UDC: 534.843:004.9 DOI: https://doi.org/10.15407/jai2021.02.015

CONSTRUCTION OF SUBSYSTEM DETERMINATION OF ATTACKS IN CYBERPHYSICAL SYSTEMS BY NEURAL NETWORK METHODS

O. Belej[1], K. Kolesnyk [2], N. Nestor[3], Yu. Fedirko[4]
[1,2,3,4] Lviv Polytechnic National University, Ukraine
5, Mytropolyt Andrei str., Building 4, Room 324, Lviv, 79015

[1]http://orcid.org/0000-0003-4150-7425
[2]http://orcid.org/0000-0001-9396-595X
[3]http://orcid.org/0000-0003-4391-2563
[4]http://orcid.org/0000-0001-9968-7313

Abstract. In this research work analyzes and compares existing methods for describing data from cyberphysical systems, methods for detecting network attacks targeting cyberphysical systems, analyzes fundamental approaches and solutions in the field of cyberphysical systems security, and makes recommendations for supplementing existing approaches using new algorithms.

The considered application of the neuroevolutionary algorithm of NeuroEvolution of Augmenting Topology using a hypercube for the analysis of multivariate time series describing the state of cyberphysical systems in order to identify abnormal conditions. After the modification, the algorithm allows almost completely configuring the target neural network without user intervention according to the specified parameters, including additionally creating intermediate network layers that were previously unavailable in the primary version of the algorithm. The method is verified on the TON_IOT DATASETS dataset. The system topology is the structure of the Internet of Things. The data are relevant, verified and correct, which allows them to be used for analysis and assessment of the accuracy of the approach under consideration. The obtained overall accuracy, proximity of solutions, values of False Positive Rate and False Negative Rate indicate the lack of retraining of the model and the high reliability of this method for detecting attacks in cyberphysical systems.

Keywords: cyber-physical systems, neuroevolutionary algorithms, network attack detection.

ПОБУДОВА ПІДСИСТЕМИ ВИЯВЛЕННЯ АТАК В КІБЕРФІЗИЧНИХ СИСТЕМАХ МЕТОДАМИ НЕЙРОМЕРЕЖІ

О.І. Белей[1], К.К. Колесник [2], Н.І. Нестор[3], Ю.А. Федірко[4]
[1,2,3,4] Національний університет «Львівська політехніка», Україна
вул. Митрополита Андрея, 5, 4 навч. корпус, 324 кімната, Львів, 79015

[1]http://orcid.org/0000-0003-4150-7425
[2]http://orcid.org/0000-0001-9396-595X
[3]http://orcid.org/0000-0003-4391-2563
[4]http://orcid.org/0000-0001-9968-7313

Анотація. У цьому дослідженні проведено аналіз і порівняння існуючих методів опису даних з кіберфізичних систем, методів виявлення мережевих атак, спрямованих на кіберфізичні системи, аналіз фундаментальних підходів і рішень у сфері безпеки кіберфізичних систем, а також вироблено рекомендації щодо доповнення існуючих підходів шляхом застосування нових алгоритмів.

Розглянуте нами застосування нейроеволюційного алгоритму розширеної топології з використанням гіперкубу для аналізу багатовимірних часових рядів, що описують стан кіберфізичних систем, щодо виявлення

аномальних станів. Після модифікації алгоритм дозволяє практично повністю налаштувати цільову нейронну мережу без втручання користувача за заданими параметрами, включаючи додатково створення проміжних мережевих шарів, які раніше були недоступні в основній версії алгоритму. Верифікація методу проводиться з набору даних TON_IOT DATASETS. Топологія системи є структурою Інтернету речей. Дані є релевантними, перевіреними та коректними, що дозволяє використовувати їх для аналізу та оцінки точності аналізованого підходу. Отримані загальна точність, близькість рішень, величини False Positive Rate та False Negative Rate свідчать про відсутність перенавчання моделі та високу надійність даного методу для виявлення атак в кіберфізичних системах.

Ключові слова: кіберфізичні системи, нейроеволюційні алгоритми, виявлення мережевих атак.

