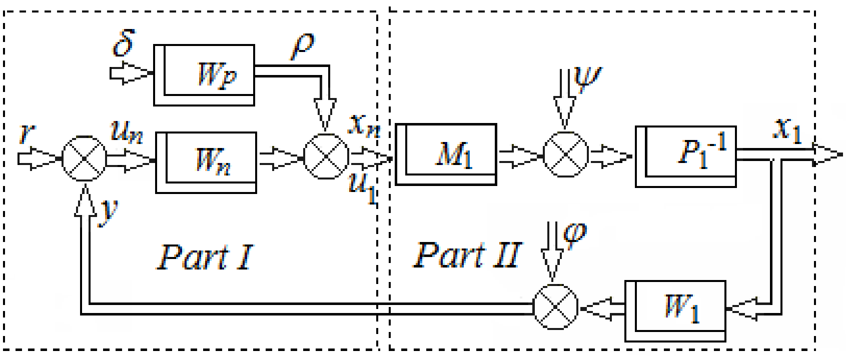


Figure 3 – Block diagram of a two-level closed-loop control system

DOI 10.15588/1607-3274-2023-4-18

if the control system is designed to ensure the movement of an unmanned aerial vehicle along a given trajectory, then the central part of the master controller is the pilot-operator. Its dynamic properties are characterised by two transfer functions matrices [9, 10]. The first W_n determines the relationship of the operator's reactions to changes in the components of the misalignment vector u_n. The second matrix W_p describes the relationship between the operator's actions to prevent (remnant) and the sensing impulses of his nervous system δ.

The block diagram (Fig. 3) second part combines the local control system, the kinematic link, and the main sensors (Fig. 2). Its inputs are also three vectors: control signals u_1, disturbances ψ, and measurement noise φ. The output of the second part is a vector of main feedback signals y. The effect of changing the components of the signal vector at the output of the local system x_1 on the components of the vector y is characterised by the transfer function matrix W_1.

The structure and parameters of the transfer function matrix W_1 and the model of measurement noise dynamics can be determined separately from the system (Fig. 3) as a result of dynamic sensor certification according to the methodology given, for example, in monograph [4].

At the same time, it is possible to determine the transfer function matrices of the master controller, which it has in real operating conditions, only as a result of solving the problem of identifying a closed-loop control system in an appropriate manner.

Taking into account the statement of Peter Eykhoff, substantiated in article [11], about the possibility of unambiguous identification of only the transfer function of an element of a closed-loop control system, the following identification problem is formulated.

Let the polynomial matrices P_1, M_1, the transfer function matrix W_1, the transposed spectral density matrices of disturbances $S'_{\psi\psi}$ and measurement noise $S'_{\varphi\varphi}$ be given, and it is known that all signals in the control loops of the system (Fig. 4) are centred stationary random signals. It is also assumed that the vectors r, u_n, x_n are measured with sufficient accuracy. The optimal identification task is that, as a result of processing experimental data (records of vectors r, u_n, x_n) and a priori information about those elements of the system in the Fig. 4, the dynamics of which is known, to find algorithm of searching for such matrices W_n and W_p, at which the identification error vector ε_x components weighted variances sum would be minimal.

2 REVIEW OF THE LITERATURE

The analysis of methods for identifying closed-loop control systems based on the study of such literature sources as [2, 6–8] and [12], allows dividing them into four parts (Fig. 4).

Direct Methods [6, 12] are used to determine the model of the dynamics of the control object and sometimes the disturbances acting during the experiment, while ignoring the presence of feedback. The main restrictions on the use of this set of methods are:

– the requirement of a low intensity of disturbances with a high intensity of the useful signal;

– requirement of knowledge of the disturbance dynamics models;

– the requirement of low sensor inertia and high signal-to-noise ratio;

– the need to pre-determine the order of the control object;

– limitation of the control object dynamics to stable dynamic links only.

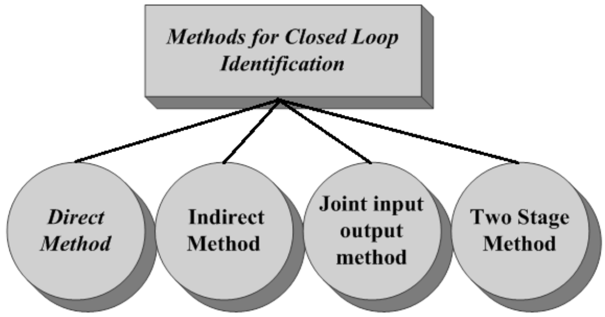


Figure 4 – Types of methods for identifying elements of closed-loop systems

An attempt to overcome the shortcomings of direct methods led to the development of a variational method for identifying the dynamics of multidimensional control objects, which is described in [13]. This method does not require a priori information about the model of disturbance dynamics and the order of the control object.

Indirect Method [6, 12] involves conducting an experiment, obtaining records of signals acting on the inputs and outputs of the system, identifying the dynamics model of a closed-loop system, calculating the dynamics model of an equivalent open system, and searching for the dynamics model of the control object. Algorithms that implement this method have an advantage over the direct method due to the absence of the requirement to have a model of the dynamics of disturbances. At the same time, the main disadvantages of these methods include:

– the need for prior knowledge of the controller's control law (transfer function);

– limitation of the class of systems that can be identified to one-dimensional ones;

– requirement of low sensor inertia and high signal-to-noise ratio;

– the need to pre-determine the order of the control object.

