UDC 004.056: 004.8
IRSTI 27.17.03

https://doi.org/10.55452/1998-6688-2025-22-4-79-96

[1]Kikbayev N.E.,
Master's student, ORCID ID: 0009-0000-0145-5718,
e-mail: kikbaevnurbek@gmail.com
[1]Zhexebay D.M.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com
[2]Xin Y.,
Professor, ORCID ID: 0000-0001-6169-0795,
e-mail: qyuxiao@purdue.edu
[3]Tynymbayev S.T.,
Professor, ORCID ID: 0000-0002-7808-5273,
e-mail: s.tynym@gmail.com
[3]Aitmagambetov A.Z.,
Professor, ORCID ID: 0000-0002-9326-9476,
e-mail: a.aitmagambetov@iitu.edu.kz
[1]Abdizhalilova L.B.,
Master's student, ORCID ID: 0009-0000-5965-7195,
abdijalil.lazzat@bk.ru
[1]*Skabylov A.A.
PhD, ORCID ID: 0000-0002-5196-8252,
*e-mail: Alisher.skabylov@kaznu.edu.kz

[1]Al-Farabi Kazakh National University, Almaty, Kazakhstan
[2]Northwestern Polytechnical University, Xi'an, China
[3]International Information Technology University, Almaty, Kazakhstan

COMPARATIVE STUDY OF MACHINE LEARNING METHODS FOR DETECTING ANOMALIES IN NETWORK TRAFFIC

Abstract

The demand for intrusion detection systems (IDSs) that can promptly identify both known and new types of attacks is on the rise due to the rapid expansion of cyber threats and the consequent increase in network traffic. The utilization of machine learning techniques to autonomously analyze the behavior of network packets and classify them as normal or malicious is a promising way to address this issue. The objective of this investigation is to assess the suitability of a variety of machine learning algorithms for the resolution of network security issues by employing network data analysis as an illustration. This investigation assesses the efficacy of machine learning models in detecting network intrusions using the UNSW-NB15 dataset. This study's primary objective is to assess the effectiveness of various machine learning models, including Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost, LightGBM, and Logistic Regression, in network security applications. According to the analysis, all models exhibited high classification accuracy; however, the LightGBM model attained the most remarkable results. This model exhibited the highest values of Accuracy (95.86%), Precision (96.02%), and F1-measure (96.99%), confirming its capacity to effectively manage complex and heterogeneous data. Overall, the study underscores the significance of selecting the most appropriate model based on the security system's objectives and the specifics of the data.

Key words: machine learning, network traffic, LightGBM, cybersecurity, IDS, data analysis.

1 Introduction

The world of today is undergoing a rapid transformation, and the number and complexity of cyber threats are increasing as a result of the increasing digitalization. Organizations and consumers worldwide are increasingly emphasizing information security. The active integration of machine learning and artificial intelligence into network security systems is being driven by the increasing ineffectiveness of traditional protection methods in this environment. Not only do these technologies enhance the precision of malware and network attack detection, but they also adjust to the constantly evolving tactics of attackers. New opportunities for the development of more robust and proactive mechanisms to safeguard data and infrastructure are presented by the introduction of intelligent approaches to information systems protection.

A model for the automatic detection of P-wave seismic signals in the Almaty region was devised by D. Zhexebay and his colleagues using convolutional neural networks (CNNs) [1]. The authors determined that the implementation of deep learning enhances the dependability of earthquake early warning systems and can be effectively employed to reduce the risks associated with natural disasters. K. Moulaei et al. conducted a comparative analysis of various machine learning algorithms to predict lethal outcomes in patients hospitalized with COVID-19 [2]. The authors concluded that machine learning models, particularly RF, can be effectively employed to identify patients at a high risk of lethality and to assist in clinical decision-making.

Advanced machine learning models are becoming more precise and resilient in their ability to map vulnerability to natural disasters, such as flooding. A novel method of Flood Susceptibility Mapping (FSM) was proposed by S. T. Seydi et al with the use of the Cascade Forest Model [3]. Zhao et al employed machine learning algorithms to investigate the capabilities of the Google Earth Engine (GEE) platform in land use and land cover classification (LULC) tasks [4].

Malware detection continues to be a significant obstacle in the realm of contemporary network security, necessitating a comprehensive strategy. The accuracy and comprehensiveness of threat detection can be substantially enhanced through the integration of machine learning techniques with both static and dynamic analysis. M. Ijaz, M. H. Durad, and M. Ismailin their paper, examined the efficacy of machine learning techniques for malware detection through both static and dynamic analysis [5]. The problem of polymorphic malware detection was examined by M. S. Akhtar and T. Feng through the application of various machine learning algorithms [6]. R. Baker del Aguila and his associates investigated the feasibility of employing resource-efficient machine learning models for static malware analysis [7].

Intrusion Detection Systems (IDS) assist in the development of anti-cyberattack defenses by analyzing network traffic and detecting suspicious activity. Some new and complex forms of attacks may not be effectively detected by traditional rule-based methods. Machine learning (ML) techniques are being integrated into intrusion detection systems to enhance their capacity to autonomously identify attacks in order to resolve this issue.

Traditional machine learning (ML), ensemble learning, and deep learning methods were evaluated by C. Zhang et al. [8]. Based on the KDD CUP99 and NSL-KDD datasets, experimental evaluations were conducted using the following algorithms: Decision Tree, naive Bayesian algorithm, SVM, Random Forest, XGBoost, CNN, and RNN. The results indicated that the naive Bayesian algorithm was more effective in detecting new forms of attacks, while the ensemble learning methods demonstrated high accuracy. Furthermore, the efficacy of deep learning methods was determined to be contingent upon their structure, hyperparameters, and training duration. The primary challenges and prospective research directions in network attack detection are also emphasized by the authors.