Introduction

Due to the presence of a significant number of factors of different nature, the functioning of the information system and the attack detection systems (ADS) is probabilistic. Therefore, it is important to substantiate the type of probabilistic laws of specific operating parameters. Particular emphasis should be placed on the task of substantiating the loss function of the information system, which is set in accordance with its target function and in the field of system parameters. In this case, the target function should be determined not only at the expert level, but also in accordance with a set of parameters of the functioning of the entire information system and the tasks assigned to it. Then the ADS quality indicator will be defined as one of the parameters that affect the target function, and its allowable values - the allowable values of the loss function.

After substantiation of laws and functions, the real task is to obtain formalized methods of optimal structure of the ADS in the form of a set of mathematical operations. Thus, the problem of synthesis of the ADS structure can be solved. On the basis of the received mathematical operations it will be possible to calculate dependences of indicators of quality of functioning of ADS on parameters of its functioning, and also on parameters of functioning of information system, ie the real analysis of quality of functioning of ADS will be possible.

The difficulty of applying to the formalized apparatus of analysis and synthesis of information systems for ADS is that a particular information complex and its subsystem - ADS consist of inhomogeneous elements that can be described by different sections of the theory, ie this object of study is aggregated. Therefore, mathematical models can probably be obtained only for individual components of the ADS, which complicates the analysis and synthesis of the ADS as a whole, but further refinement of the formalized apparatus of analysis and synthesis will optimize the ADS.

Based on the above, we can conclude that in practice there is considerable experience in solving problems of intrusion detection. The ADS used is largely based on empirical schemes of the intrusion detection process, further improvement of the ADS is associated with the specification of methods of synthesis and analysis of complex systems, the theory of pattern recognition in application to the ADS.

Problem statement

In the field of cyber-physical systems (CPS) security in machine learning, according to the empirical experience of the authors and the established practice of this field, preference is usually given to neural networks of different configurations and evolutionary algorithms. By the way, although neural networks have taken a dominant place in this field, and genetic algorithms have become less popular due to possible problems with overcoming local extremes, but, for example, in [1] genetic algorithms have shown their effectiveness.

The advantages of using machine learning in solving CPS safety problems have been confirmed theoretically and in practice in such works as [2].

In principle, the chosen neural network or genetic algorithm affects only the accuracy, speed and resource requirements of the analyzer through their internal devices and solutions. In turn, the emphasis is placed on neural networks due to the maximum flexibility of the analyzer parameters for each specific theoretical-descriptive and physical

model. For example, [3] describes in detail the reasons for choosing the configuration of the neural network, layers, degree of sieving and other necessary details.

The general approach to solving previously formulated problems is to obtain data from the system, compare them display them according to the adopted model and further predict the future state of the system. The result is compared with the current state. If there is a difference in the results that exceeds the threshold value, the behavior of the system is considered abnormal.

Advantages of the solution:

— large variability of the applied designs and, as a result, a wide choice between speed, quality and requirements to resources of power of system;
— the possibility of the deepest and most reliable detection of anomalies in the low-level component of the system (LLCS) by applying the cyclicity of the analyzer described below;
— the ability to implement the most in-depth analysis and increase system security.

Disadvantages of the solution:

— the initial complexity of setting up the analyzer;
— the need to train the system;
— a priori increased requirements for system resources compared to all other solutions;
— the impossibility of transferring the studied model to a new topology, the need for retraining.

This solution can be applied to all the previously mentioned methods of CPS data representation, which allows any degree of homogeneity of the system structure, but the latter is subject to increased requirements in the field of resource consumption.

In case of need of hypersensitivity to the LLCS analysis this decision is easily supplemented by the mechanism of cyclicity.

The most convenient applications are those that do not involve frequent changes in the network topology in terms of changing projects and / or their implementation. In cases of frequent reconfiguration of network parameters, there is additional time and computational costs of system resources for re

It is also worth noting that there are configurations of neural networks that can predict not only a single future value of the system, but also, due to the buffering of time variables, periods. This approach allows you to solve the problem of cyclicity of the analyzer, namely to teach the neural network is not a sequential data set, selective, for example, over time, and a data set corresponding to a particular cycle of many devices.

