APPROACH TO INTELLIGENT MONITORING OF CYBER ATTACKS

Alexey N. Nazarov
Expert ITU, Russia,
a.nazarov06@bk.ru

Dmitry V. Pantiukhin,
National Research University Higher School of Economics;
Moscow Institute of Physics and Technology, Moscow, Russia

Ilya M. Voronkov,
International Centre of Informatics and Electronics, ICIE;
National Research University Higher School of Economics, Moscow, Russia

Mikhail A. Nazarov,
CEO LLC "SmartTech", Moscow, Russia

DOI: 10.36724/2664-066X-2020-6-6-2-9

ABSTRACT

The results of many years of research on the subject of intellectual counteraction to cyberattacks are presented. Cloud solutions for the synthesis of the monitoring cluster of cyberattacks are based on the latest achievements with the use of neuron-fuzzy formalism. The main features of the synthesis of protection functions are determined and the features of the implementation of the security system of the object of risk in cyberspace are analyzed. Methodological approaches to solving the system problem of determining all ways of penetration of the attack on the object of risk and the formation of variants of their coatings are proposed. The peculiarities of applicability of the branch and boundary method for solving this problem are discussed.

KEYWORDS: *security function, cluster, method, Hadoop, neural network, monitoring.*

Information about authors

Alexey N. Nazarov, *Doctor of Technical Science, Professor, Expert ITU, Moscow, Russia*
Dmitry V. Pantiukhin, *Master of Science, Senior lecturer, National Research University Higher School of Economics; Lecturer, Moscow Institute of Physics and Technology, Moscow, Russia*
Ilya M. Voronkov, *Master of Science, Head of Departament, International Centre of Informatics and Electronics, ICIE; Invited lecturer, National Research University Higher School of Economics, Moscow, Russia*
Mikhail A. Nazarov, *CEO LLC "SmartTech", Moscow, Russia*

1. Introduction

For an arbitrary object of risk of an information and telecommunication system (ITCS) subjected to an information attack, in General, there is [1-3] a complete system (list) of security functions in the causal sense.

Table 1
Security Functions

Designation of security functions	Appointment of security functions
X_1	Preventing the occurrence of conditions conducive to the generation of (occurrence) destabilizing factors (hereinafter - DF)
X_2	Warning immediate manifestations of DF
X_3	Detection manifested DF
X_4	Prevention of exposure to risk in the manifested and revealed DF
X_5	Prevention of exposure to risk on the manifest, but the undetected DF
X_6	Detecting the impact of DF on the subject of risk
X_7	Localization (restriction) found the impact of DF on the subject of risk
X_8	Localization of undetected exposure to risk by DF
X_9	Dealing with the consequences of the localized impact of the detected object on the destabilizing factors risk
X_{10}	Dealing with the consequences of undetected localized exposure to risk by DF

The system of security functions allows to unite on the universal methodical basis and ways, means, technologies of information security from various, not overlapping on the physical, natural essence subject areas. Protection functions depend on a large number of destabilizing factors [1]. For specific risk objects, these functions are the basis of the security policy and are developed and investigated by the security service of the risk object. Attackers also analyze known security features of the object at risk to identify vulnerabilities. Each of the parties in relation to the object of risk pursues opposite goals. Protection functions are developed and implemented by each party on the basis of existing measures and means to ensure information security of risk objects. The essence of each of the protection functions listed in [1] is unique. And technologies of realization of functions of protection can and should be modified at reception of new knowledge about possibilities of information attack.

As shown in [1-3] in an information attack, the protection functions are interconnected by causal transitions.

