Современная наука и инновации. 2026. № 1. С. 55-68
ТЕХНИЧЕСКИЕ НАУКИ
ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

Modern Science and Innovations. 2026;(1):55-68
TECHNICAL SCIENCES
INFORMATION, COMPUTING AND MANAGEMENT

Научная статья
УДК 004.8:004.056.5
https://doi.org/10.37493/2307-910X.2026.1.4

Методы машинного обучения для выявления атак на систему управления промышленного Интернета вещей

Юлия Алексеевна Андрусенко[1], Глеб Алексеевич Семенов[2*], Артем Алексеевич Соломянко[3], Алиса Андреевна Кущенко[4], Кристина Юрьевна Серебренникова[5]

[1, 2, 3, 4, 5] Северо-Кавказский федеральный университет (д. 1, ул. Пушкина, Ставрополь, 355017, Россия)
[1] iuandrusenko@ncfu.ru, https://orcid.org/0000-0002-3392-7270
[2] glebsemenov2003@gmail.com, https://orcid.org/0009-0004-1850-9021
[3] artemixol@xmail.ru, https://orcid.org/0009-0002-3378-6743
[4] alisakuschenko@yandex.ru, https://orcid.org/0009-0008-9480-4072
[5] serebrennikova0512@gmail.com
*Автор, ответственный за переписку

Аннотация. ***Введение.*** В статье рассматривается задача обнаружения кибератак в инфраструктуре промышленного Интернета вещей (IIoT), актуальность которой обусловлена ростом числа подключённых промышленных устройств и увеличением атак на критические системы управления. ***Материалы и методы.*** Для проведения исследования использован набор данных, сформированный на основе сетевого трафика и содержащий 84 признака и 67 267 записей. В работе применялись методы предварительной обработки данных, отбор информативных признаков, анализ главных компонентов (PCA), а также методы машинного обучения, включая Decision Tree, K Nearest Neighbor, Tree Ensemble, PNN Learner и Gradient Boosted Trees. Для оценки качества моделей использовалась кросс-валидация и ограничение глубины деревьев. ***Результаты и обсуждение.*** Проведено сравнение эффективности различных моделей классификации. Установлено, что применение PCA и оптимизация параметров моделей позволяет повысить точность обнаружения атак. Наилучший результат показала модель Tree Ensemble, обеспечившая точность классификации 97,5% при использовании 17 главных компонентов. ***Заключение.*** Полученные результаты подтверждают перспективность применения методов машинного обучения для выявления кибератак в системах промышленного Интернета вещей и могут быть использованы при построении систем мониторинга и защиты IIoT-инфраструктуры.

Ключевые слова: машинное обучение; Интернет вещей; кибератаки; метод главных компонент; дерево решений, переобучение модели, кросс-валидация, промышленный Интернет вещей, KNIME, модель, глубина дерева, эффективность модели, CICFlowMeter.
Для цитирования: Андрусенко Ю. А., Семенов Г. А., Соломянко А. А., Кущенко А. А., Серебренникова К. Ю. Методы машинного обучения для выявления атак на систему управления промышленного Интернета вещей // Современная наука и инновации. 2026. № 1. С. 55–68. https://doi.org/10.37493/2307-910X.2026.1.4

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов

Статья поступила в редакцию 01.12.2025;
одобрена после рецензирования 01.02.2026;
принята к публикации 01.03.2026.

Research article

Machine learning methods for identifying attacks on the control system of the industrial Internet of things

Yuliya A. Andrusenko[1], Gleb A.Semenov[2*], Artem A. Solomyanko[3], Alisa A. Kushchenko[4], Kristina Y. Serebrennikova[5]

[1, 2, 3, 4, 5] North Caucasus Federal University (1, Pushkin st., Stavropol, 355017, Russia)
[1] iuandrusenko@ncfu.ru, https://orcid.org/0000-0002-3392-7270
[2] glebsemenov2003@gmail.com, https://orcid.org/0009-0004-1850-9021
[3] artemixol@xmail.ru, https://orcid.org/0009-0002-3378-6743
[4] alisakuschenko@yandex.ru, https://orcid.org/0009-0008-9480-4072
[5] serebrennikova0512@gmail.com
*Corresponding author

Abstract. ***Introduction.*** The article addresses the problem of detecting cyberattacks in the infrastructure of the Industrial Internet of Things (IIoT), the relevance of which is driven by the rapid growth of connected industrial devices and the increasing number of attacks on critical control systems. The aim of the study is to analyze and compare machine learning methods to improve the effectiveness of attack detection in IIoT networks. ***Materials and methods.*** The study is based on a network traffic dataset containing 84 features and 67,267 records. Data preprocessing techniques, informative feature selection, and Principal Component Analysis (PCA) were applied. Several machine learning algorithms were investigated, including Decision Tree, K-Nearest Neighbors, Tree Ensemble, PNN Learner, and Gradient Boosted Trees. Model performance was evaluated using cross-validation and tree depth limitation to mitigate overfitting. ***Results and discussion.*** A comparative analysis of classification models was conducted. The results show that the use of PCA and model parameter optimization significantly improves attack detection accuracy. The best performance was achieved by the Tree Ensemble model, which reached a classification accuracy of 97.5% when using 17 principal components. ***Conclusion.*** The obtained results confirm the effectiveness of machine learning approaches for building intrusion detection and security monitoring systems in Industrial Internet of Things environments.

Keywords: machine learning; Internet of Things (IoT); cyberattacks; Principal Component Analysis (PCA); decision tree; model overfitting; cross-validation; Industrial Internet of Things (IIoT); KNIME; model; tree depth; model performance, CICFlowMeter.
For citation: Andrusenko YA, Semenov GA, Solomyanko AA, Kushchenko AA, Serebrennikova KY. Machine learning methods for identifying attacks on the control system of the industrial Internet of things. *Modern Science and Innovations*. 2026;(1):55-68. https://doi.org/10.37493/2307-910X.2026.1.4

Conflict of interest: the autors declare no conflicts of interests.

The article was submitted 01.12.2025;
approved after reviewing 01.02.2026;
accepted for publication 01.03.2026.

Introduction. The Internet of Things (IoT) is a system of interconnected computer networks and connected physical objects (things) with embedded sensors and software for collecting and exchanging data, with the ability to be remotely monitored and controlled in an automated manner, without human intervention.

Industrial Internet of Things (IIoT) is a system of interconnected computer networks and connected industrial facilities with built-in sensors and software for collecting and exchanging data, with the ability to remotely monitor and control in an automated mode, without human intervention [10]. General model of the Industrial Internet of Things architecture (Fig. 1).