Z. Ahmad and his colleagues examine the most recent advancements and obstacles associated with network intrusion detection systems (IDS) in their paper [9]. They outline the potential of deep learning (DL) and machine learning (ML) techniques for the effective detection of attacks. The authors underscore the necessity of enhancing the precision of intrusion detection systems,

minimizing false positives, and identifying novel types of attacks. Furthermore, they recommend future research directions and classify intrusion detection systems according to machine learning/ deep learning.

In the context of a large network data and the Internet of Things (IoT), M. Asif et al. suggest a MapReduce-based intelligent model (MR-IMID) for security [10]. In order to avert future assaults, this model detects network attacks in real-time and stores the data in a database. The MR-IMID model demonstrated an accuracy of 97.7% during training and 95.7% during validation, surpassing previous methods.

A machine learning (ML)-based intelligent intrusion detection system (IDS) for detecting cyber attacks targeting Internet of Things devices is examined by authors D. Musleh, M. Alotaibi, F. Alhaidari, A. Rahman and R. Mohammad [11]. The investigation assessed numerous machine learning models (including Random Forest, KNN, SVM, and stack models) and various feature extraction algorithms (including DenseNet and VGG-16). The model that demonstrated the highest accuracy (98.3%) when utilizing the IEEE Dataport dataset was a combination of VGG-16 and layering.

The Intrusion Detection System (IDS) model proposed by H. Attou, A. Guezzaz, S. Benkirane, M. Azrour, and Y. Farhaoui [12] is a security-enhancing approach to cloud computing systems that employs Random Forest (RF) and feature engineering. The accuracy of the model was 98.3% on the Bot-IoT dataset and 99.99% on the NSL-KDD dataset. The authors observe that this method outperformed other state-of-the-art methods in terms of precision, accuracy, and completeness.

A machine learning-based intrusion detection system (ML-IDS) model for detecting attacks in IoT networks is presented by Y. Saheed, A. Abiodun, S. Misra, M. Holone, and R. Colomo-Palacios [13]. The UNSW-NB15 dataset was employed in the study, and dimensionality reduction techniques were implemented through PCA and minimum and maximum normalization. The results of the testing of six distinct machine learning algorithms were as follows: PCA-XgBoost achieved an accuracy of 99.99%, PCA-Cat Boost achieved an accuracy of 99.99%, PCA-KNN achieved an accuracy of 99.98%, PCA-SVM achieved an accuracy of 99.98%, PCA-QDA achieved an accuracy of 99.97%, and PCA-NB achieved an accuracy of 97.14%. The authors observe that this method is in competition with existing methods [14].

A. Turukmane and R. Devendiran suggest a sophisticated Intrusion Detection System (IDS) that is automated and based on machine learning as a solution to the detection of network intrusions in their research paper [15]. They employed techniques such as Min-Max normalization, zero-value processing, class inequality removal using ASmoT, and meaningful feature extraction using M-SvD to analyze the CSE-CIC-IDS 2018 and UNSW-NB15 datasets. Furthermore, the Mud Ring Multilayer SVM (M-MultiSVM) was employed to classify the attack classes, and the Opposition-based Northern Goshawk Optimization (ONgO) was employed to select the most appropriate features [16]. Consequently, the proposed system obtained an accuracy of 99.89% (CSE-CIC-IDS 2018) and 97.535% (UNSW-NB15). The K-Nearest Neighbor, Naïve Bayes, Support Vector Machine were tested in detecting 19 different types of attacks based on data from the UNSW-NB15 dataset, and the SVM algorithm showed the best results with an accuracy of 97.78%, exceeding the others. Despite its ability to achieve 98.9% accuracy on the KDD'99 Cup dataset, the SVM algorithm proved less effective than the Random Forest algorithm, which demonstrated the highest result of 99.81% [17]. The study also analyzed the classification of machine learning algorithms used in intrusion detection systems (IDS) in areas such as IoT, Big Data, Fog computing, and 5G networks. However, the following study notes that the advantage of this algorithm is revealed when computational resources are limited [18].

The comparative analysis of state-of-the-art ensemble and classical machine learning algorithms on the real and diverse UNSW-NB15 dataset is the unique feature of this study. The concentration is on the efficacy of the models in the face of cyber threats.

2 Materials and methods

2.1 Description of machine learning

Machine learning (ML) is a subfield of artificial intelligence that enables systems to autonomously learn from data and make predictions or decisions without the need for explicit programming. It is predicated on the development of models that can recognize patterns in historical data and apply those patterns to the analysis of new input data. The machine learning process is comprised of numerous critical stages, each of which is crucial to the development of the effective model depicted in Figure 1.