An attempt to overcome the disadvantages of indirect methods led to the development of a variational method for identifying the dynamics of multidimensional control objects, which is described in [14, 15]. This method overcomes all the disadvantages of the indirect method, but limits the class of useful signals, disturbances, and interferences that act during the experiment. All these signals must belong to centred stationary random processes or to an additive mixture of a stationary random process and a deterministic time function.

DOI 10.15588/1607-3274-2023-4-18

Joint input-output method [6] involves combining the signals acting at the input and output of the system into a single signal vector. This vector is considered to be the output of some imaginary dynamic system, at the input of which there is a virtual test signal with the known dynamics, for example, “white noise”. It is believed [12] that the main advantage of this method is associated with the absence of the need for a priori information about the system and disturbances. However, it also has certain limitations (disadvantages):

– the need to measure all useful signals, disturbances and interferences that are present during the experiment;

– the experiment must reproduce real conditions of the system;

– the requirement of low sensor inertia and high signal-to-noise ratio;

– the need to pre-determine the order of the control object.

Two Stage Method [6, 8] consists of reducing a closed-loop system to an equivalent open system and then parametrically identifying this open system. The main difficulty in applying this method is the need to fulfil several requirements:

– having a priori knowledge of the system’s block diagram;

– the block diagram should have one input and one output;

– sensors must have low inertia and measurement noise;

– the controller must follow a linear control law.

The purpose of the study, the results of which are presented in this article, is to extend the indirect identification method to the case of estimating the dynamics model of one of the elements of a two-level closed-loop control system based on passive experiment data.

To achieve this goal, we solved the problem of determining the set of necessary a posteriori information about the signal vectors in the control paths of the system (Fig. 4), as well as the substantiation of the algorithm for identifying the dynamics of one of the elements of a closed-loop system, provided that the identified dynamics model delivers an extreme of the selected quality indicator.

3 MATERIALS AND METHODS

According to the problem statement and the block diagram (Fig. 3), the identification error vector must satisfy the equation

$$\varepsilon_x = x_n - \Phi\zeta, \tag{2}$$

where Φ is a block matrix of transfer functions of size $\times(n+m)$ type

$$\Phi=[\Phi_{11}\ \Phi_{12}], \tag{3}$$

that relate the output of the identified master controller model (the reconstructed vector x_n) to the input vectors δ, r, ψ, φ, ζ is a generalised input vector of the form

$$\zeta = \begin{bmatrix} \delta \\ r+\varphi_0 \end{bmatrix}, \tag{4}$$

where φ_0 is a dummy measurement noise vector equal to

$$\varphi_0 = \begin{bmatrix} -W_1 P_1^{-1} & -E_m \end{bmatrix} \begin{bmatrix} \psi \\ \varphi \end{bmatrix}. \tag{5}$$

The sum of the weighted variances of identification errors can be determined in the frequency domain [18] by applying the Wiener-Khinchin theorem in vector form [19] to expression (2) as

$$J = \frac{1}{j}\int_{-j\infty}^{j\infty} tr\left\{\left(S'_{x_n x_n} - S'_{\zeta x_n}\Phi_* - \Phi S'_{x\zeta} + \Phi S'_{\zeta\zeta}\Phi_*\right)R\right\}\cdot ds, \tag{6}$$

where $S'_{x_n x_n}$ is a transposed matrix of spectral densities of the vector x_n determined as a result of statistical processing of records of the components of this vector; $S'_{\zeta x}$ is a transposed matrix of mutual spectral densities between the generalised input vector ζ and the vector x_n

$$S'_{\zeta x} = \begin{bmatrix} S'_{\delta x_n} & S'_{r x_n} + S'_{\varphi_0 x_n} \end{bmatrix}, \tag{7}$$

The index * denotes the Hermitian conjugation of the matrix; $S'_{x\zeta}$ is a transposed matrix of cross spectral densities between the vectors x_n and ζ, which is equal to

$$S'_{x\zeta}\left(S'_{\zeta x}\right)_*;$$

$S'_{\zeta\zeta}$ is a transposed spectral density matrix of the generalised input vector (4), which has the following form in the case of vectors’ δ, r, ψ, φ different and independent origin sources

$$S'_{\zeta\zeta} = \begin{bmatrix} S'_{\delta\delta} & O_{n\times m} \\ O_{m\times n} & S'_{rr} + S'_{\varphi_0\varphi_0} \end{bmatrix}; \tag{8}$$

$O_{m\times n}$ is a zero matrix of size $m\times n$; R is a positively defined weight matrix.

