ТЕХНИЧЕСКИЕ НАУКИ
И ТЕХНОЛОГИИ

АҚПАРАТТЫҚ-КОММУНИКАЦИЯЛЫҚ ТЕХНОЛОГИЯЛАР
ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ
INFORMATION AND COMMUNICATION TECHNOLOGIES

DOI 10.51885/1561-4212_2024_1_234
IRSTI 81.93.29

B.T. Rzayev[1], I.S. Lebedev[2], Zh.T. Beldeubayeva[1], I.M. Uvaliyeva[4]
[1]S. Seifullin Kazakh Agrotechnical Research University, Astana, Kazakhstan
E-mail: pathinchaos@gmail.com
E-mail: zh.beldeubayeva@mail.ru
[2]St. Petersburg Federal Research Center of the RAS, Saint Petersburg, Russian Federation
E-mail: isl_box@mail.ru
[4]D. Serikbayev East Kazakhstan Technical University, Ust-Kamenogorsk, Kazakhstan
*E-mail: iuvalieva@ektu.kz**

IDENTIFICATION OF ROOTKITS IN NETWORK TRAFFIC WITH USING THE BAGGING OF CLASSIFIERS

КЛАССИФИКАТОРЛАР БЭГГИНГІН ҚОЛДАНУ НЕГІЗІНДЕ ЖЕЛІЛІК ТРАФИКТЕГІ РУТКИТТЕРДІ АНЫҚТАУ

ИДЕНТИФИКАЦИЯ РУТКИТОВ В СЕТЕВОМ ТРАФИКЕ НА ОСНОВЕ ПРИМЕНЕНИЯ БЭГГИНГА КЛАССИФИКАТОРОВ

Abstract. *The paper proposes an approach to identify anomalies in network traffic based on the use of machine learning classifiers. The solution allows you to determine the resulting state class by averaging the votes of individual classifiers. The approach was evaluated on the NSL-KDD public dataset. A comparison of the performance of classifiers and their averaged evaluation using the Weka tool was performed. The NSL-KDD set has been optimized, with an emphasis on "rootkit" type attacks, as one of the most difficult types of attacks to detect. Using the bagging-based approach implemented in the Weka application, it was possible to obtain accuracy results – 99.94%. During the experiment, a tendency of increasing accuracy in the application of bagging on open data was revealed as the volume of training data increases. The proposed approach can be applied in the design of systems for detecting attacks and other abnormal states of information systems. The results of the accuracy of the average assessment require further research in order to improve the indicators. It is possible to modernize the approach of averaging the votes of classifiers by excluding/adding other classifiers, qualitative selection of attributes and their features, increasing the number of training samples for classification.*

Keywords: *bagging; NSL-KDD; vote; Weka; anomaly detection; rootkits; information security.*

Аңдатпа*. Мақалада машиналық оқыту классификаторларын қолдану негізінде желілік трафиктегі ауытқуларды анықтау әдісі ұсынылған. Шешім жеке классификаторлар дауыстарын орташалау арқылы алынған күй класын анықтауға мүмкіндік береді. Тәсіл жалпыға қол жетімді NSL-KDD деректер жиынтығы негізінде бағаланды. Weka құралын қолдана отырып, классификаторлардың өнімділігі мен олардың орташа бағалануы салыстырылды. NSL-KDD жиынтығы "руткит" типті шабуылдарға баса назар аудара отырып, шабуылдардың ең қиын түрлерінің бірі ретінде оңтайландырылды. Weka қосымшасында жүзеге асырылған бэгингке негізделген тәсілді қолдана отырып, 99,94% классификация дәлдігіне қол жеткізілді. Эксперимент барысында оқыту деректерінің көлемі ұлғайған сайын багинг дәлдігінің жоғарлау үрдісі анықталды. Ұсынылған тәсілді шабуылдарды анықтау жүйелерін және ақпараттық*

жүйелердің басқа штаттан тыс күйлерін жобалау кезінде қолдануға болады. Орташа бағалау дәлдігінің нәтижелері көрсеткіштерді жақсарту мақсатында қосымша зерттеулерді қажет етеді. Классификаторлардың дауыстарын орташалау тәсілін басқа классификаторларды алып тастау/қосу, атрибуттар мен олардың ерекшеліктерін сапалы таңдау, классификация үшін оқыту үлгілерінің санын көбейту арқылы жаңартуға болады..

