DOI: 10.15514/ISPRAS-2025-37(5)-6

Detection of SQL Injection Attacks through the Network Logs Using Machine Learning Methods

[1] *M.A. Lapina, ORCID: 0000-0001-8117-9142 <mlapina@ncfu.ru>*
[1] *N.R. Kapshuk, ORCID: 0009-0004-3644-7530 <kapshuknik06@gmail.com>*
[2] *M.A. Rusanov, ORCID: 0009-0000-7069-7542 <mix.rusanoff@yandex.ru>*
[1] *E.F. Timofeeva, ORCID: 0000-0001-5824-4778 <teflena@mail.ru>*

[1] *Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University, 1, Pushkina str., Stavropol, 355017, Russia.*

[2] *Institute of Information Technology, Moscow University of Finance and Law, building 1, 17, Serpukhovskiy val str., Moscow, 115191, Russia.*

Abstract: The article examines machine learning methods for detecting the introduction of SQL code into the network logs using the KNIME program, based on finding patterns between incoming features and subsequent forecasting in a binary classification problem. Unlike existing works, this article examines the effectiveness of five tree-based machine learning methods. The content and sequence of work stages are presented. The highest results were shown by the Random Forest method (accuracy – 97.58%; area under the ROC curve is 0.976).

Keywords: machine learning; KNIME; classification; dataset; data selection; SQL injection; threat detection on the network; detection of suspicious patterns; protection of web applications.

For citation: Lapina M.A., Kapshuk N.R., Rusanov M.A., Timofeeva E.F. Detection of SQL injection attacks through the network logs using machine learning methods. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 5, 2025, pp. 81-92. DOI: 10.15514/ISPRAS-2025-37(5)-6.

Обнаружение атак с использованием SQL-инъекций по сетевым журналам с помощью методов машинного обучения

[1] *М.А. Лапина, ORCID: 0000-0001-8117-9142 <mlapina@ncfu.ru>*
[1] *Н.Р. Капшук, ORCID: 0009-0004-3644-7530 <kapshuknik06@gmail.com>*
[2] *М.А. Русанов, ORCID: 0009-0000-7069-7542 <mix.rusanoff@yandex.ru>*
[1] *Е.Ф. Тимофеева, ORCID: 0000-0001-5824-4778 <teflena@mail.ru>*

[1] *Факультет математики и компьютерных наук имени профессора Н.И. Червякова,*
Северо-Кавказский федеральный университет,
Россия, 355017, г. Ставрополь, ул. Пушкина, д.1.

[2] *Институт информационных технологий,*
Московский финансово-юридический университет,
Россия, 115191, г. Москва, ул. Серпуховский вал, д. 17, корп. 1.

Аннотация: В статье рассматриваются методы машинного обучения для обнаружения внедрения SQL-кода в сетевые журналы с помощью программы KNIME, основанные на поиске закономерностей между входящими признаками и последующем прогнозировании в задаче бинарной классификации. В отличие от существующих работ, в этой статье рассматривается эффективность пяти методов машинного обучения на основе деревьев. Представлено содержание и последовательность этапов работы. Наиболее высокие результаты показал метод "Случайный лес": точность – 97,58%; площадь под кривой ошибок (ROC-кривой) – 0,976.

Ключевые слова: машинное обучение; программа KNIME; классификация; набор данных; отбор данных; SQL-инъекция; обнаружение угроз в сети; обнаружение подозрительных шаблонов; защита веб-приложений.

Для цитирования: Лапина М.А., Капшук Н.Р., Русанов М.А., Тимофеева Е.Ф. Обнаружение атак с использованием SQL-инъекций по сетевым журналам с использованием методов машинного обучения. Труды ИСП РАН, том 37, вып. 5, 2025 г., стр. 81–92 (на английском языке). DOI: 10.15514/ISPRAS–2025–37(5)–6.