The transposed matrices of spectral densities S'_{rr} and cross spectral densities $S'_{\zeta x}$ from expressions (7), (8) can be found as a result of approximating the estimates of these matrices obtained, for example, using the CPSD function of the Matlab package [20], on the class of fractional rational functions of complex argument [16]. The transposed spectral density matrix of the fictitious noise vector φ_0, obtained by using the Wiener-Khinchin theorem applied to the vector (5), can be represented as

DOI 10.15588/1607-3274-2023-4-18

$$S'_{\varphi_0\varphi_0} = W_1 P_1^{-1} S'_{\psi\psi} P_{1*}^{-1} W_{1*} + S'_{\varphi\varphi}. \quad (9)$$

The expression for calculating the transposed cross spectral densities matrix $S'_{\varphi_0 x_n}$, is determined as a result of the structural scheme (Fig. 3) transformations by the following equation

$$S'_{\varphi_0 x_n} = S'_{u_n x_n} + S'_{x_n x_n} M_{1*} P_{1*}^{-1} W_{1*} - S'_{r x_n}; \quad (10)$$

where the transposed matrices of spectral and cross spectral densities $S'_{x_n x_n}$, $S'_{u_n x_n}$, $S'_{r x_n}$, as well as the matrix $S'_{\zeta x}$ can be estimated from the records of the components of the vectors r, u_n, x_n.

The only difficulty is to find the equation of the relationship between the transposed matrix $S'_{\delta x_n}$ and the original data of the identification task. The analysis of the block diagram in Fig. 3 shows the equation

$$x_n = W_n \left(r + \varphi_0 - P_1^{-1} M_1 x_n \right) + W_p \delta. \quad (11)$$

In addition, the control systems statistical dynamics postulates [21] allow proving the following identity for the transfer function matrix W_p (Fig. 1):

$$S'_{\delta x_n} \left(S'_{\delta\delta} \right)^{-1} S'_{x_n \delta} = W_p S'_{\delta\delta} W_{p*}. \quad (12)$$

Thus, as a result of applying Wiener-Khinchin theorem to the left and right sides of equation (11), taking into account identity (12), the following matrix coupling equation is determined

$$S'_{\delta x_n} \left(S'_{\delta\delta} \right)^{-1} S'_{x_n \delta} = S'_{x_n x_n} - S'_{r x_n} \left(S'_{rr} \right)^{-1} S'_{x_n r} - A \left(S'_{\varphi_0\varphi_0} \right)^{-1} A_*, \quad (13)$$

where

$$A = S'_{u_n x_n} + S'_{x_n x_n} M_{1*} P_{1*}^{-1} W_{1*} - S'_{r x_n}.$$

Factoring the right-hand side of equation (13) on the left [22] along with taking into account the known form of the spectral density matrix of uncorrelated white noises of single intensity $S'_{\delta\delta}$, allows finding the transposed matrix $S'_{\delta x_n}$. The materials presented in [14] prove that there is a connection between matrices Φ_{11}, Φ_{12} and other matrices from the structural diagram (Fig. 3). It is formalised by the following equations:

$$\Phi_{11} = F_1^{-1} W_p, \quad (14)$$

$$\Phi_{12} = F_1^{-1} W_n, \quad (15)$$

where

$$F_1 = E_n + W_n W_1 P_1^{-1}. \quad (16)$$

Equations (14)–(16), given the known matrices Φ_{11} and Φ_{12}, allow uniquely finding the matrices of the transfer functions W_n and W_p. Thus, the problem of optimal identification of the closed-loop system elements is reduced to finding the transfer functions matrix Φ corresponding to a physically possible system while the functional (6) being minimal, given the known matrices of spectral and cross spectral densities $S'_{x_n x_n}$ and $S'_{\zeta x}$.

The solution to the problem was found by the well-known Wiener-Kolmogorov method of minimising the quadratic functional (6) on the class of transfer function matrices of physically possible systems Φ in the frequency domain. In accordance with the chosen method, the first variation of the functional (6) was found in the following form

$$\delta J = \frac{1}{j} \int_{-j\infty}^{j\infty} tr \left\{ -R S'_{\zeta x} + R \Phi S'_{\zeta\zeta} \right\} \delta\Phi_* ds + \frac{1}{j} \int_{-j\infty}^{j\infty} tr\, \delta\Phi \left\{ -S'_{x\zeta} R + S'_{\zeta\zeta} \Phi_* R \right\} ds. \quad (17)$$

The search for physically possible Lyapunov variational matrices requires factorisation of the weight matrix R on the right and factorisation of the transposed spectral density matrix of the generalised input vector (8) on the left [22]. As a result of these operations, the stable matrices Γ and D, together with their inverses, are found to satisfy by the equations

$$\Gamma_* \Gamma = R, \quad D D_* = S'_{\zeta\zeta}. \quad (18)$$

Substituting the obtained equations (18) into the variation of (17) allows representing the latter as

$$\delta J = \frac{1}{j} \int_{-j\infty}^{j\infty} tr \left\{ \Gamma_* \left[-\Gamma S'_{\zeta x} D_*^{-1} + \Gamma \Phi D \right] D_* \right\} \delta\Phi_* ds + \frac{1}{j} \int_{-j\infty}^{j\infty} tr\, \delta\Phi \left\{ D \left[-D^{-1} S'_{x\zeta} \Gamma_* + D_* \Phi_* \Gamma_* \right] \right\} ds. \quad (19)$$

As a result of the separation (splitting) of the product of matrices $-\Gamma S'_{\zeta x} D_*^{-1}$ is represented as the sum of three matrices:

$$N_0 + N_+ + N_- = -\Gamma S'_{\zeta x} D_*^{-1}, \quad (20)$$

DOI 10.15588/1607-3274-2023-4-18

where N_0 is the matrix of the results of dividing the polynomials of the numerators by the polynomials of the denominators of the product on the right side of the expression (20); N_+ is the matrix of fractional rational functions of the complex argument $s = j\omega$, with the poles located in the left half-plane of the complex plane; $N\_$ is the matrix of fractional rational functions with poles in the right half-plane.

