UDC 004.891.032.26:629.7.01.066

# IMAGE CLASSIFIER RESILIENT TO ADVERSARIAL ATTACKS, FAULT INJECTIONS AND CONCEPT DRIFT – MODEL ARCHITECTURE AND TRAINING ALGORITHM

**Moskalenko V. V.** – PhD, Associate Professor, Associate professor of Computer Science department, Sumy State University, Sumy, Ukraine.

**Moskalenko A. S.** – PhD, Senior lecturer of Computer Science department, Sumy State University, Sumy, Ukraine.

**Korobov A. G.** – PhD, Senior lecturer of Computer Science department, Sumy State University, Sumy, Ukraine.

**Zaretsky M. O.** – Postgraduate student of Computer Science department, Sumy State University, Sumy, Ukraine.

## ABSTRACT

**Context.** The problem of image classification algorithms vulnerability to destructive perturbations has not yet been definitively resolved and is quite relevant for safety-critical applications. Therefore, object of research is the process of training and inference for image classifier that functioning under influences of destructive perturbations. The subjects of the research are model architecture and training algorithm of image classifier that provide resilience to adversarial attacks, fault injection attacks and concept drift.

**Objective**. Stated research goal is to develop effective model architecture and training algorithm that provide resilience to adversarial attacks, fault injections and concept drift.

**Method.** New training algorithm which combines self-knowledge distillation, information measure maximization, class distribution compactness and interclass gap maximization, data compression based on discretization of feature representation and semi-supervised learning based on consistency regularization is proposed.

**Results.** The model architecture and training algorithm of image classifier were developed. The obtained classifier was tested on the Cifar10 dataset to evaluate its resilience over an interval of 200 mini-batches with a training and test size of mini-batch equals to 128 examples for such perturbations: adversarial black-box $L_\infty$-attacks with perturbation levels equal to 1, 3, 5 and 10; inversion of one randomly selected bit in a tensor for 10%, 30%, 50% and 60% randomly selected tensors; addition of one new class; real concept drift between a pair of classes. The effect of the feature space dimensionality on the value of the information criterion of the model performance without perturbations and the value of the integral metric of resilience during the exposure to perturbations is considered.

**Conclusions.** The proposed model architecture and learning algorithm provide absorption of part of the disturbing influence, graceful degradation due to hierarchical classes and adaptive computation, and fast adaptation on a limited amount of labeled data. It is shown that adaptive computation saves up to 40% of resources due to early decision-making in the lower sections of the model, but perturbing influence leads to slowing down, which can be considered as graceful degradation. A multi-section structure trained using knowledge self-distillation principles has been shown to provide more than 5% improvement in the value of the integral mectric of resilience compared to an architecture where the decision is made on the last layer of the model. It is observed that the dimensionality of the feature space noticeably affects the resilience to adversarial attacks and can be chosen as a tradeoff between resilience to perturbations and efficiency without perturbations.

**KEYWORDS:** image classification, robustness, resilience, graceful degradation, adversarial attacks, faults injection, concept drift.

## ABBREVIATIONS

CE is a Cross-Entropy Function;

CIFAR is a Canadian Institute for Advanced Research dataset;

CMA-ES is the covariance matrix adaptation evolution strategy optimization algorithm;

CNN is a Convolutional Neural Network;

GAN is a Generative Adversarial Network;

FIFO is a First In, First Out queue organization;

MED is a Median value of array;

IRQ is a Interquartile Range value;

KL is a Kullback-Leibler divergence function.

## NOMENCLATURE

$D_U$ is the unlabeled images for training and testing;

$D_S$ is the labeled images for training and testing;

$n$ is a number of unlabeled examples;

$K$ is a size of set of classes;

$n_k$ is a number of labeled examples of $k$-th class;

$e_{\xi_1}$ is a $\xi_1$-th parameter which impacts on feature representation;

$f_{\xi_2}$ is a $\xi_2$-th parameter which impacts on efficiency of decision rules;

$\alpha_k$ is a false positive rate for $k$-th class;

$\beta_k$ is a false negative rate for $k$-th class;

$D_{1,k}$ is a true positive rate or sensitivity for $k$-th class;

$D_{2,k}$ is a true negative rate or specificity for $k$-th class;

$J$ is a function of information criteria;

$\bar{J}$ is a class-wise averaged value of information-based classifier efficiency criterion;

$\bar{J}_0$ is a performance at normal functioning that introduced for mapping integral metric of resilience to a value between 0 and 1;

DOI 10.15588/1607-3274-2022-3-9

$T_c$ control time period which can be set a priori and estimated as the mean time between adverse events or maximum allowable recovery time;

$G$ a search domain for optimal parameter values;

$T$ a confidence threshold;

$\eta$ a coefficient to regulate tradeoff between performance without perturbation and resilience under perturbations;

$\mu_k$ membership function that represent confidence in the forecast of input sample belonging to the $k$ -th class;

$z_i$ is a binary feature representation of $i$ -th example at the feature extractor output;

$dist(\cdot)$ is a Euclidean Squared distance;

$\bar{z}_k$ is a trainable $k$ -th class prototype;

N is a dimension of high-level feature space;

$r_k$ is a trainable scale factor for radius of hyperspherical decision boundary (container) of $k$ -th class, $r_k \in (0; 1)$ ;

$L_{INF}$ is a loss function based on information criterion;

$H_o$ is a priori entropy for two alternative decision systems;

$H_\gamma$ is a posteriori entropy, which characterizes the residual uncertainty after decision-making;

$TP_k$ is a numbers of true positives for decision rule of $k$ -th class;

$TN_k$ is a numbers of true negatives for decision rule of $k$ -th class;

$FP_k$ is a numbers of false positives for decision rule of $k$ -th class;

$FN_k$ is a numbers of false negatives for decision rule of $k$ -th class;

$\varepsilon$ is a constant added for numerical stability, $\varepsilon = 10^{-6}$ ;

$y_i$ is class labels for $i$-th example after one-hot encoding;

$n_{MB}$ is a size of mini-batch;

$\hat{y}_i$ is the value of the smoothed membership function for the $i$-th sample to each class;

$relu$ is an activation function RELU;

$\odot$ is the component-wise multiplication sign (Hadamard product);

$\bar{d}$ is the averaged value of the normalized distance between the prototypes of the classes;

$\bar{r}$ is the averaged value of the scaling factor of the radius of the class container;

$z'$ is a feature presentation of the first augmented version of the input sample $x_i$ ;

$z''$ is a feature presentation of the second augmented version of the input sample $x_i$ ;

$q_k^{\mu}(\cdot)$ is an assessment of the probability of belonging the feature representation of input image to the $k$ -th class container;

$\tau$ is a temperature parameter that controls the dynamic range of the similarity function;

$q_k^{dist}(\cdot)$ is an assessment of the probability of belonging the feature representation of input image to the to $k$ -th class;

$S$ number of sections of multi-sectional classifier model;

$e$ is a column matrix of ones, $e = [1, 1, ..., 1]^T$ ;

$Hadamard$ is a square matrix whose entries are either +1 or −1 and whose rows are mutually orthogonal;

$\lambda_{INF}$ is a coefficient for regulating the influence of the information criterion based component to the resulting loss;

$\lambda_{CCL}$ is a coefficient for regulating the impact of contrastive-center loss to the resulting loss;

$\lambda_C$ is a coefficient for regulating the impact of average distance between class prototypes and average radius of separate hypersurface class boundaries (container) to the resulting loss;

$\lambda_{FSD}$ is a coefficient for regulating the impact of feature-level self-knowledge distillation to the resulting loss;

$\lambda_{CSD}$ is a coefficient for regulating the impact of classifier-level self-knowledge distillation to the resulting loss;

$\lambda_D$ is a coefficient for regulating the impact of discretization error of feature representation to the resulting loss;

$\lambda_{UCE}^{out}$ is a coefficient for regulating the impact of consistency regularization based on unlabeled examples which hits out of class containers to the resulting loss;

$\lambda_{UCE}^{in}$ is a coefficient for regulating the impact of consistency regularization based on unlabeled examples which hits into class containers to the resulting loss;

$\lambda_{UL2}$ is a regularization coefficient for regulating the impact of Euclidean distance between feature representations from last layer and intermediate sections to the resulting loss.

## INTRODUCTION

Image classification is one of the most widespread tasks in the field of artificial intelligence. Classification analysis of visual objects is often a component of safety-critical applications, such as autopilots of public transport and combat drones and medical diagnostics. It is used in production processes, monitoring traffic flows, inspection of infrastructure and industrial facilities and other similar tasks. Therefore, there is a need to ensure the resilience of artificial intelligence algorithms to destructive perturbations such. In the case of artificial intelligence for

DOI 10.15588/1607-3274-2022-3-9

image classification, specific perturbations such as adversarial attacks or noise, faults or fault injection attacks, as well as concept drift and out-of-distribution increase aleatoric and epistemic uncertainty and its involve a decrease in the productivity of the intellectual algorithm [1–3].

The resilience of the image classifier to perturbations is primarily ensured by achieving robustness for absorption of a certain level of destructive influences and implementing the graceful degradation mechanism to achieve the most effective behavior in conditions of incomplete certainty [1]. Data analysis models need to be continuously improved to take into account the non-stationary environment and new challenges. That is why the ability of the model to quickly recover performance by adapting to destructive effects and improve to increase the efficiency of subsequent adaptations are equally important components of resilience [2]. Recovery and improvement mechanisms are developed within the framework of the continual learning and meta-learning frameworks [4, 5].

Achieving a certain level of resilience is predicated upon the introduction of a certain resource and functional redundancy into the system, but in practice there are always resource constraints [6]. When designing and operating resilient systems taking into account resource constraints, the principles of rational resilience (affordable resilience) are often used. This involves achieving an effective balance between the system's lifecycle costs and the technical characteristics of the its resilience [7]. Researchers are trying to improve the resource efficiency of the inference by using biologically inspired cognitive mechanisms or adaptive computation based on cascade and multi-branch models [8, 9].

Separate components of resilience to certain types of destructive influences have been researched in many scientific papers, but the complex influence of multiple destructive factors at once had still not been considered [1–3]. In addition, machine learning algorithms for classification analysis of images that simultaneously implement such components of resilience as robustness, graceful degradation, recovery and improvement have not yet been proposed. Not all implementations of these components are compatible, especially under resource constraint conditions.

**The object of research** is the process of training and inference for image classifier that functioning under influences of destructive perturbations.

**The subjects of the research** are model architecture and training algorithm of image classifier that provide resilience to adversarial attacks, fault injection attacks and concept drift.