Түйін сөздер*: бэггинг; NSL-KDD; дауыс беру; Weka; ауытқуларды анықтау; руткиттер; ақпараттық қауіпсіздік.*

Аннотация. *В статье предлагается подход к выявлению аномалий в сетевом трафике, основанный на использовании классификаторов машинного обучения. Решение позволяет определить результирующий класс состояния путем усреднения голосов отдельных классификаторов. Подход был оценен на основе общедоступного набора данных NSL-KDD. Было проведено сравнение производительности классификаторов и их усредненной оценки с использованием инструмента Weka. Набор NSL-KDD был оптимизирован с акцентом на атаки типа "руткит", как один из наиболее сложных для обнаружения типов атак. Используя подход, основанный на бэггинге, реализованный в приложении Weka, удалось получить результаты точности классификации - 99,94%. В ходе эксперимента была выявлена тенденция повышения точности применения бэггинга на публичных данных по мере увеличения объема обучающих данных. Предложенный подход может быть применен при проектировании систем обнаружения атак и других нештатных состояний информационных систем. Результаты точности средней оценки требуют дальнейших исследований с целью улучшения показателей. Можно модернизировать подход усреднения голосов классификаторов путем исключения/добавления других классификаторов, качественного отбора атрибутов и их особенностей, увеличения количества обучающих выборок для классификации.*

Ключевые слова: *бэггинг; NSL-KDD; голосование; Weka; обнаружение аномалий; руткиты; информационная безопасность.*

Introduction. The operation of corporate telecommunications networks (CTN) requires continuous monitoring of system failures, conflicts, network equipment errors and information security (IS) incidents. The monitoring systems unfolded for these purposes are collected around the clock and show the events taking place in the CTN.

Meanwhile, the quantity and quality of the methods and techniques used by attackers is growing every day: new types of sophisticated attacks appear, most of which cannot be recognized by existing systems. Along with this, manufacturers of computing equipment and information systems (IS) are not in good loss-vulnerability, which help attackers to constitute their goals.

According to the statistics of the international software development company in the field of information security Positive Technologies [1], attackers are constantly looking for techniques with which they can bypass antiviruses and protection mechanisms built into operating systems (OS). Since the beginning of 2020, attempts have been identified to use the new vulnerability CVE-2020-0601 in Windows CryptoAPI to sign malware (the vulnerability allows bypassing the certificate verification mechanism). Another example is malware for remote management of SysUpdate. This is a unique development of the Bronze Union ART Group, which attackers use to deliver other malicious software (payload) to their controlled devices. As a rule, this payload is not detected by antiviruses, since the file has an undefined format and the antivirus cannot recognize it. Another example is the FakeChmMsi malware with a complex delivery chain of the Gh0st Trojan, during which the DLL hijacking technique is used twice, making it difficult to analyze the malicious software with antivirus protection tools. More than a third (34%) of all attacks on legal entities using malicious software is an attack by encryption Trojans. The operators of these and some other cryptographers have created their own websites on which they publish information stolen from victims in case of refusal to pay a ransom. The share of attacks directed at individuals was 14%. Half of all stolen data are usernames and passwords. This is due to the high proportion of spyware (56%) in malicious campaigns against individuals [2].

One of the causes of such incidents is the fact that existing systems for detecting harmful da-

ta use the signature approach, which implies the identification of viral programs based on the known properties of the virus - its signatures. Although most safety systems develop signatures a secret, the bulk of the signature is in the public domain, and the attackers are well aware of them. To bypass the restrictions, attackers constantly modify their programs and hacking methods, and the signature approach no longer works [3, 4], in the meantime, they will learn about a new type of virus or attack, and companies will release updates to their systems, a lot of precious time will pass, which will be enough to compromise the data and "cover their tracks". Thus, relying solely on signature-based detection may leave organizations vulnerable to new and unknown threats.

In addition, attackers can use various evasion techniques to bypass signature-based detection systems, such as polymorphic malware, obfuscation, and encryption [5]. Polymorphic harmful [6] can change its appearance every time it infects a new system, which makes it difficult to detect using systems based on signatures. Placement includes a change in the malicious software code in order to make it difficult to detect. Encryption [7] can also be used to conceal the malicious code, which complicates the detection of systems based on signatures.

All unknown and new types of potentially negative effects on the CTN at the initial stage of identification are called anomalies, due to the fact that they are clearly different from the normal functioning of the network, but its origin, structure and hazard level for the CTN are unknown. To identify such anomalies, you need an integrated approach and the use of intelligent data processing systems.

Literature Review. Depending on the setting of the task of identifying potentially harmful data in the CTN, three main directions can be distinguished:

– Statistical Methods;

– Machine Learning Methods;

– Rule Based Methods.