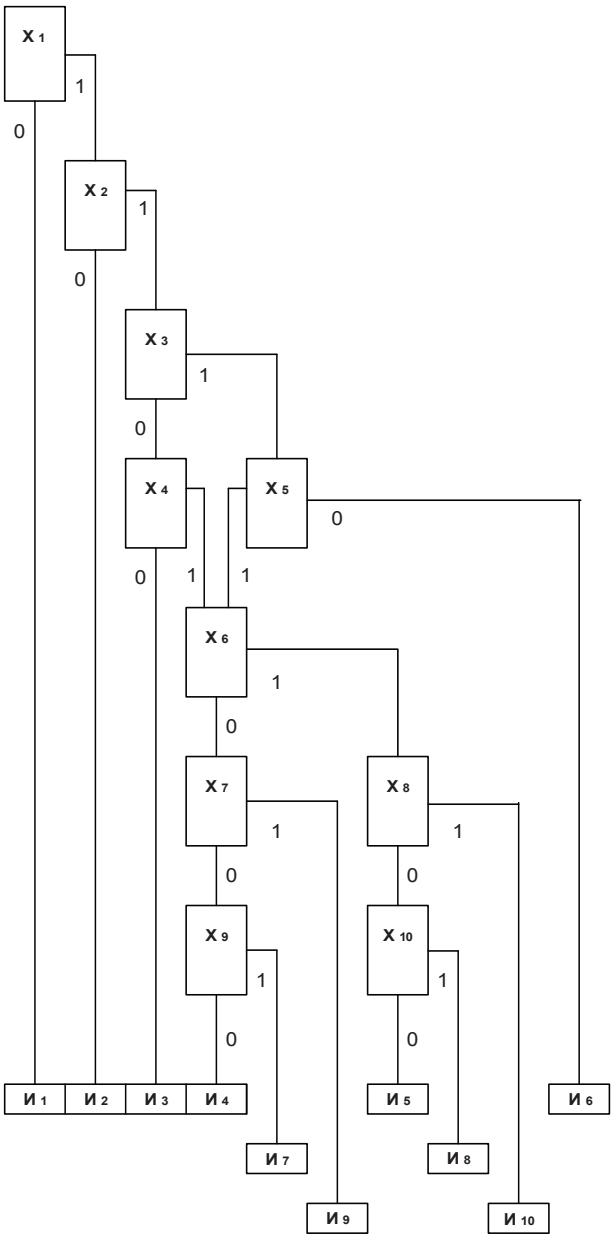

Fig. 1. The causal diagram of the security functions $X_1 \div X_{10}$ and results of attack $И_1 \div И_{10}$

Table 2
Final events in Fig. 1

$И_1 \div И_6$	Defence Provided
$И_7, И_8$	Defence Broken
$И_9, И_{10}$	Defence Destroyed

Therefore, the natural interpretation of such transitions is in the form of a digraph, which, in turn, can be represented as a matrix of incidents [7].

2. Aspects of security functions synthesis for any object of risk

General approaches to the practical creation of protection functions based on the fulfillment of the condition of reachability of an acceptable, sufficient level of security are given in [4]. In General, in order to protect the object, they develop an information security policy, guided by which they build barriers or boundaries of protection that prevent the possibility of carrying out an attack. As the boundaries of protection can be: protected area, access system to objects and buildings, user authentication, organization of password access to information of a certain category, etc. [3,7].

Each protection boundary can be represented as a directed graph [7], the set of vertices of which corresponds to the set of arguments (variables) of protection functions, and the set of arcs corresponds to the set of procedures for processing these variables. Such formalization naturally formalizes and concretizes the system of functions of protection of the object of risk.

Continuous automated monitoring of the risk object is possible on the basis of methods [4-6] that allow to quickly assess the level of security of the risk object, identify the attack and develop protection functions.

The task of further development of automated monitoring of the risk object is to simulate scenarios of trajectories of information attacks through the space in which the risk object of the infocommunication system is located. Which in turn is reduced to solving the problem of calculating the graph of the object of risk.

The solution of the problem fits into the following scheme of optimization stages:

– b sed on the graph model to overcome obstacles information attack to form all versions of attack success in relation to the risk object and the logical functions represented by the matrix of incidence (the achievability of the target of attack);

– crea e a variety of options for placing obstacles with the help of the problem of coverage;

– to carry out synthesis of variants of optimum placement by means of protection on the basis of the decision of a problem of conditional optimization;

– crea e additional coating options to improve the safety of certain critical elements;

– to carry out synthesis of additional variants of placement of means of protection providing different requirements of safety of elements of object of risk.

If the system of protection of the object of risk includes several consecutive boundaries, ordered in ascending order, then to access it it is necessary to overcome all the boundaries sequentially, starting with the lowest [7]. The security policy of the object of risk consists of a set of mechanisms for the protection of each obstacle.

Each protection mechanism is characterized by a vector of system characteristics, which include: the ability to quickly change the rules of interaction between users and data, the cost of its development and implementation, the required computing resources, etc.

At development or a choice of mechanisms of protection for some boundary of object of risk the means allocated for development and operation of this boundary of protection, losses in ITCS caused by success of attack, structure and qualification of developers, resources, structure and characteristics of a boundary etc.Thus, any mechanism of protection should be based on scientifically-practically proved methodical recommendations should be considered and methodically accurately applied.

The amount of options entry risk object specifies the number of access of the offender to the facility. And each penetration is prevented by interrupting the path at least in one edge of the graph. It is necessary to exclude (interrupt) all ways of penetration. This is the problem of finding the minimum cross-section on a graph, which is solved by determining the minimum coverage on the incidence matrix.