Substituting the result (20) into the variation of (19) allows obtaining the following result:

$$\delta J = \frac{1}{j}\int_{-j\infty}^{j\infty} tr\{\Gamma_*[N_0 + N_+ + \Gamma\Phi D]D_*\}\delta\Phi_* ds + \frac{1}{j}\int_{-j\infty}^{j\infty} tr\{\Gamma_* N_- D_*\}\delta\Phi_* ds + \frac{1}{j}\int_{-j\infty}^{j\infty} tr\delta\Phi\{D[N_{0*} + N_{+*} + D_*\Phi_*\Gamma_*]\Gamma\}ds + \frac{1}{j}\int_{-j\infty}^{j\infty} tr\{DN_{-*}\Gamma\}\delta\Phi ds. \quad (21)$$

According to the Cauchy residual theorem, the second and fourth integrals in expression (21) are zero, and the condition for ensuring the minimum of the functional (6) on the class of stable and minimally phase variations of Φ is as follows:

$$\Gamma\Phi D = -(N_0 + N_+). \quad (22)$$

The solution of equation (22) with respect to the transfer function matrix Φ:

$$\Phi = -\Gamma^{-1}(N_0 + N_+)D^{-1}, \quad (23)$$

allows determining the blocks Φ_{11} and Φ_{12} on the basis of relation (3) and proceed to the search for the matrices of transfer functions W_n and W_p. Substituting expression (16) into relation (15) allows writing the following rule for identifying the transfer function matrix W_n:

$$W_n = \Phi_{12}\left(E_m + W_1 P_1^{-1} M_1 \Phi_{12}\right)^{-1}. \quad (24)$$

The transformation of equations (14) and (16) defines the rule for identifying the transfer function matrix W_p in the following form:

$$W_p = \left(E_m + W_n W_1 P_1^{-1} M_1\right)\Phi_{11}. \quad (25)$$

Thus, equations (7)–(10) and (13) have been defined, which allow forming a set of a posteriori information about signal vectors in the control paths of a closed-loop system necessary for identification. In addition, the rules (23)–(25) are substantiated, which define an algorithm for identifying two-level closed-loop control system's one of the elements dynamics model based on passive experiment data, on the condition that the identified dynamics model delivers an extremum of the quadratic quality index (6). Thus, it is optimal [23].

4 EXPERIMENTS

The basis for checking the correctness of the new identification rules is the principle of comparing the given transfer functions W_{n0} and W_{p0} of the form:

$$W_{n0} = \frac{0.3(s+2)}{s},\ W_{p0} = \frac{0.032}{s+0.053}, \quad (26)$$

with the transfer functions W_n and W_p calculated as a result of applying algorithms (24), (25). As the initial data for identification, we used the structural diagram of the system (Fig. 3) with the following dynamic characteristics of it's known elements:

$$M_1 = 2,\ P_1 = 10(s+0.2),\ W_1 = 1, \quad (27)$$

as well as the following spectral densities

$$S'_{rr} = \frac{0.01}{-s^2+0.01},\ S'_{\psi\psi} = \frac{0.1}{-s^2+1},\ S'_{\varphi\varphi} = 0.001,\ S'_{\delta\delta} = 1. \quad (28)$$

Let us assume that the following spectral and cross spectral densities are obtained as a result of processing the statistical data on the records of signals r, u_n, x_n:

$$S'_{x_n x_n} = \frac{\left|(s+0.35)(s+0.057)(s^2+5.3s+7.1)\right|^2}{\left|(s+1)(s+0.053)(s^2+0.26s+0.12)\right|^2} \times 0.9\cdot 10^{-5}, \quad (29)$$

$$S'_{r x_n} = \frac{0.003(s+2)(s+0.2)}{|s+0.1|^2(s^2+0.26s+0.12)}, \quad (30)$$

$$S'_{u_n x_n} = \frac{0.0003(-s+0.08)|s+0.21|^2(s+1.99)}{\left|(s+0.053)(s+0.1)(s^2+0.13s+0.06)\right|^2} \times |s+3.16|^2(s^2+0.11s+0.004). \quad (31)$$

In this case, to form the transposed matrix (8), the following spectral density $S'_{\varphi_0\varphi_0}$ was found as a result of substituting the corresponding data from (27) and (28) into expression (9):

$$S'_{\varphi_0\varphi_0} = \frac{0.001\left|s^2+1.43s+1\right|^2}{\left|(s+0.1)(s+0.2)\right|}. \quad (32)$$