Obviously, the learning cycle, the time slot sampling width must be set for the longest time of one cycle of all devices considered in the system, if this cycle of the device is equal to

$GACD(t_1, t_2, ..., t_n) = t_k (GACD(t_1, t_2, ..., t_n) = t_k$,

the greatest common divisor), otherwise the cycle duration is given by the product of cycles N devices so that by the end of the cycle all devices have returned to their original physical state, otherwise the imposition of multidimensional curves of the device cycles and causes erroneous detection of the anomaly.

This approach will detect physical anomalies of the LLCS in the absence of abnormal behavior in the high-level system component (HLCS), even in cases where the anomaly at the LLCS level was detected by statistical analysis mechanisms or self-similarity criteria. For example, you can consider the hardening process at a metallurgical plant. Thus, the temperature of the induction furnace during the cycle is a complex curve due to a certain technical process, which in any case can not be violated by metal shrinkage or other physical phenomena that occur when the system deviates from the specified algorithm. In case of HLCS violation, the data continue to be considered legitimate due to compensated change of values. Among other things, equal cyclic deviation from the average value of the multidimensional curve, changed by some noise with an average value tending to zero, malware added, but the physical process is disrupted. For example, a conditional trend line or values averaged over a period of time are maintained, but in the case of cyclicity, the analyzer can detect deviations of specific physical devices from a given multidimensional curve and detect an anomaly.

Analysis of the last publications

Speaking of methods of detecting network attacks, it is worth mentioning that any CPS operates on the basis of two types of flows - physical (low-level) and logical (high-level) [4]. Analysis of the low-level component of the system (LLCS), consisting in the processing of data obtained from measuring instruments, sensors and sensors, allows you to assess the correctness of the system processes. It also allows you to study in real time the occurrence and manifestation of abnormal behavior in the early stages due to the lack of high-level abstractions and ease of access to primary data. Similarly, together with the analysis of the LLCS, the analysis of the high-level component of the system (HLCS) is performed. It is due to the need to take into account the logic of operations, including the detection of abnormal behavior in the logical space, when the physical parameters remain in the correct state.

As mentioned earlier, there are currently a large number of different methods, approaches and implementations for detecting network attacks on CPS [5], but preference is usually given to either the use of analysis methods based on statistical tools [6], or the use of machine methods. training [7].

The aim of research

Machine learning methods in most cases are used in conjunction with multidimensional time series, whose mathematical apparatus allows to achieve a high degree of reliability of the results, high response speed, low magnitude of errors of the first and second kind. This is due to working with continuously generated data in the LLCS, which are presented in the form of multidimensional time poisons. Further, aggregating time series into multidimensional ones, this approach allows to more fully characterize the behavior of CPS in the dynamics and simplify the further processing of data sets.

In the practical part of this work, the application of the NeuroEvolution of Augmenting Topology algorithm using a hypercube for the analysis of multidimensional time series describing the state of CPS for the detection of anomalous states is considered.

The method is verified on the TON_IOT DATASETS data set [8]. The topology of the system is the structure of the Internet of Things (IoT). The data are relevant, verified and correct, which allows you to use them to analyze and assess the accuracy of this approach.

The data set includes the status and transmitted data of each of the 7 network devices: each device operates with two main variables and two secondary, load and current state value. This period of the system includes 1 period of 48 hours of operation in normal condition and 3 periods of 48 hours, during which various types of attacks on the system were discretely carried out, including attacks such as DoS, DDoS, Backdoor.

Data were collected from both LLCS and HLCS, namely, the final data sample included 4 bases: the state of each object, the degree of loading of each object, the physical data measured by the object, and the final recipient of data.

The main material

After a careful analysis of existing methods of presenting data in CPS, their advantages, disadvantages, areas of application, as well as existing methods of detecting network attacks, you can proceed to the direct creation and implementation of their own method of detecting network attacks on CPS.

