DOI: https://doi.org/10.15276/hait.03.2021.4
UDC 004.056

Detection and classification of network attacks using the deep neural network cascade

Irina M. Shpinareva[1)]
ORCID: https://orcid.org/0000-0001-9208-4923; ishpinareva@gmail.com. Scopus ID: 8532376700
Anastasia A. Yakushina[1)]
ORCID: https://orcid.org/0000-0001-9510-159X; ayakushina07@gmail.com
Lyudmila A.Voloshchuk[1)]
ORCID: https://orcid.org/0000-0002-2510-0038; lavstumbre@gmail.com
Nikolay D. Rudnichenko[2)]
ORCID: https://orcid.org/0000-0002-7343-8076; nickolay.rud@gmail.com. Scopus ID: 57191406873
[1)] Odessa I. I. Mechnikov National University. 2, St. Dvoryanskaya. Odessa, 65026, Ukraine
[2)] Odessa National Polytechnic University. 1, Shevchenko Ave. Odessa, 65044, Ukraine

ABSTRACT

This article shows the relevance of developing a cascade of deep neural networks for detecting and classifying network attacks based on an analysis of the practical use of network intrusion detection systems to protect local computer networks. A cascade of deep neural networks consists of two elements. The first network is a hybrid deep neural network that contains convolutional neural network layers and long short-term memory layers to detect attacks. The second network is a CNN convolutional neural network for classifying the most popular classes of network attacks such as Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. At the stage of tuning and training the cascade of deep neural networks, the selection of hyperparameters was carried out, which made it possible to improve the quality of the model. Among the available public datasets, one of the current UNSW-NB15 datasets was selected, taking into account modern traffic. For the data set under consideration, a data preprocessing technology has been developed. The cascade of deep neural networks was trained, tested, and validated on the UNSW-NB15 dataset. The cascade of deep neural networks was tested on real network traffic, which showed its ability to detect and classify attacks in a computer network. The use of a cascade of deep neural networks, consisting of a hybrid neural network CNN + LSTM and a neural network CNN has improved the accuracy of detecting and classifying attacks in computer networks and reduced the frequency of false alarms in detecting network attacks.

Keywords: Deep learning; NIDS; CNN; LSTM; deep neural networks; hybrid neural networks

***For citation*: Shpinareva I. M., Yakushina A. A., Voloshchuk L. A, Rudnichenko N. D. Detection and classification of network attacks using the deep neural network cascade. *Herald of Advanced Information Technology.* 2021; Vol. 4 No. 3: 244–254. DOI: https://doi.org/10.15276/hait.03.2021.4**

INTRODUCTION

According to the international research on information security [1] by the EY Global Information Security Survey (GISS), the number of attacks of various types on the Internet increased by an average of 10 percent within the period of 2019-2020.

Therefore, for the protection of local computer networks, the so-called network intrusion detection systems (NIDS) are of particular importance. When designing NIDS, two approaches are used, based on misuse detection and anomaly detection [2, 3]. NIDS based on the first approach are no longer sufficient because they are unable to quickly detect new network attacks due to the requirement of frequent updates of the knowledge base (signatures). NIDS based on the second approach do not have this drawback, since they compare the parameters of the observed and normal behavior of the system using deep learning technology [3]. However, low accuracy of anomaly detection and high probability of false positives are the main disadvantages of behavior-based NIDS. The practical use of deep neural networks, consisting of a hierarchy of cascading layers for the detection of anomalies, shows significant results in increasing the accuracy of detecting network attacks and reducing the frequency of false alarms [4]. However, to date, there is no universal deep neural network model for processing large amounts of network traffic data in real-time.

Therefore, the development of a cascade of deep neural networks for detecting and classifying a network attack with high accuracy and low false alarm rate is an urgent scientific and practical task.

ITERATURE REVIEW

The issues of using deep neural networks (DNN) for detecting network attacks have been actively discussed in recent years, while an important aspect of the studied subject area is the assessment of the possibility of practical implementation of the developed models.

In the first papers [5, 6], network attacks were detected by a multilayer feedforward network. Tang et al. [5] propose a DNN model for flow-based anomaly detection. The DNN model consists of one input levels, three hidden layers and one output level. Neural network testing is performed on the NSL-KDD dataset. The authors note that the proposed DNN model detects zero-day attacks and performs better than other machine learning methods. In [7], a cascade of feedforward multilayer neural networks trained and tested on the KDD99 dataset was used to detect and classify attacks. The results of the classification of malicious programs of this NIDS made it possible to achieve 98.46 % accuracy.

Kolosnjaji et al. in [8] show the detection of network attacks by a neural network based on convolutional and recurrent network layers, which makes it possible to obtain the best characteristics for detecting malicious programs. Using their proposed method, they obtain hierarchical feature extraction architecture. This neural network architecture combines the advantages of a convolution operation from a convolutional layer and modeling the sequence of a recurrent network layer. The results of the combined neural network malware classification achieved an accuracy of 89.4 % for the KDD 99 dataset. However, with the gradual increase in the complexity of the network environment, the KDD 99 dataset is outdated and, therefore, it is no longer known how effective these models will be for detecting attacks [21].

Network security protection lies in the detection of not only malware, but also a malicious web shell. Zhang et al. in [9] suggest processing each word with word2vec in HTTP requests. As a result, the web request is presented as a fixed-size matrix. Then, they create a shell classification model based on the CNN structure. Several groups of experiments are performed, and the proposed method shows the best results when compared with the corresponding classical classifiers.

Wang et al. in [10] present a method for classifying malware traffic using CNN, which is tested on the USTC-TRC2016 streaming dataset and has an average classification accuracy of 99 %. Network traffic for CNN is presented as a two-dimensional image. However, the processing of Pcap files for 2D rendering can affect the speed of network packet analysis.

Kim et al. in [11] compare the architectures of the recurrent networks RNN and LSTM, with the help of which network attacks are detected. Both models are trained and tested on the UNSW-NB15 set. The constructed LSTM model has a higher false detection rate while training than RNN.

Le et al. [12] built an LSTM classifier for intrusion attack detection. Their goal is to find the most suitable optimizer for LSTM gradient descent training, where they compare widely used optimization techniques: Adagrad, Adadelta, RMSprop, Adam, Adamax, and Nadam. The NIDS is found to be effective based on the LSTM model with the Nadam optimizer.

For productive detection of network attacks, Liu et al. [13] propose a DNN-based end-to-end discovery method. The authors consider two payload classification models: PL-CNN and PL-RNN. They developed a data preprocessing method that retains enough information while maintaining efficiency. Training and testing was carried out on the DARPA dataset. Today, the basic parameters of the 1998 DARPA dataset do not match the parameters of modern traffic. This makes it doubtful the effectiveness of these networks in detecting an attack in a real network infrastructure.