**The research goal** is development an effective model architecture and training algorithm of image classifier that provide resilience to adversarial attacks, fault injections and concept drift.

## 1 PROBLEM STATEMENT

Let $D_U = \{x_j^U \mid j = \overline{1,n}\}$ is set of unlabeled images and $D_S = \{x_{k,\,j}^S \mid k = \overline{1,K};\ j = \overline{1,n_k}\}$ is set of labeled images for classifier training and testing, where $n$ is number of unlabeled examples and $\mathrm{n_k}$ is number of examples of $k$-th class. Is this case class index $\mathrm{m}$ can be composite form for implement hierarchical labeling for class hierarchy. Moreover, the structure of the vector of model parameters is known

$$g =< e_1,..,e_{\xi_1},..,e_{\Xi_1},\ f_1,..,f_{\xi_2},..,f_{\Xi_2} >, \qquad (1)$$
$$\Xi_1 + \Xi_2 = \Xi \ ,$$

In this case, the constraints $R_{\xi_1}(e_1,...,e_{\xi_1},...,e_{\Xi_1}) \le 0$, $R_{\xi_2}(f_1,...,f_{\xi_2},...,f_{\Xi_2}) \le 0$ are impose on parameters. These inequalities may include resource constraints, necessitating the development of resource-efficient algorithms.

It is necessary to find by machine learning an optimal values of parameters $g$ (1) that provide tradeoff between maximum of class-wise averaged value of information-based efficiency criterion $\overline{J}$ and value of integrated metric $R$ for resilience quantification on control time period $T_c$:

$$\overline{J} = \frac{1}{K}\sum_{k=1}^{K} J(\alpha_k, \beta_k, D_{1,k}, D_{2,k}) \qquad (2)$$

$$R = \frac{\frac{1}{|P|}\sum_P \int_{t=0}^{T_c} \overline{J}(t)dt}{\int_{t=0}^{T_c} \overline{J}_0(t)dt}\ , \qquad (3)$$

$$g^* = \arg\max_G \left\{ \eta \overline{J}(g) + (1-\eta) R(g) \right\} . \qquad (4)$$

## 2 REVIEW OF THE LITERATURE

The problem of image representation and image classification analysis remains an active research topic due to its relevance in safety-critical applications which require resilience to challenging operating conditions [2, 10]. Basic principles of system resilience to destructive perturbations are formulated in [6, 7]. These presuppose the existence of mechanisms of perturbation absorption, perturbation detection, graceful degradation, restoration of productivity and improvement. Research [1, 2, 3] studied vulnerability of artificial intelligence algorithms, identifying the following destructive effects: noise and adversarial attacks, faults and fault injection in the environment of intelligent algorithm deployment, concept drift and emergence of novelty, i.e., test examples that out of distribution of training data.

DOI 10.15588/1607-3274-2022-3-9

The ability to absorb destructive perturbations is called robustness. There are many methods and approaches to increase robustness to adversarial attacks. Some researchers separate methods for ensuring robustness to competitive attacks into the following categories : gradient masking methods, robustness optimization methods and methods of detecting adversarial examples [11]. Gradient masking includes some input data preprocessing methods (jpeg compression, random padding and resizing), thermometer encoding, adversarial logit pairing), defensive distillation, randomly choosing a model from a set of models or using dropout, and the use of generative models (ie, PixelDefend [12] and Defense-GAN [13]). However [14] demonstrated inefficiency of gradient masking methods. Robust optimization approach includes adversarial training, regularization methods which minimize the effects of small perturbations of the input (such as Jacobian regularization or L2-distance between feature representations for natural and perturbed samples), and provable defenses (ie, Reluplex algorithm [15]). Finally, yet another approach lies in developing an adversarial examples detector to reject such examples at the input of the main model. However, Carlini and Wagner [16], rigorously demonstrate that the properties of adversarial examples are difficult and resource-intensive to detect. In [11] it was proposed to divide the methods of protection against adversarial attacks into two groups, implementing two separate principles : methods of increasing intra-class compactness and inter-class separation of feature vectors and methods of marginalization or removal of non-robust image features. This work [17] emphasize the possibility for further development of these basic principles and their combination, taking into account other requirements and constraints.

There are three main approaches to ensure robustness to the injection of faults in the computing environment where neural networks are deployed : introduction of explicit redundancy [18], learning algorithm modification [19] and architecture optimization [19]. Faults are understood as accidental or intentional bit flips in memory which stores the weights or the original value of the neuron. The introduction of explicit redundancy is achieved, as a rule, by duplication of critical neurons and synapses, uniform distribution of synaptic weights and removal of unimportant weights and neurons. It is also possible to increase the robustness of the neural network to the injection of faults at the stage of machine learning by adding noise, perturbations or injecting direct faults during training. The same can also be achieved by including a regularization (penalty) term in the performance measure to indirectly incorporate faults in conventional algorithms [20]. Optimizing the architecture to increase robustness means minimizing the maximum error at the output of the neural network for a given number of inverted bits in memory where weights or results of intermediate calculations are stored. Authors of research [20] solved this problem with evolutionary search algorithms or Neural Architecture Search tools. However, architecture optimization is traditionally a very resource-intensive process.

Papers [21, 22] propose methods of domain randomization and adversarial domain augmentation which increase the robustness of the model under bounded data distribution shifts. Domain randomization is the generation of synthetic data with amount of variations large enough so that that real world data is viewed as simply another domain variation [21]. This can include randomization of view angles, textures, shapes, shaders, camera effects, scaling and many other parameters. Adversarial domain augmentation creates multiple augmented domains from the source domain by leveraging adversarial training with relaxed domain discrepancy constraint based on Wasserstein Auto-Encoder [22]. Transfer learning and multi-task learning also reinforce resistance to out-of-distribution perturbations. However, if there is a real concept drift in the data stream, there is a need to detect such a situation and implement reactive mechanisms to adapt [23]. There are studies on adaptation to real concept drift, but the lack of labels for test data or a significant delay in obtaining them remains a challenge.

Adversarial attacks, error injections, concept drift and out-of-distribution examples cannot always be absorbed, so the development of reactive resilience mechanisms, namely graceful degradation, recovery and improvement, remains relevant [2, 6]. The implementation of these mechanisms is often associated with the need to detect the perturbation. The most successful methods of detecting an adversarial and out-of-distribution samples and concept drift are based on the analysis of high-level feature space using a distance-based confidence score or prototype-based classifier [24, 25]. In [25], the mechanism for detecting faults affecting inference is based on the calculation of the reference value of the contrastive loss function on test diagnostic samples of data in the absence of faults. To detect faults, the current value of the contrast loss function for diagnostic data is compared with the reference value. In research [27] is proposed mechanisms of Nested Learning and Hierarchical Classification, particularly useful for the implementation of the mechanism of graceful degradation.

In [28], consider algorithms for adapting models to destructive perturbations, where the principles of active learning or contrastive learning are used to increase the speed of adaptation by reducing the requirement for labeled data in quantities. Semi-supervised learning methods are proposed in [29] for the simultaneous use of both labeled and unlabeled data in order to accelerate adaptation to concept drift. The methods of lifelong learning, which allow to continuously accumulate knowledge from different tasks and improve, as well as different reminder mechanisms helping avoid catastrophic forgetting problem are considered in [5]. Various approaches to the implementation of meta-learning to improve the effectiveness of adaptation are covered in [4]. The paper considers the principle of self-distillation for training neural networks which can implement adaptive

DOI 10.15588/1607-3274-2022-3-9

calculations and speed up the inference mode as the learning efficiency of the lower layers of the neural network grows.

Thus, there are numerous studies of separate principles of resilience of data classification models, but there are virtually no works which consider their coterminous combination. However, in systems analysis, there are studies related to the provision of affordable resilience [7] which are particularly relevant for data analysis systems operating under resource constraints.

## 3 MATERIALS AND METHODS

When building the model, we aim to implement the main characteristics of resilience: robustness, graceful degradation, recovery and improvement. The model is based on the following principles:

– hierarchical labeling and hierarchical classification to implement the principles of graceful degradation by coarsening the prediction with a more abstract class with reasonable confidence when classes at the bottom of the hierarchy are recognized with low confidence level;

– combining the mechanisms of self-knowledge distillation and nested learning to increase the robustness of the model by increasing the informativeness of the feedback for the lower layers at the training stage and accelerate inference by skipping high-level layers for simple samples at inference stage;

– prototype and compact spherical container formation for each class to simplify detection of out-of-distribution samples and concept drift;

– using memory FIFO-buffer with limited size to store labeled and unlabeled data with corresponding values of loss function obtained by inference for implementation diagnostic and recovery mechanism.

These principles should ensure resource-efficiency because the model will have small branches for intermediate decisions, which introduces minimal redundancy, since the main part of the feature extractor body is shared between intermediate classifiers. In addition, the size of the data buffers can be set to an acceptable capacity from the point of view of resource constraints.

Fig. 1 depicts the architecture of the resilient classifier with sectional design. Sections consist of ResBlocks of the well-known ResNet50 architecture. ResNet50 architecture also provided the inspiration for the Bottleneck module, serving to mitigate the impacts between each classifier of the lower sections, and to add distillation knowledge from the high-level feature map to the lower-level feature maps. The output of each section is used to construct a separate classifier. Each classifier receives feedback from the data labels and the last layer. Feedback from the last layer, denoted by a dotted line, ensures the implementation of the principle of self-knowledge distillation.

A set of prototype vectors is constructed for the classification analysis of the feature representation of each section output. Prototype vectors are not fixed, they are determined in the training process together with weights of feature extractor. To implement the graceful degradation principle, prototypes can belong to different levels in the hierarchy according to the hierarchy of labeling. In the example provided, a 2-level hierarchy is used. To increase immunity to noise and implementation of the information bottleneck, we approximate the feature representation to a discrete form, which is why the output of the feature extractor of each section uses the sigmoid layer and the corresponding regularization in the training algorithm.