The described method will be based on processing the obtained time series from CPS drives and sensors, using a modified NEAT hypercube algorithm to predict the future state of the system and calculate errors between predicted and actual values.

The NEAT hypercube algorithm itself is based on a symbiosis of two other mechanisms: neural networks and genetic algorithms that configure the neural network. The main points of implementation will be described in the following sections.

Testing of the created and implemented method of detecting network attacks on CPS will be performed on the data set TON_IOT DATASETS [8].

The initial data obtained from the sample TON_IOT DATASETS [8] were presented in csv files. Processing was performed on Python.

Data processing and aggregation included the following steps:

1. Unification of time of data presentation by time in steps of 1 second (time normalization is performed: averaging of the received data for an interval for 1 second, if any, or duplication of data in case of their absence for an interval for 1 second).
2. Assignment of identification numbers to each of 7 devices: identifier, numbering was performed arbitrarily, but with the preservation of the logical connection "sender-recipient".
3. Change of indicators for those devices that could not measure the degree of their "load", but support the ability to measure the degree of discharge of the power supply.

Given the previously discussed advantages and disadvantages of existing data presentation methods, it was decided to focus on the use of multidimensional time series. The main reasons for this choice are the variability of the data analyzers used, the possibility of manually adjusting the hyperparameters of the solver, as well as the high degree of variability of the method - the possibility of use in heterogeneous systems of different types.

When using multidimensional time series, the neural network is usually trained on the basis of actual data to predict the future state of the system and calculate the difference (error) between the predicted and the actual state. Detection of anomalous states in the system is carried out by error analysis.

As mentioned earlier, the classic multidimensional time series is as follows:

$$X = \{X^{(1)}, X^{(2)}, ..., X^{(m)}\}, \qquad (1)$$

where each value at the time is represented by a vector:

$$\{X^{(i)} = \{X_1^{(i)}, X_2^{(i)}, ..., X_n^{(i)}\} \qquad (2)$$

For convenience of work and simplification of designing of the initial template of communications (substrate) of a hypercube the initial data received from objects of system are normalized as follows:

$$x_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \qquad (3)$$

This IoT system included 7 devices, and each had 4 basic components, ie the size of the multidimensional time series was 28.

For convenience and high efficiency of the method, the sampling frequency of the processed data $\Delta x=1$ s is taken.

Figures 1 show displaying normalized system status data for 48 hours of operation in a normal state and in a state that includes abnormal behavior.

Based on previous research, it becomes clear that multidimensional time series are commonly used with neural networks, given that the latter have shown a fairly high accuracy of detecting network attacks in conjunction with this method of describing data in the CPS.

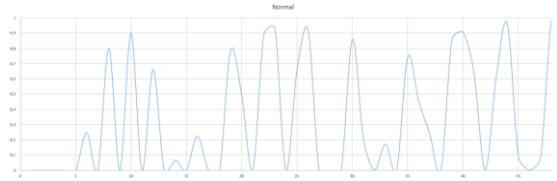

Fig. 1. Data changes during 48 hours of system operation in normal condition

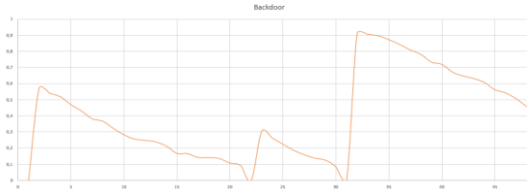

Fig. 2. Data changes during 48 hours of operation of the system with attacks such as Backdoor