To improve the overall security of the Internet, HAIXIA HOU et al. [13] propose a network attack detection method based on network LSTM with hierarchical long short-term memory that can study complex sequences of network traffic at different temporal levels. The system is evaluated on the NSL-KDD dataset. The accuracy of multiple classification by KDDTest + and KDDTest-21 is 83.85 % and 69.73 %, respectively. The accuracy is recognized as low for modern methods of detecting attacks.

There are also examples of the joint use of several types of neural networks to obtain better results, in particular, CNN + RNN [14, 16]. It is shown in [18] that hybrid HNS models use two-stage training and show the most actual results [18].

THE AIM AND OBJECTIVES OF THE RESEARCH

The aim of the study is to improve the accuracy of detection and classification of network attacks in computer networks based on the development of a cascade of deep neural networks.

To achieve this goal, the following tasks were solved within the research process:

– the models of popular deep neural networks are analyzed and the need to develop DNN models for solving the problem of detecting and classifying network attacks is shown;

– the available public data sets of network traffic were analyzed and the necessity of structural modification of UNSW-NB15 for use in training and testing of the DNN model was substantiated;

– the technology for preparing a dataset for deep learning has been developed;

– the cascade of two DNNs has been developed, consisting of a hybrid network CNN + LSTM for detecting network attacks and CNN for classifying network attacks;

– the testing and approbation of the developed DNN cascade has been performed on the UNSW-NB15 set and on a real network infrastructure against the TCP SYN Flood attack.

DATASET PREPARATION TECHNOLOGY FOR DEEP LEARNING

To train the DNN cascade for detecting and classifying network attacks based on the analysis of the available public datasets DARPA1998, KDD Cup 99, NSL-KDD, etc., one of the most relevant was chosen – the UNSW-NB15.

The UNSW-NB15 dataset was created using the IXIA PerfectStormtool at the Cyber Range lab of the Australian Cybersecurity Center. The UNSW-NB15 contains real modern and generated (synthetic) models of attack behavior in network traffic [22]. Here are its main characteristics. The UNSW-NB15 contains 2540044 network connection records, of which 55 % are for attacks, the rest for normal traffic. In the UNSW-NB15 database, each record contains 47 signs of network traffic of five types: nominal, integer, numeric, temporary, and binary. To detect anomalies, UNSW-NB15 uses a binary classification and an anomalous connection criterion as a class label, where 0 is “no attack” and 1 is “attack”. The UNSW-NB15 set contains nine classes of attacks: Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worm.

All UNSW-NB15 attributes conditionally belong to five groups [22]:

– *Current Attributes* includes identifier attributes between hosts (for example, client-to-server or server-to-client).

– *Base Attributes* includes attributes that represent the connection of protocols.

– *Content Attributes* encapsulates TCP/IP attributes; they also contain some of the attributes of the http-services.

– *Time Attributes* contain time attributes such as arrival time between packets, start/end time of a packet, and TCP feedback time.

– Additional generated attributes can be divided into two groups: *General Purpose Attributes* in which each attribute has its own purpose, according to the security of the protocol service; and *Attributes of Connections* that are built from streaming information of 100 connection records for a given time.

Training and testing of the developed DNN cascade is carried out on the UNSW-NB15 set and on a real network infrastructure.

When using the UNSW-NB15, it is technologically important to decide which set attributes to use as input for training.

The proposed technology for preparing data for deep learning based on the UNSW-NB15 set includes the following sequence of actions:

1. Removing insignificant attributes. Previously, the attributes “Source IP”, “Source Port”, “Destination IP”, “Destination Port” were excluded from the feature space on the assumption that they can be relatively easily forged by an intruder and should not be taken into account during training. The time attributes stime and ltime (recording start and end time), trans_depth (http request/response transaction depth), res_bdy_len (the size of the uncompressed data content sent from the http server service) have also been removed. The network traffic of the real network infrastructure is captured by the sniffer within 10 seconds and this data may not be complete.

2. Attribute coding: attack names, proto, state. These attributes are string values and are encoded as numeric values. The attack names are encoded so that the classifier can find out the number of the attack class to which each data tuple belongs. Current attributes: proto – indicates the type of protocol, and state – the state and its dependent protocol, e.g. ACC, CLO and CON.

As a result, instead of a single proto attribute, you get attributes of the proto_ [protocol names] type, for example: proto_icmp, proto_arp, proto_ax.25, etc. The state attribute is coded as: state_ACC, state_CLO, state_CON, state_ECO, state_FIN, etc.

3. Normalization of attribute values. In UNSW-NB15, numeric data has a different range, which creates a number of problems when training neural networks. The normalization by linear transformation is performed to compensate for these differences.

4. Generation of additional attributes on the network infrastructure of the home computer network. With the Tshark analyzer, network data is captured within 10 seconds and written to a file. Since the input data is taken from the LAN traffic, the following additional attributes are generated: is_sm_ips_ports, ct_state_ttl, ct_dst_ltm, ct_src_ltm, ct_src_dport_ltm, ct_dst_sport_ltm, ct_dst_src_ltm

As a result of all transformations of the input parameters, the output is a modified UNSW-NB15 dataset containing 176 attributes.

UNSW-NB15 is split into a training set, which is contained in the UNSW_NB15_training-set.csv file and a UNSW_NB15_testing-set.csv test set. The number of records in the training set is 175 341, and in the test set – 82 332 records of various types, attacking and ordinary. The set for testing at work was divided into testing and validation samples in a ratio of 1 to 2.

DEVELOPMENT OF A CASCADE OF DEEP NEURAL NETWORKS

To develop the DNN cascade, the accuracy of the following most common DNN models for solving classification problems was evaluated:

1) multilayer feedforward neural network (MLPs);

2) convolutional neural network (CNN);

3) long short-term memory network (LSTM);

4) hybrid neural network, consisting of layers of a convolutional network and layers of a recurrent neural network LSTM (CNN+LSTM).

Comparative grades of classification were calculated using the following metrics: percentage of correct answers (Accuracy); loss functions (Loss)

Table 1 shows the obtained values of quality metrics, averaged over the results of 50 iterations of cross-validation. The analysis shows that the best classification accuracy on the test set is provided by the CNN and CNN + LSTM DNN models.

For developing the DNN cascade (Fig. 1), consisting of a hybrid CNN + LSTM for attack detection and CNN for its classification both the of Python and Keras were used. Let's consider the functional diagram of the developed cascade. The first neural network CNN + LSTM receives a processed network dataset containing 176 attributes. At the output of the network, we obtain the probability of detecting a network attack, corresponding to a value from 0.5 to 1. Otherwise, we assume that there is no attack. When an attack is detected, a set of network data is sent to the input of the second CNN and the output is the type of attack.