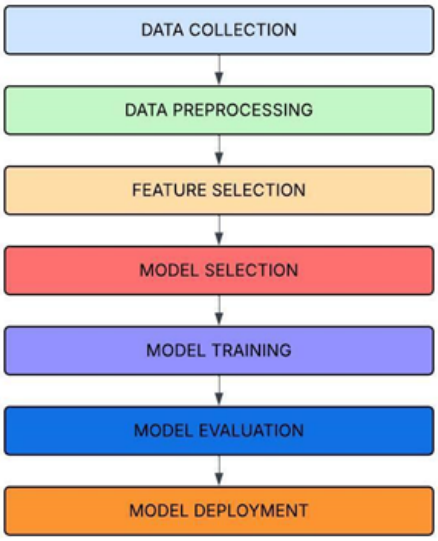

Figure 1 – Block diagram of the machine learning process

The process commences with the acquisition of data from a variety of sources, including sensors, databases, and web services. Subsequently, the data is subjected to preprocessing to eliminate outliers, duplicates, noise, normalize, and code features. Subsequently, feature selection is implemented, which involves the identification of the most critical attributes that influence the outcome in order to enhance the model's precision and simplify its structure. This is followed by the selection of an appropriate learning algorithm based on the data structure and problem type. The subsequent phase involves the model's training, which involves the analysis of the training data to identify the internal dependencies between inputs and outputs. Following the training process, the model is assessed on a test sample using a variety of metrics, including the F1-measure and accuracy, to guarantee its efficacy. The model's final stage is to be deployed in a real environment, where it begins to process real data and generate predictions.

2.2 Data preparation

In 2015, the Australian Centre for Cyber Security developed the UNSW-NB15 dataset, which served as the foundation for the investigation. This dataset was produced by utilizing IXIA PerfectStorm traffic, which offers a more accurate and current perspective on the behavior of both legitimate and malicious network traffic. The dataset comprises a total of 175,341 network sessions, of which 56,000 are normal and 119,341 are attacking. The nine primary categories of attack traffic are as follows: Fuzzers (introduction of incorrect data to disrupt the system), Analysis (analysis and scanning attacks), Backdoors (stealthy remote access), DoS (denial of service attacks), Exploits (vulnerability exploitation), Generic (cryptographic attacks), Reconnaissance (gathering information about the system), Shellcode (injection of malicious code through the shell), and Worms (self-propagating malware).

UNSW-NB15 comprises 49 attributes, each of which delineates a distinct characteristic of a network connection. Basic attributes (e.g. IP addresses, ports, and protocols), traffic content attributes (e.g. HTTP methods and FTP commands), temporal attributes (duration, connection start and end timestamps), flow attributes (amount of data transferred, number of packets), and additional automatically generated attributes (e.g. delay values, packet intervals, and TCP connection attributes) are all possible categories into which these attributes can be roughly categorized. Furthermore, class labels are present, including a binary label for normal traffic (0) and an attack_cat label for aggressor traffic (1), which denotes the category of the attack. In order to guarantee the generalizability of the models, the data were partitioned into training and test samples in an 80/20 proportion using train_test_split (random_state=42).

The id and attack_cat features were eliminated from the dataset during the preprocessing stage, as they were deemed unsuitable for binary classification tasks. In order to guarantee the accurate performance of feature scale-sensitive models, including Support Vector Machine (SVM) and K-Nearest Neighbors (KNN), categorical variables were transformed using one-hot coding and numeric features were scaled using StandardScaler. This data preparation enabled the formation of a balanced and homogeneous sample for the purpose of further training machine learning models.

In this study, the UNSW-NB15 dataset was selected as the primary benchmark for evaluating the performance of machine learning models. To justify this choice, a comparative analysis was conducted with two other widely used datasets in the field of intrusion detection: NSL-KDD and CICIDS2017. The comparison highlights the strengths of UNSW-NB15 in terms of modernity, class balance, and relevance to contemporary cybersecurity challenges.

Table 1 – Comparison of Popular NIDS Datasets

Characteristic	UNSW-NB15	NSL-KDD	CICIDS2017
Release Year	2015	2009 (based on 1999 KDD99)	2017
Number of Features	49 (original) / ~186 (after one-hot)	41	>80
Number of Attack Types	9 categories	4 categories	15+ attack scenarios
Traffic Type	Modern protocols (FTP, SSH, HTTP, DNS)	Outdated protocols (Telnet, ICMP)	Realistic mixed traffic
Class Balance	Moderate (≈60% attack traffic)	Highly imbalanced	Highly imbalanced
Threat Relevance	Modern threats	Outdated threats	Modern threats
Dataset Size	~2 million packets	~150,000 records	>3 million packets
Realism	High (generated via IXIA tools)	Low (synthetic benchmark)	Very high (real-world captures)

As shown in the comparison in Table 1, NSL-KDD is an improved version of the outdated KDD99 dataset. However, it does not reflect modern network behavior or attack patterns. CICIDS2017 offers high realism and a diverse set of attack scenarios but suffers from extreme class imbalance and excessive volume, which complicates training and evaluation processes.

In contrast, UNSW-NB15 provides a balanced trade-off between size, realism, and diversity of attack types. It includes up-to-date network traffic and protocols, a reasonable number of well-defined classes, and moderate class balance, making it ideal for building and benchmarking intrusion detection models.

2.3 Model training and evaluation

The system initially receives input network traffic, which is data regarding network connections, including both legitimate and malicious sessions. This data is gathered using specialized tools, such as IXIA PerfectStorm, to provide a realistic and current representation of the network behavior in Figure 2.

