Methods of information systems protection

UDC 004.732.056 **doi:** https://doi.org/10.20998/2522-9052.2024.1.12

Svitlana Gavrylenko[1], Vadym Poltoratskyi [1], Alina Nechyporenko[2]

[1] National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine
[2] Technical University of Applied Sciences Wildau, Wildau, Germany

INTRUSION DETECTION MODEL BASED ON IMPROVED TRANSFORMER

Abstract. The object of the study is the process of identifying the state of a computer network. **The subject** of the study are the methods of identifying the state of computer networks. **The purpose** of the paper is to improve the efficacy of intrusion detection in computer networks by developing a method based on transformer models. **The results obtained.** The work analyzes traditional machine learning algorithms, deep learning methods and considers the advantages of using transformer models. A method for detecting intrusions in computer networks is proposed. This method differs from known approaches by utilizing the Vision Transformer for Small-size Datasets (ViTSD) deep learning algorithm. The method incorporates procedures to reduce the correlation of input data and transform data into a specific format required for model operations. The developed methods are implemented using Python and the GOOGLE COLAB cloud service with Jupyter Notebook. **Conclusions.** Experiments confirmed the efficiency of the proposed method. The use of the developed method based on the ViTSD algorithm and the data preprocessing procedure increases the model's accuracy to 98.7%. This makes it possible to recommend it for practical use, in order to improve the accuracy of identifying the state of a computer system.

Keywords: data preprocessing; machine learning; neural networks; RNN; SVM; KNN; transformer models; vision transformer.

Introduction

Modern society depends more and more on existing computer technologies. At the same time, cyber threat statistics show a significant increase in the number of intrusions into computer systems and networks. According to recent statistics, cyber-attacks are becoming more frequent and sophisticated, with an estimated 1.4 million new threats emerging every day in 2023. The cost of a successful cyber-attack can range from a few hundred dollars to millions of dollars, depending on the type of attack and the targeted organization. Some of the most common cyber threats include malware, phishing scams, ransomware, and data breaches.

Dynamic changes in the rate of appearance of new classes and families of malicious software, methods of spreading and destruction of anti-virus systems complicate the possibility of detecting malicious software and do not satisfy the existing requirements for the speed and accuracy of detecting malicious software.

Moreover, protection of computer systems and networks from cyber-attacks has become an increasingly urgent problem in recent years. Although improved security features have been added to most systems, there are still a large number of vulnerabilities, including data modification or destruction, unauthorized access to systems and information, etc. In this scenario, intrusion detection systems (IDS) play a key role in protecting the network and accurately identifying attacks on the system.

Literature review and problem statement. Computer network intrusion detection systems are based on the use of machine learning (ML) methods [1, 2]. Traditional machine learning algorithms such as decision trees [3], Random Forest [4], K nearest neighbors [5], and Bayesian methods, recommendation system [6] and Neural Networks [7] have been successfully used in intrusion detection systems for many years. The different ensemble methods, based on meta-algorithms boosting, bagging and stacking are popular too [8]. However, they may not be sufficient for processing of high-dimensional big data generated by modern networks. This is because these methods are not always able to effectively extract useful features from the data. Deep learning is becoming increasingly popular in the area of intrusion detection system. It has to do with deep learning algorithms are better suited for processing big data and complex features than traditional machine learning algorithms. Deep learning-based intrusion detection models commonly use models such as convolutional neural networks (CNN) [9], recurrent neural networks (RNN) [10], long-short-term memory networks (LSTM) [11], and generative adversarial networks (GAN).

Recurrent Neural Networks (RNNs) are one of the most efficient methods for processing sequential data [12]. They include a feedback loop and transmit the current state of the model for the computation of its subsequent state. During network training, gradients may exponentially decrease or increase over time, leading to either vanishing or exploding gradients. This makes it difficult to train the model and update the weights. Moreover, for input sequences of sufficient length, recurrent neural networks become inefficient. In such models, words at the end of the sequence have a greater influence on the result, since the initial information has already passed through recurrent transformations many times. Since important information can contain both at the beginning and in the middle of a sentence, it is important to be able to save it until the end of the network calculations.

