УДК 004.89+004.056 Вестник УрФО № 3(57) / 2025, с. 51–60

Грибачёв А. С., Ручай А. Н. DOI: 10.14529/secur250306

ОПТИМИЗАЦИЯ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ЭФФЕКТИВНОГО ПРЕДОТВРАЩЕНИЯ КИБЕРАТАК В СИСТЕМАХ ОБНАРУЖЕНИЯ СЕТЕВЫХ ВТОРЖЕНИЙ[1]

В данной статье рассматривается оптимизация моделей машинного обучения (ML), предназначенных для улучшения предотвращения кибератак в системах обнаружения сетевых вторжений. В исследовании делается упор на повышение производительности и надежности этих моделей путем точной настройки их гиперпараметров с использованием передовых методов оптимизации. Используя широко известный набор данных CIC-IDS-2018, проведенисчерпывающий сравнительный анализ различных алгоритмов классификации, включая деревья принятия решений, методы опорных векторов, логистическую регрессию и продвинутые групповые подходы, такие как случайные леса и градиентный бустинг. Эмпирические данные свидетельствуют о существенном повышении производительности благодаря оптимизированным конфигурациям, особенно при выявлении сложных киберугроз. Важнейшие показатели оценки, такие как показатель F1, правильность, полнота, и точность, подтверждают превосходство оптимизированных моделей ML в защите жизненно важных инфраструктур от текущих киберугроз.

Ключевые слова: *кибербезопасность, информационная безопасность, машинное обучение, алгоритмы классификации.*

[1] Исследование поддержано грантом Российского научного фонда (проект No 22-71-10095-П).

OPTIMIZING MACHINE LEARNING MODELS FOR EFFICIENT PREVENTION OF CYBER ATTACKS IN NETWORK INTRUSION DETECTION SYSTEMS

This paper explores the optimization of machine learning (ML) models designed to improve cyber-attack prevention in network intrusion detection systems. The study emphasizes improving the performance and reliability of these models by fine-tuning their hyperparameters using advanced optimization techniques. Using the widely recognized CIC-IDS-2018 dataset, we conducted an exhaustive comparative analysis of various classification algorithms, including decision trees, support vector machines, logistic regression, and advanced ensemble approaches such as random forests and gradient boosting. Empirical evidence demonstrates substantial performance improvements from optimized configurations, especially in identifying complex and emerging cyber threats. Critical evaluation metrics, such as F1-score, accuracy, recall, and precision, confirm the superiority of optimized ML models in protecting vital infrastructures against ongoing cyber risks.

Keywords: *cybersecurity, information security, machine learning, classification algorithms.*

Introduction

In today's digital age, cyberthreats pose unprecedented challenges to organizations. Attackers are employing increasingly sophisticated strategies that render conventional protection measures inadequate [1]. Consequently, there is a growing need for intelligent systems capable of processing large volumes of network traffic to detect anomalous behavior and predict potential breaches [2-3].

Machine learning is emerging as a powerful tool for improving cyberdefenses. Advanced algorithms can uncover hidden patterns in user activity and network events, identifying deviations that indicate possible attacks. By leveraging state-of-the-art methodologies, organizations can proactively respond to new forms of cyberassaults, strengthening their defenses [4, 5].

This study aims to evaluate the efficacy of different machine learning models in detecting network intrusions. Using the CIC-IDS-2018 dataset, we compare common classification techniques and introduce an automated procedure for periodically retrainingthe model's knowledge base with new data. Through rigorous testing and systematic analysis, we will determine the most effective solution for cyber-attack prevention [6].

Related works

Numerous studies have examined machine learning-driven approaches to improve cybersecurity [7-9]. Earlier studies primarily focused on basic algorithms,such as linear analysis and artificial neural networks [10]. However, current trends favor more advanced techniques such as random forests, gradient boosting, and deep learning architectures. These methods provide greater precision and robustness, offering improved insights into complex attack patterns [11-12].

Data preprocessing is pivotal to the success of any machine learning endeavor. Researchers emphasize the importance of cleansing, normalizing, and removing anomalies prior to modeling. Studies show that proper preprocessing substantially improves the predictive power of classification models [13].