Let's consider the structural features of each of the cascade networks. To detect an attack, it is proposed to use a hybrid CNN + LSTM (Fig. 2). CNN consists of an input layer, two consecutive combinations of a convolution layer and a pooling layer, and a complete pooling layer. Convolution layers contain the ReLU activation function. The pooling layer contains a maximum function. After the layers of the convolutional network, layers of a neural network with long short-term memory are used. At the output of this neural network, a sigmoidal function is selected, which produces values from 0 to 1.

Table 1. **Efficiency of attack detection by various neural network structures**

Neural network	Structure	Loss	Accuracy
MLPs	consists of an input layer, to which a vector containing 176 attributes is fed, two hidden layers containing 128 and 64 neurons, and an output layer, consisting of 1 neuron. In the first three layers the activation function is ReLU, in the output layer it is Sigmoid.	0.649087	0.643238
CNN	consists of an input layer to which a vector containing 176 attributes is fed, two consecutive combinations of the convolution layer and the MaxPooling layer, and two feedforward neural network layers	0.134758	0.933152
LSTM	consists of two LSTM layers, a Dropout layer and two multilayer layers with relu and sigmoid activation functions	0.141581	0.932082
CNN+LSTM	consists of CNN layers and LSTM layers	0.129932	0.945461

Source: **compiled by the authors**

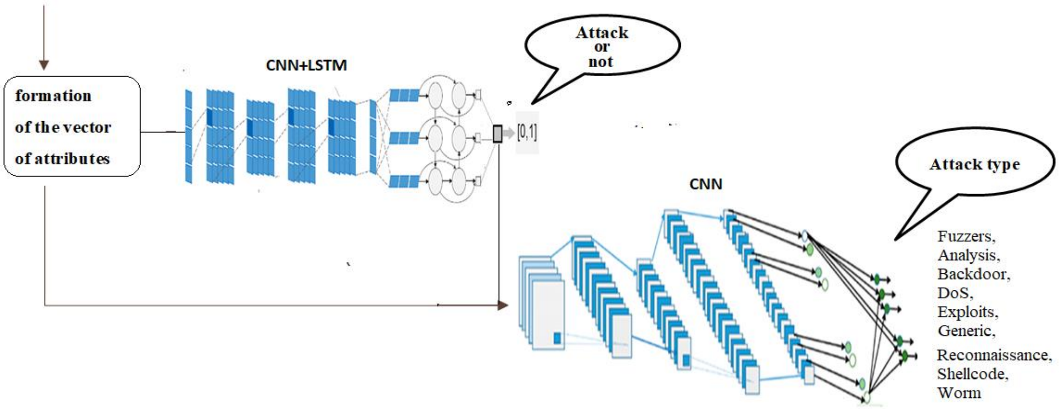

formation of the vector of attributes
CNN+LSTM
Attack or not
[0,1]
CNN
Attack type
Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worm

Fig.1. **The functional scheme of deep neural networks cascade**

Source: **compiled by the authors**

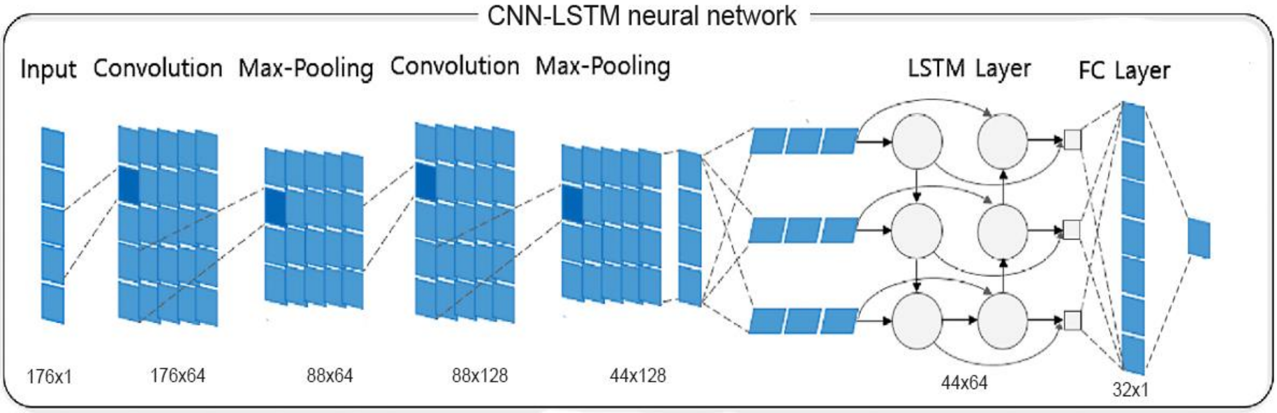


Fig.2. **The structure of CNN+LSTM hybrid neural network**
Source: **compiled by the authors**

The second part of the cascade is CNN for attack classification. This part consists of the input layer, two consecutive combinations of the convolution layer and the pooling layer, and the complete pooling layer (Fig. 3). The convolution layers contain the ReLU activation function. The pooling layer contains a maximum function. At the output of this network, the Softmax function and the output vector of size 9 are selected, where each of the vector elements shows the probability of belonging to a certain class of network attacks: Fuzzers, Analysis, Backdoor, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worm.

A backpropagation algorithm was used to train neural networks. To regulate the parameters of the CNN + LSTM network, an analysis of the optimizers was performed and the Nadam optimizer was selected (Table 2). The binary cross-entropy algorithm was chosen as the loss function.

To train the CNN neural network, the Adam optimizer with categorical cross-entropy as a loss function was used (Table 2).

Thus, a cascade of neural networks consists of a hybrid CNN + LSTM network that detects a network attack and a CNN network that classifies one.

Table 2. **Comparison of optimizers for CNN + LSTM and CNN**

	CNN+LSTM		CNN	
Optimiz-ers	Accuracy	Loss	Accur acy	Loss
Nadam	0.9454	0.1299	0.9318	0.1389
Adam	0.9345	0.1355	0.9454	0.1299
SGD	0.6405	0.6524	0.9315	`0.1390
RMS	0.9350	0.1300	0.9302	0.1478

Source: **compiled by the authors**

TESTING AND APPROBATION OF A CASCADE OF DEEP NEURAL NETWORKS

For binary classification, algorithms from the scikit-learn library were used. Balanced Accuracy (ACCBal) is used as a metric for assessing the accuracy of class classification. Fig. 4 shows the binary classification confusion matrix of the CNN + LSTM model plotted for the test dataset. It can be seen that the CNN + LSTM model provides 0.0681 false negative and 0.0615 false positives ops.