Therefore, despite significant progress in intrusion detection using machine learning and deep learning, three key challenges remain: long model training time, unbalanced datasets, and the low efficiency of multiclass

classification. Thus, the improvement and development of new methods for computer network identification are relevant tasks that require further research. In additional, there are many challenges to designing a reliable, efficient, and accurate IDS, especially when dealing with high-dimensional anomalous data with subtle and unpredictable attacks.

The aim and objectives of the study. This paper scrutinizes the possibility of using ViT and ViTSD models to detect intrusions into computer networks. ViTSD models are designed for small datasets, usually include regularization and data augmentation mechanisms, which makes them more robust with a limited number of training examples.

The object of the study is the process of identifying the state of the computer system and network.

The subject of the study are methods of identifying the state of computer system and networks.

The purpose of this study is to develop a method for detecting intrusions in computer networks based on Vision Transformer (ViT) and Improved Vision Transformer for Small-Size Datasets (ViTSD).

To achieve this, it is necessary to complete the following tasks:

– analyse input data and perform their data preprocessing;

– develop a method for increasing the accuracy of identifying the state of computer systems and network through the use of transformer model;

– develop program ViT and ViTSD models;

– analyse an impact of different machine learning methods on the model performance.

Research materials and methods

To enhance the performance of deep learning-based models, an attention mechanism has been developed. The fundamental idea behind the attention mechanism is to allow the model to learn the importance of different tokens in a sequence based on context, rather than assigning a fixed weight to each token. Thus, the model can learn how to pay attention to words that are more significant for the result, regardless of their position in the sentence.

Implementation of the attention mechanism consists of the following steps:

- initially, each element of the input sequence (e.g., a word or token) is transformed into a vector representation using embedding technology – mapping elements into a continuous space of continuous vectors. Then, weights are calculated for each element of the input sequence and the importance of each input element for the given context is determined. These weights are called "attention weights" and are calculated based on the similarity between the current item and all other items in the input sequence;
- the next step involves normalizing the relevance scores so that their sum equals one;
- then the relevance scores and the corresponding values of the input vectors are multiplied. The resulting matrix will have the same dimensions as the original matrix of word embeddings, but the values of the embedding vectors will take into account the context in which the word is situated. The obtained contextual vector (Scaled Dot-Product Attention) serves as a compressed representation of the input data, considering their importance within this context. This contextual vector can be used for further data processing, such as classification, translation, or text generation.

Also, the attention mechanism makes it possible to partially interpret the decisions made by the neural network and understand which parts of the input data are more important.

In the models, usually several "heads of attention" are used, each of which calculates its own value of attention (multi-head attention). This allows the model to focus on different aspects of the data and to detect different relationships between sequence elements and to handle sequences of different lengths.

Transformers are one of the most successful applications of the attention mechanism. The key feature of the architecture is that recurrent layers have been completely removed from the network, leaving only attention mechanisms and auxiliary layers: data normalization layers and fully connected layers. This allowed for parallel processing of sequences, handling all tokens simultaneously rather than token by token.

Transformer models consist of two parts: the encoder and the decoder (Fig. 1).