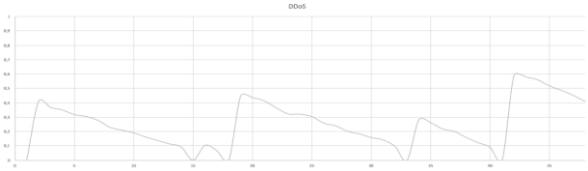

Fig. 3. Data changes during 48 hours of system operation with DDoS attacks

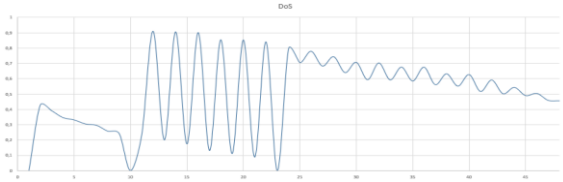

Fig. 4. data changes during 48 hours of system operation with DoS attacks

To avoid the previously mentioned disadvantages of using neural networks, namely the initial complexity of setting up the analyzer and the complexity of compiling the topology of the neural network, it was decided to use the neuroevolutionary algorithm NEAT-hypercube.

NeuroEvolution of Augmenting Topology (NEAT) is a genetic algorithm for the creation and development of neural networks. This method was developed in Austin, University of Texas. The principle of the algorithm was to change the weights and two-dimensional structure of the neural network - to find the most optimal value by methods of genetic algorithms.

It is worth noting the possibility of modular execution of NEAT algorithms. Since the implementation of the method is not limited to setting the specified hyperparameters, the performer has the ability

to configure both the data used for processing by the neural network, the neural network itself, and modify the genetic component of the algorithm to meet the needs of the task. .

NEAT-Hypercube-based NEAT is a generative coding that develops artificial neural networks based on the principles of the widely used NEAT algorithm. This is a new method of developing large-scale neural networks using geometric patterns of the subject area. It uses Compositional pattern-producing networks (CPPN). CPPN-s are a type of artificial neural network whose architecture is determined by genetic algorithms.

While neural networks often contain precisely sigmoid Gaussian activation functions, CPPN-s are usually based on more complex functions, as the former are not able to fully solve the optimization problem. The choice of functions for the canonical set can be shifted towards certain types of patterns and patterns. For example, periodic functions, such as sine, create segmented patterns with repetitions, while symmetric functions, such as Gaussian, create symmetric patterns. Linear functions can be used to create linear or fractal patterns. Thus, the architect of a CPPN-based genetic art system can change the types of patterns it generates by choosing the set of canonical functions to include.

In addition, unlike conventional neural networks, composite template networks can usually be applied to all possible input data, so they can be a complete structure. Because they are compositions of functions, CPPNs actually encode structures with infinite resolution and can be sampled for a particular solution with any optimal resolution.

The use of CPPN networks is successfully combined with multidimensional time series. As will be shown below, the modification of the NEAT-hypercube algorithm - changing the dimension of problems from two-dimensional orientation to N-dimensional allows to greatly simplify the topology of the final neural network.

In the general case, the NEAT hypercube algorithm works with input, output grids and user-configured intermediate layers, but this approach does not allow to fully automatically configure the topology of the final neural network.

An example of mapping the topology of a finite neural network to the space of a hypercube and an example of changing the topology of the neural network itself are shown in Fig. 5.

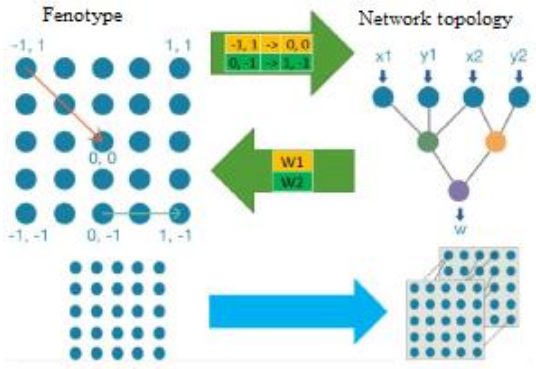

Fig. 5. Display of a neural network on a hypercube

A modification of the hypercube algorithm was to add the ability to create intermediate layers, which would otherwise have to be configured manually. This change allowed to almost completely automate the construction of the final neural network.

The novelty search function of the solution was chosen as a function of the suitability of the genetic algorithm for constructing layers. This choice is due to the

fact that the classic functions of fitness did not cope properly and the search for the optimal configuration of the intermediate layers.