The formulation of the problem of covering - all paths of penetration to cover (interrupt) the minimum number of edges, in General, is reduced to the problem of conditional optimization. Thus it is possible to achieve that redundancy of not realized possibilities of covering edges of the graph tends to a minimum, provided that each edge covers at least one path.

This problem is solved by the method of branches and boundaries. From the point of view of system analysis, the process of obtaining all penetration paths and forming many variants of their coatings is the task of decomposing a complex problem into simple subtasks. After this task, according to the theory of system analysis, the problem of synthesis of optimal placement of hardware and software protection of ITCS is solved.

3. Monitoring clust

The authors analyzed the principles, approaches and technological procedures for the organization of monitoring from fairly General premises. Methodological approaches to the creation of algorithms and software solutions in the Hadoop web-programming environment in relation to a wide class of object monitoring tasks in the web-space are developed. For the first time, the topology of the hadoop monitoring cluster, which has a common application, is schematically shown in Fig. 3. Researches are conducted and algorithms of measurements of attributes of objects of monitoring in web-space taking into account requirements of unity of measurements are offered. System requirements for Hadoop monitoring cluster design have been developed.

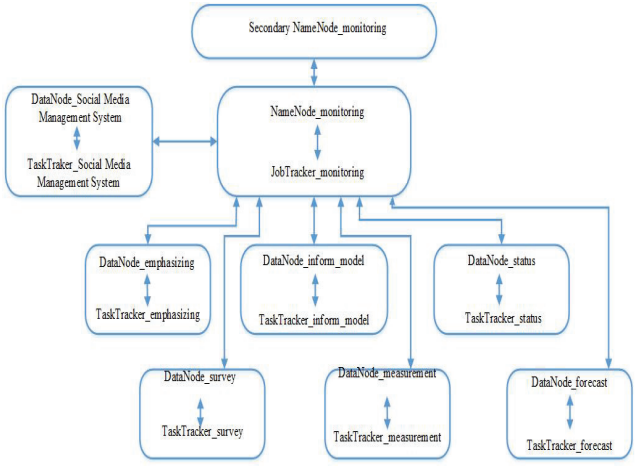

Fig. 3. Topology of the monitoring cluster Hadoop. Description daemons are given in [5]

The following hierarchical structure of the intelligent system of the ***TaskTraker***_status daemon is proposed as a complex of software modules of the automated system (AS) for assessing the state of the monitoring object in the web space. By analogy with [5] daemon contains the following functional modules: system, fuzzy production rules, describing the work ID taking into account expert assessments; neuro-fuzzy network whose structure reflects the system of fuzzy production rules; clear self-learning neural network (NS) to solve problems of classification and clustering of input vectors. As noted in [3,4], this hierarchy has a common application and is therefore suitable for different monitoring objects based on the Hadoop monitoring cluster shown in Fig. 3.

The level of identification of changes in attributes (characteristics, parameters) of the monitoring object, intended for classification by the vector of signs of changes in the elements and nodes of the monitoring object formed by the sensors of these changes, is illustrated in Fig. 4.

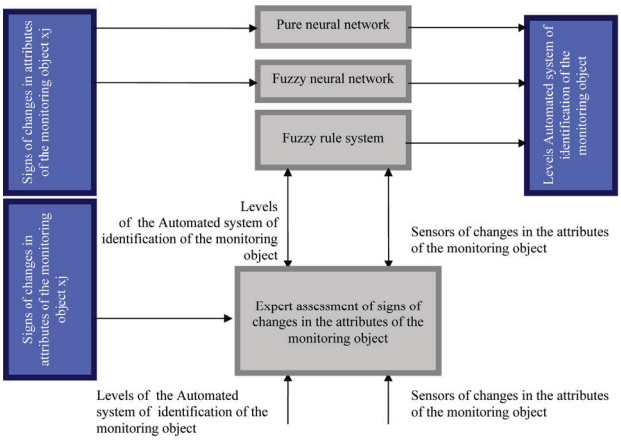

Fig. 4. Diagram of the adaptive classifier the level of identification of changes in the attributes of the monitoring object

The main system requirements for ***TaskTraker***_status daemon of AS monitoring object in the web-space, the presence of which is mandatory:

– presentation of a priori experience of experts in web-monitoring of the selected object in the form of a knowledge base described by the system of production rules;

– availability of criteria base for decision-making on changing attributes of the monitoring object;

– fuzzy logical conclus on, allowing to use the experience of experts in web-monitoring of the selected object in the form of a system of fuzzy production rules for the initial configuration of the information field (system of interneuron connections) fuzzy NS;

– plug-in aggregation services and services for processing unstructured information about changes in the attributes of the monitoring object for further analysis;

– the ability of the NS to classification and clustering;

– the ability of the NS to extract knowledge about the profile and mechanism of implementation of changes in the attributes of the monitoring object in the web space;

– the ability of the information field of the national Assembly to gain experience in the process of learning and self-learning.