Supervised learning dominates the field of cyberattack detection since labeled datasets al-

low for the precise identification of known attack types. Cross-validation is frequently employed to reliably assess model performance. For instance, kk-fold cross-validation minimizes bias by repeatedly dividing the dataset into training and testing folds, offering a comprehensive evaluation framework [14].

In article [15], the authors propose an approach thatcombines cross-validation with t-SNE visualization and applies it to four well-known datasets: UNSW-NB15, CIC-CSE-IDS2018, BoT-IoT, and ToN-IoT. The results highlight that high accuracy achieved on one dataset often fails to transfer effectively to the others due to varying class distributions. Additionally, generalization capability differs between benign and malicious traffic classes. Therefore, it is essential to carefully select training data that match dataset classes with real-world traffic profiles. While merging datasets could increase diversity, thorough validation is required prior to practical implementation.

Work [16] evaluated 15 machine learning algorithms for both binary and multi-class classification tasks focused on IoT traffic conditions. Performance was assessed based on temporal and spatial dimensions. The conclusion recommends using nonlinear algorithms, particularly tree-based ones, to train or retrain models aimed at detecting malicious network traffic in IoT intrusion detection systems.

Finally, maintaining models becomes vital in dynamic environments where attack signatures evolve rapidly. Techniques such as "on-demand retraining" allow for continuous updates, ensuring that the defensive systems are updated with the latest threats [17-18].

Methods and data

CIC-IDS-2018 Dataset

The CIC-IDS-2018 Dataset (Canadian Institute for Cybersecurit y Intrusion Detection System Dataset) is a widely recognized benchmark dataset specifically designed to evaluate intrusion detection systems (IDSs) and advanced network security mechanisms. Developed by researchers at the University of New Brunswick's Centre for Cyber Security Research, this dataset contains a comprehensive collection of both legitimate and malicious network traffic traces [19].

It contains detailed logs of network activities captured during controlled experiments that simulate various types of cyberattacks. These include brute force attacks, denial-of-service (DoS/DDoS), botnet activity, SQL injection attempts, cross-site scripting (XSS), port scans, and others. In addition to these anomalous behaviors, the dataset also captures regular user behavior (benign traffic), making it well-balanced for building and validating machine-learning-driven intrusion detection systems [20].

Each record in the dataset represents an individual flow between network endpoints and includes numerous features extracted from each flow, such as flow duration, inter-packet arrival times, packet count, bytes transferred, TCP/IP flags, source and destination IP addresses, ports, protocols, etc. All entries are structured in tabular form, facilitating easy integration with popular tools and libraries such as Pandas and Scikit-Learn [21].

Additionally, it can be employed for feature engineering tasks aimed at extracting discriminatory characteristics indicative of potential attacks.By offering a balanced mix of benign and malicious samples, alongside granularity in feature representation, the CIC-IDS-2018 dataset remains indispensable for anyone aiming to understand, simulate, or defend against sophisticated cyber assaults.

Data Preprocessing

The original dataset, obtained from the CIC-IDS-2018 repository, contained incomplete and noisy entries. Several steps were taken to prepare the data for modeling:

First, duplicates were removed to eliminate redundancy. Next, timestamps were converted into numerical representations, to facilitate time-series analyses. Missing values were filled using column medians, mitigating the impact of gaps in the data. Finally, feature scaling normalized the dataset, reducing disparities caused by varying attribute magnitudes.

After preprocessing, the cleaned dataset (base_preprocessed.csv) was ready for subsequent modeling tasks.

Data Partitioning

Proper data separation is crucial for successful machine learning because it affects the quality of model evaluation and its ability to generalize knowledge to new data. This study used a standard approach to divide the dataset into

three parts:80% of the initial dataset constituted the training data, which was essential for directly training the model and adjusting its internal parameters. The remaining 20% was distributed as follows: 10% formed the validation set, which was used for training hyperparameters and intermediate evaluation of model quality, thereby avoiding overfitting.Another 10% composed the test set, deliberately isolated from the training process and reserved exclusively for the final check of the model's operational readiness in real-life conditions. Additionally, target labels were separately extracted and recorded for each of the three subsets, stored in the files y_train_val.csv, y_valid.csv, and y_test.csv, ensuring a clear distinction between features and target variables. This division allowed for a reliable assessment of the model's quality and verified its ability to make accurate predictions on previously unseen data.