Discussing

The software implementation of the created method was tested on the TON_IOT DATASETS data sample [8].

The threshold value of the error is determined empirically, and in this case dataset the value of T was set to 0.398. At this threshold, the following values were calculated:

1.Accuracy (how close the measurement is to the true value) $\frac{(TP+TN)}{(P+N)}$.

2.Precision (how close the measurements of the same object are to each other) $\frac{TP}{(TP+FP)}$.

3.True Positive Rate $\frac{TP}{(TP+FN)}$.

4.True Negative Rate $\frac{TN}{(TN+FP)}$.

5.False Positive Rate $\frac{FP}{(FP+TN)}$.

6.False Negative Rate $\frac{FN}{(TN+TP)}$.

7.Positive Predictive Value $=1-\frac{FP}{(FP+TP)}$.

8.Negative Predictive Value $=\frac{TN}{(TN+FN)}$.

9.F1 Score $=\frac{2TP}{(2TP+FP+FN)}$.

10.Matthews Correlation Coefficient (MCC)

$$=\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}.$$

In this case:

1.TP - the number of true detections of the normal state of the system (True Positive).

2.TN - the number of true detections of attacks on the system (True Negative).

3.FP - number of unrecognized attacks (False Positive).

4.FN - the number of normal system states recognized as attacks (False Negative).

5.P is the total number of normal states of the CPS (Positive).

6.N - total number of CPS states, including attacks (Negative).

The following are the values for all considered time intervals in the form of tables. For convenience, the values are broken down by attack type and by considered time intervals. After each time interval, as well as after all calculations and calculations, brief conclusions about the accuracy of the developed method follow.

Table 1. Received data on the “no attacks” segment near 48 hours

All	1209600
Positive	1209600
Negative	0
True Positive	1083802
True Negative	0
False Positive	0
False Negative	125798

Table 2. Accuracy of the method in the "no attacks" segment near 48 hours

Accuracy	0,8960
Precision	1,0000
True Positive Rate	0,8960
True Negative Rate	-
False Positive Rate	-
False Negative Rate	0,1040
Positive Predictive Value	1,0000
Negative Predictive Value	0,0000
F1 Score	0,9451
Matthews Correlation Coefficient	-

It is difficult to correctly judge the accuracy of the method on the test set. Although the overall precision is equal to one, this does not guarantee perfect operation of the method. This value is due to the absence of false positives of the FP type, since there were basically no attacks on this set, and the set itself was the reference one. The proximity of the solution (Accuracy) allows us to conclude that some errors (FN) were still present on the trained set - the model was not retrained.

Table 3. Received data on the "DoS attack" segment near 48 hours

All	1209600
Positive	394496
Negative	815104
True Positive	363748
True Negative	760838
False Positive	54266
False Negative	30748

Analyzing the DoS interval, we can say that the method showed itself positively here. These words are confirmed by both the high proximity of the solutions (Accuracy) and the high overall classification accuracy (Precision). The following values should be noted separately: False Positive Rate and False Negative Rate - their values were less than 0.1 and are very close to each other. These indicators indicate that the frequency of false detections is a small fraction of the total, and there is no bias towards FP or FN.

Table 4. Accuracy of the method in the "DoS attack" segment 48 hours

Accuracy	0,9297
Precision	0,8702
True Positive Rate	0,9221
True Negative Rate	0,9334
False Positive Rate	0,0666
False Negative Rate	0,0779
Positive Predictive Value	0,8702
Negative Predictive Value	0,9612
F1 Score	0,8954
Matthews Correlation Coefficient	0,8433

Table 5. Received data on the "DDoS attack" segment near 48 hours

All	1209600
Positive	486456
Negative	723144
True Positive	451619
True Negative	667022
False Positive	56122
False Negative	34837

Table 6. Accuracy of the method in the "DDoS attack" segment near 48 hours

Accuracy	0,9248
Precision	0,8895
True Positive Rate	0,9284
True Negative Rate	0,9224
False Positive Rate	0,0776
False Negative Rate	0,0716
Positive Predictive Value	0,8895
Negative Predictive Value	0,9504
F1 Score	0,9085
Matthews Correlation Coefficient	0,8453

As in the case of the interval, which includes DoS attacks, the method also worked well during the interval of DDoS attacks. The value of the overall accuracy (Precision) has slightly increased, but otherwise one can draw conclusions similar to the case with DoS - within a given interval, the method coped with its task perfectly.