DOI 10.15588/1607-3274-2023-4-18

Thus, the transposed spectral density matrix of the generalised input vector, compiled in accordance with expression (9), taking into account the result of (32), is as follows:

$$S'_{\zeta\zeta}=\begin{bmatrix}1 & 0\\ 0 & \dfrac{0.001\left|(s+3.148)(s+0.38)\right|^2}{\left|(s+0.1)(s+0.2)\right|^2}\end{bmatrix}. \qquad (33)$$

From expression (7), it follows that to determine the transposed matrix of mutual spectral densities between the generalised input vector ζ and the vector x_n, it is necessary to find the spectral densities $S'_{\varphi_0 x_n}$ and $S'_{\delta x_n}$. Substitution of the corresponding matrices from relations (27), (29)–(31) into expression (10) allows us to determine that

$$S'_{\varphi_0 x_n}=\frac{0.0003(s+2)\left|s^2+1.43s+1\right|^2}{(-s+0.2)\left(s^2+0.26s+0.12\right)\left|s+0.1\right|^2}. \qquad (34)$$

The calculation of the right-hand side of the coupling equation (13) and taking into account the value of $S'_{\delta\delta}$, from expressions (28) substantiates the following relationship:

$$S'_{\delta x_n}S'_{x_n\delta}=\frac{0.000997\left|s+0.2\right|^2}{\left|(s+0.053)\left(s^2+0.26s+0.12\right)\right|^2}. \qquad (35)$$

Factoring the right-hand side of equation (35) from the left provided the spectral density $S'_{\delta x_n}$ in the following form

$$S'_{\delta x_n}=\frac{0.032(s+0.2)}{(s+0.053)\left(s^2+0.26s+0.12\right)}. \qquad (36)$$

Thus, the transposed matrix of mutual spectral densities between the generalised input vector ζ and the vector x_n, taking into account relations (7), (30), (34) and (36), is represented as

$$S'_{\zeta x}=\left[\frac{0.032(s+0.2)}{(s+0.053)\left(s^2+0.26s+0.12\right)} \quad \frac{0.0003(s+2)\left|(s+3.148)(s+0.38)\right|^2}{(-s+0.2)\left(s^2+0.26s+0.12\right)\left|s+0.1\right|^2}\right]. \qquad (37)$$

Since vector x_n has only one component, the weighting coefficient is used instead of the weighting matrix *R:*

$$R=1. \qquad (38)$$

From expression (38) and the definition of the factorisation operation of the fractional rational function on the right [22], it follows that

$$\Gamma=1. \qquad (39)$$

The left factorisation of matrix (33) allowed defining the following fractional rational matrix *D*, all features of which are located in the left half-plane of the complex variable $s=j\omega$,

$$D=\begin{bmatrix}1 & 0\\ 0 & \dfrac{0.032(s+3.148)(s+0.38)}{(s+0.1)(s+0.2)}\end{bmatrix}. \qquad (40)$$

The initial data for separation were determined by expression (20), taking into account the results of (37), (39), (40) in the following form

$$N_0+N_++N_-=\left[\frac{-0.032(s+0.2)}{(s+0.053)\left(s^2+0.26s+0.12\right)} \quad \frac{-0.0095(s+3.148)(s+2)(s+0.38)}{(s+0.1)\left(s^2+0.26s+0.12\right)}\right]. \qquad (41)$$

Since the fractional rational functions on the right-hand side of equation (41) have features with a negative real part, the result of the separation coincides with the initial data for it, namely

$$N_0+N_+=N_0+N_++N_-. \qquad (42)$$

Substituting the results (39), (40), (42) into rule (23) allows finding the following matrix Φ, which ensures the minimum of the functional (6),

$$\Phi=\left[\frac{0.032(s+0.2)}{(s+0.053)\left(s^2+0.26s+0.12\right)} \quad \frac{0.3(s+0.2)(s+2)}{\left(s^2+0.26s+0.12\right)}\right]. \qquad (43)$$

The analysis of the right-hand side of equation (43) shows that two relations are fulfilled:

$$\Phi_{11}=\frac{0.032(s+0.2)}{(s+0.053)\left(s^2+0.26s+0.12\right)}; \qquad (44)$$

$$\Phi_{12}=\frac{0.3(s+0.2)(s+2)}{\left(s^2+0.26s+0.12\right)}. \qquad (45)$$

They provide possibility of finding optimal estimates of the transfer functions W_n and W_p. Using equation (24), taking into account the given data (27) and the result (45), we prove that

$$W_n=\frac{0.3(s+2)}{s}. \qquad (46)$$

At the same time, substituting data from (27), (44) and (46) into the relation (25) allowed identifying the second transfer function in the form of

$$W_p=\frac{0.032}{s+0.053}. \qquad (47)$$

DOI 10.15588/1607-3274-2023-4-18

Comparison of the obtained transfer functions (46) and (47) with the given transfer functions (26) proves the correctness of the rules justified in the problem solution process.

Thus, the research goal has been achieved. The rules that extend the effect of the indirect identification method to the case of estimating the two-level closed-loop control system's one of the elements dynamics model based on passive experiment data have been determined.

5 RESULTS

As a result of the research, the identification problem was formalized, equations (7)–(10) and (13) which allow forming a set of a posteriori information about the input-output signals necessary for identification are defined,.