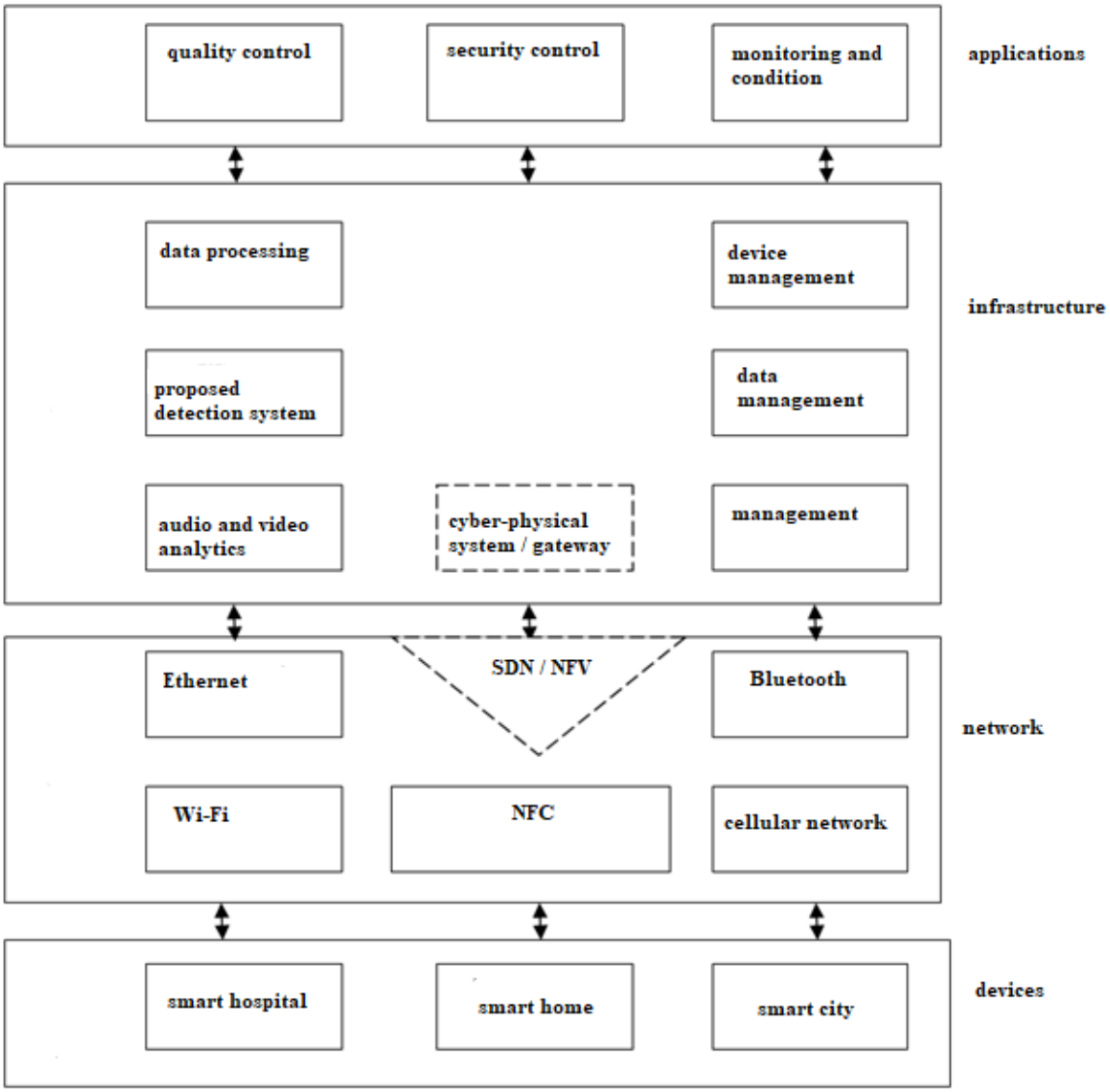

Figure 1. A generalizing model of the industrial Internet of things architecture.

The history of the emergence of the term IIoT officially began with the birth of the concept of “inter

The Internet of Things (IoT) was first coined in 1999 by researcher Kevin Ashton, who introduced the idea during a presentation to Procter & Gamble. Although IoT technologies have been around for over two decades, active interest in them emerged much more recently and continues to grow. Cisco experts associate the period 2008–2009 with the starting point for the Industrial Internet of Things (IIoT). According to their analysis, it was during these years that a key turning point occurred: the number of networked machines and sensors exceeded the Earth's population for the first time. This transitional period, when the "Internet of People" gave way to the "Internet of Things," laid the foundation for the integration of smart devices into production chains and critical infrastructure management. In the industrial context, this marked the beginning of an era in which the primary participants in network interactions were no longer personal gadgets, but equipment, sensors, and automated enterprise systems. Machine learning (ML) methods are central to the creation of algorithms aimed at classifying and detecting cyberattacks [4].

Machine learning is an approach in which an artificial intelligence system autonomously improves its performance by processing arrays of training data without direct developer intervention [1]. ML technologies provide effective tools for automated network traffic monitoring and anomaly detection. However, finding optimal models and methods for classifying network packets remains a pressing issue.

Materials and research methods. To conduct a study of machine learning methods aimed at detecting cyberattacks in the Industrial Internet of Things (IIoT) control system, a network traffic dataset and the KNIME graphical programming analytics platform [3, 11] were used. This section presents the characteristics of the dataset used, the data preprocessing methods, the tools used, and the approaches to training machine learning models [5, 14].

The study utilized a dataset obtained from pcap source files using the CICFlowMeter tool. This tool allows for the extraction of statistical characteristics of network flows and the formation of a structured dataset suitable for analysis and machine learning [5].

The dataset used includes both numerical and categorical features, such as Flow ID, Src IP, Dst IP, and Timestamp. The total dataset size is 67,267 rows and 84 columns [5]. The data represents various classes of network attacks, including DoS, DDoS, Probe, and web attacks, which are detected at the network level [3].

The presented dataset contains a significant number of features that are not key for analysis and the construction of effective machine learning models. Therefore, informative feature selection was performed during the preprocessing stage.

To reduce the data dimensionality, a special Column Filter node in the KNIME environment [14] was used. As a result of selection, the number of features was reduced from 84 to 20 without significant loss of information. The columns retained after selection are presented in (Table 1).