One of the common approaches to the detection of anomalies in network traffic is Statistical Methods [8, 9]. This includes an analysis of the statistical properties of data on network traffic to detect unusual patterns or behaviors. For example, an anomaly can be detected if the traffic volume or frequency of certain types of traffic is significantly deviated from the expected levels. An example of the statistical method of detecting anomalies is the use of sliding medium [10, 11] or exponential smoothing [12] to identify trends and anomalies in network traffic data.

Machine Learning Methods [13]–[15] can also be used to detect anomalies in network traffic. These methods include training the model on a large set of network traffic data and using the model to identify unusual patterns or behaviors in new data. For example, clustering methods can determine the current state of IoT devices [16]. Also, an example of a machine learning method for detecting anomalies is the use of neural networks [17, 18], which are able to study complex patterns and relationships in the data.

Rule Based Methods [19, 20] can also be used to detect anomalies in network traffic. These methods include the definition of a set of rules or threshold values that launch alert when fulfilling certain conditions. For example, a rule can be determined to launch a warning if the number of unsuccessful attempts to enter the system exceeds a certain threshold over the specified period of time. An example based on the rules of the method of detecting anomalies is the use of Snort [21] open source invasion systems, which uses a set of predetermined rules to detect various types of network threats.

In conclusion, it should be noted that each study can make a significant contribution in the implementation of abnormalities detection systems, which are an important tool for network administrators and security specialists. They help detect unusual behavior and events in network traffic, reveal threats to security and network performance problems. With the advent of ma-

chine learning and deep learning methods, the accuracy and effectiveness of abnormalities detecting systems continue to increase.

Materials and methods of research. The variety of elements of the Internet of things, a large number of objects, protocols of interaction, data processing technologies, heterogeneity of formats, constantly changing architecture and changes in configuration can lead to various failures and failures of functioning that affect the functioning parameters. Analysis of the values of the statistical parameters of network traffic when performing various operations and commands allows the device to implement monitoring and state control systems.

In this paper, we propose a method for assessing the state based on bagging of classifiers, which, ideally, makes it possible to adjust the weight of the classifier's "voice" in accordance with the analyzed destructive effect on the CTN, "smooth out" statistical data, analyze various encodings.

The formalized description of the proposed approach will look as follows [22, 23].

Let there be many n states of the system $\{z_1, \dots, z_n\} \in Z$, which change in discrete moments of time under the influence of internal and external influences.

At the consistent moments of time $t_0, \dots, t_k$, for the state z_i, measurements of the values of the studied parameters $\{X_i\}$ were made.

The parameter $X_i = (x_1, \dots, x_n)$ contains values of the temporary row of length $n \geq 2$.

The presentation of model looks as follows (1):

$$X_i(t) = F_i[S_i(t), v_i(t)] \quad (1)$$

where the X_i vector is the result of mutually independent signals $S(t)$, which have a distortion of the noise component $v(t)$ in discrete moments of time $t = t_0, \dots, t_k$.

Observations of states are represented by a tuple of variables $X = X_1, \dots, X_n$ formed by several parameters.

Vector X is a time series of tuples of values received from registering devices.

The set of states Z is defined by vectors $X_1, X_2, \dots, X_n$, reflecting the behavior of the process in a multidimensional space. The binary set of classes C, initially divided into subsets dangerous C_1 and safe C_2, is put in accordance with the set of states.

Thus, there is a marked-up final training sample (2):

$$X = \{(x_{11}, \dots, x_{n1}), (x_{12}, \dots, x_{n2}), \dots, (x_{1m}, \dots, x_{nm})\} \quad (2)$$

It is necessary to construct a classification algorithm a_i for the input vector of values X_i, displaying $Z \to C$.

The accordance of the current observation to one of the sets C_1 or C_2 is determined based on the decisive rule $\varphi'(x)$ of the algorithm a_i. It is determined by the function $f(x)$, which generates a partition of space into two disjoint areas (3):

$$\varphi'(x) = \begin{cases} C_1, \text{при } f(x) \geq \varepsilon \\ C_2, \text{при } f(x) < \varepsilon \end{cases} \quad (3)$$

where ε is the threshold value.

In the problems of identifying anomalies in traffic, due to the specifics of the implementation of the function $f(x)$ dividing the space, an error occurs for the classification algorithm, which can be smoothed out by a sequence of k independently trained classifiers $a_i, i = 1, \dots, k$. Then,

$a_i(x_i) \to c_j \in C$ – the answer of the i-th classifier.