1. Introduction

In the modern world, in the age of information technology, databases store a lot of different information: logins and passwords, bank card numbers, home addresses, credit history and much more. Attackers interested in stealing this information are capable of using various types of cyber attacks aimed at hacking servers storing valuable information [1]. One of these methods of cyber attacks is SQL injection. SQL injection is a serious security vulnerability of web applications and systems that have access to databases [2]. The essence of this method is the introduction of malicious SQL code into an Internet resource, which in turn allows attackers to gain access to change data and steal it [3-4]. Usually, this attack is possible when input data is not filtered thoroughly enough when using SQL queries. To successfully neutralize the introduction of malicious code on sites, applications, and ensure the protection of stored information, it is necessary to detect a hacking attempt early and then prevent it. Machine learning in comparison with manual analysis allows you to do this quickly and accurately. This approach not only provides a high level of security, but also provides effective means of detecting, analyzing, and preventing threats [5].

The purpose of the study: creation and identification of the most effective machine learning model for detecting SQL injection into the network in terms of detection accuracy.

To achieve the stated goal of the study, the following tasks were identified: studying the history of the issue, analyzing the parameters of a dataset of SQL attacks, determining input data for machine learning models, creating machine learning models to solve binary classification problems, selecting the most effective models, setting PCA values (principal component analysis), determining the depth

of machine learning, using methods of combating overfitting, building ROC curves and conducting a system analysis of machine learning results.

There are few works with a similar research purpose. One of the differences between this study and others that conduct a comparative analysis of machine learning methods for detecting SQL injections [6-8] is a visual explanation of the implementation of machine learning in the KNIME program and a special methodological toolkit - tree-based machine learning methods, which increases the value of the article.

Among the existing approaches (administrative, legal, and technical), technical ones are used, in particular, using network filters [9].

When solving problems, both general scientific research methods were used: analytical, comparative, and problem-solving methods; and machine learning methods: Decision Tree, Random Forest, Simple Regression Tree, Gradient Boosted Trees, and Tree Ensemble.

Machine learning (ML) is the process of automatically learning and improving the behavior of an artificial intelligence system based on processing an array of training data without explicit programming [10]. In other words, ML is based on finding patterns between incoming features for subsequent prediction.

Analysis of the history of SQL injection attacks showed that they originated in the 1990s, when web applications began to gain popularity. The first such attack was recorded in 1998, highlighting vulnerabilities in database-driven websites that allowed attackers to manipulate SQL queries by injecting malicious code through user input fields [11]. Since then, SQL injection attacks have been used for a quarter of a century. For example, large-scale destructive attacks using SQL injections occurred at Heartland Payment Systems in 2008, when a major payment processor experienced one of the largest data leaks and about 130 million credit and debit card numbers were exposed [12]. In 2011, Sony Pictures was attacked, which compromised Sony's network and digital infrastructure, affecting around 77 million PlayStation Network accounts, costing Sony around $170 million [13].

In 2012, Yahoo Voices suffered a massive data leak, affecting its vast user base and exposing around half a million email addresses and passwords [14].

In 2015, the telecommunications giant TalkTalk suffered a cyberattack, which resulted in almost 157,000 customers losing their personal data [15].

In recent years, the number of cyberattacks has increased many times over [16]. For example, in 2023, in the United States, a group of ransomware attackers successfully injected malicious SQL code into MOVEit Transfer software and a Progress Software product designed to manage file transfers. This attack was only detected a month later [17].

SQL attacks have also been recorded in Russia. For example, in 2024, there was a massive attack on ticket purchasing services. During this period, the total number of simultaneously conducted cyberattacks increased more than 2-fold [18].

Thus, it can be concluded that SQL injections are one of the most common and dangerous methods of attack in the field of cybersecurity, and that in the modern world, network security is of critical importance.

2. Research

During the research, the KNIME platform was used to implement ML and predict potential SQL attacks. KNIME is an environment designed to create algorithms aimed at data analysis and ML, while not requiring full-fledged programming. Unlike other implementation technologies, this program has a user-friendly and flexible interface, and the algorithms are implemented using pre-configured nodes that can be used for built-in data processing pipelines. The advantages of the KNIME program are also open source and its extensibility - the ability to support integration with other platforms such as Python, R, Java and others, and open source code [19].

2.1. Dataset analysis

The dataset chosen for the implementation of the ML was the Web Network dataset [20]. It contains network traffic logs and is designed to classify web requests as "good" or "bad" based on their characteristics. By analyzing patterns in network logs, this dataset helps to identify web requests that can be categorized as "legitimate" or "malicious". It has various HTTP(S) requests, including headers, URLs, and request bodies. Each request is labeled based on its perceived legitimacy. The detection strategy is to identify certain key patterns that indicate an attack [20].