Table 1. Description of the dataset columns

Number	Attribute	Description	Data type
1	Flow Duration	Network flow duration	Numerical
2	Flow Bytes/s	Average data transfer rate	Numerical
3	Flow Packets/s	Average packet transmission rate	Numerical
4	Packet Length Mean	Average package size	Numerical
5	Packet Length Standard	Standard deviation of batch size	Numerical
6	SYN Flag Count	Number of SYN flags	Numerical
7	ACK Flag Count	Number of ACK flags	Numerical
8	RST Flag Count	Number of RST flags	Numerical
9	Down/Up Ratio	Ratio of incoming and outgoing traffic	Numerical
10	Average Packet Size	Average package size	Numerical
11	Fwd Packet Length Mean	Average packet size (forward direction)	Numerical
12	Bwd Packet Length Mean	Average packet size (reverse direction)	Numerical
13	Flow IAT Mean	Average interval between packets	Numerical
14	Flow IAT Std	Standard deviation of intervals	Numerical
15	Fwd IAT Mean	Inter-packet interval (forward direction)	Numerical
16	Bwd IAT Mean	Inter-packet interval (reverse direction)	Numerical
17	Active Mean	Average thread activity time	Numerical
18	Idle Mean	Average thread idle time	Numerical
19	Subflow Fwd Bytes	Substream data volume (forward direction)	Numerical
20	Subflow Bwd Bytes	Substream data volume (reverse direction)	Numerical

The presented dataset contains a preponderance of columns with data that is not key to the analysis and model development, so it was necessary to select the columns containing the necessary data. The "Column Filter" node was used to select the required columns. This selection reduced the number of columns from 84 to 20, which also simplifies model creation and training without losing information.

Thus, using this node allows us to improve the quality of the model, speed up its training and reduce computational costs.

Since the dataset has 84 columns, some of which are multicaliber (linearly dependent), it was necessary to simplify the model and eliminate the multicaliber. The "PCA" node was used to transform these columns. The values of these columns were reduced and converted into a new set of independent features.

Thus, the use of this node helps to reduce the dimensionality of the data, preserving 90-95% of the information, copes with the multiplication of columns, and simplifies the model.

To assess the impact of feature selection on model quality, a graph of F1 Score values before and after data selection was plotted (figure 2). The graph shows that some values after feature selection demonstrate a decrease in accuracy; however, overall, selection simplifies the model and reduces computational costs.

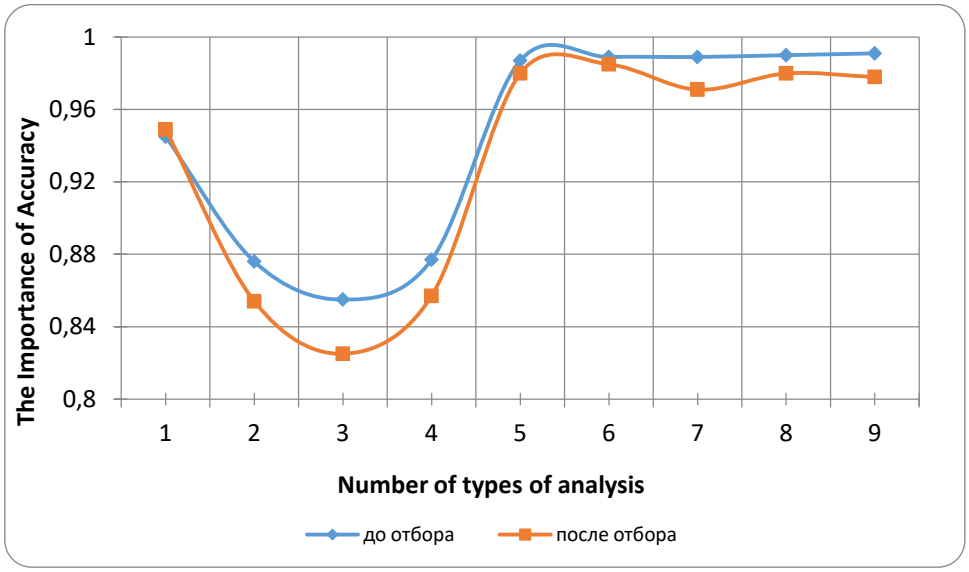

Figure 2. F1 Score graph before and after data selection

Thus, using the Column Filter node allows you to improve the quality of models, speed up the training process and reduce the complexity of calculations.

The KNIME Analytics Platform (Konstanz Information Miner) graphical programming platform was used to analyze data and build machine learning models. This environment provides a wide range of tools for data processing, model building, visualization of results, and experimental research.

The following KNIME nodes were used in the study:

File Reader, Column Filter, Normalizer, PCA, X-Partitioner, Learner, Predictor, X-Aggregator, and Scorer. Their purpose and functionality are presented in Table 2.

Table 2. Used nodes

Node name	Purpose
File Reader	Reads most common text files
Column Filter	Allows you to filter columns by data characteristics
Normalizer	Normalizes the values of all selected (numeric) columns
PCA	Performs principal component analysis on given data, reducing the dimensionality of the data while preserving information
X-Partitioner	It is the first in the cross-validation cycle
Learner	Builds a predictive model based on the received data. Responsible for training
Predictor	Responsible for applying the resulting model to test data
X-Aggregator	This is the last step in the cross-validation cycle. It collects data from the predictor, compares the predicted and actual classes, and outputs predictions for all rows and iteration statistics
Scorer	Displays the error matrix and reports a range of statistics such as recall, precision, accuracy, etc.

The study identified a problem with overfitting machine learning models, whereby algorithms performed well on the training set and significantly worse on the test data [2].

To combat overfitting, cross-validation – a method for evaluating a model on independent subsamples – was used [2]. The method involves splitting the original dataset into several parts, one of which is used for testing, and the rest for training the model. The process is repeated for all parts, and the final score is calculated as the average value across all iterations.

In the KNIME environment, cross-validation was implemented using X-Partitioner and X-Aggregator nodes, which automatically orchestrated the validation cycle. The number of validations

was set to five. Additionally, the tree depth in decision tree-based models was limited, ranging from 3 to 10, which also helped reduce the effects of overfitting.

Thus, the use of cross-validation and control of tree depth allowed us to increase the robustness of the models and provide a more objective assessment of their quality.

As part of the study, five machine learning models with different parameters were built and analyzed on the KNIME platform. him These include Decision Tree Learner, K-Nearest Neighbor, Tree Ensemble, PNN Learner , and Gradient Boosted Trees. These models were chosen due to their widespread use and effectiveness in solving network traffic classification problems.

For each model, parameter tuning was performed, and the impact of data preprocessing methods, including normalization and principal component analysis, was analyzed. Model quality was assessed using standard classification metrics such as accuracy, precision, recall, and F-measure [2].

Results and discussion. Experimental studies aimed to evaluate the effectiveness of various machine learning models in detecting attacks in Industrial Internet of Things (IoT) control systems. The impact of data preprocessing methods, model parameters, and the use of principal component analysis on classification accuracy was examined.

Decision Tree Learner is a machine learning method based on sequentially partitioning data into subsets and predicting classes based on given conditions. Decision Tree models do not use data normalization [9].

The study compared Decision Tree models using and without the principal component analysis (PCA). Several models were constructed with PCA component counts ranging from 17 to 3, with a step size of 2–3, as well as a model without PCA. The Accuracy metric values were used to plot the graph shown in Figure 3.