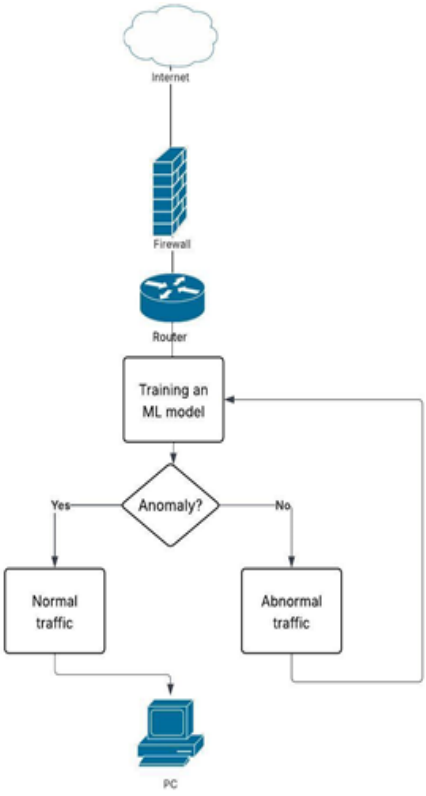

Figure 2 – Process of anomaly detection in network traffic using machine learning

The traffic is subsequently subjected to a data preprocessing stage. The set is purged of irrelevant features, including connection identifiers and assault categories, as they do not offer any valuable information for binary classification tasks. Numerical features are scaled using standard normalization, while categorical variables are transformed using one-hot encoding. This process guarantees the accurate operation of machine learning algorithms, particularly those that are vulnerable to the scope of the data, such as the support vector method or K-nearest neighbors.

Machine learning models are trained using the data that has been preprocessed. The study employs six distinct algorithms: Random Forest, K-Nearest Neighbors, Support Vector Machine, XGBoost, LightGBM, and Logistic Regression. From interpretability and fundamental accuracy to high performance and robustness to overtraining, each has its own parameters and advantages. Training is conducted on a labeled sample that is divided into normal and assault traffic.

The model that has been trained subsequently acquires new data and performs classification. The decision phase is responsible for determining whether a specific network connection is normal or anomalous. The connection is classified as anomalous if the features correspond to malevolent behavior patterns. Therefore, the system is capable of efficiently filtering network traffic, thereby detecting potential hazards in near real-time.

Six machine learning algorithms were selected for comparison:

Random Forest

The Random Forest Classifier is implemented by employing a fixed random number generator sid (random_state=42) and 100 decision trees (n_estimators=100). This method of ensemble construction achieves high accuracy and robustness against overfitting by constructing multiple trees and averaging their predictions.

RandomForestClassifier(n_estimators=100, random_state=42)

K-Nearest Neighbors (KNN)

The parameter n_neighbors=5 is employed in conjunction with the K nearest neighbors method. It categorizes objects based on the most labels among the adjacent neighbors in the feature space. It is imperative to scale the data prior to training, as the distances between nodes have a direct impact on the outcome.

KNeighborsClassifier(n_neighbors=5)

Support Vector Machine (SVM)

The probability parameter = True and the linear kernel (kernel='linear') were implemented in the support vector algorithm to facilitate the construction of the ROC curve. SVM is capable of effectively handling binary classification tasks in the presence of a substantial number of features.

SVC(kernel='linear', probability=True, random_state=42)

XGBoost

In order to construct a robust gradient-enhancing model, the XGBoost algorithm is employed with the following parameters: n_estimators=200, learning_rate=0.1, and max_depth=6. The loss during the learning phase is estimated using the eval_metric='logloss' parameter. This algorithm exhibits a high degree of accuracy and a strong capacity for generalization.

XGBClassifier(n_estimators=200, learning_rate=0.1, max_depth=6, eval_metric='logloss', random_state=42)

LightGBM

LightGBM, like XGBoost, is a gradient-based booleaning algorithm; however, it operates more efficiently on data of substantial magnitude. The parameters n_estimators=200, learning_rate=0.1, and max_depth=10 are employed to achieve a balance between performance and accuracy.

LGBMClassifier(n_estimators=200, learning_rate=0.1, max_depth=10, random_state=42)

Logistic regression

In order to ensure convergence with a substantial number of features, the logistic regression model was implemented with an increased number of iterations (max_iter=1000). Logistic regression is a fundamental model that is frequently employed as an initial benchmark and is easily interpreted.

LogisticRegression(max_iter=1000, random_state=42)

Each model was trained on a standardized training sample and subsequently tested on the deferred portion. The evaluation was conducted using the following metrics:

- Accuracy – The total proportion of correct predictions.
- Precision – The accuracy of attack classification.
- Recall (completeness) - the model's ability to find all attacks.
- F1-score is the harmonic mean between precision and recall.
- ROC AUC is the area under the ROC curve showing the quality of the binary classification at different thresholds.
- Confusion Matrix – for the visual evaluation of classification errors.

A comparative ROC curve was also generated, which enabled a visual comparison of the models' quality in terms of AUC by calculating TPR (True Positive Rate) and FPR (False Positive Rate) values for each model and interpolating them over the total interval.

3 Results and discussion

3.1 Results of the ML algorithms

Based on the results of the experiments, the LightGBM model demonstrated the most effective performance, as illustrated in Table 2. The model's exceptional capacity to differentiate between assaults and normal traffic was demonstrated by its high classification accuracy of 95.86% and an area under the ROC curve (AUC-ROC) value of 94.6%. Furthermore, the model demonstrated a precision of 96.02%, recall of 97.98%, and F1-count of 96.99%, indicating a high equilibrium between the accuracy and completeness of predictions.

The effectiveness of LightGBM in the network attack detection assignment is confirmed by these results, which are based on the features presented in Figure 3.