Let us consider the dataset in more detail.

The original dataset contains 522 rows of data and there are 29 columns in total, each of which belongs to a specific data type and contains specific information that can help in training the model. The target column is the "class" column. The ratio of safe requests to suspicious ones in this column is 321 to 201. In the course of the study, a comparative analysis of the web network dataset was carried out (Table 1).

The dataset [20] contains 12 duplicate columns with similar data. Repeated recordings can lead to incorrect operation of the models. To solve this problem, duplicate columns were not used as input data. There are also various concepts such as "method", "path", "features" and "prediction". In the "method" column only 22% of the cells have the values "GET" or "POST", and the rest contain zeros, which indicates that this column has incorrect values, so it should be excluded. The "path" column, contains various query paths, but it does not have information about the contents of the queries, since information about the number of special characters has been placed in separate columns. The "features" column contains arrays with the same data that were presented in the previous columns. And "prediction" is the results of a study conducted by the author of the dataset. Thus, we can conclude that for the most efficient ML, the following columns will be used as input data for the models: "single_q_1", "double_q_2", "dashes_3", "braces_4", "spaces_5", "percentages_6", "semicolons_7", "angle_brackets_8", "special_chars_9", "path_length_10", "body_length_11", "badwords_count_12" and "class". It is important to keep in mind that the "class" column is the target column. No transformations will occur with it, and it will be used to compare the values contained in it with the values that will be output by ML models in order to obtain a percentage of prediction accuracy.

2.2. Modeling

After selecting the input data, the ML models were constructed (Fig. 1).

The justification for the order of connecting the nodes of the generalizing model is determined by the analysis (Table 2).

A total of 5 models were created, each using its own training method. Among them: Decision Tree, Random Forest, Simple Regression Tree, Gradient Boosted Trees, Tree Ensemble. The selected machine learning methods related to trees are characterized by high accuracy, interpretability, and they can also be used to work with missing values.

It should be noted that with ML for the same model and with the same settings, different accuracy results could be obtained, and the error was about 1-2%. To get more objective results, training was performed 10 times for each ML model setting option, after which the average value between the results was returned.

With the initial (default) settings, each model showed the following accuracy results (Table 3).

In order to improve the accuracy of the models, it is necessary to perform additional adjustments. One of these adjustments is the PCA adjustment. PCA (principal component analysis) is a statistical method that allows reducing the dimensionality of data (Table 4), while preserving the greatest amount of information [21]. The values were adjusted in the PCA block.

The accuracy of the models from the PCA value is presented more clearly below (Fig. 2).

Table 1. Analysis of data from the Web Network dataset.

Name of the columns	Data type	Description
method	Categorical	Indicates the type of operation the user wants to perform (GET or POST).
path	Text	Request path
single_q	Quantitative	Number of single quotes in the query (‘)
double_q	Quantitative	Number of double quotes in the query (“)
dashes	Quantitative	Number of dashes in the query (-)
braces	Quantitative	Number of curly braces in the query ({})
spaces	Quantitative	Number of spaces in the query
percentages	Quantitative	Number of percent characters in the query (%)
semicolons	Quantitative	Number of semicolons in the query (;)
angle_brackets	Quantitative	Number of angle brackets (< >)
special_chars	Quantitative	Number of special characters in the query
path_length	Quantitative	Request path length
body_length	Quantitative	Length of the request body. Only available when using the POST method in the "method" column.
badwords_count	Quantitative	The number of suspicious words that may be frequently used in SQL injections.
single_q_1	Quantitative	Similar to "single_q"
double_q_2	Quantitative	Similar to "double_q"
dashes_3	Quantitative	Similar to "dashes"
braces_4	Quantitative	Similar to "braces"
spaces_5	Quantitative	Similar to "spaces"
percentages_6	Quantitative	Similar to "percentages"
semicolons_7	Quantitative	Similar to "semicolons"
angle_brackets_8	Quantitative	Similar to "angle_brackets"
special_chars_9	Quantitative	Similar to "special_chars"
path_length_10	Quantitative	Similar to "path_length"
body_length_11	Quantitative	Similar to "body_length"
badwords_count_12	Quantitative	Similar to "badwords_count"
class	Binary	Mark what the request is (0 – safe, 1 – suspicious). This is also the target column, since it will be compared with the predicted data.
features	Text	The previously specified cell values combined into a list.
prediction	Binary	Prediction of the previously used learning model.