Software that meets the above requirements must be developed in the Hadoop environment. In addition, the daemon ***TaskTraker***_status monitoring object in the web-space should be based on service-oriented integration methods in terms of scalability of its functional features.

4. The mechanism of fuzzy inference

This mechanism is based on the presentation of the experience of specialists (experts) on web-monitoring by the system of fuzzy production rules of the IF-THEN-type. For example:

Π_1: IF $\widetilde{x}_1$ IS A_{11} AND … $\widetilde{x}_n$ IS A_{1n}, THEN $\widetilde{y}$ IS B_1;

Π_2: IF $\widetilde{x}_1$ IS A_{21} AND … $\widetilde{x}_n$ IS A_{2n}, THEN $\widetilde{y}$ IS B_2;

…

Π_k: IF $\widetilde{x}_1$ IS A_{k1} AND … $\widetilde{x}_n$ IS A_{kn}, THEN $\widetilde{y}$ IS B_k,

where $\widetilde{x}_i$ and $\widetilde{y}$ are fuzzy input and output variables respectively; A_{ij} and B_i, $j = 1,\ldots,n$, $i = 1,\ldots,k$, the corresponding membership functions.

Combining the capabilities of NS and fuzzy inference is one of the promising approaches to the organization of artificial intelligence systems. As it was shown in [5], fuzzy logic systems compensate for the main "opacity" of the NS: in the representation of knowledge and the ability to explain the results of the intellectual system, i.e. complement the NS. Fuzzy inference formalism works in the absence of knowledge about the attributes of monitoring objects and their changes for any monitoring objects, which is important when new attributes with unknown dynamics of changes appear.

For the functional demon ***TaskTraker***_status monitoring in the web-space is a very important feature of neuro-fuzzy networks as the ability to automatically generate a system of fuzzy production rules in the process of teaching and learning, extracting hidden patterns from data, the input training samples.

Neural network training algorithms using stochastic properties of the dynamics of changes in the attributes of the monitoring object in the web space should be based on the standard method of minimizing the generalization error [3], on the basis of minimizing the quadratic residual functional on the training sample, searching for the gradient extremum of the target function of the error distribution density using the Robbins-Monroe procedure.

5. Hierarch levels and technological features of the functioning of the intelligent system daemon *TaskTraker*_status

The ability of NS to classify and cluster is used in the daemon to solve two main tasks:

1) classification of input ve or, for example, vector of signs of changes of attributes of object of monitoring;

2) xpansion of classification at appearance on an input of the classifier of the combination of signs of changes of attributes which was not earlier met.

Let the currently complete parcel space be $X = \{\tilde{x}_1, \ldots, \tilde{x}_m\}$ and a full space of conclusions $Y = \{\tilde{y}_1, \ldots, \tilde{y}_n\}$. Fuzzy causal relationships $\tilde{x}_i \to \tilde{y}_j$, $i = 1, \ldots, m$, $j = 1, \ldots, n$ between the elements of these spaces can be represented as a matrix R with elements r_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, and the premises and conclusions between them can be presented in the form of: $B = A \bullet R$, where $\bullet$ – composition operation, for example, max-min-composition.

In fuzzy inference, expert knowledge $A \to B$ reflects a fuzzy relation $R = A \to B$, where the operation $\to$ corresponds to a fuzzy implication. A fuzzy relation R can be considered as a fuzzy subset of the direct product $X \times Y$ of a complete set X and inferences Y, and the process of obtaining a fuzzy inference B result by premise A and knowledge $A \to B$ can be considered as a compositional rule: $B = A \bullet R = A \bullet (A \to B)$.

From practice [5] it follows that at the level of experience accumulation, the neuro-fuzzy classifier of vectors of characteristics, parameters of change (dynamics of attributes) of the monitoring object should be designed in the form of a three-layer fuzzy NS (Fig. 5) with the ability to reduce (compress) the number of features.