Model Evaluation and Selection

Multiple classification algorithms were tested systematically:

Logistic Regression is a fundamental classification technique that employs a linear combination of features passed through a sigmoid function to estimate the probability of an observation belonging to one of two classes (e.g., attack or no attack). A hypothesis is defined as follows:

$$z = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots, \quad \#(1)$$

the resulting probability is calculated using the sigmoid function:

$$p(y = 1|x) = \frac{1}{1 + e^{-z}}, \quad \#(2)$$

transforming the linear combination into a probability score ranging from 0 to 1. Objects are assigned to class 1 if their probability exceeds 0.5. Despite its simplicity and computational efficiency, logistic regression struggles with non-linear problems and performs poorly on high-dimensional datasets. Nevertheless, it serves as a baseline for comparison and demonstrates interpretability in its coefficients, allowing easy understanding of feature influence on classification outcomes [22-23].

Support Vector Machines (SVM) is a powerful classification method aimed at finding an optimal hyperplane that separates two classes with maximum margin—the largest possible gap between the closest points (support vectors) of opposing classes. The boundary is described by the equation as follows:

$$w^T x + b = 0, \quad \#(3)$$

where w denotes the weight vector perpendicular to the hyperplane, and b is the offset along the axis. The distance from the boundary to the nearest point (a support vector) is calculated as follows:

$$\frac{|w^T x + b|}{||w||}. \quad \#(4)$$

For non-linear problems, SVM projects data into a higher-dimensional space using special functions called kernels (common choices include linear, radial basis function (RBF), and polynomial kernels). Although SVM delivers excellent separation quality even with small numbers of features and possesses strong generalization abilities, it requires careful selection of the kernel type and regularization parameters, and its computational demand increases with larger datasets [24].

Naive Bayes is a quick and computationally efficient classification algorithm based on Bayes' theorem and the simplifying assumption that features are statistically independent. The posterior probability of assigning an object to a particular class C_k is calculated as follows:

$$P(C_K|X) = \frac{P(X|C_K) \cdot P(C_K)}{P(X)}, \quad \#(5)$$

where $(X|C_K)$, the likelihood of observing features X given class C_K, is approximated as the product of individual feature probabilities under the naive independence assumption. This approach greatly reduces computational complexity, making Naïve Bayes particularly suitable for tasks involving many features. Its advantages include speed and low memory usage, but it suffers from limitations when features exhibit dependencies or correlations, potentially degrading its performance [25].

Decision Trees are simple and intuitively understandable classification models that construct hierarchical branching structures to guide decision-making processes. They recursively split data by selecting the best features for parti-

tioning, creating nodes representing logical checks and leaf nodes symbolizing final class assignments. One major advantage lies in their transparent nature, as the reasoning behind predictions is easily traceable through the tree structure. Decision trees effectively handle both categorical and numerical data, aiming to minimize impurity measures such as entropy or Gini index during splits. However, they tend to suffer from overfitting when grown too deeply, meaning they may memorize training data rather than generalizing well to new observations. Despite this limitation, decision trees remain invaluable in domains where interpretability and visualization play a crucial role, such as diagnostic systems or customer profiling [26].

Ensemble Methods (Random Forest, Gradient Boosting) arepowerful predictors combining weak learners into stronger ones, albeit requiring greater computing capacity [27].Each algorithm was tuned individually to maximize its predictive performance.Ensemble methods significantly enhance classification accuracy and robustness by aggregating multiple weak learners into a collective model. Unlike standalone decision trees, they are less prone to overfitting, yielding reliable performance even on complex, noisy, or heterogeneous datasets. However, these methods come with increased computational costs and reduced interpretability compared to single-tree models. Nonetheless, they are widely preferred in classification and regression tasks where high accuracy and reliability are paramount [18-30].