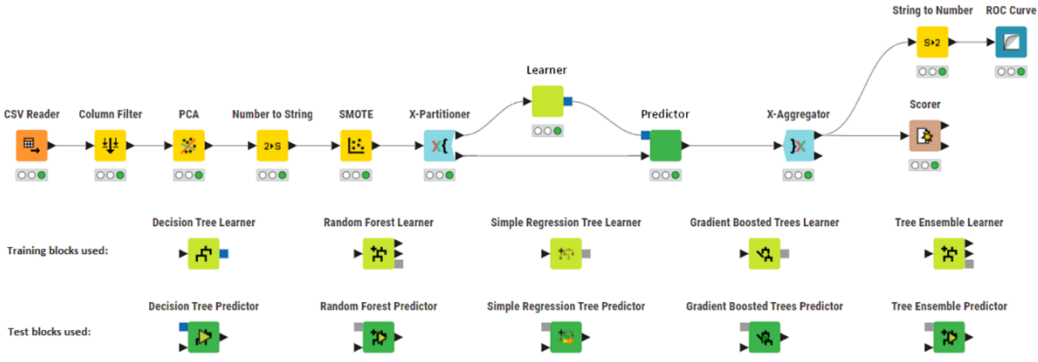

Fig 1. Generalized Machine Learning Model and Machine Learning Blocks in KNIME.

Table 2. Purpose of nodes in each model.

Node	Node name	Purpose of the node
	CSV Reader	Uploads a CSV file.
	Column Filter	Discards the specified columns.
	PCA	Reduces the dimensionality of data.
	Number to String	Converts numeric data to strings.
	SMOTE	Makes the dataset balanced.
	X-Partitioner	Splits the data into a set number of parts and performs cross-validation. During the research, the data is divided into five parts, after which, during each iteration, four parts are sent for training and one for testing.
	Learner	Learning node.
	Predictor	Testing node.
	X-Aggregator	Returns the average accuracy value of the results at each iteration after cross-validation.
	Scorer	Outputs the accuracy of the model.
	String to Number	Converts string data to numbers.
	ROC Curve	Builds a ROC curve.

Table 3. Accuracy of machine learning models on initial settings.

Machine learning method	Accuracy (in %)
Decision Tree	93,730
Random Forest	93,502
Simple Regression Tree	94,199
Gradient Boosted Trees	93,729
Tree Ensemble	94,008

Table 4. Dependence of model accuracy on the PCA block value.

PCA Block Value	Model accuracy (in %)				
	Decision Tree	Random Forest	Simple Regression Tree	Gradient Boosted Trees	Tree Ensemble
1	93,730	93,502	94,199	93,729	94,008
2	94,131	94,617	93,868	93,903	94,617
3	94,814	94,706	94,581	94,461	94,531
4	95,053	95,054	95,229	95,611	95,211
5	96,025	96,342	96,080	96,064	96,569
6	96,271	97,039	**97,038**	96,950	97,073
7	96,236	97,126	96,848	96,656	97,144
8	96,273	97,231	96,395	96,672	97,214
9	96,169	**97,598**	96,480	96,811	**97,562**
10	**96,533**	97,195	96,672	96,882	97,231
11	96,427	97,160	96,550	**97,125**	97,336
12	96,341	97,108	96,741	97,089	97,370