Table 7. Received data on the "Backdoor attack" segment near 48 hours

All	1209600
Positive	781609
Negative	427991
True Positive	658908
True Negative	301315
False Positive	126676
False Negative	122701

Table 8. Accuracy of the method on the segment "Backdoor attack" near 48 hours

Accuracy	0,7938
Precision	0,8387
True Positive Rate	0,8430
True Negative Rate	0,7040
False Positive Rate	0,2960
False Negative Rate	0,1570
Positive Predictive Value	0,8387
Negative Predictive Value	0,7106
F1 Score	0,8409
Matthews Correlation Coefficient	0,5482

In this interval, the method showed the lowest precision (Precision) and the closest solutions (Accuracy). As it is not difficult to see from False Positive Rate and False Negative Rate, almost a third of attacks were incorrectly classified by the system as a normal state of the CPS. This behavior can be partially

explained by the fact that in the studied dataset, Backdoor attacks meant repeated duplication of a packet to the sender at certain intervals. Since the power of the packet data stream rarely exceeded the value of normally sent retry packets in the event of a real loss of the primary ones, the method did not fully cope with this particular attack. It is recommended to use either additional selection and detection criteria for this implementation of attacks, or in this case introduce an event handler for retransmission of previously received packets.

Table 9. Received data on the segment "All attacks" near 144 hours

All	3628800
Positive	1662561
Negative	1966239
True Positive	1474275
True Negative	1729175
False Positive	237064
False Negative	188286

Table 10. Accuracy of the method on the segment "All attacks" near 144 hours

Accuracy	0,8828
Precision	0,8615
True Positive Rate	0,8867
True Negative Rate	0,8794
False Positive Rate	0,1206
False Negative Rate	0,1133
Positive Predictive Value	0,8615
Negative Predictive Value	0,9018
F1 Score	0,8739
Matthews Correlation Coefficient	0,7647

It is easy to see that the overall precision (Precision) and the proximity of the solution (Accuracy) have decreased relative to the same indicators, but in the case of DoS and DDoS attacks. This change is due to the lower accuracy of the method on many Backdoor attacks. Possible reasons for this behavior are a rather rare duplication of sent packets during an attack, which can be easily lost against the background of real packet loss and legitimate duplication.

Table 11. Received data on the segment "For all time" near 192 hours

All	4838400
Positive	2872161
Negative	1966239
True Positive	2558077
True Negative	1729175
False Positive	237064
False Negative	314084

Table 12. Accuracy of the method on the segment "For all time"near 192 hours

Accuracy	0,8861
Precision	0,9152
True Positive Rate	0,8906
True Negative Rate	0,8794
False Positive Rate	0,1206
False Negative Rate	0,1094
Positive Predictive Value	0,9152
Negative Predictive Value	0,8463
F1 Score	0,9027
Matthews Correlation Coefficient	0,7658

Summing up the accuracy of the method, it is also worth considering these indicators for all studied time intervals. By aggregating data for all 4 intervals - DoS, DDoS, Backdoor attacks and the learning gap (no attacks), we get very high indicators of overall accuracy (Precision) and proximity of solutions (Accuracy). Additionally, it should be noted that in the general interval, the values of True Positive Rate and True Negative Rate have undergone insignificant drops by a few percent, which, again, indicates a high level of accuracy and reliability of the method both on the validation set, on a multitude of discrete attacks, and on the aggregate. these sets.

To get a clearer idea of the accuracy of the method, an ROC analysis was performed. Figure 20 shows the approximated ROC curve over the All Time 192 hours.