Table 2 – Results of the ML algorithms

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC
Random Forest	95,83	95,96	98,0	96,97	94,6
KNN	93,77	94,45	96,53	95,48	92,2
SVM	93,27	91,12	99,86	95,29	89,5
XGBoost	95,74	95,91	97,92	96,9	94,5
LightGBM	95,86	96,02	97,98	96,99	94,6
Logistic Regression	93,42	92,09	98,83	95,34	90,3

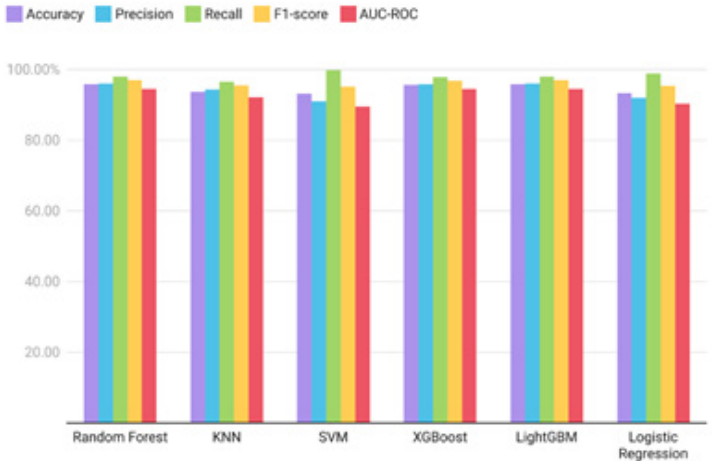

Figure 3 – Results of the ML algorithms

LightGBM showed the highest AUC-ROC value (94,6) in Figure 4. In general, LightGBM was the most effective model. The study's findings indicated that machine learning models are highly effective in the detection of attacks. The LightGBM model, in particular, demonstrated exceptional performance on all critical metrics, owing to its ensemble nature and capacity to effectively adjust to the complexity of the data. This demonstrates that it will be highly beneficial in real-world applications, particularly when analyzing large volumes of network traffic.

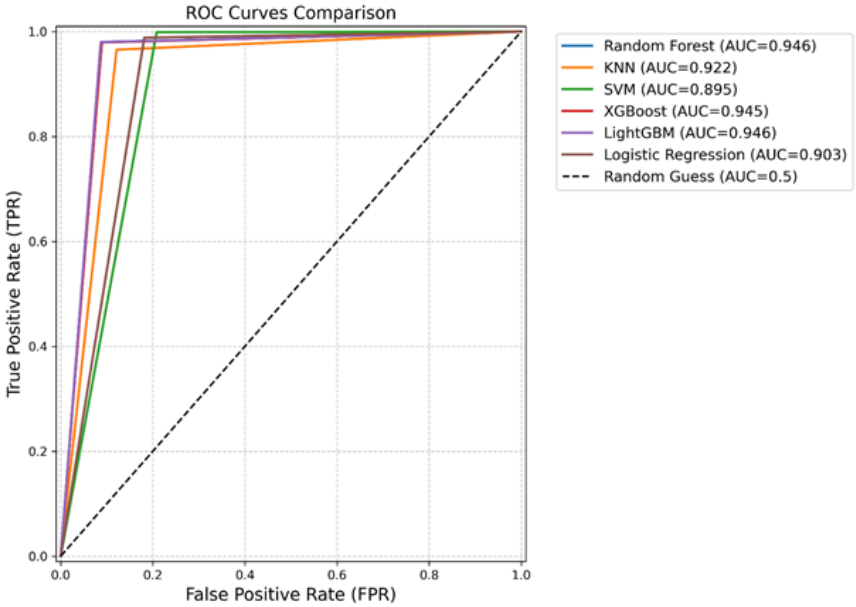

Figure 4 – Comparison of ROC curves for the six models

Similar high performance was also obtained by the Random Forest and XGBoost models. This is a result of their structural similarity (ensembles of decision trees) and their capacity to iteratively reduce errors. The SVM model was effective in maximizing assault coverage (Recall); however, it was inferior in terms of overall performance (AUC-ROC, Precision). Although the logistic regression and KNN algorithms are straightforward and basic models, they demonstrated relatively limited performance in identifying complex types of attacks, as illustrated in Figure 5.