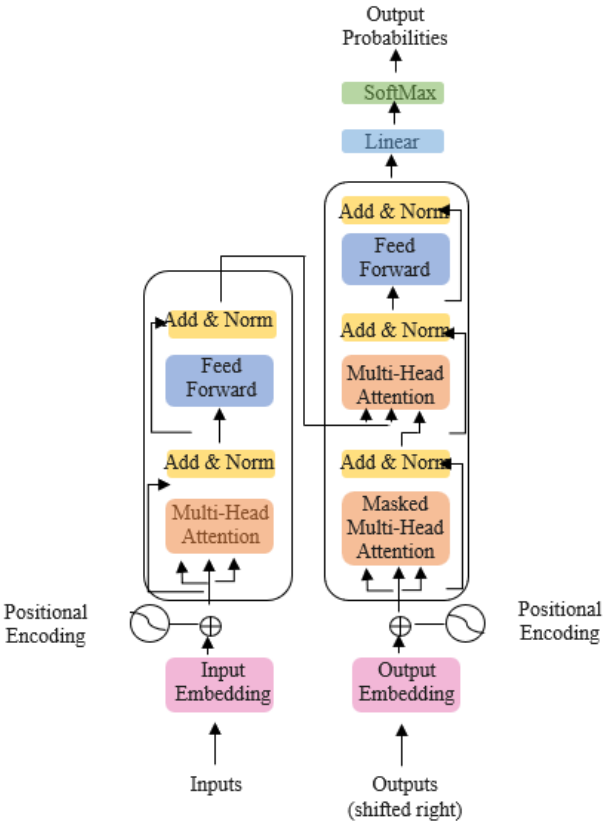

Fig. 1. Transformer model

The encoder is responsible for converting the input data into an internal multidimensional representation and consists of several identical blocks (in the original architecture 6). Each block consists of two layers. The first layer is a multi-head self-attention mechanism, and the second is an ordinary fully connected layer. Residual

connections [13] are employed around each of the two sub-layers, followed by layer normalization [14]. In other words, the output of each sublayer can be found as

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)), \quad (1)$$

where x is input data, LayerNorm is a data normalization function, Sublayer(x) is a function implemented by the sublayer itself. To facilitate these residual connections, all sublayers in the model, as well as embedding layers, generate outputs of size $d_{model} = 512$.

The decoder also consists of several identical decoding units. In addition to the two sublayers in each encoder layer, the decoder inserts a third sublayer that applies multi-head attention to the output of the encoder stack.

Similar to the encoder, residual connections around each of the sublayers are used, followed by layer normalization. The self-attention sub-layer in the decoder stack is also modified to prevent information from future positions influencing the current position. This modification, along with shifting the output embeddings by one position, ensures that the prediction for position i depends only on the known outputs at positions less than i. The internal representation of the encoder and decoder layers is shown in Fig. 2.

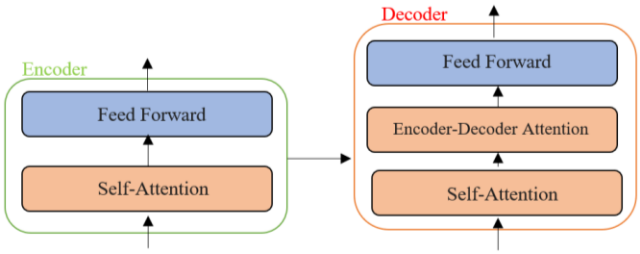

Fig. 2. Internal representation of the encoder and decoder layers

Model transformers are used in computer network intrusion detection systems (Intrusion Detection Systems, IDS) to process and analyze network traffic data and identify potential anomalies or intrusions. They are effective in analyzing network traffic, including data packets, metadata, and logs. Transformer models are able to detect such types of attacks as DDoS attacks, SQL injections, buffer overflow attacks, and others. They can help in segmenting network traffic into different categories such as normal traffic, attacks, port scans, etc. This allows for the analysis of each category separately and taking actions accordingly based on the detected threats. In addition, in IDS systems, transformers can be applied to analyze text data, such as event logs, and identify potential attacks and incidents based on text information. They are also useful in predicting future threats and intrusions based on analysis of historical data and current activity. Model transformers can be trained to perform several tasks simultaneously. For example, they can predict the next network event and simultaneously determine whether the event is an anomaly.

The use of different types of transformer models depends on the specific requirements and characteristics of the task and requires prior adaptation to intrusion detection systems (IDS).

Sequence-to-Sequence Transformer models are typically used to process text sequences and can be adapted to process network logs or other security-related text data for anomaly detection.

Time Series Transformer models, specially designed for working with time series, can be applied in IDS to process time stamps and network events. Attention-based models with attention mechanisms can be useful for detecting unusual patterns in network traffic.

The attention mechanism allows the model to focus on key elements of the sequence. Transformer Autoencoders models can be used to study the representation of "normal" traffic and detect anomalies in deviations from this representation.

If the input data includes different types of information (for example, text logs and numerical metrics), then Multimodal Transformers can handle them effectively. Using pre-trained transformer models such as BERT, GPT, etc. can improve training with small amounts of data when using Transfer Learning with Transformers models.