Random Forest builds an ensemble consisting of numerous decision trees, each trained on randomly sampled subsets of data and features. Final predictions are made either by majority voting (for classification) or averaging (for regression). Suppose we haveN decision trees$T_1, T_2, \ldots, T_N$, each trained on a random subsample of data and features. The prediction of the random forest is formed as follows:

• For regression:

$$\hat{Y}_{RF} = \frac{1}{N}\sum_{i=1}^{N} T_i(x). \qquad \#(6)$$

• For classification, choose the class with the highest number of votes:

$$\hat{C}(x) = arg_{C\in\{C_1,\ldots,C_K\}} \max\left\{\sum_{i=1}^{N} I(T_i(x) = c)\right\}, \#(7)$$

where$I(\cdot)$ is an indicator function equal to 1 if the condition holds, otherwise 0.

Gradient Boosting constructs a series of trees sequentially, where each subsequent tree tries to improve upon the errors of the previous ones. The next element in the ensemble learns from residuals (errors) produced by the preceding tree, gradually refining the overall prediction towards the correct output.Let us assume we have a sequence of trees$T_1, T_2, \ldots, T_M$. The prediction of gradient boosting is expressed as:

$$\hat{\boldsymbol{Y}}_{\boldsymbol{GB}} = \sum_{\boldsymbol{m=1}}^{\boldsymbol{M}} \boldsymbol{\alpha_m T_m(x)}, \qquad \#(7)$$

where α_m controls the contribution of each tree to the final outcome.

Explanation of Metrics

Metrics used here follow standard definitions in binary classification:

Accuracy represents the proportion of correctly classified instances out of the total number of predictions.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Predictions}, \#(8)$$

where *True Positives* are objects that truly belong to class 1 and were correctly classified as such; *True Negatives* are objects that truly belong to class 0 and were correctly classified as such; *False Positives* areobjects that were incorrectly classified as class 1 (false alarm); *False Negatives* are objects that were incorrectly classified as class 0 (missed detections).This is a measure of the overall correctness rate but doesnot reveal details about error distribution.

Recall measures the fraction of actual positive examples correctly identified by the model.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}. \#(9)$$

It refers tohow well the model captured the positive examples. Having a high recall is important, when finding all relevant positive cases is crucial, like disease diagnosis,.

Precision calculates what portion of positively predicted objects actually belong to the positive class.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}, \quad \#(10)$$

Of all the objects predicted as positive, it may not be clear how many are really positive. When misclassifying something as positive is costly (such as false diagnoses), achieving high precision is desirable.

F1-Score combines precision and recall into a single metric, taking their harmonic meaning. Thus, it balances both factors equally.

$$F1 - Score = 2 \cdot \frac{Recall \times Precision}{Recall + Precision}. \quad \#(11)$$

Values closer to 1 indicate better classification performance. It is useful when you want to balance both precision and recall, especially in situations where neither should dominate (e.g., early-stage medical diagnostics).

These four metrics provide a comprehensive evaluation of the classification model's performance. Combining their outputs provides insight into strengths and weaknesses, helping to select the best model depending on task-specific needs [31]. Given these trade-offs, ensemble methods are the preferred choice for real-world intrusion detection tasks.

Results

Several classification algorithms were thoroughly tested, including Logistic Regression, Support Vector Machines (SVM), Naïve Bayes, Decision Trees, and ensemble methods such as Random Forest and Gradient Boosting. Logistic Regression,for example, yielded training accuracies ranging from 0.902672 to 0.9998309005344181, whereas Decision Trees scored perfectly (1.0) every time. Notably, ensemble methods demonstrated remarkable stability and precision, achieving almost flawless results across epochs. Similarly, SVM exhibited nearly identical performance, highlighting its effectiveness in binary classification tasks.

Experiments confirmed the superiority of ensemble methods over other classification techniques. Specifically, Random forest and Gradient boosting delivered near-perfect accuracy rates (close to 1.0), surpassing alternatives such as Logistic regression and Decision trees. Furthermore, incorporating cross-validation into the evaluation process improved reliability. With cross-validation, the model maintained stable performance across multiple iterations, minimizing the risk of overfitting.Table 1 summarizes the results comparing performance metrics.