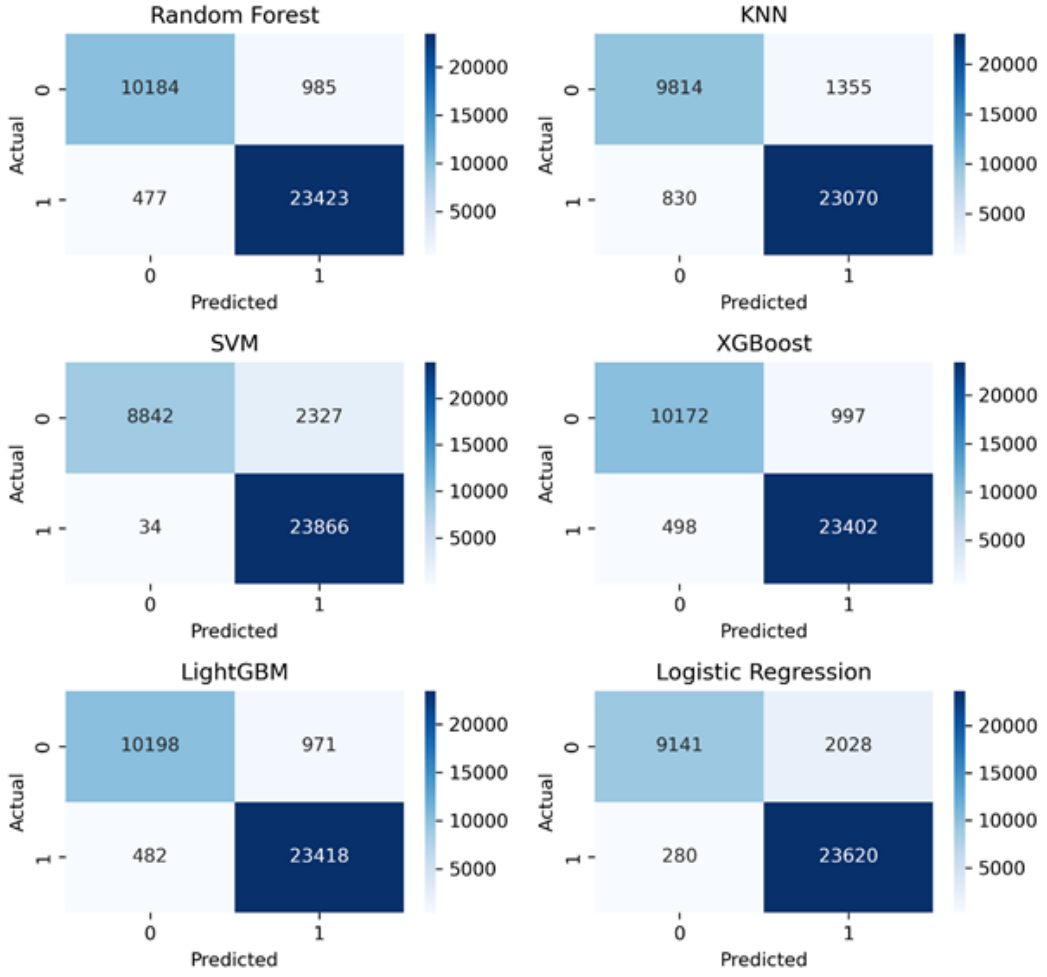

Figure 5 – Confusion matrix analysis for the six models

It is important to acknowledge that the UNSW-NB15 dataset utilized in the study encompasses specific categories of network attacks; however, it may be deficient in certain aspects of real-time complexity. Additional testing is necessary to conduct a comprehensive assessment of the performance of these models in real-world scenarios. Furthermore, it is advised that future research make use of alternative datasets (e.g., CICIDS, NSL-KDD) and assess the efficacy of the models' application to real-time systems.

Therefore, the findings of this investigation demonstrate that reinforcement models, including LightGBM, are ideal for practical application in the context of automated attack detection, thereby facilitating the development of effective network security decisions.

An analysis of classification errors by attack category (attack_cat) was conducted to conduct a more thorough assessment of the quality of the models. False positive (FP) and false negative (FN) classifications were given particular attention, as they are essential components of intrusion detection systems.

Table 3 provides a summary of the analysis' findings, including the quantity of FP and FN errors for each category of attacks.

Table 3 – Classification errors by attack category

Attack category	False Positives	False Negatives
Fuzzers	0	416
Analysis	0	40
DoS	0	2
Exploits	0	18
Generic	0	1
Reconnaissance	0	3
Shellcode	0	2
Normal	971	0

The analysis indicates that the Exploits category has the maximum number of FNs, which may be attributed to the diverse attack behavior patterns in this group outlined in Figure 6. This suggests that it is necessary to enhance the sensitivity of models to this category.

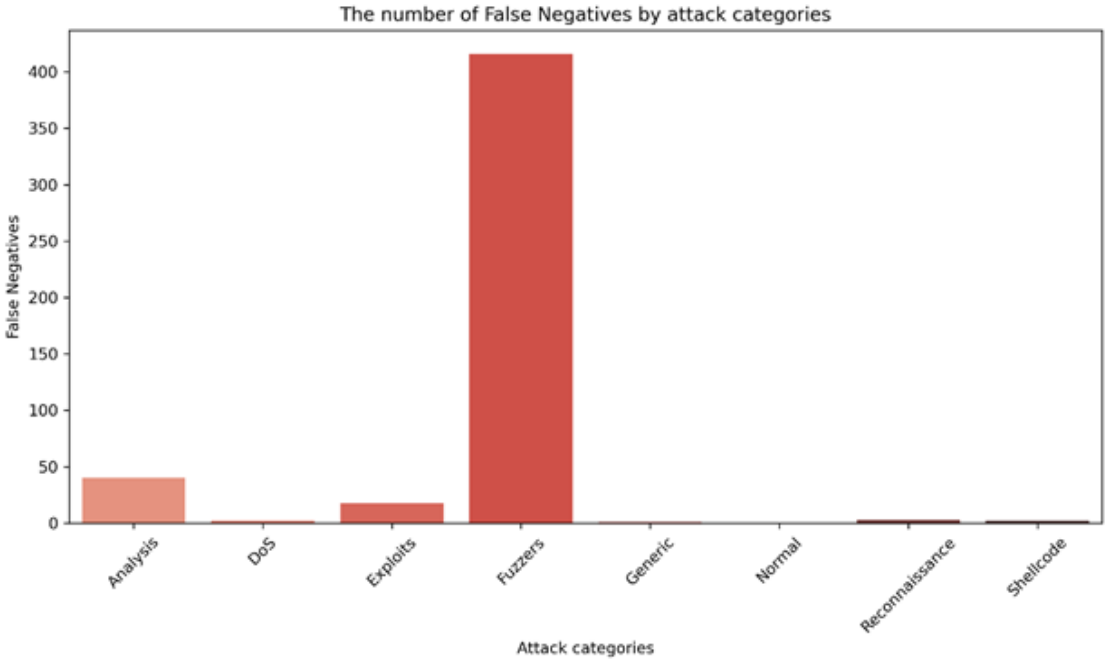

Figure 6 – Graph of the number of False Negatives by attack category

3.2 Evaluation of efficiency by resource consumption

The results of the measurements of peak memory usage, prediction time, and training time are summarized in Table 4.