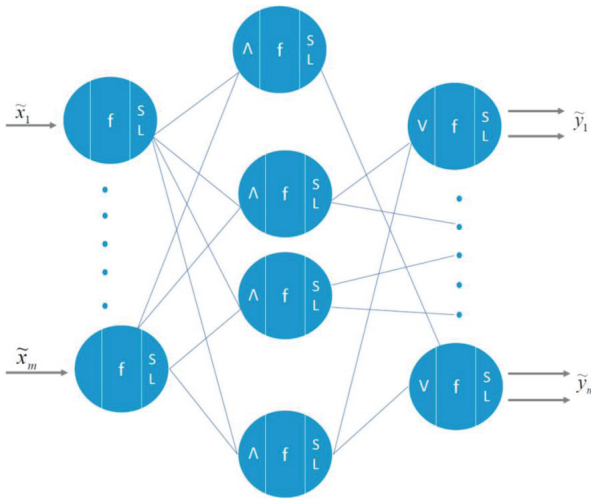

Fig. 5. Scheme of neuro-fuzzy classifier as S L-complementary pair of accessory functions

Each input vector from space can be matched with a fuzzy formal neuron (FN). The middle layer contains fuzzy FN performing a logical inference operation (for example, min) on combinations of fuzzy statements (FS) of the first NS layer to form a system of fuzzy classification conclusions.

The third layer of fuzzy NS is formed from fuzzy FN "OR" (by the number of fuzzy conclusions ,) and forms a vector of output fuzzy conclusions in accordance with the system of fuzzy rules set by experts.

5. xperimental results for application of convolution neural network for intrusions classification based on UNSW-NB15 dataset

For experimental estimation we use well-known UNSW-NB15 intrusion dataset [8]. This dataset contains both records about network connections, and the traffic itself (about 100GB) from the test computer network (3 servers). Dataset contains 9 types of attacks: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, Worms. Each records contains 47 fields – nformation about network connections and 2 fields - information about the type of intrusion (or normal packets). The total number of records is about 2 million. Information about network connections is divided into 4 text files (csv format). Files for training and testing of classifiers, containing respectively 175341 and 82332 records, are separately presented

We introduce a short review of UNSW-NB15 dataset usage (taken from [9]). Firstly, in 2015, creators of UNSW-NB15 dataset propose and compare significant feature selection approach to UNSW-NB15 and KDD99 datasets and build classifiers based on Naive Bayes and EM-clustering approaches [8]. They show that such simple approaches works with accuracy about 30-40% but some classes are not recognized at all. In [11] (2017) other authors combine Random forest classification method and feature selection approach, and reach better results both for KDD99 and UNSW-NB15 dataset. In same 2017, another author [12] combine Random Forest method with "Logitboost" [13] boosting algorithm and show better results on both datasets compared to clear Random forest methods [14] and some previous approaches. Boosting is quite attractive approach to create classifiers, as example in [16] authors compare a set of boosting methods (Bagged Tree, AdaBoost, GentleBoost, LogitBoost and RUSBoost algorithms) on UNSW-NB15 dataset and show that Bagged Tree and GentleBoost classifiers show superior performance (see [16] for details). Another comparison of boosting presented in [17] and ensembling methods in [18].

In 2016 researches apply genetic search to select appropriate features to each class and use Support Vector Machine to classify intrusions with such features [19]. They achieve high accuracy, more than 90% for all classes except of "Exploits" (~80%), and false positive rates less then 0.1%.

In 2018 quite interesting approach was introduced in [20]. Authors use multiscale wavelet transform and perceptron-like neural network with Hebbian learning for anomaly detection in records with HTTP protocol. They achieve Mean Accuracy 93.56% (wherein Mean TPR is 73.55%, Mean FPR is 4.46%, Mean TNR is 95.53%, Mean FNR: is 26.44%).

Creators of UNSW-NB15 continues their work and propose a Collaborative Anomaly Detection Framework, which is based on modification of Gaussian Mixture Model [21]. This system installed on each node of cloud network and each node in training phase create a statistical model of normal data by approximation of probability distribution function using Gaussian Mixture Model. Then, in testing phase, such model can be used to detect an anomaly, which is interpreted as intrusion.

Such approach can be applied only to detection but not to classification of intrusion. Authors achieve results 96% in accuracy and detection rate and 4-9% in false positive rate. Similar approaches but with Hidden Markov Models also was applied in [22], with Beta Mixture Model in [23], with Dirichlet Mixture Mechanism in [24].

Some comparison of machine learning approaches for intrusion classification over UNSW-NB15 dataset are made in [25]. Authors compare Support Vector Machine, Multilayer Perceptron, Restricted Boltzmann Machine, Sparse Autoencoder and deep learning architecture with embedding (like word2vec approach).