$\{P_i(c_j|x_i)\}_{j=0}^{n}$ – a posteriori probability for the i-th classifier after training.

$w_i = \frac{1}{k}$ – weight coefficients.

$a(x) = arg \max_{j=0,\ldots,n} \sum_{i=0}^{k} w_i P_i(c_j|x_i)$ – general classifier.

Results and discussion. The NSL-KDD [24, 25] public dataset was used to evaluate the proposed approach. As part of the experiment, a binary classification of the states of the telecommunications system was carried out (identification of malware such as rootkits and normal traffic). Weka software was used to conduct an experimental evaluation of the approach. The evaluation was performed for the classifiers: Naïve Bayes, Hoeffding Trees, J48, Random Forest, Random True and REPTree. The sample was divided into two parts, one of which was training, and the other was used for testing.

The data structure consisted of a vector of more than 40 attribute values, the description of which is presented in Table 1.

Table 1. Description of the features of the NSL-KDD dataset

№	Feature	Description
1	Duration	Duration of connection time
2	Protocol_type	The type of connection protocol
3	Service	The network service utilized by the destination host
4	Flag	Connection state
5	Src_bytes	The quantity of data bytes that the source sent to the recipient within a one connection
6	Dst_bytes	The quantity of data bytes that the recipient sent to the source within a one connection
7	Land	1 if the source address and port and destination address and port match, 0 if they do not match
8	Wrong_fragment	The overall number of incorrect fragments within a single connection
9	Urgent	The number of network packets with an urgency bit
10	Hot	The number of important indicators in the data
11	Num_failed_logins	How many failed attempts of login have been made
12	Logged_in	1 if the login attempt was successful, otherwise 0
13	Num_compromised	The quantity of compromised state
14	Root_shell	1 if root level access is obtained 0 if not
15	Su_attempted	If the “su root” command was applied then the value is 1; 0 if not
16	Num_root	How many attempts to connect to the root level were made or performed operations from the root level within one session
17	Num_file_creations	The quantity of operations to create a file when connecting
18	Num_shells	How many shell hints were revealed
19	Num_access_files	How many manipulations with access control files have been done
20	Num_outbound_cmds	Numbering of outgoing commands if it was ftp connection
21	Is_hot_login	1 if the connection attempt refers to the root level; otherwise 0
22	Is_guest_login	1 if the connection attempt is at the guest level; 0 if not
23	Count	How many connections were there to the same host within the last couple of seconds
24	Srv_count	How many connections have there been to the same service by port number as the actual connection in the last couple of seconds
25	Serror_rate	What is the percentage of sessions in which the flags (4) s0, s1, s2 or s3 were activated in the session combined into count (23)
26	Srv_serror_rate	What is the percentage of sessions in which the flags (4) s0, s1, s2 or s3 were activated in the session combined into srv_count (24)
27	Rerror_rate	What is the percentage of sessions in which the REJ flag (4) were activated in the session combined into count (23)

№	Feature	Description
28	Srv_rerror_rate	What is the percentage of sessions in which the REJ flag (4) were activated in the session combined into srv_count (24)
29	Same_srv_rate	What is the percentage of sessions to the same resource, in the session combined into count (23)
30	Diff_srv_rate	What is the percentage of sessions to various resources, in the connections combined into count (23)
31	Srv_diff_host_rate	What is the percentage of sessions that were to different destination hosts in the sessions combined into srv_count (24)
32	Dst_host_count	The number of sessions with the same target IP address
33	Dst_host_srv_count	The number of sessions with the same number of port
34	Dst_host_same_srv_rate	What is the percentage of sessions to the same resource, in the sessions combined into dst_host_count (32)
35	Dst_host_diff_srv_rate	What is the percentage of sessions to any services, in sessions combined in dst_host_count (32)
36	Dst_host_same_src_port_rate	What is the percentage of sessions have there been to the same source port in the sessions combined in dst_host_srv_count (33)
37	Dst_host_srv_diff_host_rate	What is the percentage of sessions have there been to different target hosts in the sessions combined in dst_host_srv_count (33)
38	Dst_host_serror_rate	What is the percentage of sessions in which the flags (4) s0, s1, s2 or s3 were activated in the sessions combined in dst_host_count (32)
39	Dst_host_srv_serror_rate	What is the percentage of sessions in which the flags (4) s0, s1, s2 or s3 were activated in the sessions combined in dst_host_srv_count (33)
40	Dst_host_rerror_rate	What is the percentage of sessions in which the flag (4) REJ flag was activated in the sessions combined in dst_host_count (32)
41	Dst_host_srv_rerror_rate	What is the percentage of sessions in which the flag (4) REJ flag was activated in the sessions combined in dst_host_srv_count (33)
42	Class	Class of data

The classifiers were evaluated based on the Precision (5) and Recall (6) indicators:

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (6)$$

where *TP* is a true-positive solution of the classifier, *TN* is a true-negative solution, *FP* is a false-positive solution, *FN* is a false-negative solution.