If the structure of the network is important, then the use of Graph Transformers models is effective. Such models work with graphs and can be used to process and analyze network topology.

For image classification, it is effective to use the Vision Transformer (ViT) [15] model and its improved Vision Transformer for Small-Size Datasets (ViTSD) [16] model, which uses the method of image tokenization with shifted patches (Shifted Patch Tokenization, SPT). This allows capturing more spatial relationships between pixels.

Since Transformer models are trained on labeled data, including normal network traffic and attack data, they serve as an effective tool for continuous network monitoring and detecting potential intrusions. However, the use of Transformer models requires significant computational power and a large amount of data for effective training. Moreover, the effectiveness of Transformer models in intrusion detection systems depends on the nature of the data, the quality of the training data, and the specific detection task that needs to be addressed.

Development of an intrusion detection method based on improved transformer

This work uses the UNSW-NB 15 data set as input, which was developed at the Australian Cyber Security Center (ACCS) Cyber Range Laboratory and contains information about normal network operation and during synthetic intrusions. UNSW-NB15 has 45 features and represents nine main groups of attacks generated by the IXIA tool: PerfectStorm Analysis, Backdoors, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode and Worms.

An important step in building a model is data preprocessing, which can significantly improve its performance. It includes cleaning scaling, balancing and feature extraction.

The analysis of the input data founded a significant number of related features (Fig. 3)[17].

	dur	state	proto	service	sbytes	spkts	dpkts	sttl	dbytes
dur	1.000000	0.079980	-0.053138	-0.104148	0.225432	0.280239	0.217507	-0.000990	0.172492
state	0.079980	1.000000	-0.228909	-0.135552	0.033448	0.050412	0.045430	-0.537155	0.025496
proto	-0.053138	-0.228909	1.000000	-0.239996	-0.012132	-0.033177	-0.040910	0.199680	-0.023490
service	-0.104148	-0.135552	-0.239996	1.000000	0.004752	-0.026797	-0.046527	0.112926	-0.033948
sbytes	0.225432	0.033448	-0.012132	0.004752	1.000000	0.965750	0.175834	-0.017866	0.010036
spkts	0.280239	0.050412	-0.033177	-0.026797	0.965750	1.000000	0.369554	-0.092536	0.198324
dpkts	0.217507	0.045430	-0.040910	-0.046527	0.175834	0.369554	1.000000	-0.163830	0.976419
sttl	-0.000990	-0.537155	0.199680	0.112926	-0.017866	-0.092536	-0.163830	1.000000	-0.114537
dbytes	0.172492	0.025496	-0.023490	-0.033948	0.010036	0.198324	0.976419	-0.114537	1.000000
rate	-0.118032	-0.394435	0.208858	0.286647	-0.025102	-0.068249	-0.083173	0.388155	-0.047978
dload	-0.047033	0.076395	-0.071932	-0.143624	-0.006428	0.074440	0.133835	-0.386224	0.100923
dttl	0.090048	0.295439	-0.221595	-0.378092	0.049891	0.054601	0.036483	-0.033338	0.012537
sload	-0.076344	-0.270630	0.139353	0.002966	-0.015228	-0.044194	-0.054145	0.252901	-0.031266
dinpkt	0.150801	0.076754	-0.025898	-0.059731	-0.001432	-0.003309	-0.007181	-0.006154	-0.007266
sloss	0.240113	0.041305	-0.019758	-0.004532	0.995027	0.973644	0.189060	-0.038088	0.014561
dloss	0.171182	0.029645	-0.030430	-0.037502	0.007091	0.198683	0.981506	-0.137737	0.997109
sinpkt	0.079840	-0.061994	-0.036653	-0.086310	-0.005399	-0.014501	-0.017141	-0.179270	-0.010201

Fig. 3. Correlation matrix

The presence of correlated features negatively affects the quality of the model. However, correlated features can contain useful information about the dependence between variables. When removing one of them, the model loses access to this information and may become less informative. To solve this problem, we previously proposed a special procedure [17]. If there are features that exhibit a correlation higher than a specified threshold (e.g., 90%), they are processed using Principal Component Analysis (PCA). To achieve this, we create data frames with the two features that have the maximum correlation and then apply the Principal Component Analysis (PCA) method. We turn each set into a new feature. After the formation of new feature, we delete the old ones and add new feature to the main data set. The process is repeated until the specified stop criteria are met. Using the proposed procedure made it possible to reduce the number of signs to 31