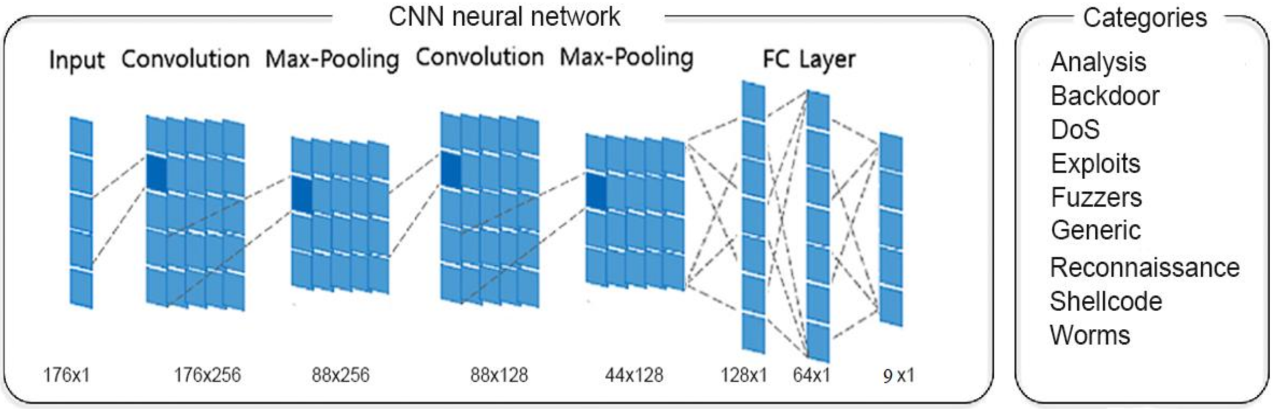


Fig.3. **Структура глубинной нейронной сети CNN**
Source: **compiled by the authors**

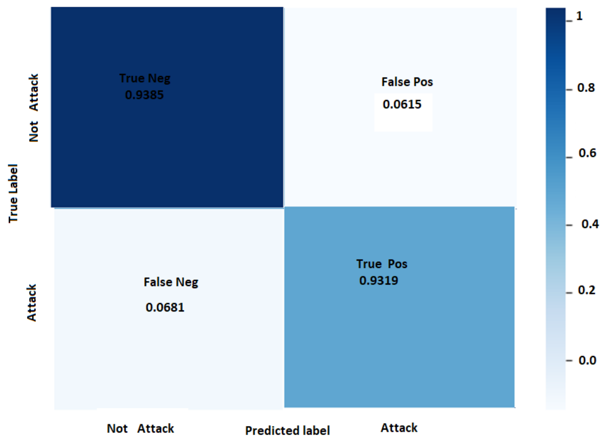


Fig.4. **The balanced confusion matrix of binary classification for CNN + LSTM**
Source: **compiled by the authors**

For the multiclass classification of network attacks, the regular traffic records were removed from the UNSW-NB15 dataset since the purpose of the classification in the second step is to clarify the type of attack. There are 9 classes of attacks in UNSW-NB15. For multiclass classification, the same algorithms from the scikit-learn library were used.

Analysis of Fig. 5 lead to the following conclusions:

a) CNN model accurately detects network attacks Fuzzers (93 %), Exploits (88 %), Generic (86%), DoS (83 %), slightly worse – attacks Reconnaissance (73 %), Shellcode (69 %), and much worse – attacks by Worms (57 %);

b) the model quite often considers that the attack belongs to the Exploits class instead of the real class, Analysis (48 %), Backdoor (47 %), Worms (26 %), Reconnaissance (20 %), Shellcode (15 %).

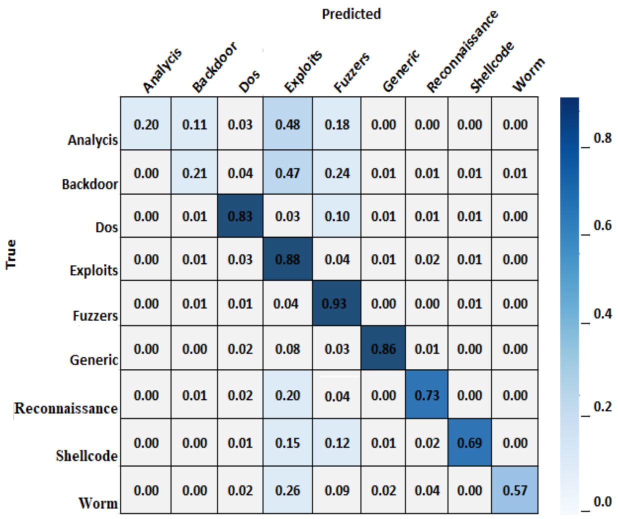


Fig.5. T**he balanced confusion matrix of multiclass classification for CNN**
Source: **compiled by the authors**

The values of the Accuracy indicators, Loss function, Precision and Recall of the CNN + LSTM neural network model for detecting a network attack during testing depending on the number of learning epochs (50 epochs in total) are shown in Figures 6 and 7, respectively. The figures show that during CNN + LSTM training, the accuracy reaches 0.935, the loss function grows up to 0.13, Precision – up to 0.955 and Recall – up to 0.942, and during approbation, the Accuracy indices increase to 0.94 and Precision to 0.97 and the loss function indices decrease to 0.12 and Recall to 0.94. This means that the CNN + LSTM model detects network attacks with an accuracy of 0.97, but there is a problem with an unbalanced dataset (55 % refers to network attacks and 45% to normal traffic).