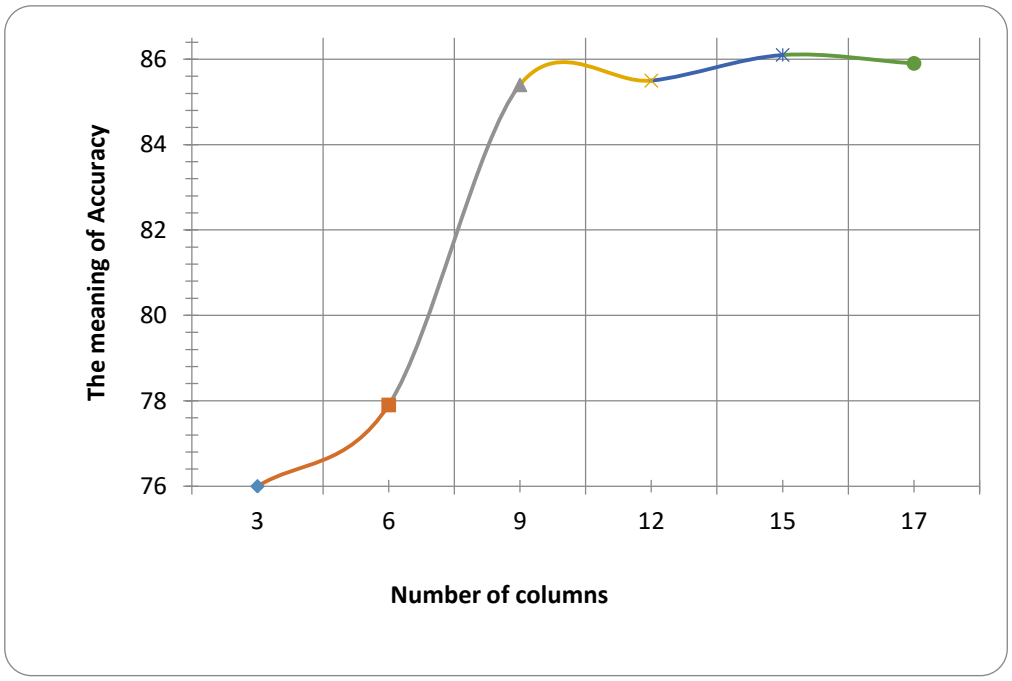

Figure 3. Accuracy parameter for the Decision Tree model using PCA with different numbers of columns

An analysis of the results showed that the use of PCA did not lead to a significant increase in classification accuracy for this model. Next, a study was conducted to examine the influence of the Decision Tree parameters without using PCA. such Parameters such as Quality measure (Gain ratio and Gini index) and Pruning method (No pruning and MDL) were used. Based on the experimental results, a graph was constructed, shown in Figure 4.

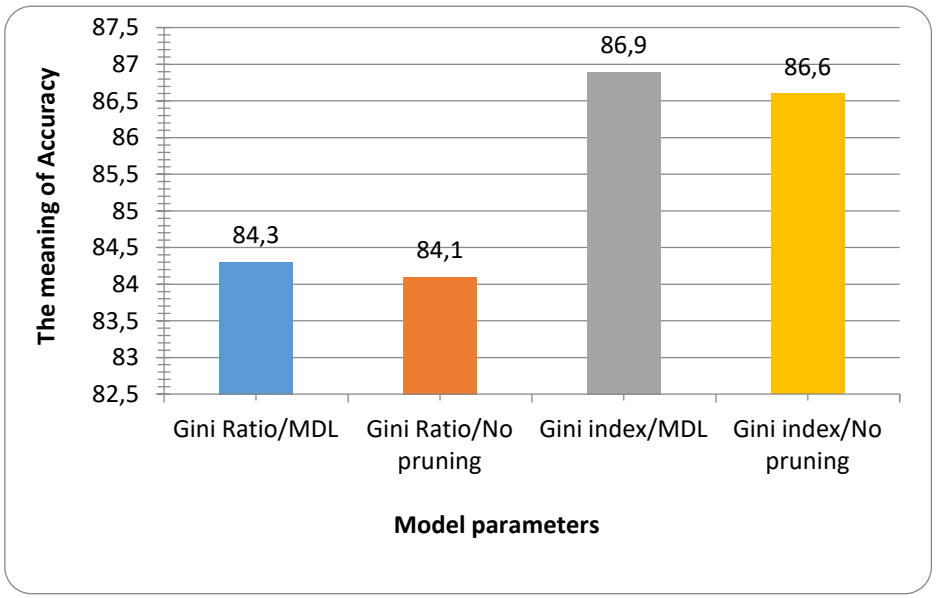

Figure 4. Accuracy parameter for the Decision model Tree without PCA

The results showed that the best accuracy for the Decision Tree model was achieved without PCA, with the Gini index and MDL parameters set to their default values. The maximum accuracy was 86.9%.

The K Nearest Neighbor (KNN) algorithm is used to solve classification problems and is based on determining the class of an object based on the majority of its nearest neighbors in the training sample [10].

The study compared KNN models with and without PCA for a fixed number of nearest neighbors. Models were constructed with the number of PCA components ranging from 17 to 5 with a step of 2, as well as a model without PCA. The Accuracy metric values were plotted in the graph shown in Figure 5.

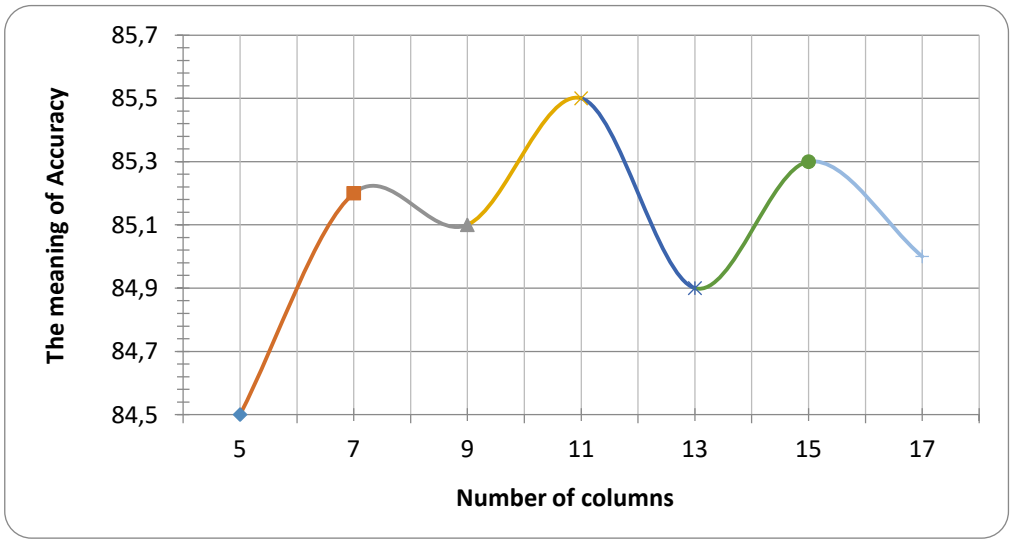

Figure 5. Accuracy parameter for KNN model using PCA with different number of columns.