They show that deep learning approach is better in average than others, but, unfortunately, does not provide information about performance on each classes for UNSW-NB15 dataset. Autoencoders is also interesting approach to intrusion detection, in [26] authors present a two stage approach with autoencoders then second stage uses a results (score – output of classification unit) from first stage. Such approach shows follow results: 89.134% in accuracy and a 0.7495% in FAR for the UNSW-NB15 dataset. Combination of deep autoencoder, Support Vector Machine and Artificial Bee Colony searching method was used in [27] and show detection accuracy about 90% and FAR about 5%.

In [28] authors provide using a Long-Short-Term-Memory (LSTM) neural network, which is a type of Recurrent Neural Network, for intrusion detection over UNSW-NB15 dataset. They achieve quite high results (Precision=98.02%; Accuracy=99.41%; TPR=97.97%; TNR=99.53%, FNR=2.03%, and FPR=0.47%) in detection and show that such approach works better than, for example, Support Vector Machine. Bidirectional LSTM was used in [29] and show average 85% in precision and 88% in recall metrics. It is also shown that some classes are not recognized at all, due to imbalance amount of data.

In [30] deep learning architecture (16 layers, including fully connected with ReLU activation, dropout, and batch normalization layers) was created and tested on various intrusion datasets, including UNSW-NB15. Results not so high as in previous researches, but this architecture in used for large set of different datasets, and clearly show a problem of imbalanced classes.

Authors of [31] provide a set of techniques (Bootstrap Aggregation, Synthetic Minority Over-sampling, Under-sampling, and Class Balancer) to deal with imbalanced classes in intrusion detection system. They show minor advantages in term of area under ROC-curve to whole dataset, but, unfortunately, did not provide any information for classes.

Latest article [32] use a multiple-layer approach consisting of a coarse layer and a fine layer, in which the coarse layer with the deep convolutional neural network model focuses on identification of N abnormal classes and a normal class. While in the fine layer, an improved model based on gcForest (caXGBoost) further classifies the abnormal classes into N-1 subclasses. The proposed framework has been compared with the existing deep learning models using dataset NBC, a combination of UNSW-NB15 and CICIDS2017 including 101 classes. The experimental results show that method outperforms other single deep learning methods in terms of accuracy, detection rate and FAR and works well with imbalanced classes.

We study that for intrusion detection tasks it is a common situation then amount of examples of data of various classes is quite different (e.g. Normal class has 2 millions records, Worms class – only 174).

In this case training process of neural networks for intrusion recognition need to be modified to achieve better recognition of classes with small amount of examples. We use an approach of class-balanced batch forming and show in experiment that it can improve recognition performance of classes with small number of examples by the expense of decreased recognition performance of classes with large number of examples [9].

We use a convolution neural network for build a classifier. It consist from 5 convolutional layers with ‘sigmoid’ activations and 3 fully connected layers, first and second with ‘sigmoid’ and last with ‘softmax’ activation. Number of inputs – 190, number of outputs – 10 (9 intrusions classes and Normal). UNSW-NB15 has 47 features field, some of them relates only to that computer network which used for dataset collection, so we drop 7 irrelevant features. We code categorical features using one-hot encoding scheme. Therefore, for categorical features we have output vector of 153 (129+8+16) length. 15 of 37 numerical features scaled by division to appropriate value to make range almost equal and remaining 22 features unchanged.

Neural network trained using ‘RMS-prop’ [10] method, which is one of the modification of gradient-based method realized in ‘keras’ [10] library. Training is an iterative process in which loss-function minimizes. On each iteration, examples of inputs and corresponding desired outputs processed in neural network. Common choice for large datasets is to use so called ‘batches’ – combination of inputs\outputs of some length. This batches varies in each iteration and random input\output examples taken to a batch. In our work, we show that for situations then amount of examples in classes is very different better to use another approach for batch forming, namely - use predefine amount of examples of each class in batch. For example, if batch size is 300 and we have 10 classes then we can form a batch that consist of 30 random examples of each class.

As a result, we show that applying a class-balanced approach it is possible to drastically increase a recognition rate for classes with small number of examples corresponding to imbalanced training [9]:

– from 0% to 38% for Worms class (174 examples),

– rom 44% to 79% for Shellcode class (1511 examples),

– f om 0% to 53% for Backdoor class (2329 examples),

– f om 10% to 13% for Analysis class (2677 examples),

– f om 73% to 82% for Reconnaissance class (13987 examples),

– from 13% to 56% for DoS class (16353 examples),

but decrease recognition rate for classes with large number of examples:

– fro 77% to 53% for Fuzzers class (24246 examples),

– rom 81% to 56% for Exploits class (44525 examples),

– rom 98% to 97% for Generic class (215481 examples),

– fro 97% to 71% for Normal class (2218761 examples).