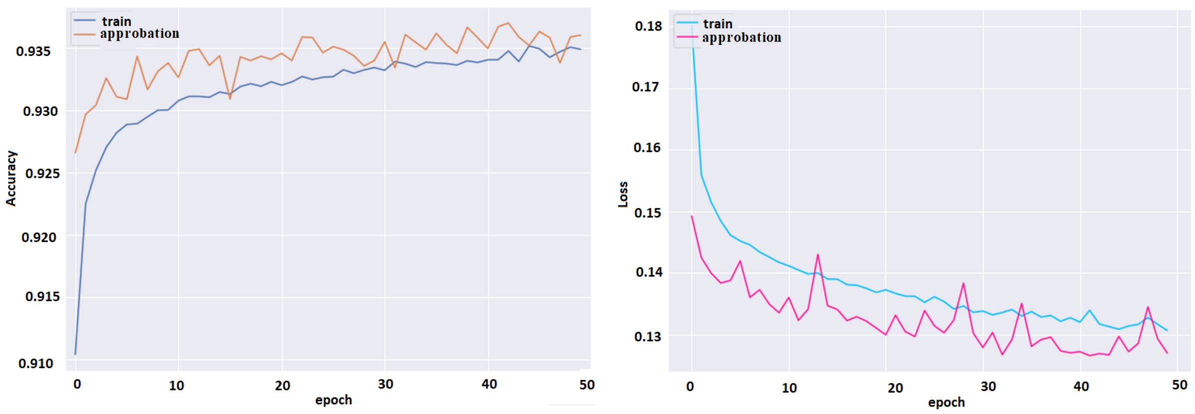


Fig. 6. **Accuracy and loss function values during CNN + LSTM deep neural networks approbation for attack detection**
Source: **compiled by the authors**

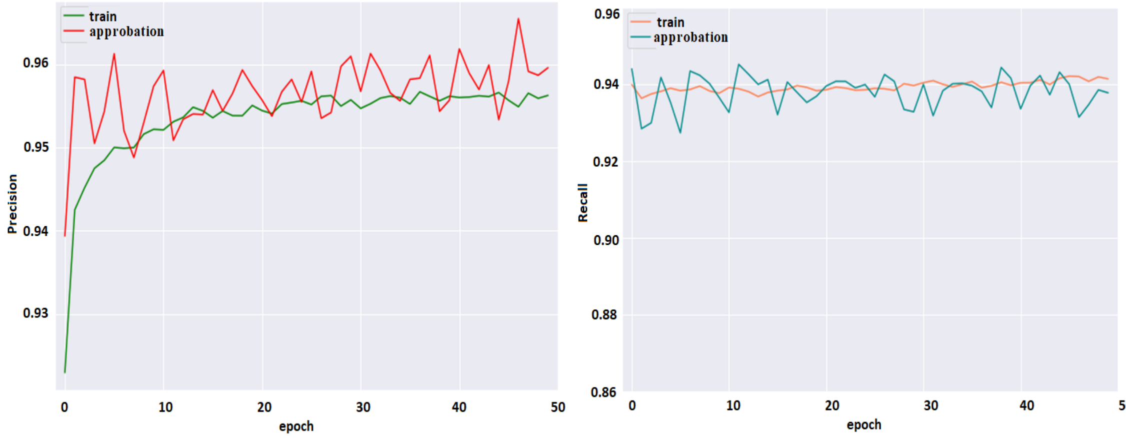


Fig. 7. **Precision and Recall values during approbation CNN + LSTM neural network of the attack detection module**

***Source:* compiled by the authors**

The values of the indicators Accuracy, Loss function, Precision, Recall, F-measure of the CNN neural network model when classifying a network attack while testing are shown in Fig. 8, Fig. 9 and Fig. 10, respectively. The figures show that Accuracy (0.92), Precision (0.93), Recall (0.945) and F-measure (0.918) increase respectively up to mentioned values, and Loss function decreases. From this we can conclude that the CNN model classifies network attacks with an accuracy of 0.93, but also depends on the balance of the dataset.

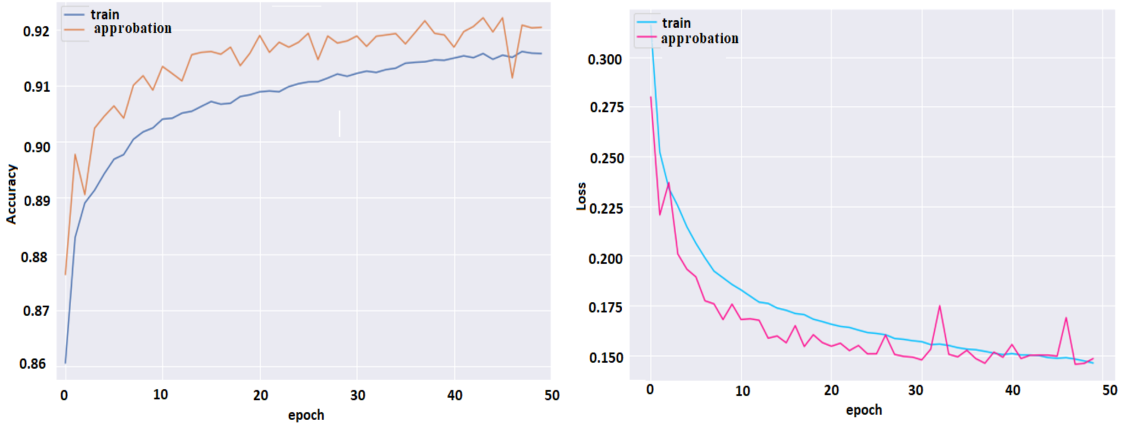


Fig. 8. **Accuracy and loss function values during approbation CNN neural network for attack classification**

***Source:* compiled by the authors**

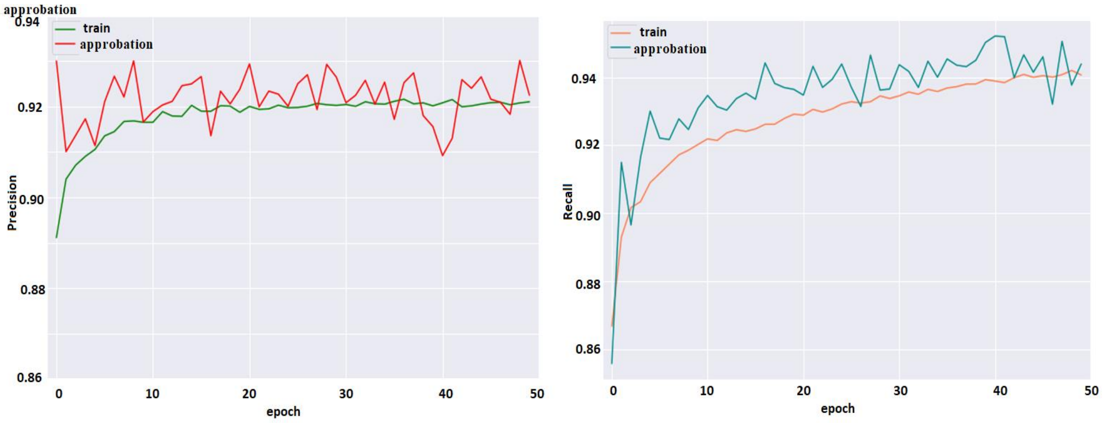


Fig. 9. **Precision and Recall values during approbation CNN neural network of the attack classification module**