After evaluating various machine learning algorithms according to predefined criteria, the results revealed that Random Forest and Gradient Boosting models outperformedother models. This outcome aligns with existing literature that has previously discussed the strengths and weaknesses of each algorithm. Specifically, Logistic Regression and Linear Support Vector Machines tend to perform better in binary classification settings, however, intrusion detection in-

Table 1

Summarized results comparing performance metrics.

Algorithm	Train Accuracy	Test Accuracy	Recall	Precision	F1-Score
Logistic Regression	0.99961	0.99959	0.9996	0.9996	0.9996
SVM	0.99991	0.99991	0.9999	0.9999	0.9999
Native Bayes	0.99997	0.99997	0.9999	0.9999	0.9999
Decision Tree	0.99999	0.99999	0.9999	0.9999	0.9999
Random Forest	0.99999	1.0	1.0	1.0	1.0
Gradient Boosting	1.0	0.99999	1.0	1.0	1.0
AdaBoost	0.99962	0.99965	0.9996	0.9996	0.9996

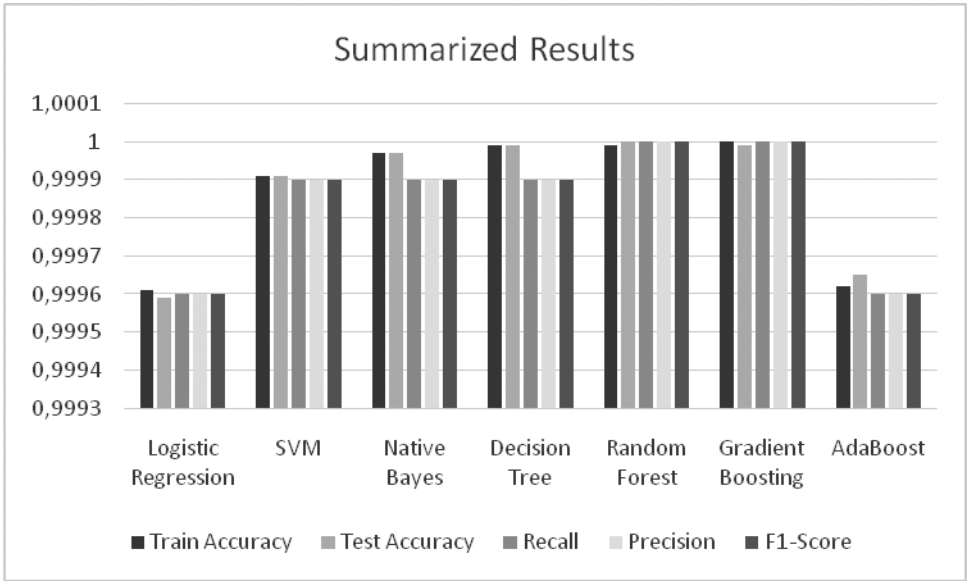

Figure 1. Summary of the results

herently involves multi-class classification. Additionally, theoretical expectations suggest that Gaussian Naïve Bayes would underperform in real-world scenarios, a hypothesis that our findings validate.

Our study confirms that Random Forest and Gradient Boosting, when coupled with training and grouping techniques, substantially improve the handling of imbalanced datasets and the prediction of cyberattacks. Across key metrics, such as precision, recall, F1-score, and accuracy, these models consistently outperformed alternative algorithms. Figure 1 illustrates a clear disparity between Random Forest, Gradient Boosting, and the remaining four models (including Logistic Regression and AdaBoost). Despite relatively high accuracy figures for certain models, this metric alone cannot fully capture their overall effectiveness.

Discussion

The authors [32] propose a NIDS system that uses machine learning to detect modern attack types with a high detection rate. Several ML algorithms were implemented and evaluated using a cutting-edge dataset containing modern attacks. The results demonstrate that the Random Forest model achieves superior performance, with a detection rate of 97%. These results suggest that machine learning has significant potential for creating highly effective NIDS systems.

Researchers in [33] introduce IWHO-XGBoost—a novel machine learning-based intrusion detection system optimized with Intelligent Wild Horse Algorithm. Trained on diverse simulated network intrusion datasets, it combines Z-score normalization and Kernel PCA for preprocessing and feature extraction. Hyperparameter optimization enhances XGBoost's intrusion identification abilities. Evaluation metrics—F1-score (98.42%), accuracy (98.45%), recall (98.4%), and precision (98.38%)—demonstrate superior detection performance compared to conventional methods.