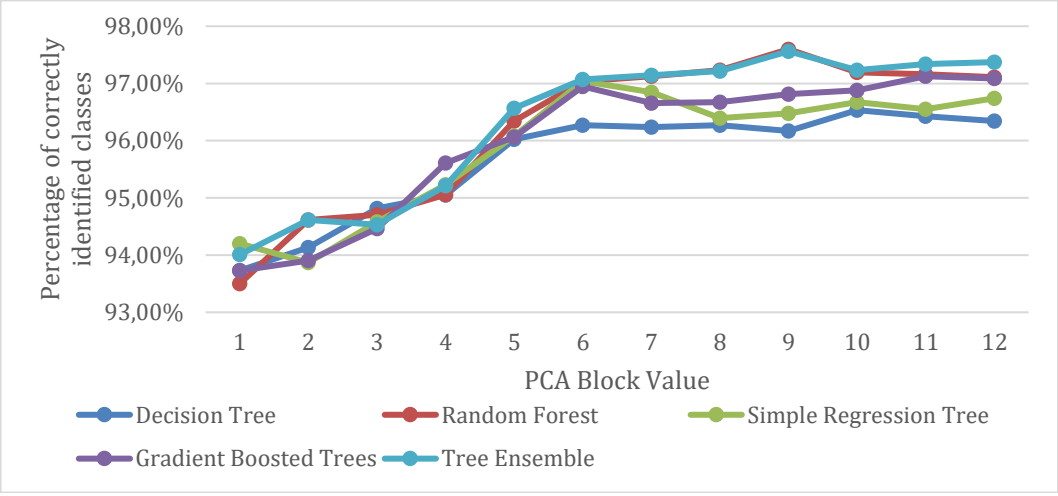

Fig. 2. Dependence of the accuracy of the models on the value of the PCA block.

Table 5. Dependence of model accuracy on learning depth.

Learning depth	Model accuracy (in %)				
	Decision Tree	Random Forest	Simple Regression Tree	Gradient Boosted Trees	Tree Ensemble
1	96,307	91,516	92,754	96,707	90,485
2	96,290	96,467	95,958	96,828	95,785
3	96,707	97,073	96,482	96,879	96,690
4	**96,898**	97,143	96,342	96,864	96,795
5	96,464	97,546	96,550	**97,003**	97,038
6	96,237	**97,580**	96,255	96,483	97,283
7	95,924	**97,580**	**96,618**	96,306	97,213
8	96,133	97,545	96,534	96,725	97,387
9	95,890	97,300	96,498	96,637	**97,473**
10	96,081	97,421	96,603	96,830	97,457

From these results, we can conclude that for a model with the "Decision Tree" learning method, the most optimal reduction is to 10 columns, for the "Random Forest" and "Tree Ensemble" methods - to 9 columns, for the "Simple Regression Tree" – to 6 columns, and for the "Gradient Booste Trees - to 11 columns (Fig. 2). For each model, the learning depth – number of hidden learning layers [22] – was adjusted with the numbers of columns that were obtained before (Table 4). The depth of learning was adjusted in the Learner block.

The results showed that the Decision Tree model achieves the best result with 4 training layers, the Random Forest model with 6-7 training layers, the Simple Regression Tree with 7 training layers, the Gradient Boosted Trees with 5 training layers, and the Tree Ensemble with 9 training layers (Fig. 3).

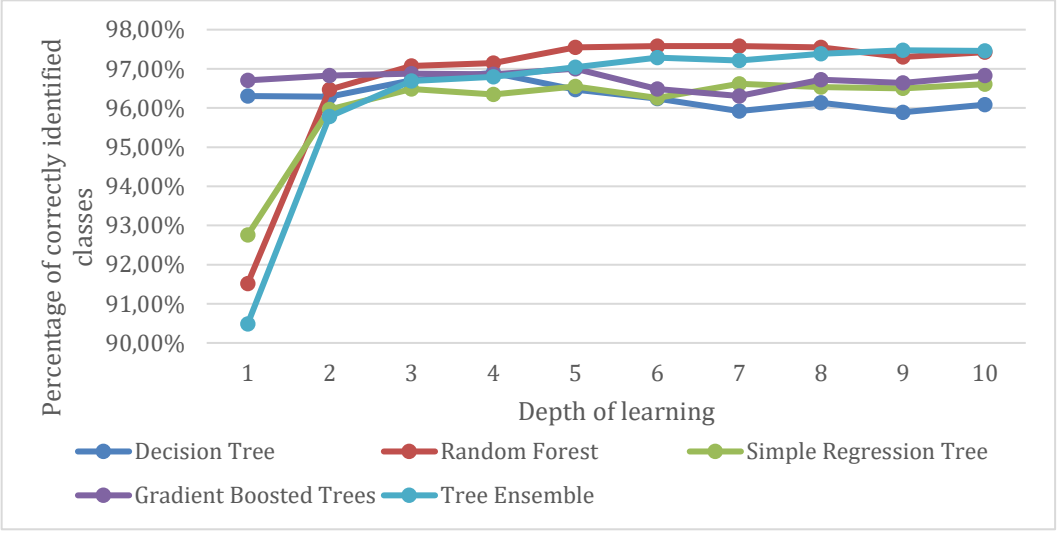

Fig. 3. Dependence of model accuracy on learning depth.