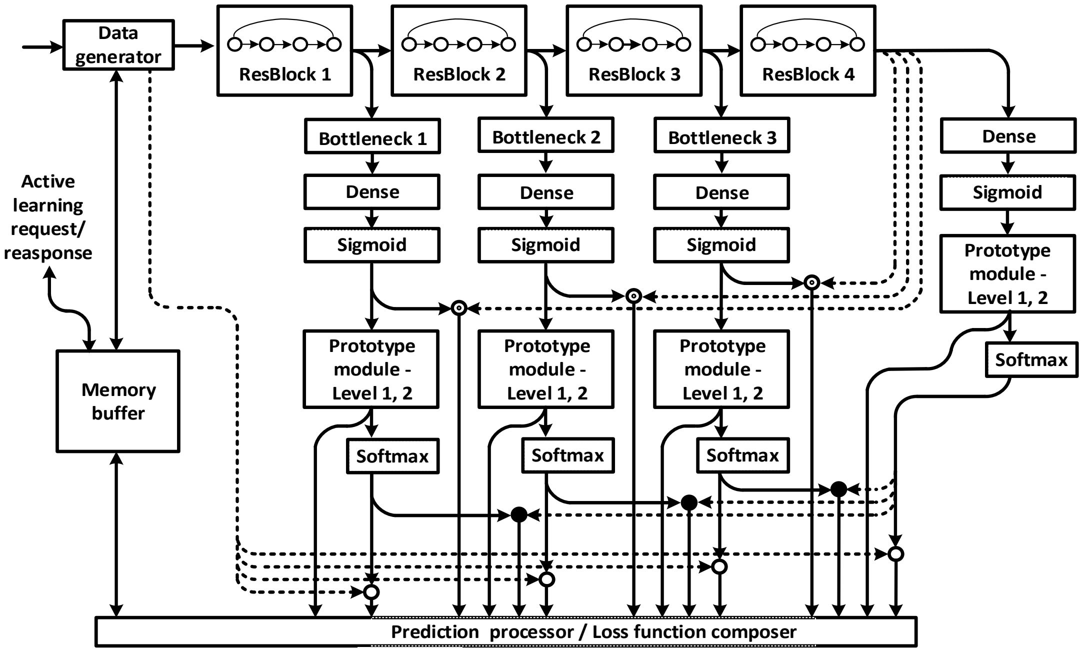

Data generator
ResBlock 1
ResBlock 2
ResBlock 3
ResBlock 4
Bottleneck 1
Bottleneck 2
Bottleneck 3
Dense
Dense
Dense
Dense
Sigmoid
Sigmoid
Sigmoid
Sigmoid
Active learning request/ reasponse
Prototype module - Level 1, 2
Prototype module - Level 1, 2
Prototype module - Level 1, 2
Prototype module - Level 1, 2
Memory buffer
Softmax
Softmax
Softmax
Softmax
Prediction processor / Loss function composer

Figure 1 – Resilient classifier model

DOI 10.15588/1607-3274-2022-3-9

The radius of hyperspherical containers of classes is optimized for each prototypical classifier. Container radii are stored in memory to detect high levels of uncertainty when making decisions. Test samples outside the class containers become candidates for incremental learning using unlabeled samples and trigger a request for manual labeling (active learning) to be performed at a later stage. Controlling for the samples outside the class container can also be used for real concept drift and out-of-distribution detection.

After updating the weights and parameters of the model, the diagnostic dataset and the corresponding value of the loss function must be stored (or updated) in memory. After that, a subset of diagnostic data should be passed along for processing together with the test samples in each batch. This will allow comparison of the past and present values of the loss function to detect errors or injection faults in the memory of the neural network weights. Where the difference between past and present values of the loss function exceeds a certain threshold $\alpha = 0.01$ a neural network fine-tuning algorithm utilizing the diagnostic data needs to be initiated to bring this difference under a threshold $\beta = 0.001$.

Multi-section structure of the model with intermediate classifiers allows implementing adaptive calculations, allowing accelerating the recognition of simple images. At the same time, as the model is continually trained, it becomes faster due to increased recognition confidence of the lower section classifiers. This, in turn, will allow the rest of the high-level sections of the model to be skipped. The following rules for classification analysis in the adaptive calculations framework are proposed:

– Neural network calculations are performed sequentially, section by section;

– high-level sections can be skipped if in the output of the current section the maximal value of the membership function to a particular class of the lower hierarchical level exceeds the confidence threshold $T$;

– if the maximal value of membership function of any of the hierarchical levels of the classifier at the output of the current section has not increased compared to the previous section, then the subsequent calculations can be omitted;

– where any of the conditions of omission of the subsequent sections are fulfilled or the classifier in question is the last classifier in the model and the maximal value of the membership function of the lower hierarchical level does not exceed the confidence threshold, the higher level in the hierarchy is checked;

– where a class with a sufficient confidence level has not been identified, a decision is refused, a request for a manual labeling is generated, and the corresponding sample is designated as suitable for unsupervised tuning.

The confidence in the forecast of $i$ -th sample belonging to the $k$ -th recognition class, is determined by the following membership function

$$\mu_k(z_i) = 1 - \frac{dist(z_i, \overline{z}_k)}{N \cdot r_k}. \tag{5}$$

If maximum value of the function (1) for an input unlabeled sample $z_i$ is less than zero, such a forecast should not be trusted and such sample should be added to the buffer of unlabeled data outside the training distribution. Where the input unlabeled sample falls into one of the containers of the recognition classes (at any of the levels), it should be added to the in-class unlabeled data buffer within the training distribution. Unlabeled sample buffers can be used for training with pseudo-labeling, soft-labeling or for consistency regularization.

Where the model was trained, but in the buffer of the new labeled data an occurrence of $n$ samples of the $c$ -th class misallocated during forward propagation to $k$ -th class container is detected, the real concept drift is recognized.

To avoid catastrophic forgetting in the context of concept drift or emergence of a new recognition class a reminder function is implicitly implemented. Such function is based on unlabeled data buffers and prototypical vectors in feature space, which are changing slowly. Upper layers knowledge distillation mechanism also serves the same purpose.

Data from unlabeled data buffer can be moved to the labeled data queue after the feedback on their actual affiliation with the classes is received. The priority of specific samples being recommended for manual labeling depends on the value of the membership function (1).

During the development of the training algorithm, we aim to ensure the robustness, graceful degradation, recovery and improvement. To this end, the training algorithm will be based on the following principles :

– accounting for the hierarchy of data labeling and hierarchy class prototypes by calculating the loss function separately for each level of the hierarchy to provide graceful degradation at inference;

– implementation of self-knowledge distillation, i.e., distillation of knowledge from the high-level layer (section) of the model down to lower layers (sections) as additional regularization components to increase robustness and provide adaptive calculations in inference mode;

– increasing the compactness of the distribution of classes and the buffer zone between classes to increase resistance to noise, outliers, and adversarial attacks in turn as additional distance-based regularization component;

– penalization of discretization error (compression to binary form) of the feature representation as a way for implementing an information bottleneck to improve the robustness and informativeness of the feature representation;

– implementation of reactive mechanisms for rapid performance recovery under perturbations based on the fine-tuning weights on diagnostic data to eliminate the effects of detected faults, reset (re-initialization) prototypes of drifting or new classes, use of new unlabeled data for consistency regularizing;

– ability to effectively use both labelled and unlabeled data samples to speed up adaptation with a limited quantity of labelled data, which usually comes with a significant lag;

– avoidance of catastrophic forgetting when adapting to perturbations without full retraining by implementing a reminding mechanism utilizing the data buffers, class prototypes and distillation feedback of the upper layers.

The proposed training method consists of two main stages :

– preparatory training the model on labeled and unlabeled data using a semi-supervised regime;

– adaptation to perturbation with semi-supervised supervision and active learning feedback.

The main criterion for learning in both cases is the information measure. The loss function based on the use of the information measure has the form:

$$L_{INF} = 1 - \overline{J} \ . \qquad (6)$$

The normalized modification of C. Shannon's entropy-based information measure is used as the criterion of the recognition efficiency of the $k$ -th class and calculated by the formula [30]

$$J_k = \frac{H_o - H_\gamma}{H_o} =$$

$$= 1 + \frac{1}{2}\left( \frac{\alpha_k}{\alpha_k + D_{2,k}} \log_2 \frac{\alpha_k}{\alpha_k + D_{2,k}} + \right.$$

$$+ \frac{D_{1,k}}{D_{1,k} + \beta_k} \log_2 \frac{D_{1,k}}{D_{1,k} + \beta_k} +$$

$$+ \frac{\beta_k}{D_{1,k} + \beta_k} \log_2 \frac{\beta_k}{D_{1,k} + \beta_k} +$$

$$\left. + \frac{D_{2,k}}{\alpha_k + D_{2,k}} \log_2 \frac{D_{2,k}}{\alpha_k + D_{2,k}} \right). \qquad (7)$$

A separate hyperspherical surface is built for each class in the radial basis feature space. The accuracy characteristics of the hyperspherical decision boundary for each class can be calculated on the basis of statistical tests as follows :

$$D_{1,k} = \frac{TP_k}{TP_k + FN_k + \varepsilon} + \varepsilon \, , \qquad (8)$$

$$D_{2,k} = \frac{TN_k}{TN_k + FP_k + \varepsilon} + \varepsilon \, , \qquad (9)$$

$$\alpha_k = \frac{FN_k}{FN_k + TP_k + \varepsilon} + \varepsilon \, , \qquad (10)$$

$$\beta_k = \frac{FP_k}{FP_k + TN_k + \varepsilon} + \varepsilon \, . \qquad (11)$$

Procedures for calculating statistical tests are not differentiable, so in the training mode their smoothed versions can be used instead [31]

$$TP \approx \sum_{i=1}^{n_{MB}} \hat{y}_i \odot y_i \, , \qquad (12)$$

$$FP \approx \sum_{i=1}^{n_{MB}} \hat{y}_i \odot (1 - y_i) \, , \qquad (13)$$

$$FN \approx \sum_{i=1}^{n_{MB}} (1 - \hat{y}_i) \odot y_i \, , \qquad (14)$$

$$TN \approx \sum_{i=1}^{n_{MB}} (1 - \hat{y}_i) \odot (1 - y_i) \, , \qquad (15)$$

$$\hat{y}_i = \left\{ relu(\mu_{i,k}) \mid k = \overline{1,K} \right\} . \qquad (16)$$

Admissible domain of criterion function (7) is bounded by inequalities $D_{1,k} \ge 0.5$ and $D_{2,k} \ge 0.5$ , or $\beta_k < 0.5$ and $\alpha_k < 0.5$. In order to take into account the admissible domain of function (7) in the optimization procedure based on error backpropagation method it is proposed to perform the following operations when calculating the loss function [30]:

$$D_{1,k} = \max(D_{1,k}, 0.5) \, , \qquad (17)$$

$$D_{2,k} = \max(D_{2,k}, 0.5) \, , \qquad (18)$$

$$\alpha_k = \min(\alpha_k, 0.5) \, , \qquad (19)$$

$$\beta_k = \min(\beta_k, 0.5) \, . \qquad (20)$$

To increase the compactness of class distribution and inter-class gap in feature space it is proposed to use the contrastive-center loss function that calculated for labeled training samples [32]

$$L_{CCL} = \frac{dist(z_i, \bar{z}_{y_i})}{\sum\limits_{k=1, k \ne y_i}^{K} dist(z_i, \bar{z}_k) + 1} \, . \qquad (21)$$

To optimize boundaries of classes it is proposed to use additional regularization component $L_C$ that connects the average distance between class prototypes and the average radius of separate hypersurface class boundaries (container)

$$L_C = \frac{\bar{r}}{\bar{d} + 1} \, , \qquad (22)$$

DOI 10.15588/1607-3274-2022-3-9

$$\bar{d} = \frac{1}{N(K-1)^2}\sum_{k=1}^{K}\sum_{c=1}^{K} dist(\bar{z}_c, \bar{z}_k), \tag{23}$$

$$\bar{r} = \frac{1}{K}\sum_{k=1}^{K} r_k . \tag{24}$$

To speed up adaptation to changes, unlabeled data examples can be used in consistency regularization [29]. In this case, unlabeled data is divided into two groups : unlabeled examples that fall into the class containers; unlabeled examples that out of all class containers.