Our results confirm that ensemble methods excel in cyber-intrusion detection. Their high accuracy makes them ideal candidates for deploying scalable, resilient cyber-defense systems. In contrast, simpler models struggle to match their performance, underscoring the value of ensembling weaker learners into robust classifiers. Performance measures such as F1-score (100.0%), accuracy (99.99%), recall (100.0%), and precision (100.0%) clearly illustrate that the proposed method significantly outperforms traditional approaches in terms of intrusion detection efficiency.

Cross-validation helped stabilize the evaluation process, ensuring consistency across tests. The automated training mechanism also played a critical role in maintaining model relevance despite shifting attack profiles.

However, caution must be exercised regarding ensemble sizes and configurations. Overly complex ensembles may lead to overfitting, necessitating vigilance in choosing appropriate parameters.

Conclusion

The present study has demonstrated the significant advantages of using advanced machine learning algorithms, such as Random Forest and Gradient Boosting,to identify cyber threats. These methods offer superior performance compared to traditional approaches by effectively handling complex data distributions and high-dimensional feature spaces commonly encountered in cybersecurity datasets.

A key innovation of this research is an automated training mechanism that enables the model to dynamically adapt to emerging attack vectors without requiring manual intervention or reconfiguration. This ensures sustained effectiveness over extended periods despite evolving adversarial strategies.

Future work may focus on exploring hybrid architecturesthat combine multiple algorithmic paradigms to enhance detection capabilities further. Additionally, more comprehensive studies on feature engineering techniques are warranted to refine input representations and maximize prediction accuracy. Investigating the process of retrofitting the model with smallportions of newdatainrealtime is also worthwhile.These advancements will contribute significantly to the development of robust and scalable solutions that can effectively mitigate modern cyber risks.

References

1. Goswami, Maloy Jyoti. (2024). AI-Based Anomaly Detection for Real-Time Cybersecurity. 3006-1075.

2. Banketov D.A., Fedorov M.A., Palmov S.V. (2024). The role of artificial intelligence in cybersecurity. Industrial Economics. 143-148.

3. GribachevA.S, Kal'shchikovV.V., RuchayA.N..Metody, algoritmyibazydannykhobnaruzheniyakomp'y uternykhincidentov. Vestnik URFU. Bezopasnost' v informatsionnoy sfere.1(51) 2024, 42-52.

4. Ejjami, Rachid. (2024). Enhancing Cybersecurity through Artificial Intelligence: Techniques, Applications, and Future Perspectives. Journal of Next-Generation Research 5 0. 1. 10.70792/jngr5.0.v1i1.5.

5. Vlasenko A.V. Obzorinstrumentovmashinnogoobucheniya i ikhprimeneniya v oblastikiberbezopasnosti / A.V. Vlasenko, P.I. Dzyoban, R.V. Zhuk // Prikaspiyskiyzhurnal: upravlenie i vysokietekhnologii. – 2020. – № 1(49). – pp. 144-155. – DOI 10.21672/2074-1707.2020.49.4.144-155. – EDN IPJDMM.

6. A.S. Gribachev, V.V. Kalschikov, A.N. Ruchay. Metody, algoritmyibazydannykhobnaruzheniyakomp'y uternykhincidentov / // VestnikUrFO. Bezopasnost' v informatsionnoysfere.– 2024. – № 1(51). – pp. 45-52. – DOI 10.14529/secur240106. – EDN JDVEUY.

7. Nagaraju, Bharath. (2025). Machine Learning For Cybersecurity: Enhancing Intrusion Detection Systems And Threat Mitigation.

8. Oyetoro, Amos & Mart, Joseph & Okoroafor, Nneoma & Amah, Joshua. (2022). Using Machine learning Techniques Random Forest and Neural Network to Detect Cyber Attacks. 10.13140/RG.2.2.27484.05763/1.