So, such approach can be applicable only for classes with small number of examples. Approach can be enlarged to control relative importance of classes by varying proportion of classes presented in batch.

Conclusion

Theoretical and technological research, which have a common application, and allow the machine to develop intelligent cloud solutions for the synthesis of monitoring systems and protection against cyber attacks on the basis of the protection functions of the object of risk on the basis of the following new results.

A machine-oriented graph approach to the synthesis of protection functions is developed, based on the graph representation of attack scenarios and a new application of the branch and boundary method. As a result of the application of this method, graph paths to the success of a cyber attack are suppressed.

For the topology of the monitoring a Hadoop cluster developed and tested guidelines for the synthesis of the demon Taktakishvili responsible for the task of assessing the status of the object of observation and identification information model, with the features of cloud computing. The principles and approaches based on neuro-fuzzy solutions that can be used as a basis for the design of intelligent automated systems for monitoring objects in the web space are proposed.

The mechanisms of decision-making based on the formalization of a priori experience of experts in the fuzzy base of fuzzy production rules are proposed. In the framework of solving the problems of classification and extension of classification of input data on the characteristics of the dynamics of attributes of the object of monitoring, the possibilities of a neuro-fuzzy classifier in the form of a three-layer fuzzy neural network are investigated.

Acknowledgement

Support from the Basic Research Program of the National Research University Higher School of Economics is gratefully acknowledged.

References

1. A. N. Nazarov. “Estimation of information safety level of modern infocommunication networks on basis of logic-probability approach,” in *Journal of Automation and Remote Control*, Vol. 68. Issue 7, 2007, pp. 1165-1176.

2. A. N. Nazarov. “Logical-and-probabilistic model for estimating the level of information security of modern information and communication networks,” *Journal of Telecommunications and Radio Engineering*, USA, 2010. Vol. 69. No. 16, pp. 1453-1463.

3. A. Nazarov & K. Sychev 2011, Models and methods for calculating the indicators of quality of functioning of the equipment units and structural parameters of the network the next generation networks, 2th edn, LLC Policom, Russia, Krasnoyarsk, 491 p.

4. A. Nazarov 2016, “Assessment of security from information attacks”, *Telecommunications*, no. 5, pp. 23-33.

5. A. Nazarov, M. Nazarov, D. Pantiuhin, S. Pokrova & A Sychev 2015, “Automation of monitoring processes in web-based neuro-fuzzy formalism”, T-Comm. Vol. 9. No. 8, pp. 26-33.

6. A Nazarov 2017, “Syntez of security functions against cyber-attacks”, T-Comm. Vol. 11. No. 9, pp. 80-85.

7. V Kostin 2017, “Synthesis of the optimal placement of hardware for physical protection systems for critical facilities”, Information Technology. Vol. 23. No. 1, pp. 41-49.

8. N. Moustafa, J. Slay, 2016 “The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set,” *Information Security Journal: A Global Perspective*. Vol. 25. No. 1-3, pp. 18-31.

9. D. Pantiukhin, A. Nazarov, I. M. Voronkov, 2019. “Intelligent methods for intrusion detection in local area networks”, *Proceedings of the 6th International Conference on Actual Problems of System and Software Engineering (APSSE 2019)* / Ed. by B. Pozin, A. R. Cavalli, A. Petrenko. P. 138-149.

10. “Keras: The Python Deep Learning Library.” [Online]. Available: https://keras.io.

11. T. Janarthanan and S. Zargari, "Feature selection in UNSW-NB15 and KDDCUP'99 datasets," 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), Edinburgh, 2017, pp. 1881-1886. doi: 10.1109/ISIE.2017.8001537

12. M. H. Kamarudin, C. Maple, T. Watson and N. S. Safa, "A LogitBoost-Based Algorithm for Detecting Known and Unknown Web Attacks," *IEEE Access*, vol. 5, pp. 26190-26200, 2017. doi: 10.1109/ACCESS.2017.2766844

13. J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: A statistical view of boosting,” Ann. Stat., vol. 28, no. 2, pp. 337-374, 2000.

14. H. Tribak, B. L. Delgado-Márquez, P. Rojas, O. Valenzuela, H. Pomares, and I. Rojas, “Statistical analysis of different artificial intelligent techniques applied to intrusion detection system,” *Proc. Int. Conf. Multimed. Comput. Syst*., 2012, pp. 434-440.