For the practical application of the models, it is crucial to evaluate not only accuracy but also memory and time efficiency, as illustrated in Figure 7.

Table 4 – Comparison of resource intensity of models

Model	Training time (s)	Prediction time (s)	Memory (MB)
Random Forest	19.296	0.466	114.5
KNN	0.091	19.070	213.9
SVM	15833.081	60.857	416.5
XGBoost	2.820	0.030	1.5
LightGBM	1.906	0.071	4.1
Logistic Regression	8.979	0.007	212.3

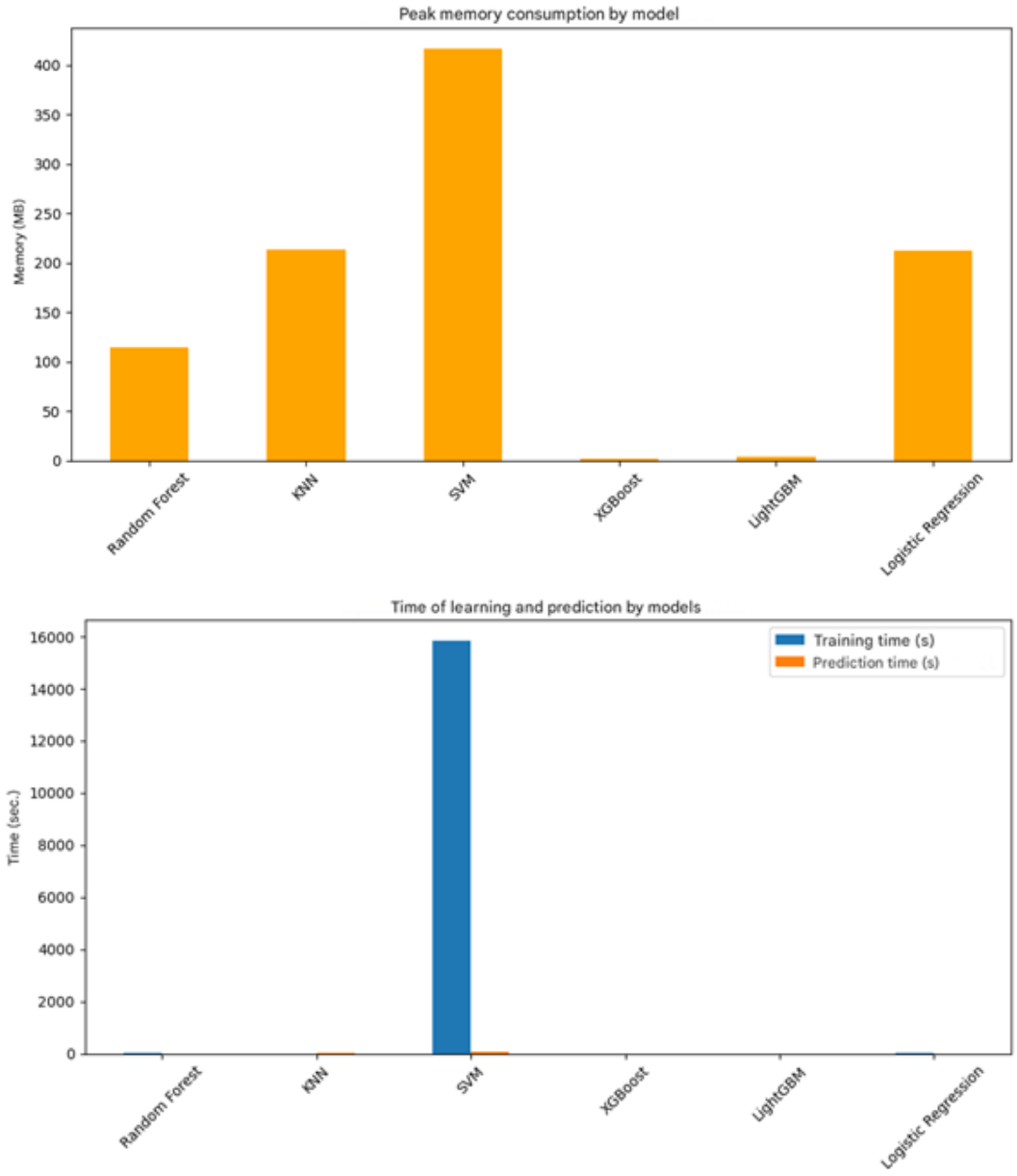

Figure 7 – Comparative plot of learning and memory time

The results indicate that LightGBM exhibits the highest accuracy/performance ratio, with minimal training time and relatively low resource consumption.

3.3 Configuring hyperparameters

Heuristics and expert settings were implemented to enhance the models' precision through manual hyperparameter selection. The parameters utilized for each model and their fitting procedure are summarized in Table 5.