The results of the evaluation of Precision and Recall obtained during the experiment are shown in (Table 2) and (Table 3).

Table 2. Precision for various classifiers

№	Class	Naïve Bayes, %	Hoeffding Tree, %	J48 %	Random Forest, %	Random Tree, %	REP Tree, %
1	normal	96,9728	100	99,9794	100	99,9588	99,9588
2	rootkit	83,3333	0	50	16,6666	50	83,3333

Table 3. Recall for various classifiers

№	Class	Naïve	Hoeffding	J48	Random	Random	REP Tree,

		Bayes, %	Tree, %	%	Forest, %	Tree, %	%
1	normal	0,970	1,000	1,000	1,000	1,000	1,000
2	rootkit	0,833	0,000	0,500	0,167	0,500	0,833

The overall accuracy score of classifiers is determined by the expression (7):

$$Accuracy = \frac{P}{N} \times 100\% \quad (7)$$

where, P is the number of entries for which the classifier made the correct decision, and N is the size of the training sample.

The accuracy values for binary classification are given in (Table 4):

Table 4. Accuracy for various classifiers

Class	Naïve Bayes, %	Hoeffding Tree, %	J48, %	Random Forest, %	Random Tree, %	REP Tree, %
ALL	96,956	99,8766	99,9177	99,8972	99,8972	99,9383

Figure 1 shows a visual representation of the accuracy results.

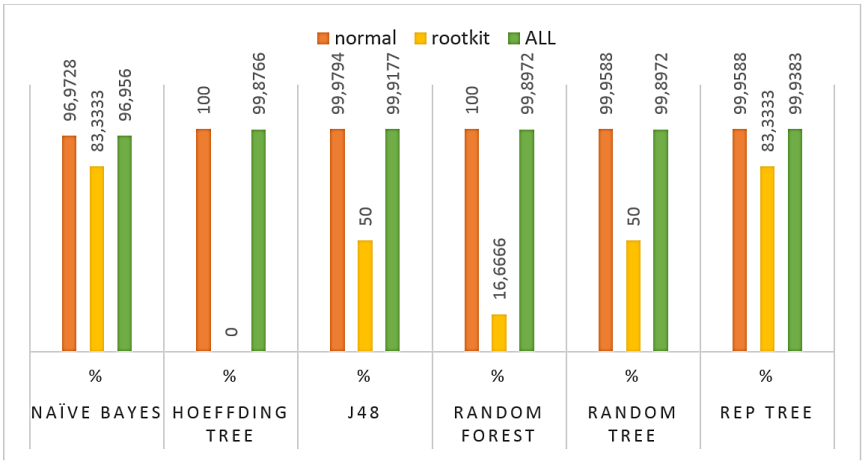

Figure 1. Accuracy indicators for various classifiers

In the second part of the experiment, a sequence of classifiers was implemented to average errors.

The set was divided in the ratio of 20/80, 40/60, 60/40 and 80/20, where the first part shows the ratio of entries for training, and the second for testing. The incoming data were simultaneously processed by all classifying algorithms. The results obtained after the application of bagging, which determines the resulting state class by averaging the voting values, are given in the Table 5.

Table 5. Accuracy for bagging of classifiers

Ratio	20/80	40/60	60/40	80/20
Correctly classified entries	99,8457	99,8629	99,8715	99,9486

Incorrectly classified entries	0,1543	0,1371	0,1285	0,0514

Thus, the results of testing an open NSL-KDD dataset with machine learning classifiers implemented in the Weka application using bagging show accuracy results of 99.94%.

During the experiment, sufficiently "strong" classifiers were selected, however, even on such a set when using bagging, as the volume of the training sample increases, a certain the growth of accuracy.

Conclusion. The exponential growth of information requires the improvement of models and methods of its analysis for the detection of destructive effects. At the same time, new types of attacks appear, and their detection by modern means becomes more problematic. It is necessary to analyze a larger number of network traffic parameters to identify such impacts.

The paper proposes an approach to identifying abnormal situations in network traffic based on the use of bagging classifiers. Given the fact that a significant number of traffic indicators are being processed, the proposed approach has shown acceptable results without pre-processing the data by smoothing out possible errors with several machine learning classifiers.