9. Y. Wang, Q. Li, and Z. Huang, "A Random Forest Approach for Detecting Botnets," in Proceedings of the International Conference on Advanced Cloud and Big Data, pp. 68-74, 2020

10. Ravi, Vinayakumar& Hb, Barathi Ganesh & Poornachandran, Prabaharan &Madasamy, Anand Kumar &Kp, Soman. (2018). Deep-Net: Deep Neural Network for Cyber Security Use Cases. 10.13140/RG.2.2.13593.06243.

11. Alharthi, Ayesha &Alaryani, Meera & Kaddoura, Sanaa. (2025). A Comparative Study of Machine Learning and Deep Learning Models in Binary and Multiclass Classification for Intrusion Detection Systems. Array. 26. 100406. 10.1016/j.array.2025.100406.

12. Jeamaon A, Khemapatapan C. Development cyber risk assessment for intrusion detection using enhanced random forest. ECTI Transactions on Computer and Information Technology (ECTI-CIT) 2024; 18:429–42.

13. Dhongade, Gauri &Chandrakar, Dr &Khande, Rajeshree. (2024). Enhancing Cyber Security: A Study of Data Preprocessing Techniques for Cyber Security Datasets. International Journal of Scientific Research in Science and Technology. 11. 71-75. 10.32628/IJSRST2411427.

14. Lumumba, Victor & Sang, Dennis &Mpaine, Mary & Makena, Njoka & Musyimi, Daniel &Kavita. (2024). Comparative Analysis of Cross-Validation Techniques: LOOCV, K-folds Cross-Validation, and Repeated K-folds Cross-Validation in Machine Learning Models. American Journal of Theoretical and Applied Statistics. 13. 127-137. 10.11648/j.ajtas.20241305.13.

15. Iwanowski, M.; Olszewski, D.; Graniszewski, W.; Krupski, J.; Pelc, F. The Choice of Training Data and the Generalizability of Machine Learning Models for Network Intrusion Detection Systems. Appl. Sci. 2025, 15, 8466. https://doi.org/10.3390/app15158466

16. Li, Shuren & Lu, Yifei & Li, Jingqi. (2022). Can We Half the Work with Double Results: Rethinking Machine Learning Algorithms for Network Intrusion Detection System. 242-247. 10.1109/CBD54617.2021.00049.

17. Ruchay A.N., Tokarev I.V., Gribachev A.S. Metody mashinnogoobucheniyaiiskusstvennogointellekta v sfereinformatsionnoybezopasnosti: analizsovremennogososto yaniyaiperspektivyrazvitiya. Vestnik URFO. Bezopasnost' v informatsionnoysfere. 4(46) 2022, 76-87.

18. Gribachev A.S., Kalschikov V.V. Issledovaniemetodov i algoritmovobnaruzheniyakomp'yuternykhincidentov. Sborniktrudov XXII Vserossiyskoynauchno-prakticheskoykonferentsiistudentov, aspirantovimolodykhuchenykh. Chelyabinsk, 2024. БЕЗОПАСНОСТЬ ИНФОРМАЦИОННОГО ПРОСТРАНСТВА, 290-294

19. Liu, Lan & Wang, Pengcheng& Lin, Jun & Liu, Langzhou. (2020). Intrusion Detection of Imbalanced Network Traffic Based on Machine Learning and Deep Learning. IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.3048198.

20. Shandilya, Shishir K & Ganguli, Chirag &Izonin, Ivan & Nagar, Atulya. (2022). Cyber Attack Evaluation Dataset for Deep Packet Inspection and Analysis. Data inBrief. 46. 108771. 10.1016/j.dib.2022.108771.

21. Luay, Majed &Layeghy, Siamak &Noorbin, Faezeh & Sarhan, Mohanad & Moustafa, Nour & Portmann, Marius. (2025). Temporal Analysis of NetFlow Datasets for Network Intrusion Detection Systems. 10.48550/arXiv.2503.04404.

22. Hristov, Alexander & Trifonov, Roumen & Hristov, Valentin. (2023). Cyberattacks Detector with Logistic Regression Module. 1-4. 10.1109/COMSCI59259.2023.10315802.

23. Sperandei S. Understanding logistic regression analysis // Bio-chemia Medica. 2014. No. 24 (1), pp. 12-8.

24. Ghanem, Kinan & Aparicio-Navarro, Francisco & Kyriakopoulos, Konstantinos & Lambotharan, S. & Chambers, Jonathon. (2017). Support Vector Machine for Network Intrusion and Cyber-Attack Detection. 1-5. 10.1109/SSPD.2017.8233268.