The results showed that the highest accuracy was achieved using PCA with 11 and 15 components, as well as a model without PCA. These variants were selected for further F-measure analysis.

The effect of the number of nearest neighbors on classification accuracy was additionally investigated. For this purpose, models with numbers of neighbors ranging from 1 to 10 were constructed for PCA variants with 11 and 15 components. The results of these experiments were plotted, as shown in Figure 6.

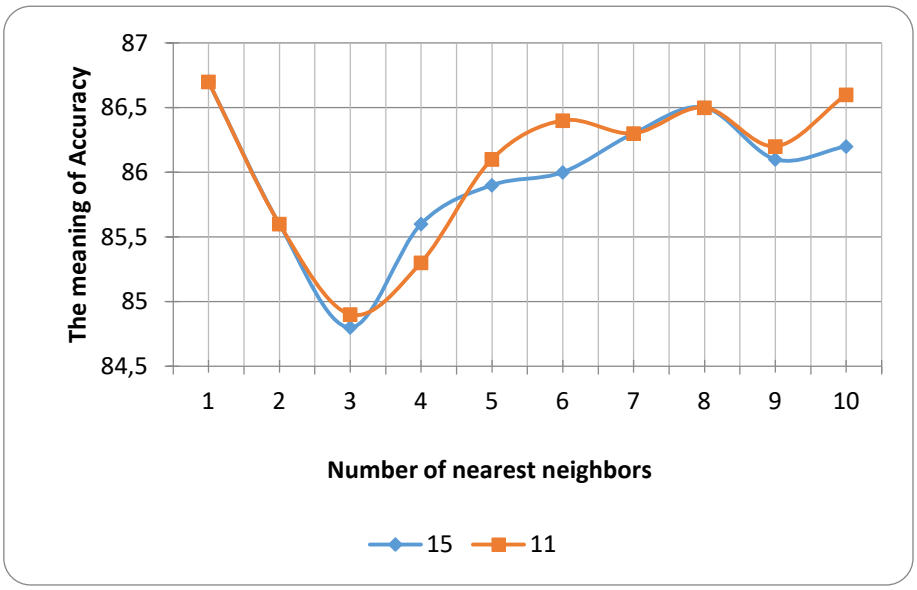

Figure 6. Accuracy parameter for KNN model using PCA with 11 and 15 columns.

The analysis showed that the highest accuracy values are achieved with the number of nearest neighbors equal to 1, while the differences between models with different numbers of PCA components are insignificant.

Tree Ensemble is a machine learning method based on the use of a set of decision trees. Data normalization was not used in this type of model [7].

In the first stage, we studied the impact of the principal component analysis (PCA) on the accuracy of the Tree Ensemble model at a tree depth of 10. Models were constructed with the number of PCA components ranging from 17 to 5 with a step of 2, as well as a model without PCA. The experimental results are presented in Figure 7.

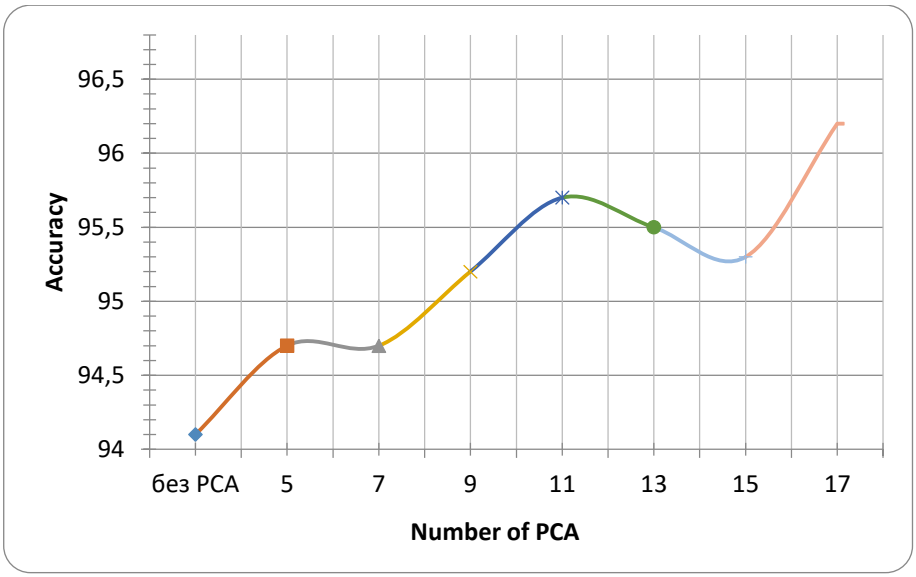

Figure 7. Accuracy parameter for Tree Ensemble model using PCA with different number of columns and without PCA

Experimental results indicate a decrease in classification accuracy in models without PCA, while the use of principal component analysis provides a significant improvement in performance. Maximum accuracy values were achieved using 17 and 11 PCA components.

The next step was to examine the effect of tree depth on classification accuracy for a model with 17 PCA components. Tree depth varied from 3 to 10. The results were plotted in Figure 8.

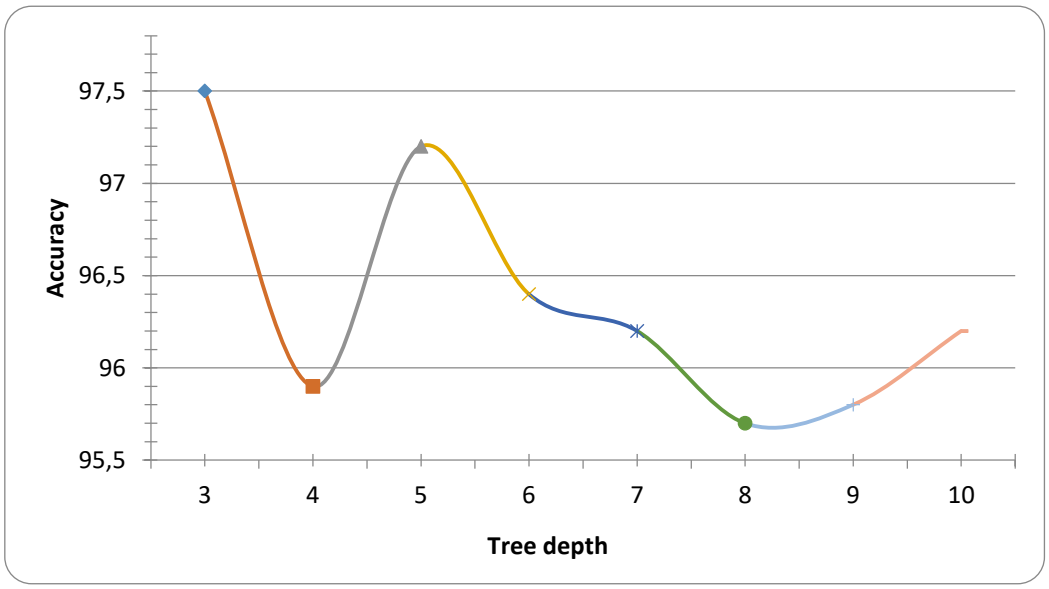

Figure 8. Accuracy parameter for Tree Ensemble model with PCA columns number 17

The obtained results indicate that the maximum classification accuracy is achieved with a tree depth of 3. In this case, the Accuracy value was 97.5%, which is the best result among all the models considered.