It is proposed to use unlabeled data that fall into the class containers in regularization component $L_{UCE}^{in}$ which can be calculated by following formulas:

$$L_{UCE}^{in} = CE\left(q^{\mu}(z_i' \ , \tau=1), q^{\mu}(z_i'' \ , \tau=1)\right), \tag{25}$$

$$q_k^{\mu}(z_i \ ,\tau) = \frac{\exp\left(\mu_k(z_i \ )/\tau \ \right)}{\sum_{c=1}^{K}\exp\left(\mu_c(z_i \ )/\tau\right)}. \tag{26}$$

Certain portions γ (<10%) of unlabeled data, which fall into class containers and have maximum values of $q\ (z_i)$, can be pseudo-labeled with the corresponding classes. Such pseudo-labeled data can be included in every mini-batches during training.

Unlabeled examples that out of all class containers may be examples of unknown classes or result of concept drift. In this case, soft-labeling $q_k^{dist}(z_i)$ based on distances to prototype of known classes should be used in consistency regularization component $L_{UCE}^{out}$:

$$L_{UCE}^{out} = CE\left(q^{\mu}(z_i \ ), q^{dist}(z_i \ )\right), \tag{27}$$

$$q_k^{dist}(z_i \ ) = \frac{\exp\left(-dist\left(z_i \ , \bar{z}_k\right)\right)}{\sum_{c=1}^{K}\exp\left(-dist\left(z_i \ , \bar{z}_c\right)\right)}. \tag{28}$$

Consistent regularization can be performed not only at the level of the classification module, but also at the level of features. The corresponding regularization component $L_{UL2}$ of the loss function is calculated by the formula

$$L_{UL2} = dist\left(z_i' \ , z_i'' \ \right). \tag{29}$$

Kullback-Leibler divergence loss $L_{CSD}$ and $L_2$ loss from hints $L_{FSD}$ and calculated based on the $S$-th (last) output of the model and the $s$-th output (intermediate) of the model are used in additionally for self-knowledge distillation

$$L_{FSD} = dist(z_i^s, z_i^S), \tag{30}$$

$$L_{CSD} = KL\left(q^{\mu}(z_i^s, \tau), q^{\mu}(z_i^S, \tau)\right). \tag{31}$$

A regularization component which penalizes the discretization error of feature representation is introduced in addition to implement the information bottleneck [30]

$$L_D = z_i^T(e - z_i). \tag{32}$$

The initial values of the parameters of the lower level class prototypes are initialized on the basis of the Hadamard matrix using the principle of label smoothing. For this first the dimensionality of the Hadamard matrix is determined $N_{Hadamard} = 2^{ceil(\log_2(N))}$, where $ceil()$ is the function rounding a number to a larger integer value. All values less than 0 are replaced by 0, ie $Z = \max\left(0, Hadamard(N_{Hadamard})\right)$ subsequently. As the next step, to facilitate the process of adapting the class prototype to the data structure, the proposed approach uses label smoothing. This is performed according to the formula $Z' = Z*0.7+0.15$, as a result of which the 1's will turn into 0.87, and the 0s into 0.15. K of the first vectors truncated by N first features , ie $\bar{z} = Z'[1:K, 1:N]$ are then selected from the resulting matrix. The trainable scale factor $r_k$ for radius of hyperspherical decision boundary (container) of $k$-th class is initialized with a value of half of Plotkin's Bound, divided by the dimensionality of the feature space

$$r_k \leftarrow \left(\frac{1}{2}\frac{N}{2}\frac{K}{K-1}\right)\frac{1}{N} = \frac{K}{4\cdot(K-1)}. \tag{33}$$

Appearance of a sample with a label indicating a new ($K+1$)-th lower-level class necessitates a formation of a new prototype for the class $\bar{z}_{K+1}$ with the corresponding initial values of the radius scale factor $r_{K+1}$. This is achieved by selecting the nearest vector from the remaining unused rows of a modified Hadamard matrix $Z'$, where the proximity is determined on the basis of Euclidean Squared distance. Initial value of Radius scale factor for the new class is also determined by formula (14), but taking into account the new number of classes.

Each coordinate of the prototype of the upper hierarchical level is initialized by copying the corresponding coordinate of one of the prototypes of the lower level, selected at random. Initial class radius of the upper hierarchical level is determined by formula (14) taking into account the number of classes at this level.

Where a real concept drift is recognized, prototypes of drifting classes are populated with random numbers from the range [0; 1].

The resulting loss function is formed by the sum of the above components, averaged by sections of the model and

DOI 10.15588/1607-3274-2022-3-9

levels of class hierarchy, with coefficients that regulate the impact of individual components depending on the training regime.

The following combined loss function averaged over hierarchical levels and model sections is suggested for supervised learning

$$L_S = \lambda_{INF}\overline{L}_{INF} + \lambda_{CCL}\overline{L}_{CCL} + \lambda_C\overline{L}_C + \lambda_{FSD}\overline{L}_{FSD} + \\ +\lambda_{CSD}\overline{L}_{CSD} + \lambda_D L_D . \quad (34)$$

When new labeled data appear, they are combined with unlabeled data from FIFO-buffer to implement continuous adaptation using the loss function

$$L_{TOTAL} = L_S + \lambda_{UCE}^{out}\overline{L}_{UCE}^{out} + \lambda_{UCE}^{in}\overline{L}_{UCE}^{in} + \lambda_{UL2}\overline{L}_{UL2} . \quad (35)$$

Default values of coefficients are proposed as follows : $\lambda_{INF} = 1.0$ , $\lambda_{CCL} = 1.0$ , $\lambda_C = 0.001$ , $\lambda_{FSD} = 0.01$ , $\lambda_{CSD} = 0.1$ , $\lambda_D = 0.001$ , $\lambda_{UCE}^{out} = 0.1$ , $\lambda_{UCE}^{in} = 0.1$, $\lambda_{UL2} = 0.01$.

## 4 EXPERIMENTS

The Cifar10 dataset was chosen for experimental research because it is publicly available and its images are small in size, which speeds up experimental research. The classes of this dataset can be arranged in a hierarchical structure. For example, the first upper level class will be the animal class, which includes the subclasses bird, cat, deer, dog, frog and horse. The second upper level class will be the vehicle class, which includes airplane, automobile, ship and truck subclasses. Therefore, 12 prototype vectors will be used at the output of the classifier of each section, of which 2 for upper level prototypes and 10 lower level prototypes. For all experiments, the chosen confidence threshold, considered sufficient to make a decision, is $T = 0.8$ . The Cifar10 dataset consists of 50,000 training images and 10,000 test 32x32 color images distributed evenly between 10 classes. For convenience of the analysis for training of base model we will use 70% of training data to form dataset_base, and use the remaining 30% for additional dataset_additional training dataset.

As a result of perturbations, there is a notable decrease in model performance. To test the ability to recover, we define recovery as the state of reaching 95% of the performance level observed prior to perturbation. The control interval is set at $T_C = 200$ to ensure testing on the full volume of test data. During recovery, each test mini-batch is preceded by a training mini-batch. The size of the mini-batch is equal to 128 examples.

To test the model for resistance to faults and the ability to recover, it is suggested to use the TensorFI2 library, which is capable of simulating software and hardware faults. In the experiment, it is proposed to consider the influence of the most difficult to absorb type of faults by generation of random bit inversion (bit-flip injection) in each layer of the model. A fixed share of tensors is randomly selected (fault rate) and 1 bit is randomly selected from them to be inverted. For diagnostics and recovery, along with test data, diagnostic data is added to the input of the model in each mini-batch. Diagnostic data are generated from the dataset_additional set and data quantity is equal to the size of 128 examples.

Different model weights have different importance and impact on model performance. In addition, a fault in the higher bits of tensor value leads to a greater distortion of the results than a fault in the lower bits. Therefore, statistical characteristics should be used to evaluate and compare the model's resilience to different proportions of damaged tensors. The statistical characteristics are derived from a large number of experiments, where bits and tensors for inversion are chosen randomly from a uniform distribution. For simplicity, we can consider the median value (MED) and interquartile value (IRQ) of the integral metric of classifier's resilience for the classes of the upper and lower hierarchical level, calculated after 1000 experiments. We can also consider the influence of the dimensionality of the feature space.

To test the model for resistance to noise and adversarial attacks, it is suggested not to rely on gradients or other features of the model architecture and learning algorithm. Instead testing will be carried out on the basis of black box attacks. To assess the level of disturbances, the resistance to which is tested, it is necessary to choose a metric. In practice, such metrics as L0-norm, L1-norm, L2-norm and L∞-norm have become widespread. However, only L0-norm and L∞-norm impose restrictions on the spatial distribution of noise, which prevents the formation of distorted samples that are incorrectly classified even by humans. In addition, the selection of the perturbation level by the metric L0-norm or L∞-norm does not depend on the size of the image, which is convenient for comparison. Covariance matrix adaptation evolution strategy (CMA-ES) using the $L\infty$ metric [33] is chosen as an evolutionary attack strategy for our experiments. Classifier efficiency measurements are performed on perturbed test samples, with each mini-batch of perturbed test data created on the basis on the actual model. At the same time, mini-batches of perturbed data from the dataset_additional set are created, and 50% of them are provided with data labels for active learning emulation. Perturbed data from the dataset_additional set is not involved in measuring the model's efficiency, but is used to adapt it to disturbances of this type.

Resilience testing to the appearance of new classes and to the concept drift is performed on the classes of lower hierarchical level. Each of the classes will be considered as a new class in turn. Likewise, real concept drift will be examined between any pair of classes.