The main advantage of the proposed approach is the possibility of scaling and combining it by adding new classification algorithms. Also, it is possible to make changes to the weight coefficients, which allows improving the accuracy of identifying potential negative impacts.

Список литературы

1. Позитивные технологии / ru.wikipedia.org. 11.08.2023. URL: https://ru.wikipedia.org/?curid=2048197&-oldid=132289638.
2. Угрозы кибербезопасности Q1 2020 / ptsecurity.com. 16.06.2020. URL: https://www.ptsecurity.-com/upload/corporate/ww-en/analytics/cybersecurity-threatscape-2020-q1-eng.pdf.
3. Яманиши К., Такеучи Дж., Маруяма Ю. Интеллектуальный анализ данных для обеспечения безопасности, предотвращения фальсификации и защиты технологий и продуктов / nec.com. 18.09.2023. URL: https://www.nec.com/en/global/techrep/journal/g05/n01/pdf/a063.pdf.
4. Аль-Джаллад К., Альджниди М., Десуки М.С. Оптимизация обнаружения аномалий с использованием больших данных и глубокого обучения для уменьшения ложноположительных результатов // J Большие данные 7. – 2020. № 68. doi: 10.1186/s40537-020-00346-1.
5. Фрейли Б., Джеймс Улучшенное обнаружение продвинутых полиморфных вредоносных программ. Докторская диссертация. Юго-Восточный университет Новой Зеландии / nsuworks.nova.edu. 26.07.2023. URL: https://nsuworks.nova.edu/gscis_etd/1008.
6. Альрзини Дж., Пеннингтон Д. Обзор методов обнаружения полиморфных вредоносных программ. Международный журнал передовых исследований в области инженерии и технологий. № 11 (12). С. 1238-1247. 26.07.2023. URL: http://iaeme.com/home/issue/ijaret?volume=11&issue=12.
7. Дханасекар Д. Обнаружение зашифрованных вредоносных программ с использованием скрытых моделей Маркова. Магистерские проекты. № 574. doi: https://doi.org/10.31979/etd.qhm6-sn26. 08.06.2023. URL: https://scholarWorks.sjsu.edu/etd_projects/574.
8. Иглесиас Васкес Ф., Зсеби, Т. Анализ характеристик сетевого трафика для обнаружения аномалий // Машинное обучение 101. – 2014. № 101. С. 59-84. doi: 10.1007/s10994-014-5473-9.
9. Лю Х., Ван Х. Обнаружение аномалий сетевого трафика в реальном времени на основе CNN // Симметрия. – 2023. 15(6):1205. doi: https://doi.org/10.3390/sym15061205.
10. Чжан М., Го Дж., Ли Х., Цзинь Р. Основанный на данных подход к обнаружению аномалий для потоковых данных временных рядов // Датчики (Базель). – 2020. 20(19):5646. doi: 10.3390/s20195646.
11. Чжоу З.-Г., Тан П. Улучшение обнаружения аномалий временных рядов на основе экспоненциально взвешенного скользящего среднего (EWMA) остатков модели сезонного тренда // Международный симпозиум IEEE по геонаукам и дистанционному зондированию (IGARSS). – 2016. С. 3414-3417. doi: 10.1109/IGARSS.2016.7729882doi: 10.1109/IGARSS.2016.7729882.