25. Wickramasinghe, I., Kalutarage, H. Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. Soft Comput 25, 2277–2293 (2021). https://doi.org/10.1007/s00500-020-05297-6

26. Lukiewicz, K. et al. (2025). Decision Trees and Machine Learning for Cybersecurity: How Model Settings Affect Attack Detection. In: Paszynski, M., Barnard, A.S., Zhang, Y.J. (eds) Computational Science – ICCS 2025 Workshops. ICCS 2025. Lecture Notes in Computer Science, vol 15907. Springer, Cham. https://doi.org/10.1007/978-3-031-97554-7_4

27. Shrivastav, Lokesh & Kumar, Ravinder. (2022). An Ensemble of Random Forest Gradient Boosting Machine and Deep Learning Methods for Stock Price Prediction. Journal of Information Technology Research. 15. 1-19. 10.4018/JITR.2022010102.

28. G. R. Bagadi and Y. B. Adimulam, "Random Forest and Gradient Boosting for Superior Intrusion Detection and Anomaly Classification," 2025 International Conference on Machine Learning and Autonomous Systems (ICMLAS), Prawet, Thailand, 2025, pp. 1618-1625, doi: 10.1109/ICMLAS64557.2025.10968997.

29. Gulhane, Kimsy& Saxena, Surabhi &Deogaonkar, Anant & Kumar, Vinod &Vichoray, Chandan & Goyal, Shweta. (2024). Overview of Machine Learning Techniques in Cybersecurity Data Science using Gradient Boosting and Random Forest Algorithm. 19-24. 10.1109/ICoICI62503.2024.10696688.

30. Kumar, Chaitanya & Sabeena, Jasmine. (2023). Cybersecurity Attack Detection using Gradient Boosting Classifier. 10.21203/rs.3.rs-3711213/v1.

31. Soto, Jean. (2023). Binary classification of malware by analyzing its behavior in the network using machine learning. Innovare: Revista de ciencia y tecnología. 12. 30-36. 10.5377/innovare.v12i1.15956.

32. Ali, Md & Thakur, Kutub& Schmeelk, Suzanna &Debello, Joan & Dragos, Denise. (2025). Deep Learning vs. Machine Learning for Intrusion Detection in Computer Networks: A Comparative Study. Applied Sciences. 15. 1903. 10.3390/app15041903.

33. Manivannan, R. & Selvaraj, Senthilkumar. (2025). Intrusion Detection System for Network Security Using Novel Adaptive Recurrent Neural Network-Based Fox Optimizer Concept. International Journal of Computational Intelligence Systems. 18. 10.1007/s44196-025-00767-x.

ГРИБАЧЁВ Антон Сергеевич, аспирант (соискатель) математического факультета, Челябинский государственный университет. 454001, г. Челябинск, ул. Братьев Кашириных, 129. E-mail: a.gribachev@yandex.ru.

РУЧАЙ Алексей Николаевич, доктор технических наук, доцент, заведующий кафедрой компьютерной безопасности и прикладной алгебры, Челябинский государственный университет. 454001, г. Челябинск, ул. Братьев Кашириных, 129.; профессор кафедры защиты информации, Южно-Уральский государственный университет (национальный исследовательский университет), 454080, г. Челябинск, пр. Ленина, 76. E-mail: ran@csu.ru.

GRIBACHEV Anton Сергеевич, PhD candidate of the Faculty of Mathematics, Chelyabinsk State University. 454001, Chelyabinsk, st. Brothers Kashirin, 129. E-mail: a.gribachev@yandex.ru.

RUCHAY Alexey Nikolaevich, Doctor ofTechnical Sciences, Associate Professor, Head of the Department of Computer Security and Applied Algebra, Chelyabinsk State University. 454001, Chelyabinsk, st. Brothers Kashirin, 129.; Professor, Department of Information Security, South Ural State University (National Research University). 454080, Chelyabinsk, Lenina avenue, 76. E-mail: ran@csu.ru.