2.3. Combating overfitting

It is important to consider that during the training of neural networks, so-called overfitting may occur, which can lead to worse forecasting results. This phenomenon occurs when the model adjusts too much to the training data, which is why it begins to work poorly with new data. Combating overfitting is an integral task in the field of ML. One of the methods for solving this problem is cross-validation - a method for assessing the quality of a model by dividing the data into several parts, after which the model is trained and predicted on different subsets of data. In this case, the dataset [20] is divided into five equal parts using the "X-partitioner" block. The model is then trained through five iterations, where one part of the data is used in testing and the rest in training. After that, the "X-aggregator" block returns the average accuracy value. To clearly see the forecasting quality of each model, it is necessary to use ROC curves (receiver operating characteristic) - graphs that are used to assess the quality of binary classifiers [23]. An important part of the ROC curve is the area under it, where a value of "1" indicates an ideal classifier, and a value of "0.5" indicates a large amount of randomness during forecasting. Based on the ROC curve graphs (Fig. 4), it can be concluded that the resulting ML models were quite effective in predicting the potential introduction of malicious SQL code into the network. The resulting models showed high efficiency in detecting patterns between input features and subsequent prediction, as evidenced by the results.

Based on all the obtained results, a comparative table of machine learning methods was compiled (Table 6).