The ViT and ViTSD models work with images where each pixel is represented by three channel values. Therefore, in this paper was proposed a procedure for converting tabular input data into a special image format, which is necessary for the models to work. After loading the data set, we normalized the values from 0 to 255, according to the RGB format. Further, from the initial data, we form N arrays of size *k* x *k*, where *k* is the number of features and duplicate three times. Using the above procedure, the raw data of UNSW-NB 15 is transformed into a structure (2744, 31, 31, 3). As result we got 2744 "images" of size 31x31x3.

To investigate the effectiveness of ViT and ViTSD models, as well as the proposed feature correlation reduction procedure, software models were developed in the Python environment using Google Colab. The quality of the model was estimated by accuracy, training and testing time of model (Tabl. 1).

Table 1 – **Indicators of quality of ViT and ViTSD models with and without using of the feature correlation reduction procedure**

Quality indicators	Using the method PCA		Without prior data preprocessing	
	ViT	ViTSD	ViT	ViTSD
Accuracy	0.973	**0.987**	0.761	0.952
Training time, s	1056	1176	816	817
Testing time, s	20	14	20	10

As you can see in the table, applying the proposed feature correlation reduction procedure for the ViT method improved the accuracy of the model from 0.761% to 0.973% with the same testing time. For the ViTSD model, the accuracy also increased by 3.5 %, although the testing time also increased.

To evaluate the quality of the proposed Vision Transformer and ViTSD intrusion detection models, classical models based on SVM and KNN methods were investigated.

A comparative analysis of the results of the developed intrusion detection models based on ViT and ViTSD and classical models based on SVM and KNN methods is given in Tabl. 2.

Table 2 – **Results of machine and deep learning models using the data preprocessing procedure**

Quality indicators		Accuracy	Training time, s	Recognition time, s
With the use of data preprocessing procedure	SVM	0.910	107	21
	KNN	0.933	0.06	10
	ViT	0.973	1056	20
	ViTSD	**0.987**	1176	14
Without using the data preprocessing procedure	SVM	0.908	128	18
	KNN	0.931	0.02	12
	ViT	0.761	816	20
	ViTSD	0.952	817	10

As you can see in the table, the best accuracy is achieved by using the Vision Transformer For Small-Size Datasets model with preprocessing of the input data using the proposed feature space reduction procedure.

Conclusions

As part of the research, an analysis of various approaches to improving the quality of computer network intrusion detection models was performed. Traditional machine learning algorithms and deep learning methods have been analysed, and the advantages of using transformer models have been considered. The efficiency of using the ViT and ViTSD models was proved.

To enhance the performance of the model, a procedure for reducing the correlation of input data has been proposed, achieved through the recursive application of Principal Component Analysis (PCA). This has led to a reduction in the size of input data and decrease training time for the model.

The procedure for converting tabular raw data into a special image format, which is necessary for the operation of deep learning models Vision Transformer (ViT) and Vision Transformer for Small-size Datasets (ViTSD), has been developed.

A method for detecting intrusions into computer networks is proposed, which differs from known ones by using the Vision Transformer for Small-size Datasets (ViTSD) deep learning algorithm and a special procedure for reducing the correlation of the input data and converting tabular output data into a special image format.

The UNSW-NB 15 set, which was developed in the Cyber Range Laboratory of the Australian Cyber Security Center (ACCS) and contains information about the normal functioning of the network and during synthetic intrusions, was used as the source data.

To investigate the effectiveness of the proposed methods, software models were developed in the Python environment using Google Colab. The performance of the model was evaluated using accuracy, model training and testing time. A comparative analysis of the proposed method and classic models based on the SVM and KNN methods was performed.

It was found that the use of the developed method based on the ViTSD algorithm and data preprocessing procedure led to an increase in the model's accuracy to 98.7%, which is 22.6% better compared to the ViT method.