PNN Learner (Probabilistic Neural Network) implements a probabilistic neural network and is designed for classification based on numerical data [6].

The study assessed the impact of principal component analysis on the accuracy of the PNN Learner model. Models were constructed using PCA with components ranging from 17 to 5, as well as a model without PCA. The experimental results are presented in Figure 9.

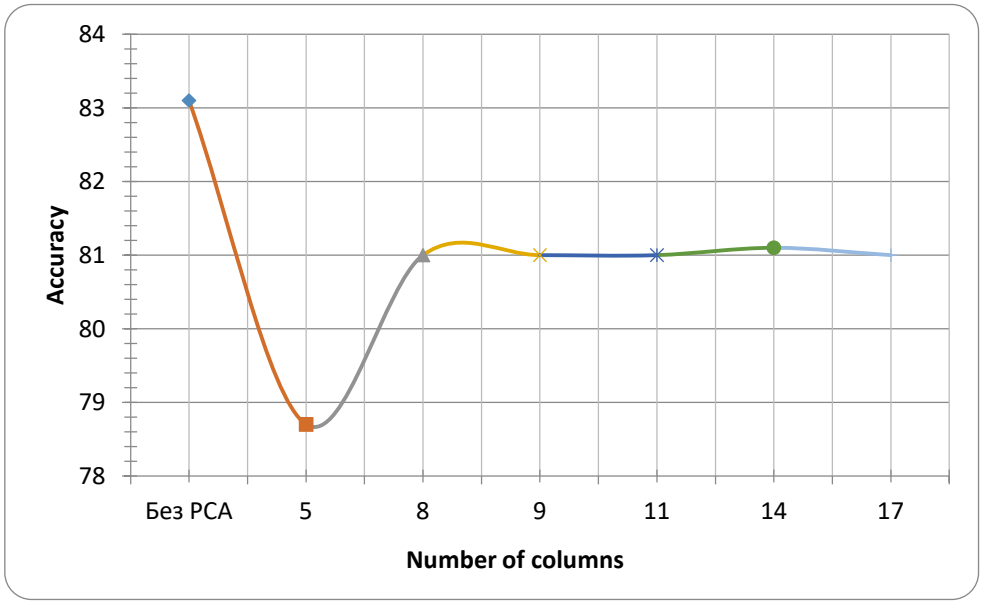

Figure 9. Accuracy parameter for the PNN Learner model using PCA with different numbers of columns and without PCA

The obtained results indicate that the highest Accuracy metric value is achieved in the model without PCA. Therefore, further research was conducted without the use of principal component analysis.

The effect of the number of training epochs on classification accuracy was additionally studied. The number of epochs varied from 10 to 20. The experimental results are presented in Figure 10.

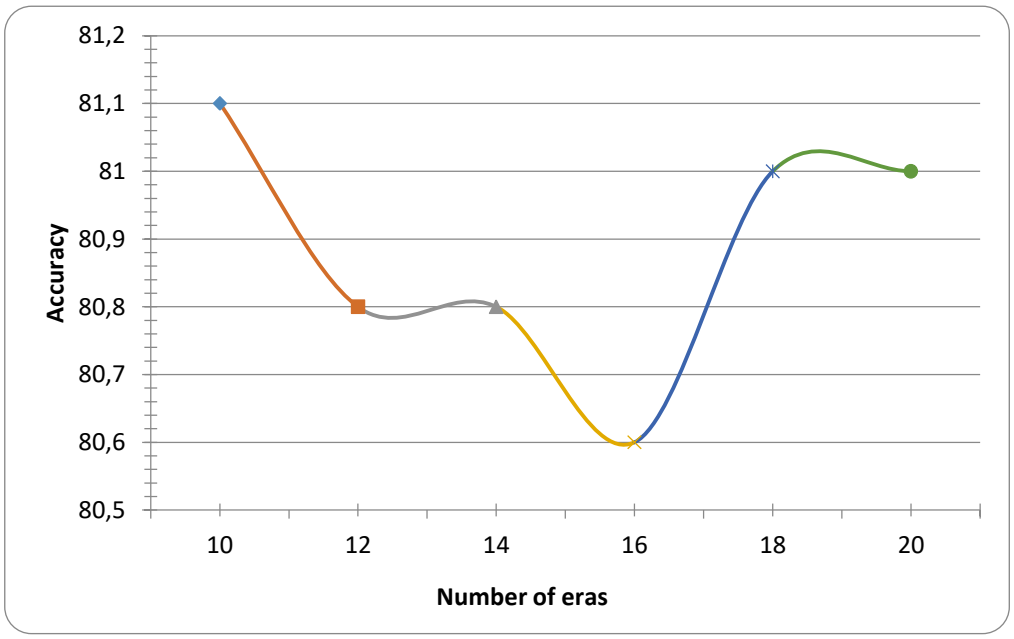

Figure 10. Accuracy parameter for the PNN Learner model without PCA

The obtained data showed that optimal results were achieved with a number of epochs equal to 10, with the maximum accuracy value being 89.2%.

Gradient Boosted Trees (GBT) is a gradient-boosted decision tree algorithm designed for solving classification problems. Data normalization was not used in this type of model [8].

In the first stage, we studied the impact of PCA on the accuracy of the GBT model at a tree depth of 10. Models were constructed with the number of PCA components ranging from 17 to 4 with a step of 3, as well as a model without PCA. The experimental results are presented in Figure 11.