The rules (23)–(25) for identifying the two-level closed-loop control system's one of the elements dynamics model, which minimizes the sum of the identification errors variances in the frequency domain (6), are obtained and verified.

6 DISCUSSION

The conditions and restrictions on the use of the new frequency method for closed-loop system elements optimal identification were determined as follows:

– the system operates under the influence of one-dimensional or multidimensional centred stationary useful signals, disturbances and measurement noises, the dynamics of which may differ from white noise;

– models of dynamics of system elements that are not subject to identification should be known in advance;

– models of the dynamics of external influences on the system that act during the identification experiment must be specified;

– it is necessary to ensure the possibility of measuring the input-output signals of the closed-loop system element to be identified.

The signals in the control paths of closed-loop systems analysis proves the possibility of these signals mathematical expectations changing effect, even under the conditions of existence only centred stationary input influences on the system. On this basis, further development of research can be directed at overcoming such effects.

CONCLUSIONS

The urgent problem of mathematical support development is solved for identifying the two-level closed-loop control system's one of the elements dynamics model with the minimum of the identification errors variances sum.

The application of the Wiener-Kolmogorov ideas allows overcoming the contradictions between the assumptions made in the formulation of the identification method and a two-level closed-loop system design and operating conditions by developing and applying new rules for optimal identification of this system elements.

The scientific novelty of obtained results is that a result of the study, a new method for identifying a complex multidimensional element of a two-level closed-loop control system was determined for the first time. The justified method has two main distinguishing features. The first feature is that to solve the identification problem it is enough to measure one of the two vectors at the inputs of the identification object and the vector of signals at its outputs. The second distinctive feature is that as a result of solving the identification problem, two transfer function matrices are determined. The first characterizes the influence of the vector of controlled input signals on the output signals of the object, and the second determines the influence of the vector of uncontrolled input signals on the output of the object.

The practical significance of the results lies in the fact that the a priori conditions for obtaining initial data for identifying the elements of a closed-loop control system are substantiated. This allows for effective planning of an identification experiment, as well as justifying the list of signals that are to be recorded during this experiment.

Prospects for further research are to develop methods and rules for overcoming the effect of violation of stationarity in closed-loop control systems, which occurs even under the action of centered stationary random influences, when carrying out identification.

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 food production.

The preparation of this article would not have been possible without the creation of a creative environment at the Department of Aircraft Design, Aircraft Engines and Airworthiness Maintenance of the Flight Academy of the National Aviation University, headed by Professor Yulija Sikirda.

REFERENCES

1. Zadeh L.A. From Circuit Theory to System Theory, *Proc. IRE*, 1962, Vol. 50, pp. 856–865. DOI: 10.1007/978-3-540-71795-9_4.
2. 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
3. Eykhoff P. eds. Trends and progress in system identification: IFAC series for graduates, research workers and practising engineers, *Elsevier Science & Technology*, 2013, 418 p. ISBN-10: 1483116247. ISBN-13: 978-1483116242.
4. Blokhin L. N., Osadchyi S. I., Didyk O. K., Rudyuk H. I. Technologies of construction of modern competitive complexes for controlling stochastic motion of objects. Kirovohrad, Publisher-Lysenko V. F., 2015, 284 p. ISBN 978-617-7197-30-9
5. Hjalmarsson H. System Identification of Complex and structured systems, *European Journal of Control*, 2009, Vol. 15, № 3, pp. 275–310.
6. Afrooz Ebadat. Experiment Design for Closed-loop System Identification with Applications in Model Predictive Control and Occupancy Estimation: thesis ... doctor of philosophy.