REFERENCES

1. Gavrylenko, S., Chelak, V. and Hornostal, O. (2020), “Research of Intelligent Data Analysis Methods for Identification of Computer System State”, *Proceedings of the 30th International Scientific Symposium Metrology and Metrology Assurance* (*MMA*), Sozopol, Bulgaria, pp. 1–5, doi: https://doi.org/10.1109/MMA49863.2020.9254252
2. Semenov, S., Mozhaiev, O., Kuchuk, N., Mozhaiev, M., Tiulieniev, S., Gnusov, Y., Yevstrat, D., Chyrva, Y. and Kuchuk, H. (2022), “Devising a procedure for defining the general criteria of abnormal behavior of a computer system based on the improved criterion of uniformity of input data samples”, *Eastern-European Journal of Enterprise Technologies*, vol. 6(4), pp. 40-49, doi: https://doi.org/10.15587/1729-4061.2022.269128
3. Bhupendra I., Anamika Y. and Atul, S. (2017), “Decision Tree Based Intrusion Detection System for NSL-KDD Dataset, *Conference: International Conference on Information and Communication Technology for Intelligent Systems* (ICTIS 2017), doi: https://doi.org/10.1007/978-3-319-63645-0_23
4. Zhang, J., Zulkernine, M. and Haque, A. (2008), “Random-Forests-Based Network Intrusion Detection Systems”, *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 5, pp. 649–659, doi: https://doi.org/10.1109/TSMCC.2008.923876
5. Liao, Y. and Vemuri, R. (2002), “Use of K-Nearest Neighbor classifier for intrusion detection”, *Computers and Security*, Volume 21, Issue 5, pp. 439–448, doi: https://doi.org/10.1016/S0167-4048(02)00514-X
6. Meleshko, Ye., Drieiev, O., Yakymenko, M. and Lysytsia, D. (2020), “Developing a model of the dynamics of states of a recommendation system under conditions of profile injection attacks”, *Eastern-European Journal of Enterprise Technologies*, vol. 4, no 4(106), pp. 14–24,. doi: https://doi.org/10.15587/1729-4061.2020.209047

7. Yaloveha, V., Podorozhniak, A. and Kuchuk, H. (2022), “Convolutional neural network hyperparameter optimization applied to land cover classification”, *Radioelectronic and Computer Systems*, vol. 1(2022), pp. 115–128, doi: https://doi.org/10.32620/REKS.2022.1.09
8. Gavrylenko, S. and Hornostal, O. (2023), “Application of heterogeneous ensembles in problems of computer system state identification”, *Advanced Information System*, vol. 7, no. 4, pp. 5–12, doi: https://doi.org/10.20998/2522-9052.2023.4.01
9. Wei, W. (2017), “Malware Traffic Classification Using Convolutional Neural Network for Representation Learning”, *Recent Advances in Machine Learning and Applications*, doi: https://doi.org/10.1109/ICOIN.2017.7899588
10. Ashfaq, M. (2021), “HCRNNIDS: Hybrid Convolutional Recurrent Neural Network-Based Network Intrusion Detection System. Processes”, doi: https://doi.org/10.3390/pr9050834
11. Laghrissi, F., Douzi, S. and Douzi, K. (2021), ”Intrusion detection systems using long short-term memory (LSTM)”, *J Big Data*, vol. 8(65), doi: https://doi.org/10.3390/pr9050834
12. Sak, H., Senior, A. and Beaufays, F. (2014), “Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling”, *Interspeech*, pp. 338–342, doi: https://doi.org/10.21437/Interspeech.2014-80
13. Khan, T., Alhussein, M., Aurangzeb, V., Arsalan, K., Naqvi, S. and Nawaz, S. (2020), “Residual Connection-Based Encoder Decoder Network (RCED-Net) for Retinal Vessel Segmentation”, *IEEE Access*, vol. 8, pp. 131257–131272, doi: https://doi.org/10.1109/ACCESS.2020.3008899
14. Ba, J., Kiros, J. and Hinton, G. (2016), “Layer Normalization”, *arXiv*, doi: https://doi.org/10.48550/arXiv.1607.06450
15. Dosovitskiy A., Beyer L., Kolesnikov A., Weissenborn D., Zhai X., Unterthiner T., Mostafa Dehghani, Minderer M., Heigold G., Sylvain Gelly, Uszkoreit, J. and Houlsby, N. (2021), “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”, *Conference paper at ICLR*, *arXiv*, arXiv:2010.11929, doi: https://doi.org/10.48550/arXiv.2010.11929
16. Seung, L., Seunghyun, L. and Byung, S. (2021), “Vision Transformer for Small-Size Datasets”, *arXiv*, arXiv:2112.13492v1 [cs.CV], 27 Dec 2021, doi: https://doi.org/10.48550/arXiv.2112.13492
17. Gavrylenko, S. and Poltoratskyi, V. (2023), “Method of increasing the efficiency of data classification at the account of reducing the correlation of the sign”, *Control, Navigation and Communication Systems*, No. 4 (74), pp. 71–75, doi: https://doi.org/10.26906/SUNZ.2023.4.070