***Source:* compiled by the authors**

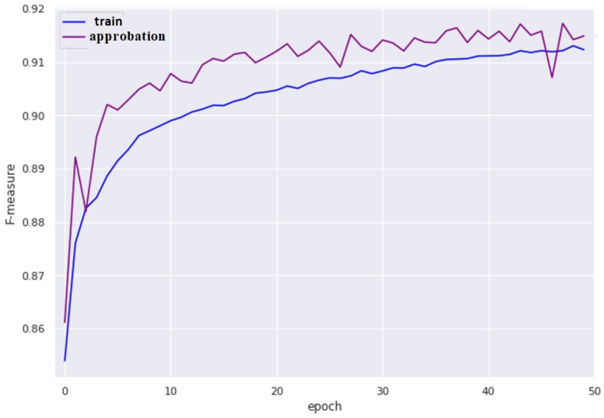


Fig.10. **The value of the F-measure during approbation of the CNN neural network of the attack classification module**
***Source:* compiled by the authors**

To test a cascade of DNN in a computer network, a Tshark analyzer was used. The analyzer allows you to intercept network traffic and write to a Pcap file. Network traffic is captured within 10 seconds. From the .pcap file using the argus utility, the data is written to the .argus file. Then the data from the file is passed to the subprocess.run preprocessor for further processing. A set of attributes is extracted from network packets and saved to a .csv file. Thus, a dataset is formed.

The hping3 utility simulated the TCP SYN Flood attack. A cascade of deep neural networks recognized a DoS attack (Fig. 11). Hybrid neural network CNN + LSTM detected the presence of a network attack with an accuracy of 99.96. And the convolutional neural network classified this attack with an accuracy of 95.87.

CONCLUSIONS

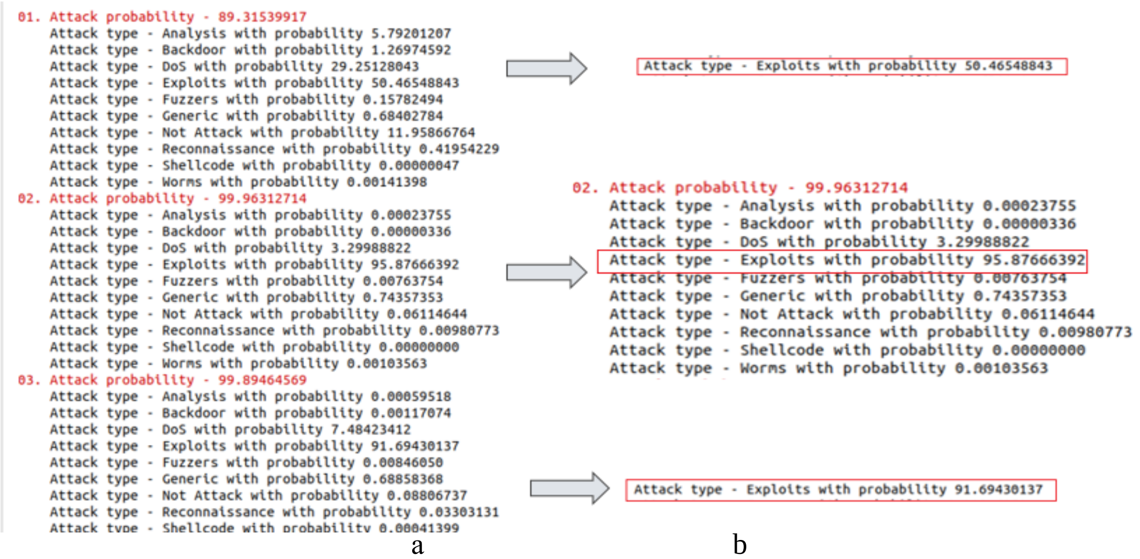
Thus, having analyzed the popular models of deep neural networks MLPs, CNN, LSTM, the authors came to the conclusion that in order to solve the problems of detecting and classifying network attacks, it is necessary to develop a cascade of two DNNs. Network attack detection should be performed with a hybrid DNN consisting of CNN and LSTM layers, while attack classification – by CNN.

Based on the analysis of the widespread available datasets, taking into account modern network traffic, UNSW-NB15 was selected and the need for structural modification of the dataset was substantiated. For the input set, a technology for preparing data for training and testing DNN has been developed. As a result, 176 features were obtained out of 47 ones of the network dataset UNSW-NB15.

At the stage of tuning and training the DNN cascade, the selection of hyperparameters was carried out, which made it possible to improve the quality of the model.

The developed cascade of deep neural networks was tested on the UNSW-NB15 validation set and on the real network infrastructure of a home computer network. A TCP SYN Flood attack was simulated and the DNN cascade recognized a DoS attack.

The use of a cascade of deep neural networks made it possible to improve the accuracy of detecting and classifying attacks in computer networks and to reduce the frequency of false alarms in detecting attacks in comparison with previously published results of studies of DNNs. But perhaps these metrics would be better if the dataset was balanced.



```
01. Attack probability - 89.31539917
    Attack type - Analysis with probability 5.79201207
    Attack type - Backdoor with probability 1.26974592
    Attack type - DoS with probability 29.25128043
    Attack type - Exploits with probability 50.46548843
    Attack type - Fuzzers with probability 0.15782494
    Attack type - Generic with probability 0.68402784
    Attack type - Not Attack with probability 11.95866764
    Attack type - Reconnaissance with probability 0.41954229
    Attack type - Shellcode with probability 0.00000047
    Attack type - Worms with probability 0.00141398
02. Attack probability - 99.96312714
    Attack type - Analysis with probability 0.00023755
    Attack type - Backdoor with probability 0.00000336
    Attack type - DoS with probability 3.29988822
    Attack type - Exploits with probability 95.87666392
    Attack type - Fuzzers with probability 0.00763754
    Attack type - Generic with probability 0.74357353
    Attack type - Not Attack with probability 0.06114644
    Attack type - Reconnaissance with probability 0.00980773
    Attack type - Shellcode with probability 0.00000000
    Attack type - Worms with probability 0.00103563
03. Attack probability - 99.89464569
    Attack type - Analysis with probability 0.00059518
    Attack type - Backdoor with probability 0.00117074
    Attack type - DoS with probability 7.48423412
    Attack type - Exploits with probability 91.69430137
    Attack type - Fuzzers with probability 0.00846050
    Attack type - Generic with probability 0.68858368
    Attack type - Not Attack with probability 0.08806737
    Attack type - Reconnaissance with probability 0.03303131
    Attack type - Shellcode with probability 0.00041399
```

```
Attack type - Exploits with probability 50.46548843

02. Attack probability - 99.96312714
    Attack type - Analysis with probability 0.00023755
    Attack type - Backdoor with probability 0.00000336
    Attack type - DoS with probability 3.29988822
    Attack type - Exploits with probability 95.87666392
    Attack type - Fuzzers with probability 0.00763754
    Attack type - Generic with probability 0.74357353
    Attack type - Not Attack with probability 0.06114644
    Attack type - Reconnaissance with probability 0.00980773
    Attack type - Shellcode with probability 0.00000000
    Attack type - Worms with probability 0.00103563

Attack type - Exploits with probability 91.69430137
```

Fig. 11. **An example of testing a DNN cascade on a real network infrastructure**
a – CNN + LSTM approbation; b – CNN approbation
***Source:* compiled by the authors**

However, there are still many problems with using DNN to detect network attacks. First, it is difficult to modify DNNs as classifiers to detect attacks in real time. Moreover, with the development of IoT, cloud and big data technologies, the question of how to use them to improve the efficiency of attack detection methods using DNN remains open and interesting and indicates prospects for further research in this direction [4].