12. Тан Х., Ван К., Цзян Г. Модель обнаружения аномалий временных рядов, основанная на мультифункциональности // Компьютерная нейробиология. – 2022. doi: 10.1155/2022/2371549.
13. Вехтер Э. В., Касап С., Колозали Ш., Чжай Х., Эхсан С., Макдональд-Майер, К. Д. Использование машинного обучения для обнаружения аномалий в системе на чипе под воздействием гамма-излучения // Ядерная инженерия и технологии. – 2022. 54 (11). С. 3985-3995. Предварительная онлайн-публикация. doi: https://doi.org/10.1016/j.net.2022.06.028.
14. Нассиф А.Б., Талиб М.А., Насир К., Дакалбаб, Ф.М. Машинное обучение для обнаружения аномалий: систематический обзор // Доступ к IEEE, 9. – 2021. С. 78658-78700. doi: 10.1109/ACCESS.2021.3083060.
15. Тудуму С., Бранч П., Джин Дж. и др. Всесторонний обзор методов обнаружения аномалий для больших данных большой размерности // J Большие данные 7. – 2020. № 42. doi: https://doi.org/10.1186/s40537-020-00320-x.
16. Сухопаров М. Е., Лебедев И. С. Определение состояния информационной безопасности устройств Интернета вещей в информационных и телекоммуникационных системах // Системы управления, связи и безопасности. – 2020. № 3. С. 252-268. doi: 10.24411/2410-9916-2020-10310.
17. Стаар Б., Лютьен М., Фрейтаг М. Обнаружение аномалий с помощью сверточных нейронных сетей для промышленного контроля поверхности // Procedia CIRP. – 2019. Том 79. С. 484-489. ISSN 2212-8271. doi: https://doi.org/10.1016/j.procir.2019.02.123.
18. Алдахул Н., Абдул Карим Х., Ба Вазир А.С. Моделирование слияния глубоких нейронных сетей для обнаружения аномалий // J Большие данные 8. – 2021. № 106. doi: https://doi.org/10.1186/s40537-021-00496-w.
19. Эльфаки А. Использование метода, основанного на правилах, для обнаружения аномалий в линейке программных продуктов // Исследовательский журнал прикладных наук, инженерии и технологий. 7. – 2014.
20. Даффилд Н., Хаффнер П., Кришнамурти Б., Рингберг, Х. Обнаружение аномалий в потоках IP на основе правил // IEEE INFOCOM 2009. – 2009. Рио-де-Жанейро. Бразилия. С. 424-432. doi: 10.1109/INFCOM.2009.5061947.
21. Шмит М., Венгик Р., Сковронски М., Шмит, А. Обнаружение аномалий дорожного движения с помощью Snort. 2007.
22. Семенов В.В., Лебедев И.С., Сухопаров М.Е. Подход к классификации состояния информационной безопасности элементов киберфизических систем путем применения побочного электромагнитного излучения // Научно-технический журнал информационных технологий, механики и оптики. – 2018. – Том 18. – № 1. – С. 98-105. doi: 10.17586/2226-1494-2018-18-1-98-105.
23. Сухопаров М. Е., Лебедев И. С. Определение статуса информационной безопасности устройств Интернета вещей в информационных и телекоммуникационных системах // Системы управления, связи и безопасности. – 2020. – № 3. – С. 252-268. doi: 10.24411/2410-9916-2020-10310.
24. Бхупендра И., Анамика Ю. Анализ производительности набора данных NSL-KDD с использованием ANN // 2015 Международная конференция по инженерным системам обработки сигналов и связи. – 2015. – С. 92-96. doi: 10.1109/SPACES.2015.7058223.
25. Дханабал Л., Шантараджа Д-р С.П. Исследование набора данных NSL-KDD для системы обнаружения вторжений, основанной на алгоритмах классификации // Международный журнал передовых исследований в области компьютерной и коммуникационной инженерии. – 2015. Том 4 (6). – С. 446-452. doi: 10.17148/IJARCCE.2015.4696.

References

1. Positive Technologies. ru.wikipedia.org. 11.08.2023. Available at: https://ru.wikipedia.org/?curid=2048197&oldid=132289638.
2. Positive Technologies Q1 2020. ptsecurity.com. 16.06.2020. Available at: https://www.ptsecurity.-com/ru-ru/research/analytics/cybersecurity-threatscape-2020-q1.
3. Yamanishi K., Takeuchi J., Maruyama Y. Data Mining for Security, Falsification Prevention and Protection Technologies and Products. NEC Journal of Advanced Technology. nec.com. 18.09.2023. Available at: https://www.nec.com/en/global/techrep/journal/g05/n01/pdf/a063.pdf.
4. Al Jallad K., Aljnidi M., Desouki M.S. Anomaly detection optimization using big data and deep learning to reduce false-positive. J Big Data 7, 2020, no. 68. doi: 10.1186/s40537-020-00346-1.
5. Fraley B. James Improved Detection for Advanced Polymorphic Malware. Doctoral dissertation.

Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing. (1008). 2017. nsuworks.nova.edu. 26.07.2023. Available at: https://nsuworks.nova.edu/gscis_-etd/1008.
6. Alrzini J., Pennington D. A Review of Polymorphic Malware Detection Techniques. International Journal of Advanced Research in Engineering and Technology, 2020, no. 11 (12), pp. 1238-1247. Available at: http://iaeme.com/home/issue/ijaret?volume=11&issue=12.
7. Dhanasekar D. Detecting Encrypted Malware Using Hidden Markov Models. Master's Projects. 2017. doi: https://doi.org/10.31979/etd.qhm6-sn26. Available at: https://scholarWorks.sjsu.edu/etd_-projects/574
8. Iglesias Vázquez F., Zseby T. Analysis of network traffic features for anomaly detection. Machine Learning, 2014, no. 101. doi: 10.1007/s10994-014-5473-9.
9. Liu H., Wang H. Real-Time Anomaly Detection of Network Traffic Based on CNN. Symmetry, 2023, no. 15(6):1205. doi: https://doi.org/10.3390/sym15061205.
10. Zhang M., Guo J., Li X., Jin R. Data-Driven Anomaly Detection Approach for Time-Series Streaming Data. Sensors (Basel), 2020, no. 20(19):5646. doi: 10.3390/s20195646.
11. Zhou Z.-G., Tang P. Improving time series anomaly detection based on exponentially weighted moving average (EWMA) of season-trend model residuals. 2016. doi: 10.1109/IGARSS.2016.7729882.
12. Tang H., Wang Q., Jiang G. Time Series Anomaly Detection Model Based on Multi-Features. Computational Intelligence and Neuroscience, 2022. doi: 10.1155/2022/2371549.
13. Wächter E. W., Kasap S., Kolozali Ş., Zhai X., Ehsan S., McDonald-Maier K. D. Using Machine Learning for Anomaly Detection on a System-on-Chip under Gamma Radiation. Nuclear Engineering and Technology, 2022, no. 54(11), 3985-3995. Available at: https://doi.org/10.1016/j.net.2022.06.028.
14. Nassif A.B., Talib M.A., Nasir Q., Dakalbab F.M. Machine Learning for Anomaly Detection: A Systematic Review. IEEE Access, 2021, no. 9, 78658-78700, pp. 1-1. doi: 10.1109/ACCESS.2021.3083060.
15. Thudumu S., Branch P., Jin J. et al. A comprehensive survey of anomaly detection techniques for high dimensional big data. J Big Data 7, 2020, no. 42. Available at: https://doi.org/10.1186/s40537-020-00320-x.
16. Sukhoparov M. E., Lebedev I. S. Identification of the state of information security of Internet of Things devices in information and telecommunication systems. Control systems, communications and security, 2020, no. 3, pp. 252-268. doi: 10.24411/2410-9916-2020-10310.
17. Staar B., Lütjen M., Freitag M. Anomaly detection with convolutional neural networks for industrial surface inspection. Procedia CIRP, 2019. – Vol. 79, – Pp. 484-489. ISSN 2212-8271. Available at: https://doi.org/10.1016/j.procir.2019.02.123.
18. AlDahoul N., Abdul Karim H., Ba Wazir A.S. Model fusion of deep neural networks for anomaly detection. J Big Data 8, 2021, 106. doi: https://doi.org/10.1186/s40537-021-00496-w.
19. Elfaki A. Using a Rule-based Method for Detecting Anomalies in Software Product Line. Research Journal of Applied Sciences, Engineering and Technology. 7, 2014.
20. Duffield N., Haffner P., Krishnamurthy B., Ringberg, H. Rule-Based Anomaly Detection on IP Flows. IEEE INFOCOM 2009, 2009, Rio de Janeiro. Brazil. – Pp. 424-432. doi: 10.1109/INFCOM.2009.5061947.
21. Szmit M., Wężyk R., Skowroński M., Szmit, A. Traffic anomaly detection with Snort, 2007.
22. Semenov V.V., Lebedev I.S., Sukhoparov M.E. Approach to classification of the information security state of elements for cyber-physical systems by applying side electromagnetic radiation. Scientific and Technical Journal of Information Technologies, Mechanics and Optics, 2018, vol. 18, no. 1, pp. 98–105 (in Russian). doi: 10.17586/2226-1494-2018-18-1-98-105.
23. Sukhoparov M. E., Lebedev I. S. Identification the Information Security Status for the Internet of Things Devices in Information and Telecommunication Systems. Systems of Control, Communication and Security, 2020, no. 3, pp. 252-268 (in Russian). doi: 10.24411/2410-9916-2020-10310.
24. Bhupendra I., Anamika, Y. Performance Analysis of NSL-KDD dataset using ANN. 2015 International Conference on Signal Processing And Communication Engineering Systems (SPACES), 2015, pp. 92-96. doi: 10.1109/SPACES.2015.7058223.
25. Dhanabal L., Shantharajah Dr. S.P. A Study on NSL-KDD Dataset for Intrusion Detection System Based on Classification Algorithms. International Journal of Advanced Research in Computer and Communication Engineering, 2015, vol. 4 (6), pp. 446-452. doi: 10.17148/IJARCCE.2015.4696.