DOI 10.15588/1607-3274-2023-4-18

Royal Institute of Technology (KTH). Stockholm, Sweden, 2017, 231 p. ISSN 1653-5146. ISBN 978-91-7729-464-1.
7. Wang Jianhong, Ramirez-Mendoza Ricardo A. The practical analysis for closed-loop system identification, *Cogent Engineering*, 2020, Vol. 7, Issue 1, pp. 1–16. DOI:10.1080/23311916.2020.1796895
8. Lei Zhou, Jiangang Li, Jun Sheng and Jun Cao Zexiang Li Closed-loop identification for motion control system, *8th World Congress on Intelligent Control and Automation.* Jinan, China, 7–9 July 2010, proceedings, 2010, pp. 477–488. DOI: 10.1109/WCICA.2010.5553795
9. Jirgl M., Havlikova M., Bradac Z. The Dynamic Pilot Behavioral Models, *25th DAAAM International Symposium on Intelligent Manufacturing and Automation*. Vienna, Austria, 26–29 November 2014, proceedings, DAAAM, 2014, pp. 1192–1197 https//doi:org/ 10.1016/j.proeng.2015.01.483
10. Petrova Yu. V. Methodology of initial selection of operators by taking visual stochastic information (according to the models of the dynamics of the "visual channel" and the criterion parameters of a person when working on a statistical...): PhD thesis: 05.13.01. Kyiv, 2006, 20 p.
11. Eykhoff P. eds.: Trends and progress in system identification: IFAC series for graduates, research workers and practising engineers, *Elsevier Science & Technology*, 1981, Vol. 1, pp. 239–303
12. Urban Forssell. Closed-loop Identication: Methods, Theory, and Applications. Linkoping Sweden: Department of Electrical Engineering Linkoping University, 1999, 263 p. ISBN 91-7219-432-4. ISSN 0345-7524
13. Blokhin L. N., Osadchiy S. I., Bezkorovainyi Yu. N. Technology of Structural Identification and Subsequent Synthesis of Optimal Stabilization Systems for Unstable Dynamic Objects, *Journal of Automation and Information Sciences.* – 2007, Vol. 39, №11, pp. 57–66.
14. Osadchyi S. I. Technologies and algorithm for structural identification of a mathematical model of a multidimensional moving object with arbitrary dynamics as part of a closed control system, *Collection of scientific works of Kharkiv Air Force University*, 2009, Vol. 4 № 22, pp. 98–101.
15. 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.
16. Azarskov V. N., Blokhin L. N., Zhitetsky L. S., eds.: Blokhin L. N. Methodology of designing optimal systems of stochastic stabilisation, Monograph. K., Book publishing house NAU, 2006, 440 p. ISBN 966-598-325-3.
17. Uicker J. J., Pennock G. R., Shigley J. E. Theory of Machines and Mechanisms. New York, Oxford University Press, 2003, 977 p.
18. Kučera V. 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. Korn G., Korn T. Handbook of Mathematics (for scientists and engineers). Moscow, Nauka, 1977, 831 p.
20. MathWorks. "Cross-Power Spectral Density (cpsd)" [Electronic resource]. Access mode: https://nl.mathworks.com/help/signal/ref/cpsd.html
21. Blokhin L. M., Burichenko M. Y., Bilak N. V. et al. Statistical dynamics of control systems. K., NAU, 2014, 300 p.
22. 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
23. Feldbaum A.A. Fundamentals of the theory of optimal automatic systems. Moscow, Nauka, 1966, 623 p.

Received 23.09.2023.
Accepted 20.11.2023.

УДК 62.505:629.524

ЧАСТОТНИЙ МЕТОД ОПТИМАЛЬНОЇ ІДЕНТИФІКАЦІЇ ЕЛЕМЕНТІВ ЗАМКНЕНОЇ СИСТЕМИ

Осадчий С. І. – д-р техн. наук, професор кафедри конструкції повітряних суден, авіадвигунів та підтримання льотної придатності Льотної академії Національного авіаційного університету, Кропивницький, Україна.

Зозуля В. А. – канд. техн. наук доцент кафедри цифрової економіки та системного аналізу Державного торговельно-економічного університету, Кропивницький, Україна.

Каліч В. М. – канд. техн. наук, професор кафедри автоматизації виробничих процесів Центральноукраїнського національного технічного університету, Кропивницький, Україна.

Тимошенко Г. С. – старший викладач кафедри аеронавігації, метеорології та організації повітряного руху Льотної академії Національного авіаційного університету, Кропивницький, Україна.

АНОТАЦІЯ

Актуальність. Стаття присвячена подоланню протиріч між припущеннями, прийнятими при формулюванні методу ідентифікації, та конструкцією і умовами функціонування замкненої системи керування. У статті здійснено приведення архітектури дворівневої замкненої систем керування до структурної схеми, яка має дві частини. Перша частина поєднує у собі головний контролер та систему зв'язку з головними сенсорами. Друга частина складається з локальної систему керування, кінематичної ланки та головних сенсорів.

Мета роботи. Метою дослідження, результати якого представлені у цій статті, є поширення дії непрямого методу ідентифікації на випадок оцінювання моделі динаміки головного контролера дворівневої замкненої системи керування за даними пасивного експерименту.

Метод. У статті використано метод ідентифікації в частотній області багатовимірних стохастичних систем стабілізації рухомих об'єктів з довільною динамікою. Початкова інформація про зміни сигналів «вхід-вихід» отримана за даними пасивного експерименту під час натурних випробувань, яка спотворена недосконалістю вимірювальних приладів та системи реєстрації.

Результати. Визначено новий метод ідентифікації елементів дворівневої замкненої системи керування, яка функціонує в умовах дії багатовимірних стаціонарних центрованих випадкових впливів.

DOI 10.15588/1607-3274-2023-4-18

Висновки. Обґрунтовані правила дозволяють коректно визначати матриці передатних функцій обраного елемента замкненої системи при виконанні визначеного переліку умов. Проведений аналіз сигналів контурів керування замкнутими системами доводить можливість існування ефекту зміни цих сигналів статистичними засобами навіть за умов дії на систему лише зосереджених стаціонарних вхідних впливів. Виходячи з цього, подальший розвиток досліджень може бути спрямований на подолання таких ефектів.

КЛЮЧОВІ СЛОВА: ідентифікація, матриця передавальних функцій, спектральна щільність, дисперсія похибки, функціонал якості.