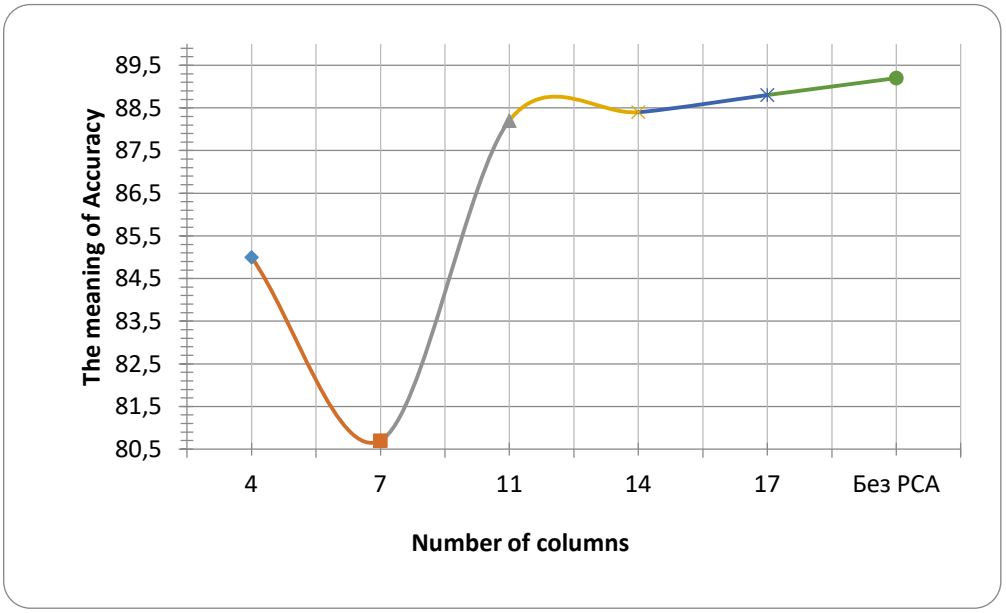

Figure 11. Accuracy parameter for the Gradient Boosted Trees model using PCA with different numbers of columns and without PCA

The results indicate that the highest Accuracy metric value is achieved in the model without PCA. High accuracy values were also obtained using PCA with 17 and 14 components.

In the next step, the effect of tree depth on classification accuracy was investigated without using PCA. Tree depth varied from 4 to 10. The experimental results are presented in Figure 12.

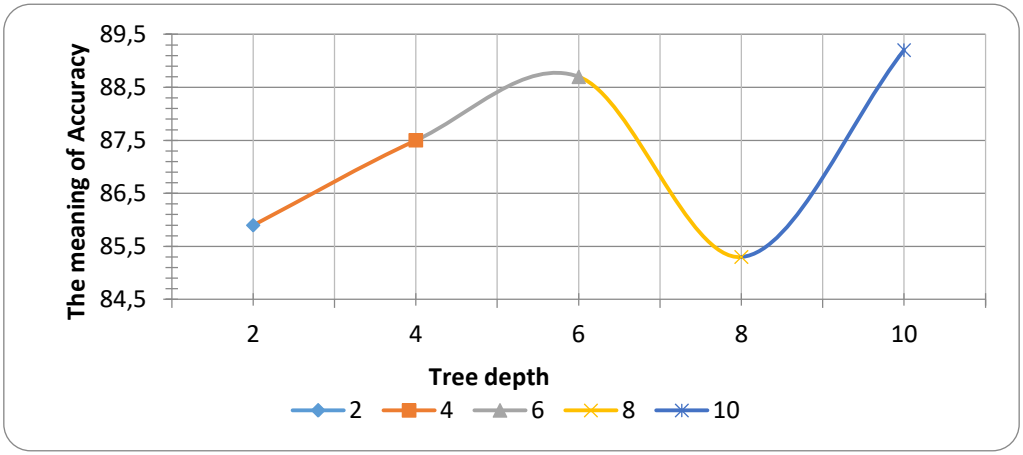

Figure 12. Accuracy parameter for GBT (Gradient Boosted Trees) models without PCA

The obtained results showed that the maximum accuracy value of 89.2% is achieved at a tree depth of 10.

To visually compare the effectiveness of the machine learning models considered, a summary table of results was generated (Table 3), which presents the values of the Accuracy metric for all the studied algorithms, taking into account the use of the principal component method.

Table 3. Research results.

Model	Accuracy, %	PCA
Decision Tree	86,5	-
K Nearest Neighbor	86,7	11
Tree Ensemble	97,5	17
PNN	81,1	-
Gradient Boosted Trees	89,2	-

The data presented in the table indicate that, among the models considered, the Tree Ensemble method provides the highest classification accuracy when using PCA with a number of components equal to 17. The resulting Accuracy metric value of 97.5% exceeds the performance of other machine learning algorithms.

It should be noted that ensemble methods demonstrate higher resistance to the peculiarities of the original data compared to individual models, which makes them particularly promising for the tasks of detecting cyberattacks in the infrastructure of the Industrial Internet of Things [12].

Conclusion. The study examined and compared various machine learning models used to detect cyberattacks in Industrial Internet of Things (IoT) control systems based on network traffic analysis. The experiments used a dataset containing the main attack types typical of the IIoT environment, allowing for an objective assessment of the effectiveness of the classification algorithms.

The obtained results indicate that attack detection accuracy is significantly determined by both the choice of machine learning model and the data preprocessing methods used. It was found that the use of principal component analysis (PCA) improves classification efficiency for a number of algorithms by reducing data dimensionality and eliminating multicollinearity. However, this approach is not universal and requires individual tuning for each model.

A comparative analysis showed that the best accuracy results were achieved using the Tree Ensemble model in combination with the principal component analysis (PCA) with 17 components and a tree depth of 3. In this case, the Accuracy metric value was 97.5%, which exceeds the results of the other machine learning models considered.

The obtained results confirm the potential of using machine learning methods to detect and analyze cyberattacks on Industrial Internet of Things devices and can be used in building monitoring and intrusion detection systems in IIoT infrastructure. Future work should focus on

expanding and updating datasets, fine-tuning model parameters, and using modern machine learning libraries and tools to improve the resilience and effectiveness of security systems [13].