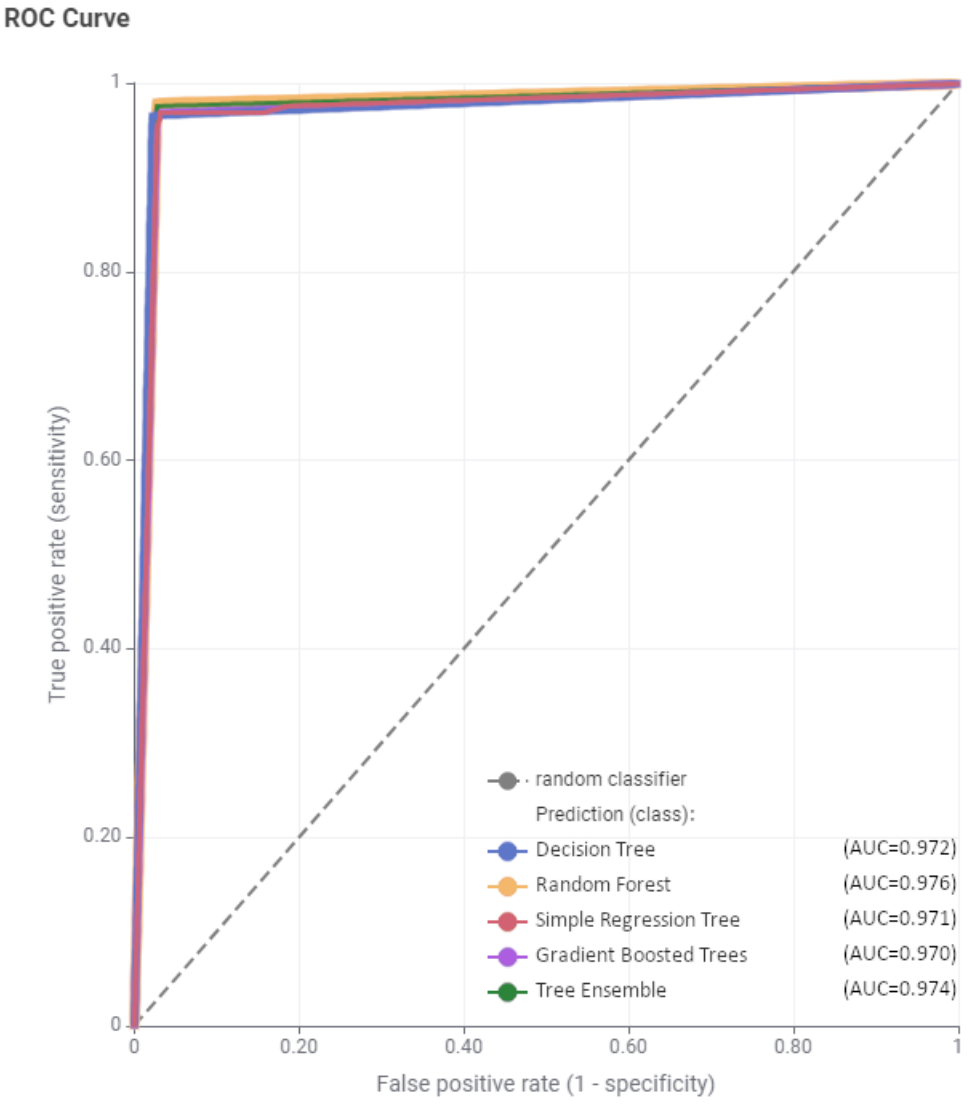

Fig. 4. ROC curve graphs.

Table 6. Best results of the models.

Machine learning method	PCA Block Value	Learning depth (number of learning layers)	Highest accuracy (in %)	AUC (Area Under the Curve)
Decision Tree	10	4	96,898	0,972
Random Forest	**9**	**6-7**	**97,580**	**0,976**
Simple Regression Tree	6	7	96,618	0,971
Gradient Boosted Trees	11	5	97,003	0,970
Tree Ensemble	9	9	97,473	0,974

2.4. Analysis of the results

The general characteristics of the models as a whole indicate the high efficiency of each of the ML methods used. The analysis of accuracy and ROC curves shows that a model using a Random Forest as a machine learning method provides the highest quality of binary classification. Optimization of the model parameters to achieve maximum classification accuracy was achieved by adjusting the values of the PCA block and the depth (number of layers) of the ML.

3. Conclusion

The conducted research on the creation of machine learning models to detect the introduction of malicious SQL code into the network differed from similar studies of this problem by implementing models in the KNIME program. An analysis of other work related to the detection of potential SQL injections using ML was carried out. The data set was analyzed, and ML tree-based models were compiled. Each ML method used was configured in such a way as to increase the percentage of correctly identified classes, and a comparative analysis was performed. The "Random Forest" model showed the best result with the highest accuracy of 97.58%, and the area under the ROC curve graph compiled to assess the quality of this model is 0.976. Thus, the ML in the KNIME program allows you to create effective models for detecting the potential introduction of malicious SQL code into the network.

References

[1]. "Stab me if you can" – how websites and SQL databases are attacked with injections – Dmitry Ushakov on TenChat.ru. URL: https://tenchat.ru/media/2607916-protkni-menya-yesli-smozhesh--kak-atakuyut-vebsayty-i-bazy-dannykh-sql-inyektsiyami (date of access: 17.04.2025).

[2]. Khomyarchuk M. V. Modern trends and innovations in web security: challenges, solutions and prospects //Science and modern education: current issues. – 2023. – p. 28.

[3]. Oglov V. A. Investigation of sql injection attacks and analysis of web site security //Bulletin of the Magistracy. - 2024. – p. 15.

[4]. Manukyan A. R. Problems of ensuring cybersecurity at the present stage //Law and management. – 2024. – No. 10. – pp. 313-316.

[5]. Peev D. D., Pankov K. N. The use of computer vision and machine learning technologies in the field of secure information systems //Signal synchronization, generation and processing systems. – p. 28.

[6]. Yudova E. A., Laponina O. R. Comparative analysis of approaches to detecting SQL injections using machine learning methods //International Journal of Open Information Technologies. - 2023. - Vol. 11. - No. 6. - pp. 175-181.

[7]. Kasim Ö. An ensemble classification-based approach to detect attack level of SQL injections //Journal of Information Security and Applications. – 2021. – T. 59. – C. 102852.

[8]. Erdődi L., Sommervoll Å. Å., Zennaro F. M. Simulating SQL injection vulnerability exploitation using Q-learning reinforcement learning agents //Journal of Information Security and Applications. – 2021. – T. 61. – C. 102903.

[9]. Zaozersky A. A. Technical approaches to information protection //BBK 1 N 34. – P. 6505.

[10]. Chesalov A. Y. Glossary on artificial intelligence: 2500 terms/ A. Y. Chesalov - "Publishing solutions", 2022. - 670 p.

[11]. SQL attack. URL: https://ru.easiio.com/sql-attack/ (date of access: 03.04.2025).