## 5 RESULTS

Fig. 2 shows an example of model performance recovery curves for classes of the lower hierarchical level with the feature space dimension $N$=64 after fault injection. The vertical axis corresponds to the value of the information criterion averaged over the set of the classes, and the horizontal axis corresponds to the number of test iterations of the trained model on the dataset_base set. The first 50 iterations take place without fault injection, and on the 51st iteration, 4 versions of the model are generated with a different proportion of tensors with an inverted bit in a random position, i.e. $fault\_rate \in \{0,1; 0,3; 0,5; 0,6\}$. Therefore only 4 recovery curves of the model's performance are presented below.

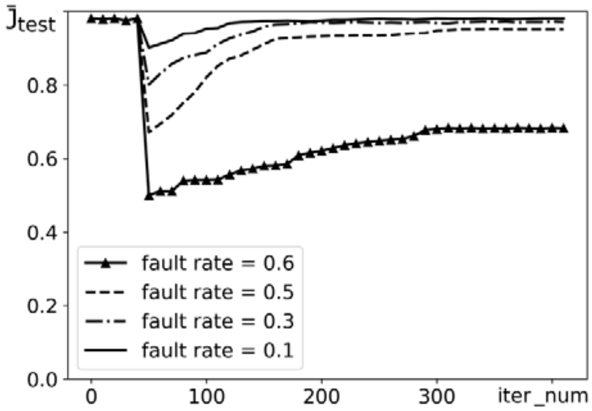


Figure 2 – Example of performance recovery curves after fault injection computed for low-level classes with information measure as performance metric

Table 1 below shows the experimental data after testing the resilience of the model to the faults injection, where $\bar{J}_0$ is the average value of the information criterion before the impact of the fault injection, averaged over the set of the classes, N is the selected dimension of the features. In this case, the table shows the data collected for different hierarchical levels of the model. The hierarchical level number is denoted by the symbol $H$.

Table 1 – Experimental data of model resilience to the faults injection testing

| H | N | fault rate | MED(R) | IRQ(R) | $\bar{J}_0$ |
|---|---|---|---|---|---|
| 1 | 64 | 0.1 | 0.981 | 0.021 | 0.992 |
| 1 | 64 | 0.3 | 0.952 | 0.019 | 0.992 |
| 1 | 64 | 0.5 | 0.883 | 0.020 | 0.992 |
| 1 | 64 | 0.6 | – | – | 0.992 |
| 2 | 64 | 0.1 | 0.978 | 0.022 | 0.978 |
| 2 | 64 | 0.3 | 0.945 | 0.021 | 0.978 |
| 2 | 64 | 0.5 | 0.873 | 0.038 | 0.978 |
| 2 | 64 | 0.6 | – | – | 0.978 |
| 1 | 128 | 0.1 | 0.981 | 0.019 | 0.985 |
| 1 | 128 | 0.3 | 0.955 | 0.018 | 0.985 |
| 1 | 128 | 0.5 | 0.919 | 0.020 | 0.985 |
| 1 | 128 | 0.6 | – | – | 0.985 |
| 2 | 128 | 0.1 | 0.979 | 0.021 | 0.971 |
| 2 | 128 | 0.3 | 0.951 | 0.022 | 0.971 |
| 2 | 128 | 0.5 | 0.880 | 0.019 | 0.971 |
| 2 | 128 | 0.6 | – | – | 0.971 |

Analysis of the table 1 shows that if the share of damaged tensors reaches 60%, it becomes impossible to ensure recovery during processing $T_C$ mini-batches. Fig. 2 shows the performance recovery curves, where the curve corresponding to the damage of 60% of the tensors after 200 iterations does not improve and does not show a recovery of 95% of the performance prior to perturbance. In addition, the analysis of the table 1 shows that increasing the dimensionality of the feature space leads to both a slight decrease in the performance of the model without disturbances, and a slight improvement in the median value of the integral metric of resilience. The corresponding interquartile value of the integral metric of resilience is in the interval [0.01; 0.04].

Fig. 3 shows an example of recovery curves of model performance for classes of the lower hierarchical level with the feature space dimension $N$=64 after the application of adversarial attacks. The vertical axis corresponds to the value of the information criterion averaged over the set of the classes, and the horizontal axis corresponds to the number of iterations of testing the trained model on the dataset_base set. The first 50 iterations are tested without adversarial attacks, and on the 51st iteration, data sets with 4 different threshold values of the disturbance level are generated, i.e. $threshold \in \{1; 3; 5; 10\}$ . Therefore, 4 performance recovery curves are displayed.

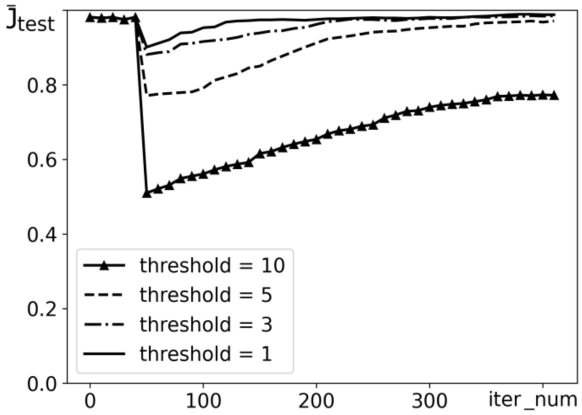


Figure 3 – Example of performance recovery curves after adversarial attack computed for low-level classes with information measure as performance metric

Table 2 shows the result of the experimental testing the model's resilience to adversarial L∞ -attacks.

Analysis of the table 2 shows that if the adversarial perturbation level is less than 10, it becomes impossible to obtain recovery by processing $T_C$ mini-batches. Fig. 3 shows performance recovery curves, where the curve corresponding to a perturbation level of 10 after 200 iterations does not provide 95% performance recovery. In addition, the analysis of the table 2 shows that an increase in the dimensionality of the feature space leads to a slight decrease in the efficiency of the model on unperturbed data, but also to a noticeable improvement in the median value of the integral index of resilience, with corresponding interquartile value of resilience being in the interval [0.01; 0.03]. Therefore, according to formula

© Moskalenko V. V., Moskalenko A. S., Korobov A. G., Zaretsky M. O., 2022
DOI 10.15588/1607-3274-2022-3-9

(4), the dimension of space $N = 128$ is a more optimal compromise option than lower dimension $N = 64$ .

Table 2 – Experimental data of the model resilience to adversarial attacks testing

| H | N | threshold | MED(R) | IRQ(R) | $\overline{J}_0$ |
|---|---|---|---|---|---|
| 1 | 64 | 1 | 0.980 | 0.017 | 0.992 |
| 1 | 64 | 3 | 0.955 | 0.019 | 0.992 |
| 1 | 64 | 5 | 0.885 | 0.017 | 0.992 |
| 1 | 64 | 10 | 0.667 | 0.027 | 0.992 |
| 2 | 64 | 1 | 0.978 | 0.028 | 0.978 |
| 2 | 64 | 3 | 0.954 | 0.021 | 0.978 |
| 2 | 64 | 5 | 0.879 | 0.017 | 0.978 |
| 2 | 64 | 10 | – | – | 0.978 |
| 1 | 128 | 1 | 0.988 | 0.018 | 0.985 |
| 1 | 128 | 3 | 0.967 | 0.018 | 0.985 |
| 1 | 128 | 5 | 0.925 | 0.022 | 0.985 |
| 1 | 128 | 10 | 0.701 | – | 0.985 |
| 2 | 128 | 1 | 0.983 | 0.021 | 0.971 |
| 2 | 128 | 3 | 0.962 | 0.021 | 0.971 |
| 2 | 128 | 5 | 0.905 | 0.026 | 0.971 |
| 2 | 128 | 10 | – | – | 0.971 |

A comparison of the averaged information efficiency criterion and the integral metric of resilience for different hierarchical levels shows that the upper-level classifier is characterized by a lower level of uncertainty and exhibits a higher level of resilience to disturbances, which allows it to be used in graceful degradation mechanisms in case of adversarial attacks.

Fig. 4 shows the performance recovery curve for the worst-case variant of the new class and the worst-case pair of drifting classes in terms of the model's resilience to these perturbations.

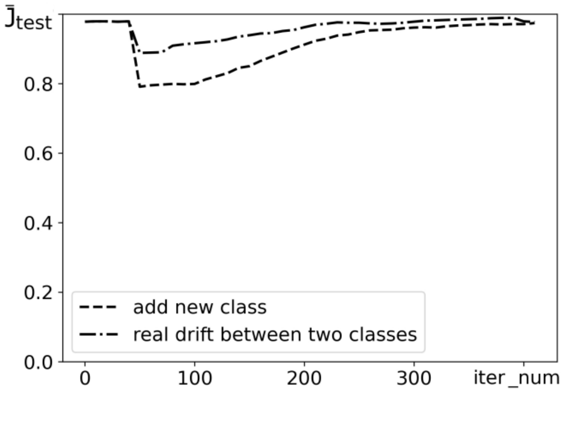


Figure 4 – Worst cases of performance recovery curves after add new class and real concept drift between pair of classes

Analysis of Fig. 4 shows that in both cases the $T_C$ quantity of mini-batches (iterations) was sufficient for recovery. In comparison, learning from scratch required more than 100 times more mini-batches (taking into account 10 learning epochs and a mini-batch size of 128 samples). The worst performing new class from the point of view of the integral metric of resilience was the "bird" class (*R*=0.88). The worst pair of drifting classes from the point of view of the integral metric of resilience were "truck" and "automobile" classes with corresponding *R*=0.95.

Thus, the ability of the proposed algorithm to restore performance after exposure to perturbations has been experimentally proven. Described method of adaptation to adversarial attacks ensures absorption of disturbances of this type and amplitude and ensures performance recovery. Superior efficiency and resilience of the algorithm during the analysis of classes of a higher hierarchical level was also confirmed; this forms the basis for implementation of graceful degradation mechanisms.