ЛІТЕРАТУРА

1. Zadeh L. A. From Circuit Theory to System Theory / L. A. Zadeh // Proc. IRE. – 1962. – Vol. 50. – P. 856–865. DOI: 10.1007/978-3-540-71795-9_4.
2. Teorija i praktika identifikacії 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
3. Trends and progress in system identification: IFAC series for graduates, research workers and practising engineers; eds.: P. Eykhoff / Elsevier Science & Technology, 2013. – 418 p. ISBN-10: 1483116247. ISBN-13: 978-1483116242.
4. Tehnologii konstruirovanija sovremennyh konkurentosposobnyh kompleksov upravlenija stohasticheskim dvizheniem ob’ektov/ [L. N. Blohin, S. I. Osadchij, O. K. Didyk, G. I. Rudjuk]. – Kirovograd: Izdatel’-Lisenko V.F.,2015.– 284s. ISBN 978-617-7197-30-9
5. Hjalmarsson H. System Identification of Complex and structured systems / H. Hjalmarsson // European Journal of Control. – 2009. – Vol. 15, № 3. – P. 275–310.
6. 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.
7. Wang Jianhong. The practical analysis for closed-loop system identification / Wang Jianhong, Ricardo A. Ramirez-Mendoza // Cogent Engineering. – 2020. – Vol. 7, Issue 1. – P. 1–16. DOI:10.1080/23311916.2020.1796895
8. Lei Zhou. Closed-loop identification for motion control system / Lei Zhou, Jiangang Li, Jun Sheng and Jun Cao Zexiang Li // 8th World Congress on Intelligent Control and Automation. Jinan, China, 7–9 July 2010: proceedings. – 2010. – P. 477–488. DOI: 10.1109/WCICA.2010.5553795
9. Jirgl M. The Dynamic Pilot Behavioral Models / Miroslav Jirgl, Marie Havlikova, Zdenek Bradac // 25th DAAAM International Symposium on Intelligent Manufacturing and Automation. Vienna, Austria, 26–29 November 2014: proceedings. – DAAAM, 2014. – pp. 1192–1197 https//doi:org/ 10.1016/j.proeng.2015.01.483
10. Petrova Ju. V. Metodologija pochatkovogo vidboru operatoriv shljahom znimannja vizual’noї stohastichnoї informacії (za modeljami dinamiki “zorovogo kanalu” ta kriterial’nimi parametrami ljudini pri її roboti na statist...): avtoref. dis. … kand. tehn. nauk: 05.13.01 / Ju. V. Petrova: – Kiїv, 2006. – 20 p.
11. Trends and progress in system identification: IFAC series for graduates, research workers and practising engineers; eds.: P. Eykhoff / Elsevier Science & Technology. – 1981. – Vol. 1. – P. 239–303
12. Urban Forssell. Closed-loop Identication: Methods, Theory, and Applications / Urban Forssell. Linkoping Sweden: Department of Electrical Engineering Linkoping University, 1999. – 263 p. ISBN 91-7219-432-4. ISSN 0345-7524
13. Blokhin L. N. Technology of Structural Identification and Subsequent Synthesis of Optimal Stabilization Systems for Unstable Dynamic Objects / L. N. Blokhin, S. I. Osadchiy, Yu. N. Bezkorovainyi// Journal of Automation and Information Sciences. – 2007. – Vol. 39, №11. – P. 57–66.
14. Osadchij S. I. Tehnologiї i algoritm strukturnoї identifikacії matematichnoї modeli bagatovimirnogo ruhomogo ob’єkta z dovil’noju dinamikoju u skladi zamknenoї sistemi upravlinnja / S. I. Osadchij // Zbirnik naukovih prac’ Harkivs’kogo universitetu Povitrjanih sil. – 2009. – Vip. 4, № 22. – P. 98–101.
15. 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. – pp. 128–134.
16. 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.
17. Uicker J. J. Theory of Machines and Mechanisms / J. J. Uicker, G. R. Pennock, J. E. Shigley. – New York : Oxford University Press, 2003. – 977 p.
18. Kučera Vladimír. The H2 control problem: a general transfer-function solution / V. Kučera // International Journal of Control – 2007. – Vol. 80, № 5. – P. 800–815 DOI:10.1080/00207170701203590
19. Korn G. Handbook of Mathematics (for scientists and engineers) / G. Korn, T. Korn. – Moscow : Nauka, 1977. – 831 p.
20. MathWorks. “Cross-Power Spectral Density (cpsd)” [Electronic resource]. – Access mode: https://nl.mathworks.com/help/signal/ref/cpsd.html
21. Statistichna dinamika sistem upravlinnja / [L. M. Blohin, M. Ju. Burichenko, N. V. Bilak et al]. – K. :NAU, 2014. – 300 p.
22. 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
23. Fel’dbaum A. A. Osnovy teorii optimal’nyh avtomaticheskih sistem/ A. A. Fel’dbaum. – M. : Nauka, 1966. – 623 p.

DOI 10.15588/1607-3274-2023-4-18

Наукове видання

Радіоелектроніка,

інформатика,

управління

№ 4/2023

Науковий журнал

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 04.12.2023. Формат 60×84/8.
Папір офс. Різогр. друк. Ум. друк. арк. 23,95.
Тираж 300 прим. Зам. № 975.

69063, м. Запоріжжя, НУ «Запорізька політехніка», друкарня, вул. Жуковського, 64

Свідоцтво суб'єкта видавничої справи
ДК № 6952 від 22.10.2019.