[12]. The Hearland Breach | A cautionary Tale foe E-Commerce. URL: https://blog.comodo.com/e-commerce/the-heartland-breach-a-cautionary-tale-for-e-commerce/ (date of access: 03.04.2025).

[13]. Indonesian Journal of Electrical Engineering and Computer Science Vol. 21, No. 2, February 2021, pp. 1121-1131.

[14]. Yahoo Hack Leaks 453,000 Voices Passwords. URL: https://www.darkreading.com/cyberattacks-data-breaches/yahoo-hack-leaks-453-000-voice-passwords (date of access: 03.04.2025).

[15]. Unknown persons hacked the British TalkTalk provider – Xakep. URL: https://xakep.ru/2015/10/27/talktalk-hacked/ (date of access: 03.04.2025).

[16]. Nathan C., Steven F., Human Aspects of Information Security and Assurance, p.329, New York: Springer International Publishing (2022).

[17]. Current threats: The second quarter of 2023. URL: https://www.ptsecurity.com/ru-ru/research/analytics/cybersecurity-threatscape-2023-q2/ (date of access: 03.04.2025).
[18]. Major cyber attacks and leaks in Russia in 2024. URL: https://blog.cortel.cloud/2024/05/23/krupnye-kiberataki-i-utechki-pervoj-poloviny-2024-goda-v-rossii/?ysclid=m929qx878m857705097 (date of access: 03.04.2025).
[19]. KNIME Analytics Platform | KNIME. URL: https://www.knime.com/knime-analytics-platform (date of access: 15.05.2025).
[20]. Web Network. URL: https://www.kaggle.com/datasets/willianoliveiragibin/web-network (date of access: 21.03.2025).
[21]. How to use the PCA method to reduce the dimension of data / Habr. URL: https://habr.com/ru/companies/otus/articles/769274 / (date of access: 04.03.2025).
[22]. Machine Learning Glossary | Google for Developers. URL: https://developers.google.com/machine-learning/glossary#d (date of access: 15.05.2025).
[23]. Kostromitin M. A. The fight against retraining of neural networks: causes, effects and methods of prevention //BBK 1 N 34. - p. 2809.

Information about authors

Мария Анатольевна ЛАПИНА – кандидат физико-математических наук, доцент кафедры вычислительно математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова Северо-Кавказского федерального университета. Сфера научных интересов: цифровые технологии, анализ данных, искусственный интеллект, кибербезопасность, управление информационной безопасностью, криптография.

Maria Anatolyevna LAPINA – Cand. Sci. (Phys.-Math.), Associate Professor at the Department of Computational Mathematics and Cybernetics, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University. Research interests: digital technologies, data analysis, artificial intelligence, cybersecurity, information security management and cryptography.

Николай Романович КАПШУК – студент кафедры вычислительно математики и кибернетики факультета математики и компьютерных наук имени профессора Н.И. Червякова Северо-Кавказского федерального университета. Сфера научных интересов: информационная безопасность, технологии сетевой безопасности, машинное обучение, нейронные сети, цифровые технологии.

Nikolay Romanovich KAPSHUK – student at the Department of Computational Mathematics and Cybernetics, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University. Research interests: information security, network security technologies, machine learning, neural networks, digital technologies.

Михаил Андреевич РУСАНОВ – аспирант Института информационных технологий, Московский финансово-юридический университет. Сфера научных интересов: информационная безопасность, управление информационной безопасностью, машинное обучение, нейронные сети, обнаружение аномалий.

Mikhail Andreevich RUSANOV – postgraduate student at the Institute of Information Technology, Moscow University of Finance and Law. Research interests: information security, information security management, machine learning, neural networks, anomaly detection.

Елена Федоровна ТИМОФЕЕВА – доцент кафедры математического анализа алгебры и геометрии факультета математики и компьютерных наук имени профессора Н.И. Червякова Северо-Кавказского федерального университета. Сфера научных интересов: математическое моделирование, численные методы, задачи гидродинамики.

Elena Fedorovna TIMOFEEVA – Associate Professor of the Department of Mathematical Analysis of Algebra and, Faculty of Mathematics and Computer Science named after Professor N.I. Chervyakov, North Caucasus Federal University. Research interests: mathematical modeling, numerical methods, problems of hydrodynamics.