The indicator area under the graph of the curve (AUROC) can be treated as the equivalence of the probability that a binary classifier, when performing an assessment, assigns more weight to a randomly selected positive characteristic or indicator than to a randomly selected negative one. Under ideal conditions, this indicator tends to 1, and in the case of equiprobability "guessing" on the set -

to 0.5. This line is shown in Fig. 6 in orange. For this case, the AUROC indicator was 0.89, which emphasizes the sufficient accuracy of the method and successfully.

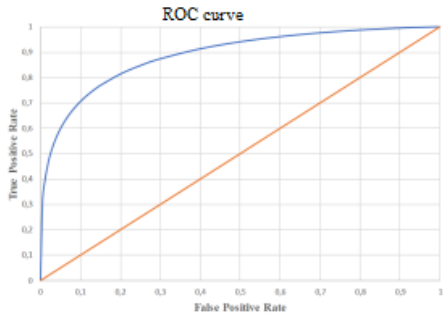

Fig. 6. Approximated ROC curve for the "All time" time interval

Conclusions

As a result of the work, a method for detecting network attacks on CPS was created and implemented. The accuracy of the method was also assessed. The principle of operation is to identify deviations between the current values of the CPS condition and the predicted results. Forecasting is performed by a neuroevolutionary algorithm of the NeuroEvolution of Augmenting Topology family.

The result of this work is the creation, implementation and experimental research of the implemented method for detecting network attacks carried out on the CPS. The method includes the use of a neuroevolutionary algorithm of the NEAT family: modified NEAT-hypercube.

After the modification, the algorithm allows almost completely configuring the target neural network without user intervention according to the specified parameters, including additionally creating intermediate network layers that were previously unavailable in the primary version of the algorithm.

The detection of network attacks carried out on the CPS was carried out in several stages:

1. Primary data processing and presentation of them in the form of multidimensional time series.
2. Configuring the neural network of the genetic component.
3. Training a configured neural network on a test set.
4. Predicting the future state of the system based on current data.
5. Calculation of the error between the predicted and real states of the system.
6. Comparison of the received error with the minimum threshold value T.

Testing was performed on the TON_IOT DATASETS dataset. The obtained overall accuracy (Precision; 0.9152) and the proximity of solutions (0.8861), as well as the values of False Positive Rate (0.1206) and False Negative Rate (0.1094) indicate the absence of model overfitting and high reliability of this method.

A further direction of the development of the topic is the creation of a data flow model of cyberphysical systems based on a hypercube with the possibility of self-healing according to an adaptive graph structure.

References

1. Kim, S.; Park, K.-J. A Survey on Machine-Learning Based Security Design for Cyber-Physical Systems. *Appl. Sci.* 2021, *11*, 5458. https://doi.org/10.3390/app11125458
2. C. A. R. de Sousa, "An overview on weight initialization methods for feedforward neural networks," *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 52-59, doi: 10.1109/IJCNN.2016.7727180.
3. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding / K. Hundman, V. Constantinou, Ch. Laporte, I. Colwell, T. Soderstrom //KDD '18: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. – 2018. – pp. 387–395
4. Filonov P., Lavrentyev A., Vorontsov A. Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model / P. Filonov, A. Lavrentyev, A. Vorontsov // NIPS Time Series Workshop, 2016.
5. Nanduri A., Sherry L. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN) / A. Nanduri, L. Sherry //Integrated Communications Navigation and Surveillance (ICNS), 2016. – IEEE, 2016. – pp. 5C2-1-5C2-8.
6. Grouped Convolutional Neural Networks for Multivariate Time Series /S. Yi, J. Ju, M.-K. Yoon, J. Choi//URL: https://arxiv.org/pdf/1703.09938.pdf.
7. Stouffer, K. , Falco, J., Scarfone, K. Guide to Industrial Control Systems (ICS) Security – Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS), and

other control system configurations such as Programmable Logic Controllers (PLC), Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, URL: https://doi.org/10.6028/NIST.SP.800-82

8. TON_IOT DATASETS. – URL: https://ieee-dataport.org/documents /toniot-datasets.

Received 29.11.21
Accepted 15.12.21