Надійшла (received) 22.11.2023
Прийнята до друку (accepted for publication) 06.02.2024

ВІДОМОСТІ ПРО АВТОРІВ/ ABOUT THE AUTHORS

Гавриленко Світлана Юріївна –доктор технічних наук, професорка, професорка кафедри комп’ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Svitlana Gavrylenko – Doctor of Technical Sciences, Professor, Professor of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: gavrilenko08@gmail.com; ORCID ID: https://orcid.org/0000-0002-6919-0055;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57189042150.

Полторацький Вадим Олександрович – магістрант кафедри комп’ютерної інженерії та програмування, Національний технічний університет "Харківський політехнічний інститут", Харків, Україна;
Vadym Poltoratskyi – master's student of Computer Engineering and Programming Department, National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine;
e-mail: Vadim.poltoratsky@gmail.com; ORCID ID: https://orcid.org/0009-0003-5312-4939.

Нечипоренко Аліна Сергіївна –доктор технічних наук, професорка, професорка кафедри кафедра молекулярної біології та функціональної геноміки, Технічний університет прикладних наук Вільдау, Вільдау, Німеччина;
Alina Nechyporenko – Doctor of Technical Sciences, Professor, Professor of Department of Molecular Biotechnology and Functional Genomics, Technical University of Applied Sciences Wildau, Wildau, Germany;
e-mail: nechyporenko@th-wildau.de; ORCID ID: https://orcid.org/0000-0002-4501-7426;
Scopus ID: https://www.scopus.com/authid/detail.uri?authorId=57189386760.

Модель виявлення вторгнень на основі покращеного трансформера

С. Ю. Гавриленко, В. О. Полторацький, Нечипоренко А. С.

Анотація. Об'єктом дослідження є процес ідентифікації стану комп'ютерної мережі. **Предметом** дослідження є методи ідентифікації стану комп’ютерних мереж. **Метою** статті є підвищення якості виявлення вторгнень у комп’ютерні мережі шляхом розробки методу на основі моделей-трансформерів. **Отримані результати**. У роботі проаналізовано традиційні алгоритми машинного навчання та методи глибокого навчання, розглянуто переваги використання моделей-трансформерів. Запропоновано метод виявлення вторгнень в комп’ютерні мережі, який відрізняється від відомих використанням алгоритму глибокого навчання Vision Transformer for Small-size Datasets (ViTSD), містить процедури зменшення кореляції вихідних даних та перетворення табличних вихідних даних у спеціальний формат, необхідний для роботи моделей. Досліджені методи реалізовані програмно з використанням хмарного сервісу GOOGLE COLAB на основі Jupyter Notebook. **Висновки**. Проведені експерименти підтвердили працездатність запропонованого методу. Отримано, що використання розробленого методу на основі алгоритму ViTSD та процедури попередньої обробки даних надає можливість підвищити точність моделі до 98,7%. Це надає можливість рекомендувати його для практичного використання з метою підвищення точності ідентифікації стану комп’ютерної системи.

Ключові слова: попередня обробка даних; машинне навчання; нейронні мережі; RNN; SVM; KNN; моделі-трансформери; Vision Transformer.