## 6 DISCUSSION

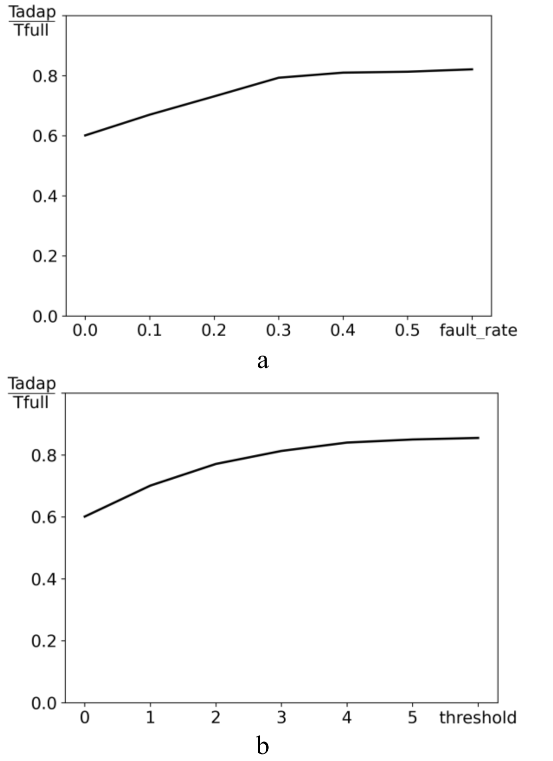
The proposed model of the classifier has a multi-section structure designed to implement adaptive calculations and increase the generalization capabilities of the model due to self-knowledge distillation. Integral metric of model resilience using the outputs of each section and the model using the output of only the last layer of the model are compared to identify the influence of the multi-section structure on the resilience of the model. The model with the feature space dimension $N$ =64 is considered.

Table 3 – Comparison of the integral metric of resilience for the model using the outputs of individual sections and the model with a single output in the last layer

| Only single output | Perturbation | MED(R) | IRQ(R) |
|---|---|---|---|
| True | Fault injection (*fault_rate*=0.3) | 0.891 | 0.034 |
| True | Adversarial attack (*threshold*=3) | 0.912 | 0.053 |
| False | Fault injection (*fault_rate*=0.3) | 0.955 | 0.018 |
| False | Adversarial attack (*threshold*=3) | 0.965 | 0.021 |

Analysis of the table 3 shows that the median value of the integral metric of resilience for the model using the outputs in all sections is 5–6% higher compared to the model with a single output on the last layer.

It is assumed that as the multi-sectional model architecture is trained, its computational efficiency of inference is improved by saving resources on simple examples without perturbations. Fig. 5 shows the dependence of the ratio of the average time spent in the adaptive mode $T_{adap}$ to the time of inference across the entire network $T_{full}$ on the *fault_rate* (Fig. 5a) and maximum amplitude of the adversarial $L_\infty$ -attack (Fig. 5b).



a

b

Figure 5 – Dependence of the average time ratio in the adaptive mode to the time of inference across the entire network on the factor of influence : a – *fault_rate*; b – maximum amplitude of the adversarial $L_\infty$ -attack

Analysis of Fig. 5 confirms the hypotheses that the average inference time increases when the amplitude of the adversarial attack and the frequency of faults increase and vice versa. This can also be considered a mechanism of graceful degradation.

## CONCLUSIONS

The **scientific novelty** of obtained result are the new model architecture and the learning algorithm of a multilayer classifier with the property of resilience to the injection of faults, adversarial attacks, and concept drift.

The model with the proposed architecture has a multi-section structure. At the output of each section, a hierarchy of optimized prototypes and radii of hyperspherical separation boundaries (containers) of classes is built, which ensures the absorption of some part of disturbances and the graceful degradation.

A new learning algorithm that combines ideas and principles of self-knowledge distillation, maximization of compactness of class distribution and interclass buffer zone, discretization of feature representation and consistency regularization is proposed. Self-knowledge distillation is aimed at improving the efficiency of an inference by adaptive computing and the mechanism of graceful degradation. Consistency regularization is carried out both at the level of classification output and at the level of features and is used to increase the robustness and speed of adaptation to destructive perturbations due to the effective use of unlabeled data. At the same time, the main component of the loss function is the information criterion of the classifier's effectiveness, expressed as a functional of smoothed probability estimates for errors of the first and second kind, true positives and true negatives tests.

During testing of the proposed algorithm on the Cifar10 dataset, it was found that if the proportion of damaged tensors reaches 60%, it is not possible to ensure recovery during the processing of mini-batches both for the upper and lower levels of class hierarchy. Similarly, if the adversarial $L_\infty$ -attack perturbation level is 10, it fails to recover during mini-batches processing at the lower class hierarchy level, but for the upper class hierarchy level it is able to achieve 95% recovery of the performance obtained on unperturbed samples. In addition, it was observed that increasing the dimensionality of the feature space leads to a noticeable improvement in the median value of the integral mectric of resilience. At the same time, the interquartile value of the integral metric of resilience is in the interval [0.01; 0.03].

A comparison of the averaged information efficiency criterion and the integral metric of resilience for different class hierarchy levels shows that the upper level of class hierarchy is characterized by a lower level of uncertainty and exhibits a higher level of resilience to disturbances, which allows it to be used in graceful degradation mechanisms under the influence of adversarial attacks.

The median value of the integral metric of resilience of model that uses the outputs of all sections is 5–6% higher compared to the model that has a single output on the last layer. The multi-section structure of the model saves 40% of time on the test dataset, but in the case of perturbation influences, the processing slows down a bit.

The proposed learning algorithms provide adaptation to the appearance of a new class and a real concept drift between a pair of classes in $T_C$ =200 iterations with a mini-batch size of 128 examples. The worst class in the Cifar10 dataset, from the point of view of the integral metric of resilience, if we consider it as a new class, is the "bird" class, for which the value R=0.88 was reached. The worst pair of drifting classes from the point of view of the integral metric of resilience are the "truck" and "automobile" classes, for which the value of *R*=0.95 was reached.

**The practical significance** of the achieved outcomes is formation of a new methodological basis for the development of classification analysis algorithms with resiliece to adversarial attacks, fault injection and concept drift.

The **prospects for further research** are the development of criteria, models, and methods for measuring and certifying the resilience of image classification analysis models.

## ACKNOWLEDGMENTS

The research was concluded in the Intellectual Systems Laboratory of Computer Science Department at

Sumy State University with the financial support of the Ministry of Education and Science of Ukraine in the framework of state budget scientific and research work of DR No. 0122U000782 “Information technology for providing resilience of artificial intelligence systems to protect cyber-physical systems”.

**Contribution of authors :** development of conceptual provisions and methodology of research, development of mathematical model and training algorithm, analysis of research results – V. V. Moskalenko; software development and conducting experiments for testing resilience to faults injection – A. S. Moskalenko; software development and conducting experiments for testing resilience to adversarial attacks – A. G. Korobov; software development and conducting experiments for testing resilience to real concept drift.

## REFERENCES

1. Eigner O., Eresheim S., Kieseberg P., Klausner L., Pirker M., Priebe T., Tjoa S., Marulli F., Mercaldo F. Towards Resilient Artificial Intelligence: Survey and Research Issues, *2021 IEEE International Conference on Cyber Security and Resilience (CSR), Virtual conference, 26–28 July,* 2021, pp. 536–542. DOI: 10.1109/CSR51186.2021.9527986.
2. Olowononi F. O., Rawat D. B., Liu C. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS, *IEEE Communications Surveys Tutorials,* 2021, Vol. 23, No. 1, pp. 524–552. DOI: 10.1109/COMST.2020.3036778.
3. Dymond J. Graceful Degradation and Related Fields, *A review for Applied Research Centre at the Alan Turing Institute*, 2021, pp. 1–32. DOI: 10.48550/arXiv.2106.11119.
4. Hospedales T., Antoniou A., Micaelli P., Storkey A. Meta-Learning in Neural Networks: A Survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2021, 20 p. DOI: 10.1109/TPAMI.2021.3079209.
5. Parisi G., Kemker R., Part J., Kanan C. , Wermter S. Continual lifelong learning with neural networks: A review, *Neural Networks*, 2019, No. 113, P. 54–71. DOI: 10.1016/j.neunet.2019.01.012
6. Fraccascia L., Giannoccaro I., Albino V. Resilience of Complex Systems: State of the Art and Directions for Future Research, *Complexity*, 2018, pp. 1–44. DOI: 10.1155/2018/3421529.
7. Madni A. Affordable Resilience, *Transdisciplinary Systems Engineering*, 2017, pp. 133–159. DOI: 10.1007/978-3-319-62184-5_9.
8. Zhang L., Bao C., Ma K. Self-Distillation: Towards Efficient and Compact Neural Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021, Vol. 44 (8), pp. 4388–4403. DOI: 10.1109/TPAMI.2021.3067100.
9. Marquez E., Hare J., Niranjan M. Deep Cascade Learning, *IEEE Transactions on Neural Networks and Learning Systems,* 2018, Vol. 29(11), pp. 5475–5485. DOI : 10.1109/TNNLS.2018.2805098.
10. Makarichev V., Lukin V., Illiashenko O., Kharchenko V. Digital Image Representation by Atomic Functions: The Compression and Protection of Data for Edge Computing in IoT Systems, *Sensors*, 2022, Vol. 22(10), P. 3751. DOI : 10.3390/s22103751.
11. Smith L. N. A useful taxonomy for adversarial robustness of Neural Networks, *Trends in Computer Science and Information Technology,* 2020, pp. 037–041. DOI: 10.48550/arXiv.1910.10679.
12. Song Y.. Kim T., Nowozin S., Ermon S., Kushman N. PixelDefend: Leveraging Generative Models to Understand and Defend against Advers arial Examples, *Sixth International Conference on Learning Representations, Vancouver CANADA, 30 Apr. –3 May*, 2018, 20 p. DOI: 10.48550/arXiv.1710.10766.
13. Samangouei P., Kabkab M., Chellappa R. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models, *Sixth International Conference on Learning Representations (ICLR 2018), Vancouver CANADA, 30 Apr – 3 May,* 2018, 17 p. DOI: 10.48550/arXiv.1805.06605.
14. Athalye A., Carlini N., Wagner D., Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. [online] arXiv.org, 2022. Access mode: https://arxiv.org/abs/1802.00420 [Accessed 1 June 2022]. DOI: 10.48550/arXiv.1802.00420.
15. Xu J., Li Z., Du B., Zhang M., Liu J. Reluplex made more practical: Leaky ReLU [Text], *IEEE Symposium on Computers and Communications (ISCC), Rennes, France, July 7–July 10 2020,* IEEE, 2022, 7 p. DOI: 10.1109/ISCC50000.2020. 9219587.
16. Carlini N., Wagner D. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas Texas USA, 3 Nov. 2017*. NY, United States, 2017, pp. 3–14. DOI: 10.1145/3128572.3140444.
17. Silva S., Najafirad P. Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey, *IEEE Transactions on Knowledge and Data Engineering*, 2020, 20 p. DOI: 10.48550/arXiv.2007.00753.
18. Huang K., Siegel P. H., Jiang A. Functional Error Correction for Robust Neural Networks, *IEEE Journal on Selected Areas in Information Theory,* 2020, 24 p. DOI: 10.48550/arXiv.2001.03814.
19. Hacene G. B., Leduc-Primeau F., Soussia A. B., Gripon V., Gagnon F. Training modern deep neural networks for memory-fault robustness, *IEEE International Symposium on Circuits and Systems (ISCAS 2019), Sapporo, Hokkaido*. Japan, 26–29 May 2019, 5 p. DOI: 10.1109/ISCAS.2019.8702382.
20. Li W., Ning X., Ge G., Chen X., Wang Y. , Yang H. FTT-NAS: Discovering Fault-Tolerant Neural Architecture, *Proceeding of 25th Asia and South Pacific Design Automation Conference (ASP-DAC)*. Beijing, China, 13–16 Jan. 2020, IEEE Press, pp. 211–216. DOI: 10.1109/ASP-DAC47756.2020.9045324.
21. Valtchev S., Wu J. Domain randomization for neural network classification, *Journal of Big Data*, 2021, Vol. 8, Article No. 94, 12 p. DOI: 10.1186/s40537-021-00455-5.
22. Qiao F., Zhao L., Peng X. Learning to Learn Single Domain Generalization, *Computer Vision and Pattern Recognition,* 2020, pp. 1–13. DOI : 10.48550/arXiv.2003.13216.
23. Priya S., Uthra R. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data, *Complex & Intelligent Systems*, 2021, 17 p. DOI: 10.1007/s40747-021-00456-0.
24. Jiang H., Kim B., Guan M. Y., Gupta M. R. To Trust Or Not To Trust A Classifier, *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 5546–5557. DOI: 10.48550/arXiv.1805.11783.