Список источников

1. Машинное обучение [Электронный ресурс]. URL: https://www.sap.com/central-asia-caucasus/products/artificial-intelligence/what-is-machine-learning.html (дата обращения: 11.05.2025).
2. Воронина В. В., Михеев А. В., Ярушкина Н. Г., Святов К. В. Теория и практика машинного обучения: учеб. пособие. Ульяновск: УлГТУ, 2017. 290 с.
3. Лапина М. А., Мовзалевская В. В., Токмакова М. Е., Бабенко М. Г., Саджид Мохаммад. Применение технологий машинного обучения для обнаружения веб-атак // Вопросы кибербезопасности. 2024. № 4 (62). С. 92–103. https://doi.org/10.21681/2311-3456-2024-4-92-103
4. Лапина М. А., Подручный Н. В., Русанов М. А., Бабенко М. Г. Исследование методов машинного обучения для выявления сетевых атак // Труды Института системного программирования РАН. 2025. Т. 37. № 4–2. С. 147–174.
5. IoT SDN IDS Датасет [Электронный ресурс]. URL: https://www.kaggle.com/datasets/hebadhirar/iot-sdn-ids-dataset (дата обращения: 11.05.2025).
6. Jolliffe I. T., Cadima J. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A. 2016. Art. 374. https://doi.org/10.1098/rsta.2015.0202
7. Biau G., Scornet E. A random forest guided tour // Test. 2016. Vol. 25. P. 197–227. https://doi.org/10.1007/s11749-016-0481-7
8. Natekin A., Knoll A. Gradient boosting machines, a tutorial // Frontiers in Neurorobotics. 2013. Vol. 7. Art. 21. https://doi.org/10.3389/fnbot.2013.00021
9. Song Y., Lu Y. Decision tree methods: applications for classification and prediction // Shanghai Archives of Psychiatry. 2015. Vol. 27. No. 2. P. 130–135. https://doi.org/10.11919/j.issn.1002-0829.215044
10. Zhang S., Li X., Zong M., Zhu X., Wang R. Efficient kNN classification with different numbers of nearest neighbors // IEEE Transactions on Neural Networks and Learning Systems. 2018. Vol. 29. No. 5. P. 1774–1785. https://doi.org/10.1109/TNNLS.2017.2673241
11. Карачанская Е. В., Соседова Н. И. Метод выявления аномалий сетевого трафика, основанный на его самоподобной структуре // Безопасность информационных технологий. 2019. Т. 26. № 1. С. 98-110. https://doi.org/10.26583/bit.2019.1.10
12. Балыбердин А. В., Крылов Г. О. Повышение точности выявления аномалий для систем обнаружения вторжения с помощью ансамблевого обучения // Безопасность информационных технологий. 2025. Т. 32. № 1. С. 153–171. https://doi.org/10.26583/bit.2025.1.11
13. Лавров Б. О., Иванов М. А. Интеллектуальное обнаружение аномалий в контейнеризованных приложениях: методы, архитектура и инструменты // Безопасность информационных технологий. 2025. Т. 32. № 4. С. 149–164. https://doi.org/10.26583/bit.2025.4.11
14. KNIME Analytics Platform. Available from: https://www.knime.com/knime-analytics-platform [Accessed 11 May 2025].

References

1. Machine Learning. Available from: https://www.sap.com/central-asia-caucasus/products/artificial-intelligence/what-is-machine-learning.html [Accessed 11 May 2025]. (In Russ.).
2. Voronina VV, Mikheev AV, Yarushkina NG, Svyatov KV. Theory and Practice of Machine Learning. Ulyanovsk: Ulyanovsk State Technical University; 2017. 290 p. (In Russ.).
3. Lapina MA, Movzalevskaya VV, Tokmakova ME, Babenko MG, Sajid Mohammad. Application of Machine Learning Technologies for Web Attack Detection. Voprosy Kiberbezopasnosti. 2024;(4):92-103. (In Russ.). https://doi.org/10.21681/2311-3456-2024-4-92-103
4. Lapina MA, Podruchny NV, Rusanov MA, Babenko MG. Research of Machine Learning Methods for Detecting Network Attacks. Proceedings of the Institute for System Programming of the Russian Academy of Sciences. 2025;37(4-2):147-174. (In Russ.).
5. IoT SDN IDS Dataset. Available from: https://www.kaggle.com/datasets/hebadhirar/iot-sdn-ids-dataset [Accessed 11 May 2025]. (In Russ.).
6. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philosophical Transactions of the Royal Society A. 2016;374. https://doi.org/10.1098/rsta.2015.0202

7. Biau G, Scornet E. A random forest guided tour. Test. 2016;25:197-227. https://doi.org/10.1007/s11749-016-0481-7
8. Natekin A, Knoll A. Gradient boosting machines, a tutorial. Frontiers in Neurorobotics. 2013;7:21. https://doi.org/10.3389/fnbot.2013.00021
9. Song Y, Lu Y. Decision tree methods: applications for classification and prediction. Shanghai Archives of Psychiatry. 2015;27(2):130-135. https://doi.org/10.11919/j.issn.1002-0829.215044
10. Zhang S, Li X, Zong M, Zhu X, Wang R. Efficient kNN classification with different numbers of nearest neighbors. IEEE Transactions on Neural Networks and Learning Systems. 2018;29(5):1774-1785. https://doi.org/10.1109/TNNLS.2017.2673241
11. Karachanskaya EV, Sosedova NI. A Method for Detecting Network Traffic Anomalies Based on Its Self-Similar Structure. Information Technology Security. 2019;26(1):98-110. (In Russ.). https://doi.org/10.26583/bit.2019.1.10
12. Balyberdin AV, Krylov GO. Improving the Accuracy of Intrusion Detection Systems Using Ensemble Learning. Information Technology Security. 2025;32(1):153-171. (In Russ.). https://doi.org/10.26583/bit.2025.1.11
13. Lavrov BO, Ivanov MA. Intelligent Anomaly Detection in Containerized Applications: Methods, Architectures, and Tools. Information Technology Security. 2025;32(4):149-164. (In Russ.). https://doi.org/10.26583/bit.2025.4.11
14. KNIME Analytics Platform. Available from: https://www.knime.com/knime-analytics-platform [Accessed 11 May 2025]. (In Russ.).

Информация об авторах

Юлия Алексеевна Андрусенко – старший преподаватель, Северо-Кавказский федеральный университет, Ставрополь, Researcher ID: PJB-8661-2026.
Глеб Алексеевич Семенов – студент, Северо-Кавказский федеральный университет, Ставрополь, Researcher ID: PII-8176-2026.
Артем Алексеевич Соломянко – студент, Северо-Кавказский федеральный университет, Ставрополь, Researcher ID: PJB-0124-2026.
Алиса Андреевна Кущенко – студент, Северо-Кавказский федеральный университет, Ставрополь, Researcher ID: PSM-3830-2026.
Кристина Юрьевна Серебренникова – студент, Северо-Кавказский федеральный университет, Ставрополь, Researcher ID: PJB-0103-2026.

Вклад авторов: все авторы внесли равный вклад в подготовку публикации.

Information about the authors

Yuliya A. Andrusenko – Senior Lectuter, North Caucasian Federal University, Stavropol, Researcher ID: PJB-8661-2026.
Gleb A. Semenov – student, North Caucasian Federal University, Stavropol, Researcher ID: PII-8176-2026.
Artem A. Solomyanko – student, North Caucasian Federal University, Stavropol, Researcher ID: PJB-0124-2026.
Alisa A. Kushchenko – student, North Caucasian Federal University, Stavropol, Researcher ID: PSM-3830-2026.
Kristina Y. Serebrennikova – student, North Caucasian Federal University, Stavropol, Researcher ID: PJB-0103-2026.

Author contributions: the authors contributed equally to this article.