REFERENCES

1. “EY Global Information Security Survey”. 2020. – Available from: https://assets.ey.com/content/dam/ey-sites/ey-com/en_gl/topics/advisory/ey-global-information-security-survey-2020-single-pages.pdf. – [Accessed Dec. 2020].

2. Markus R., et al. “A survey of network-based intrusion detection data sets”. *Computers & security*. 2019; Vol. 8, No. 6: 147–167. DOI: https://doi.org/10.1016/j.cose.2019.06.005.

3. Khraisat, Ansam, Gondal, Iqbal,Vamplew, Peter & Kamruzzaman, Joarder. “Survey of intrusion detection systems: techniques, datasets and challenges”. *Cybersecurity*. 2019; Vol. 20: 2–20. DOI: https://doi.org/10.1186/s42400-019-0038-7.

4. Wu Yirui, Wei Dabao & Feng Jun. “Network Attacks Detection Methods Based on Deep Learning Techniques: A Survey”. *Security and Communication Networks*. 2020. 17 p. DOI: https://doi.org/10.1155/2020/8872923.

5. Tang, T. A., Mhamdi, L., McLernon, D., Zaidi, S. A. R. & Ghogho, M. “Deep learning approach for network intrusion detection in software defined networking.” *Proceedings of 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM).* IEEE. Reims: France. 2016. p. 258–263. DOI: https://doi.org/10.1109/WINCOM.2016.7777224.

6. Li, D., Baral, R., Li, T., Wang, H., Li, Q. & Xu, S. “Hashtrandnn: a framework for enhancing robustness of deep neural networks against adversarial malware samples.” 2018. – Available from: http://arxiv.org/abs/ 1809.06498. – [Accessed Nov.2020].

7. Maksimov, M. & Shpinareva, I. “Attack Detection System on Local Computer Network.” *Proceedings of the International Scientific Young Scientists Conference CSYSC-2018.* Ivano-Frankivsk: Ukraine. 2018. p.102–105.

8. Kolosnjaji, B., Zarras, A., Webster, G. & Eckert, C. “Deep learning for classification of malware system call sequences.” *Proceedings of Australasian Joint Conference on Artificial Intelligence. Publ. Springer*. Hobart: Australia. 2016. p.137–149. DOI: https://doi.org/10.1007/978-3-319-50127-7_11.

9. Zhang, M., Xu, B., Bai, S., Lu, S. & Lin, Z. “A deep learning method to detect web attacks using a specially designed CNN”. *Proceedings of 24th International Conference on Neural Information Processing*. Guangzhou: China. November 2017. p. 828–836. DOI: https://doi.org/10.1007/978-3-319-70139-4_84.

10. Wang, W., Zhu, M., Zeng, X., Ye, X. & Sheng, Y. “Malware traffic classification using convolutional neural network for representation learning”. *Proceedings of 2017 International Conference on Information Networking*. Da Nang: Vietnam. January 2017. DOI: https://doi.org/10.1109/ ICOIN.2017.7899588.

11. Kim, J., Kim, J., Thu, H. L. T. & Kim, H. “Long short term memory recurrent neural network classifier for intrusion detection”. *Proceedings of 2016 International Conference on Platform Technology and Service (PlatCon)*. Jeju: Korea. February 2016. p. 1–5. DOI**:** https://doi.org/10.1109/ PlatCon.2016.7456805.

12. Le, T., Kim, J. & Kim, H. “An effective intrusion detection classifier using long short-term memory with gradient descent optimization”. *Proceedings of 2017 International Conference on Platform Technology and Service (PlatCon).* Jeju: Korea. February 2017. p. 1–6. DOI: https://doi.org/ 10.1109/PlatCon.2017.7883684.

13. Liu, H., Lang, B., Liu, M. & Yan, H. “CNN and RNN based payload classification methods for attack detection”. *Knowledge-Based Systems*. 2019; Vol. 163: 332–341. DOI: https://doi.org/10.1016/ j.knosys.2018.08.036.

14. Hou Haixia, Xu Ingying, Chen Menghan & Liu Zhi. “Hierarchical Long Short-Term Memory Network for Cyberattack Detection.” *IEEE Access*. 2020; Vol. 8: 90907–90913. DOI: https://doi.org/10.1109/ACCESS.2020.2983953.

15. Lopez-Martin, M., Carro, B., Sanchez-Esguevillas, A. & Lloret, J. “Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things”. *IEEE Access*. 2017; Vol. 5: 18042–18050. DOI: https://doi.org/10.1109/ACCESS.2017.2747560.

16. Rezaei, S., Kroencke, B., Liu X. Large-Scale. “Mobile App Identification Using Deep Learning”. *IEEE Access*. 2020; Vol. 8: 348–362. DOI: https://doi.org/10.1109/ACCESS.2019.2962018.

17. Zeng, Y., Qi, Z. et al. “TEST: an End-to-End Network Traffic Examination and Identification Framework Based on Spatio-Temporal Features Extraction”. 2019. – Available from: arXiv:1908.10271. – [Accessed Oct.2020].

18. Chalapathy, R. & Chawla, S. “Deep learning for anomaly detection: A survey.” 2019. – Available from: https://arxiv.org/abs/1901.03407. – [Accessed Oct. 2020].

19. Ojha, V. K., Abraham, A. & Snasel, V. “Metaheuristic Design of Feedforward Neural Networks: A Review of Two Decades of Research”. *Engineering Applications of Artificial Intelligence*. 2017. 60 p. DOI: https://doi.org/10.1016/j.engappai.2017.01.013.

20. Sahu, S. K., Sarangi, S. & Jena, S .K. “A Detail Analysis on Intrusion Detection Datasets”. *Proceedings of the 2014 IEEE International Advance Computing Conference (IACC)*. 2014. p. 1348. DOI: https://doi.org/10.1109/IAdCC.2014.6779523.