15. R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," *2017 International Conference on Data and Software Engineering (ICoDSE)*, Palembang, 2017, pp. 1-6. doi: 10.1109/ICODSE.2017.8285847

16. V. Timčenko and S. Gajin, "Ensemble classifiers for supervised anomaly based network intrusion detection," *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP),* Cluj-Napoca, 2017, pp. 13-19. doi: 10.1109/ICCP.2017.8116977

17. B. Patel, Z. Somani, S. A. Ajila and C. Lung, "Hybrid Relabeled Model for Network Intrusion Detection," *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Halifax, NS, Canada, 2018, pp. 872-877. doi: 10.1109/Cybermatics_2018.2018.00167

18. N. Moustafa, B. Turnbull and K. R. Choo, "An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things," *IEEE Internet of Things Journal*. Vol. 6. No. 3, pp. 4815-4830, June 2019. doi: 10.1109/JIOT.2018.2871719

19. H. Gharaee and H. Hosseinvand, "A new feature selection IDS based on genetic algorithm and SVM," *2016 8th International Symposium on Telecommunications (IST),* Tehran, 2016, pp. 139-144. doi: 10.1109/ISTEL.2016.7881798

20. S. Siddiqui, M. S. Khan and K. Ferens, "Multiscale Hebbian neural network for cyber threat detection," *2017 International Joint Conference on Neural Networks (IJCNN),* Anchorage, AK, 2017, pp. 1427-1434. doi: 10.1109/IJCNN.2017.7966020

21. N. Moustafa, G. Creech, E. Sitnikova and M. Keshk, "Collaborative anomaly detection framework for handling big data of cloud computing," *2017 Military Communications and Information Systems Conference (MilCIS),* Canberra, ACT, 2017, pp. 1-6. doi: 10.1109/MilCIS.2017.8190421

22. N. Moustafa, E. Adi, B. Turnbull and J. Hu, "A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems," *IEEE Access*, vol. 6, pp. 32910-32924, 2018. doi: 10.1109/ACCESS.2018.2844794

23. N. Moustafa, J. Slay and G. Creech "Novel Geometric Area Analysis Technique for Anomaly Detection using Trapezoidal Area Estimation on Large-Scale Networks," *IEEE Transactions on Big Data*. doi: 10.1109/TBDATA.2017.2715166

24. N. Moustafa, K. R. Choo, I. Radwan and S. Camtepe, "Outlier Dirichlet Mixture Mechanism: Adversarial Statistical Learning for Anomaly Detection in the Fog," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 1975-1987, Aug. 2019. doi: 10.1109/TIFS.2018.2890808

25. J. Yan, D. Jin, C. W. Lee and P. Liu, "A Comparative Study of Off-Line Deep Learning Based Network Intrusion Detection," *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*, Prague, 2018, pp. 299-304. doi: 10.1109/ICUFN.2018.8436774

26. F. A. Khan, A. Gumaei, A. Derhab and A. Hussain, "A Novel Two-Stage Deep Learning Model for Efficient Network Intrusion Detection," *IEEE Access*, vol. 7, pp. 30373-30385, 2019. doi: 10.1109/ACCESS.2019.2899721

27. Q. Tian, J. Li and H. Liu, "A Method for Guaranteeing Wireless Communication Based on a Combination of Deep and Shallow Learning," *IEEE Access*, vol. 7, pp. 38688-38695, 2019. doi: 10.1109/ACCESS.2019.2905754

28. S. Xiao, J. An and W. Fan, "Constructing an Intrusion Detection Model based on Long Short-term Neural Networks," *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, Singapore, 2018, pp. 355-360. doi: 10.1109/ICIS.2018.8466445

29. S. Yang, "Research o etwork Behavior Anomaly Analysis Based on Bidirectional LSTM," *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chengdu, China, 2019, pp. 798-802. doi: 10.1109/ITNEC.2019.8729475

30. R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat and S. Venkatraman, "Deep Learning Approach for Intelligent Intrusion Detection System," *IEEE Access*, vol. 7, pp. 41525-41550, 2019. doi: 10.1109/ACCESS.2019.2895334

31. C. Wheelus, E. Bou-Harb and X. Zhu, "Tackling Class Imbalance in Cyber Security Datasets," *2018 IEEE International Conference on Information Reuse and Integration (IRI),* Salt Lake City, UT, 2018, pp. 229-232. doi: 10.1109/IRI.2018.00041

32. X. Zhang, J. Chen, Y. Zhou, L. Han and J. Lin, "A Multiple-layer Representation Learning Model for Network-Based Attack Detection," *IEEE Access*. doi: 10.1109/ACCESS.2019.2927465