DOI 10.15588/1607-3274-2022-3-9

25. Shu Y., Shi Y. , Wang Y. , Huang T., Tian Y. P-ODN: Prototype-based Open Deep Network for Open Set Recognition, *Scientific Reports*, 2020, No. 10, Article No. 7146. DOI: 10.1038/s41598-020-63649-6.
26. Wang C., Zhao P., Wang S., Lin X. Detection and recovery against deep neural network fault injection attacks based on contrastive learning, *3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD,* Virtual Event, USA ,14 Aug 2021, 5 p.
27. Achddou R., Di Martino J., Sapiro G. Nested Learning for Multi-Level Classification, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Toronto, Canada, 6–11 June 2021, pp. 2815–2819. DOI: 10.1109/ICASSP39728.2021.9415076.
28. Margatina K., Vernikos G., Barrault L., Aletras N. Active Learning by Acquiring Contrastive Examples, *Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021*, pp. 650–663. DOI: 10.48550/arXiv.2109.03764.
29. Park J., Yun S., Jeong J., Shin J. OpenCoS: Contrastive Semi-supervised Learning for Handling Open-set Unlabeled Data, *International Conference on Learning Representations ICLR Virtual, 3–7 May 2022*, 14 p. DOI: 10.48550/arXiv.2107.08943.
30. Konkle T., Alvarez G. A self-supervised domain-general learning framework for human ventral stream representation, *Nature Communications*, 2022, Vol. 13, Article No. 491, 12 p. DOI: 10.1038/s41467-022-28091-4.
31. Moskalenko V., Zaretskyi M., Moskalenko A., Korobov A., Kovalsky Y. Multi-stage deep learning method with self-supervised pretraining for sewer pipe defects classification, *Radioelectronic and computer systems*, 2021, No. 4, pp. 71–81. DOI: 10.32620/reks.2021.4.06.
32. Li G., Pattabiraman K., DeBardeleben N. TensorFI: A Configurable Fault Injector for TensorFlow Applications, *2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW).* Memphis, TN, USA, 15–18 Oct. 2018, pp. 313–320. DOI: 10.1109/ISSREW.2018.00024.
33. Kotyan S., Vargas D. Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary [Text], *PLOS ONE*, 2022, No. 17(4), Article No. e0265723, 22 p. DOI: 10.1371/journal.pone.0265723.

Received 15.07.2022.
Accepted 21.08.2022.

УДК 004.891.032.26:629.7.01.066

# КЛАСИФІКАТОР ЗОБРАЖЕНЬ ІЗ РЕЗІЛЬЄНТНІСТЮ ДО ПРОТИБОРЧИХ АТАК, ІНЖЕКЦІЇ НЕСПРАВНОСТЕЙ ТА ДРЕЙФУ КОНЦЕПЦІЙ – АРХІТЕКТУРА МОДЕЛІ ТА АЛГОРИТМ НАВЧАННЯ

**Москаленко В. В.** – канд. техн. наук, доцент, доцент кафедри комп’ютерних наук, Сумський державний університет, Суми, Україна.

**Москаленко А. С.** – канд. техн. наук, старший викладач кафедри комп’ютерних наук, Сумський державний університет, Суми, Україна.

**Коробов А. Г.** – канд. техн. наук, старший викладач кафедри комп’ютерних наук, Сумський державний університет, Суми, Україна.

**Зарецький М. О.** – молодший науковий співробітник Лабораторії інтелектуальних систем, Сумський державний університет, Суми, Україна.

## АНОТАЦІЯ

**Актуальність**. Проблема вразливості алгоритмів класифікаційного аналізу зображень до деструктивних збурень досі не була повністю вирішена і є досить актуальною для критичних до безпеки застосувань. Тому об’єктом дослідження є процес навчання та формування рішень для класифікатора зображень, що функціонує під впливом деструктивних збурень. Предметом дослідження є архітектура моделі та алгоритм навчання класифікатора зображень, що забезпечують стійкість до протиборчих атак, інжекції несправностей і дрейфу концепцій.

**Мета дослідження** – є розроблення ефективних архітектури моделі та алгоритму навчання, які забезпечують стійкість до протиборчих атак, інжекції несправностей та дрейфу концепцій.

**Методи дослідження.** Архітектура моделі та алгоритм навчання реалізовані шляхом поєднання ідей і принципів самодистиляції знань, максимізації інформаційної міри та компактності розподілу класів, максимізації міжкласового зазору, стиснення даних на основі дискретизації ознакового подання, а також навчання з частковим залученням учителя на основі регулярізації узгодженості.

**Результати.** Розроблено архітектуру моделі і алгоритм навчання класифікатора зображень. Отриманий класифікатор було випробувано на наборі даних Cifar10 для оцінювання його резільєнтності на інтервалі в 200 міні-пакетів із розміром навчального і тестового міні-пакету в 128 зразків для таких збурень : протиборчі $L^{\infty}$-атаки чорної шухляди з рівнями 1, 3, 5 та 10; інверсія одного випадково обраного біту в тензорі для 10%, 30%, 50% та 60% випадково обраних тензорів; додавання одного нового класу; реальний дрейф концепцій між парою класів. Розглянуто вплив розмірності простору ознак на значення інформаційного критерію ефективності моделі без збурень та на значення інтегрального показника резільєнтності під час впливу збурень.

**Висновки.** Запропоновані архітектура моделі і алгоритм навчання забезпечують поглинання частини збурюючого впливу, витончену деградацію за рахунок ієрархічності класів та адаптивних обчислень, а також швидку адаптацію на обмеженій кількості розмічених даних. Показано, що адаптивні обчислення дозволяють економити до 40% ресурсів за рахунок раннього прийняття рішень на нижніх секціях моделі, однак збурюючий вплив призводить до уповільнення, що можна розглядати як витончену деградацію. Доведено, що багатосекційна структура, що навчається з використанням принципів дистиляції само-знань, забезпечує більш ніж на 5% покращення значення інтегрального показника резільєнтності порівняно з архітектурою, де рішення приймається на останньому шарі моделі. Помічено, що розмірність простору ознак помітно впливає на стійкість до протиборчих атак і може обиратися як компроміс між резільєнтністю до збурень та ефективність без впливу збурень.

DOI 10.15588/1607-3274-2022-3-9

**КЛЮЧОВІ СЛОВА**: класифікація зображень, робастність, резільєнтність, витончена деградація, протиборчі атаки, інжекція несправностей, дрейф концепцій.

УДК 004.891.032.26:629.7.01.066

## КЛАССИФИКАТОР ИЗОБРАЖЕНИЙ С РЕЗИЛЬЕНТНОСТЬЮ К СОСТЯЗАТЕЛЬНЫМ АТАКАМ, ИНЖЕКЦИИ НЕИСПРАВНОСТЕЙ И ДРЕЙФУ КОНЦЕПЦИЙ – АРХИТЕКТУРА МОДЕЛИ И АЛГОРИТМ ОБУЧЕНИЯ

**Москаленко В. В.** – канд. техн. наук, доцент, доцент кафедры компьютерных наук, Сумской государственный университет, Сумы, Украина.

**Москаленко А. С.** – канд. техн. наук, старший преподаватель кафедры компьютерных наук, Сумской государственный университет, Сумы, Украина.

**Коробов А. Г.** – канд. техн. наук, старший преподаватель кафедры компьютерных наук, Сумской государственный университет, Сумы, Украина.

**Зарецький М. О.** – младший научный сотрудник Лаборатории интеллектуальных систем, Сумской государственный университет, Сумы, Украина.

### АННОТАЦИЯ

**Актуальность**. Проблема уязвимости алгоритмов классификационного анализа изображений к деструктивным возмущениям до сих пор не была полностью решена и достаточно актуальна для критических к безопасности применений. Поэтому объектом исследования является процесс обучения и формирования решений классификатора изображений, функционирующем под влиянием деструктивных возмущений. Предметом исследования является архитектура модели и алгоритм обучения классификатора изображений, обеспечивающие устойчивость к состязательным атакам, инжекции неисправностей и дрейфу концепций.

**Цель исследования –** разработка эффективных архитектуры модели и алгоритма обучения, которые обеспечивают устойчивость к противоборствующим атакам, инжекции неисправностей и дрейфа концепций.

**Методы исследования.** Архитектура модели и алгоритм обучения реализуются путем сочетания идей и принципов самодистилляции знаний, максимизации информационной меры и компактности распределения классов, максимизации межклассового зазора, сжатия данных на основе дискретизации признакового представления, а также обучения с частичным привлечением учителя на основе регуляризации согласованности.