21. Tavallaee, M., Bagheri, E., Lu, W. & Ghorbani, A. A. “A detailed analysis of the KDD CUP 99 data set”. *IEEE Symposium on Computational Intelligence for Security and Defense Applications*. CISDA. 2009. p. 1–6. DOI: https://doi.org/10.1109/CISDA.2009.5356528.

22. Moustafa. N. & Slay. J. “UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)”. *MilCIS*. 2015. p. 1–6. DOI: https://doi.org/10.1109/MilCIS.2015.7348942.

23. Bernstein, R. & Dulkin, A. “Systems and methods for detection of anomalous network behavior.” 2016. – Available from: https://patents.google.com/patent/US9565203. – [Accessed Oct.2020].

24. Monowar, H., Dhruba, K. & Jugal, K. “Network anomaly detection: methods, systems and tools”. *IEEE Communications Surveys & Tutorials*. 2014. p. 303–336. DOI: https://doi.org/10.1109/SURV.2013.052213.00046.

25. Creech Gideon & Hu Jiankun. “A semantic approach to host-based intrusion detection systems using contiguousand discontiguous system calls patterns”. *IEEE Transactions on Computer*s. 2014. p. 807–819. DOI: https://doi.org/10.1109/TC.2013.13.

Conflicts of Interest: the authors declare no conflict of interest

Received 26.12.2020
Received after revision 28.02.2021
Accepted 16.03.2021

DOI: https://doi.org/10.15276/hait.03.2021.4
УДК 004.056

Використання методів поглиблених навчання для виявлення і класифікації мережевих атак

Ірина Михайлівна Шпінарева[1)]
ORCID: https://orcid.org/0000-0001-9208-4923; ishpinareva@gmail.com. Scopus ID: 8532376700
Анастасія Олексіївна Якушина[1)]
ORCID: https://orcid.org/0000-0001-9510-159X ; ayakushina07@gmail.com
Людмила Арнольдовна Волощук[1)]
ORCID: https://orcid.org/0000-0002-2510-0038; lavstumbre@gmail.com
Микола Дмитрович Рудніченко[2)]
ORCID: https://orcid.org/0000-0002-7343-8076; nickolay.rud@gmail.com. Scopus ID: 57191406873
[1)] Одеський національний університет імені І. І. Мечникова, вул. Дворянська, 2. Одеса, 65026, Україна
[2)] Одеський національний політехнічний університет, проспект Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

На основі аналізу практичного використання систем виявлення атак для захисту локальних комп'ютерних мереж показана актуальність розробки каскаду глибинних нейронних мереж для виявлення і класифікації мережевих атак. Каскад глибинних нейронних мереж, складається з двох мереж. Перша мережа – гібридна глибинна нейронна мережа, що складається з шарів згорткової нейронної мережі і шарів довгої короткострокової пам'яті для виявлення атак. Друга мережа – згорткова нейронна мережа для класифікації найбільш популярних класів мережевих атак, таких як: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode і Worms. На етапі налаштування і навчання каскаду глибинних нейронних мереж здійснено підбір гіперпараметрів, що дозволило добитися підвищення якості моделі. Серед доступних публічних наборів даних з урахуванням сучасного трафіку обраний один з актуальних наборів UNSW-NB15. Для розглянутого набору даних розроблена технологія попередньої обробки даних. Каскад глибинних нейронних мереж навчений, протестований і апробований на наборі даних UNSW-NB15. Проведена апробація каскаду глибинних нейронних мереж на реальному мережевому трафіку, яка показала його спроможність виявляти і класифікувати атаки в комп'ютерній мережі.Використання каскаду глибинних нейронних мереж, що складається з гібридної нейронної мережі CNN + LSTM і нейронної мережі CNN дозволило поліпшити точність виявлення і класифікації атак в комп'ютерних мережах і зменшити частоту помилкових тривог виявлення мережевих атак.

Ключові слова: глибоке навчання; NIDS; CNN, LSTM; глибокі нейронні мережі; гібридні нейронні мережі

ABOUT THE AUTHORS

Irina M. Shpinareva – PhD in Physico-mathematical sciences, Associate Professor of the Department of Mathematical Support of Computer Systems. Odessa I. I. Mechnikov National University. 2, St. Dvoryanskaya. Odessa, 65026, Ukraine
ORCID: https://orcid.org/0000-0001-9208-4923; ishpinareva@gmail.com. Scopus ID: 8532376700
Research field: Information security technology; cryptography; machine learning; deep learning; big data

Ірина Михайлівна Шпінарева – кандидат фізико-математичних наук, доцент кафедри Математичного забезпечення комп’ютерних систем. Одеський національний університет імені І. І.Мечникова, вул. Дворянська, 2. Одеса, 65026, Україна

Anastasia A. Yakushina – Master of the Department of Mathematical Support of Computer Systems.
Odessa I. I. Mechnikov National University. 2, St. Dvoryanskaya, Odessa, 65026, Ukraine
ORCID: https://orcid.org/0000-0001-9510-159X , ayakushina07@gmail.com
***Research field*:** Information security technology; machine learning; deep learning; data mining; big data; data visualization

Анастасія Олексіївна Якушина – магістр кафедри Математичного забезпечення комп'ютерних систем. Одеський національний університет імені І. І.Мечникова, вул. Дворянська, 2. Одеса, 65026, Україна

Lyudmila A. Voloshchuk – PhD (Eng), Associate Professor of the Department of Mathematical Support of Computer Systems. Odessa I. I. Mechnikov National University. 2, St. Dvoryanskaya. Odessa, 65026, Ukraine.
ORCID: https://orcid.org/0000-0002-2510-0038, lavstumbre@gmail.com
Research field: Computer networking; cloud technology; information security technology; machine learning; big data

Людмила Арнольдовна Волощук – кандидат технічних наук, доцент кафедри Математичного забезпечення комп’ютерних систем. Одеський національний університет імені І. І. Мечникова, вул. Дворянська, 2. Одеса, 65026, Україна

Nikolay D. Rudnichenko – PhD (Eng), Associate Professor of the Department of Information Technology.Odessa National Polytechnic University. 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-7343-8076; nickolay.rud@gmail.com. Scopus ID: 57191406873
***Research field*:** Machine learning; data mining; big data; data visualization; risk assessment; complex technical systems it management

Микола Дмитрович Рудніченко – кандидат технічних наук, доцент, доцент кафедри Інформаційних технологій. Одеський національний політехнічний університет, проспект Шевченка,1. Одеса, 65044, Україна