**Результаты.** Разработана архитектура модели и алгоритм обучения классификатора изображений. Полученный классификатор был испытан на наборе данных Cifar10 для оценивания его резильентности на интервале в 200 мини-пакетов с размером обучающего и тестового мини-пакета в 128 образцов для таких возмущений: состязательные L∞-атаки чёрного ящика с уровнями 1, 3, 5 и 10; инверсия одного случайно выбранного бита в тензоре для 10%, 30%, 50% и 60% случайно выбранных тензоров; добавление одного нового класса; реальный дрейф концепции между парой классов. Рассмотрено влияние размерности пространства признаков на значение информационного критерия эффективности модели без возмущений и значение интегрального показателя резильентности во время воздействия возмущений.

**Выводы.** Предлагаемые архитектура модели и алгоритм обучения обеспечивают поглощение части возмущающего воздействия, изощренную деградацию за счет иерархичности классов и адаптивных вычислений, а также быструю адаптацию на ограниченном количестве размеченных данных. Показано, что адаптивные вычисления позволяют экономить до 40% ресурсов за счет раннего принятия решений на нижних секциях модели, однако возмущающее влияние приводит к замедлению, что можно рассматривать как изощренную деградацию. Доказано, что многосекционная структура, обучающаяся с использованием принципов самодистилляции знаний, обеспечивает более чем на 5% улучшение значения интегрального показателя резильентности по сравнению с архитектурой, где решение принимается на последнем слое модели. Замечено, что размерность пространства признаков заметно влияет на устойчивость к противоборствующим атакам и может выбираться как компромисс между резильентностью к возмущениям и эффективностью без возмущений.

**КЛЮЧЕВЫЕ СЛОВА:** классификация изображений, робастность, резильентность, утонченная деградация, состязательные атаки, инжекция неисправностей, дрейф концепций.

### ЛІТЕРАТУРА/ЛИТЕРАТУРА

1. Towards Resilient Artificial Intelligence: Survey and Research Issues / [O. Eigner, S. Eresheim, P. Kieseberg et al.] // 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Virtual conference, 26–28 July 2021. – P. 536–542. DOI: 10.1109/CSR51186.2021.9527986.
2. Olowononi F. O. Resilient Machine Learning for Networked Cyber Physical Systems: A Survey for Machine Learning Security to Securing Machine Learning for CPS / F. O. Olowononi, D. B. Rawat, C. Liu // IEEE Communications Surveys Tutorials. – 2021. – Vol. 23, No. 1. – P. 524–552. DOI: 10.1109/COMST.2020.3036778.
3. Dymond J. Graceful Degradation and Related Fields / J. Dymond // A review for Applied Research Centre at the Alan Turing Institute. – 2021. – P. 1–32. – DOI: 10.48550/arXiv.2106.11119.
4. Meta-Learning in Neural Networks: A Survey / [T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey] // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – 20 p. DOI: 10.1109/TPAMI.2021.3079209.
5. Continual lifelong learning with neural networks: A review / [G. Parisi, R. Kemker, J. Part et al] // Neural Networks. – 2019. – No. 113. – P. 54–71. DOI: 10.1016/j.neunet.2019.01.012
6. Fraccascia L. Resilience of Complex Systems: State of the Art and Directions for Future Research / L. Fraccascia, I. Giannoccaro, V. Albino // Complexity. – 2018. – P. 1–44. DOI: 10.1155/2018/3421529.
7. Madni A. Affordable Resilience / A. Madni // Transdisciplinary Systems Engineering. – 2017. – P. 133–159. DOI: 10.1007/978-3-319-62184-5_9.

DOI 10.15588/1607-3274-2022-3-9

8. Zhang L. Self-Distillation: Towards Efficient and Compact Neural Networks / L. Zhang, C. Bao, K. Ma // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2021. – Vol. 44 (8). – P. 4388–4403. DOI: 10.1109/TPAMI.2021.3067100.
9. Marquez E. Deep Cascade Learning / E. Marquez, J. Hare, M. Niranjan // IEEE Transactions on Neural Networks and Learning Systems. – 2018. – Vol. 29(11). – P. 5475–5485. DOI : 10.1109/TNNLS.2018.2805098.
10. Digital Image Representation by Atomic Functions: The Compression and Protection of Data for Edge Computing in IoT Systems / [V. Makarichev, V. Lukin, O.Illiashenko, V. Kharchenko] // Sensors. – 2022. – Vol. 22(10). – P. 3751. DOI : 10.3390/s22103751.
11. Smith L. N. A useful taxonomy for adversarial robustness of Neural Networks / L. N. Smith // Trends in Computer Science and Information Technology. – 2020. – P. 037–041. DOI: 10.48550/arXiv.1910.10679.
12. PixelDefend: Leveraging Generative Models to Understand and Defend against Advers arial Examples / [Y. Song, T. Kim, S. Nowozin et al.] // Sixth International Conference on Learning Representations, Vancouver CANADA, 30 Apr. –3 May, 2018. – 20 p. – DOI: 10.48550/arXiv.1710.10766.
13. Samangouei P. Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models / P. Samangouei, M. Kabkab, R. Chellappa // Sixth International Conference on Learning Representations (ICLR 2018), Vancouver CANADA, 30 Apr – 3 May, 2018. – 17 p. DOI: 10.48550/arXiv.1805.06605.
14. Athalye A. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. [online] arXiv.org / A. Athalye, N. Carlini, D.Wagner, – 2022. – Access mode: https://arxiv.org/abs/1802.00420 [Accessed 1 June 2022]. DOI: 10.48550/arXiv.1802.00420.
15. Reluplex made more practical: Leaky ReLU [Text] / [J. Xu, Z. Li, B. Du et al.] // IEEE Symposium on Computers and Communications (ISCC), Rennes, France, July 7–July 10 2020. – IEEE, 2022. – 7 p. DOI: 10.1109/ISCC50000.2020.9219587.
16. Carlini N. Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods / N. Carlini, D. Wagner // Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas Texas USA, 3 Nov. 2017. – NY, United States, 2017. – P. 3–14. DOI: 10.1145/3128572.3140444.
17. Silva S. Opportunities and Challenges in Deep Learning Adversarial Robustness: A Survey / S. Silva, P. Najafirad // IEEE Transactions on Knowledge and Data Engineering. – 2020. – 20 p. DOI: 10.48550/arXiv.2007.00753.
18. Huang K. Functional Error Correction for Robust Neural Networks / K. Huang, P. H. Siegel, A. Jiang // IEEE Journal on Selected Areas in Information Theory. – 2020. – 24 p. DOI: 10.48550/arXiv.2001.03814.
19. Training modern deep neural networks for memory-fault robustness / [G. B. Hacene, F. Leduc-Primeau, A. B. Soussia et al.] // IEEE International Symposium on Circuits and Systems (ISCAS 2019), Sapporo, Hokkaido, Japan, 26–29 May 2019. – 5 p. DOI: 10.1109/ISCAS.2019.8702382.
20. FTT-NAS: Discovering Fault-Tolerant Neural Architecture / [W. Li, X. Ning, G. Ge et al.] // Proceeding of 25th Asia and South Pacific Design Automation Conference (ASP-DAC), Beijing, China, 13–16 Jan. 2020. – IEEE Press – P. 211–216. DOI: 10.1109/ASP-DAC47756.2020.9045324.
21. Valtchev, S. Domain randomization for neural network classification / S. Valtchev, J. Wu // Journal of Big Data. – 2021. – Vol. 8, Article No. 94. – 12 p. DOI: 10.1186/s40537-021-00455-5.
22. Qiao F. Learning to Learn Single Domain Generalization / F. Qiao, L. Zhao, X. Peng // Computer Vision and Pattern Recognition. – 2020. – P. 1–13. DOI : 10.48550/arXiv.2003.13216.
23. Priya S. Deep learning framework for handling concept drift and class imbalanced complex decision-making on streaming data / S. Priya, R. Uthra // Complex & Intelligent Systems – 2021. – 17 p. DOI: 10.1007/s40747-021-00456-0.
24. To Trust Or Not To Trust A Classifier / [H. Jiang, B. Kim, M. Y. Guan, M. R. Gupta] // Proceedings of the 32nd International Conference on Neural Information Processing Systems. – 2018. – P. 5546–5557. DOI: 10.48550/arXiv.1805.11783.
25. P-ODN: Prototype-based Open Deep Network for Open Set Recognition / [Y. Shu, Y. Shi, Y. Wang et al.] // Scientific Reports. – 2020. – No. 10. – Article No. 7146. DOI: 10.1038/s41598-020-63649-6.
26. Detection and recovery against deep neural network fault injection attacks based on contrastive learning / [C. Wang, P. Zhao, S. Wang, X. Lin] // 3rd Workshop on Adversarial Learning Methods for Machine Learning and Data Mining at KDD, Virtual Event, USA ,14 Aug 2021. – 5 p.
27. Achddou R. Nested Learning for Multi-Level Classification / R. Achddou, J. Di Martino, G. Sapiro // IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, Canada, 6–11 June 2021. – P. 2815–2819. DOI: 10.1109/ICASSP39728.2021.9415076.
28. Active Learning by Acquiring Contrastive Examples / [K. Margatina, G. Vernikos, L. Barrault, N. Aletras] // Conference on Empirical Methods in Natural Language Processing, Online and Punta Cana, Dominican Republic, Nov. 2021. – P. 650–663. DOI: 10.48550/arXiv.2109.03764.
29. OpenCoS: Contrastive Semi-supervised Learning for Handling Open-set Unlabeled Data / [J. Park, S. Yun, J. Jeong, J. Shin] // International Conference on Learning Representations ICLR Virtual, 3–7 May 2022. – 14 p. DOI: 10.48550/arXiv.2107.08943.
30. Konkle T. A self-supervised domain-general learning framework for human ventral stream representation / T. Konkle, G. Alvarez // Nature Communications. – 2022. – Vol. 13, Article No. 491. – 12 p. DOI: 10.1038/s41467-022-28091-4.
31. Multi-stage deep learning method with self-supervised pretraining for sewer pipe defects classification / [V. Moskalenko, M. Zaretskyi, A. Moskalenko et al.] // Radioelectronic and computer systems. – 2021. – No. 4. – P. 71–81. DOI: 10.32620/reks.2021.4.06.
32. Li G. TensorFI: A Configurable Fault Injector for TensorFlow Applications / G. Li, K. Pattabiraman, N. DeBardeleben // 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Memphis, TN, USA, 15–18 Oct. 2018. – P. 313–320. – DOI: 10.1109/ISSREW.2018.00024.\
33. Kotyan, S. Adversarial robustness assessment: Why in evaluation both L0 and L∞ attacks are necessary [Text] / S. Kotyan, D. Vargas // PLOS ONE. – 2022. – No. 17(4), Article No. e0265723. – 22 p. DOI: 10.1371/journal.pone.0265723.

DOI 10.15588/1607-3274-2022-3-9