Table 5 – Model hyperparameter settings

Model	Selection method	Parameters	Values
Random Forest	Manual selection	n_estimators	100
KNN	Manual selection	n_neighbors	5
SVM	Manual selection	kernel, probability	linear, True
XGBoost	Manual selection	n_estimators, max_depth, learning_rate	200, 6, 0.1
LightGBM	Manual selection	n_estimators, max_depth, learning_rate	200, 10, 0.1
Logistic Regression	Manual selection	max_iter	1000

To ensure optimal model performance, hyperparameter tuning was performed using Optuna, a Bayesian optimization framework. The study optimized n_estimators, learning_rate, and max_depth for LightGBM, as well as n_estimators, max_depth, and min_samples_split for Random Forest over 30 optimization trials each, using 3-fold cross-validation and ROC-AUC as the objective function in Table 6.

Table 6 – Results of LightGBM and Random Forest

Model	Accuracy	Precision	Recall	F1-score	AUC-ROC	Best Parameters
LightGBM	95,97	96,09	98,06	97,07	99,40	n_estimators=279, learning_rate=0.0910, max_depth=14
Random Forest	95,92	95,81	98,32	97,05	99,40	n_estimators=297, max_depth=26, min_samples_split=5

Due to their superior baseline performance, hyperparameter optimization was initially applied to LightGBM and Random Forest classifiers. Other models were evaluated with standard or manually selected parameters, as their optimization yielded marginal or less impactful improvements.

Both LightGBM and Random Forest, after Bayesian optimization via Optuna, achieved almost identical performance with an AUC of 99,4. LightGBM demonstrated slightly higher precision (96,09) and a better F1-score (97,07), while Random Forest had slightly better recall (98,32). These results confirm the high effectiveness of ensemble models in detecting network anomalies in the UNSW-NB15 dataset.

3.4 Importance of attributes

In order to enhance the interpretability of the models, feature importance diagrams were generated for the LightGBM and Random Forest models contained in Table 7.

Table 7 – Top 10 traits by importance (LightGBM)

№	sign	Importance value
1	smean	575
2	sbytes	527
3	ct_srv_src	432
4	ct_srv_dst	266
5	ct_dst_src_ltm	246
6	sload	229
7	dbytes	201
8	synack	198
9	stcpb	195
10	dmean	187

The 10 most critical features, as indicated by the LightGBM version, are illustrated in the Figure 8.

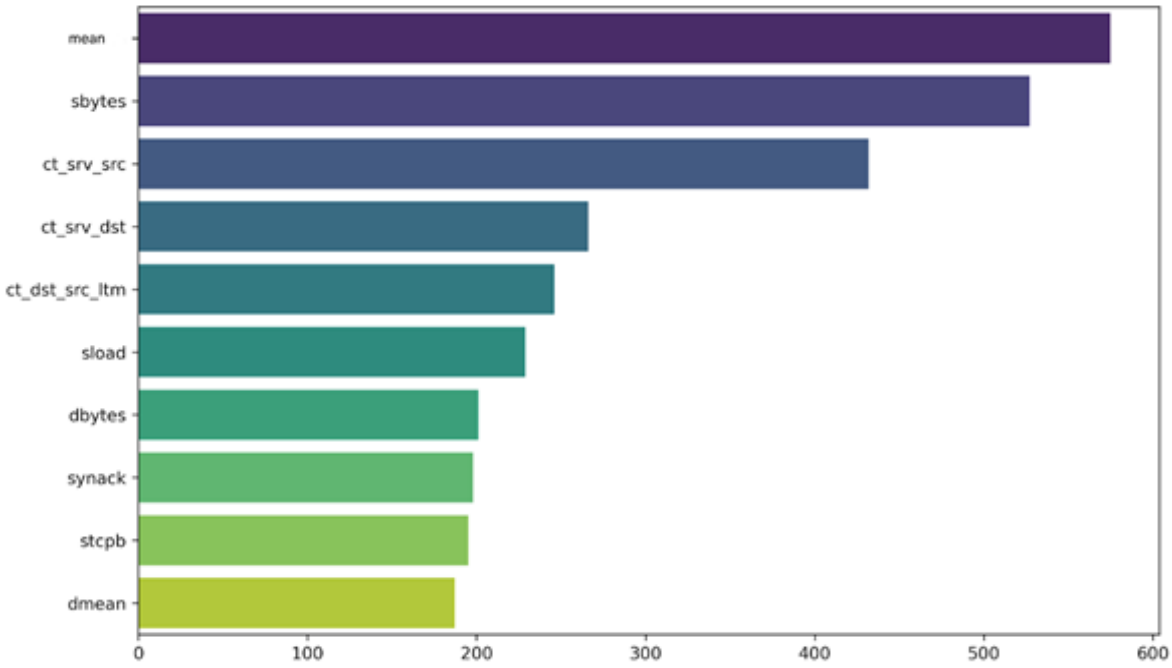

Figure 8 – Top 10 attributes by importance of LightGBM

Critical attributes include:

- sbytes (bytes sent)
- dbytes (bytes received)
- ct_state_ttl (connection state and TTL)
- ct_dst_sport_ltm (port destination frequency)
- sttl (packet lifetime)

These features have the greatest impact on the final solution of the model, which confirms their importance in the task of detecting anomalous behavior in the network.

3.5 Error Analysis

A refined evaluation of classification results underscores the significance of analyzing false positives (FP) and false negatives (FN) in intrusion detection. For the LightGBM model, 950 FPs and 463 FNs were observed, while the Random Forest model yielded 1028 FPs and 403 FNs. Although overall error rates were moderate, false negatives are particularly critical as they represent undetected attacks.

Key causes of false negatives include overlapping feature distributions between benign and malicious traffic, underrepresentation of specific attack types, and high intra-class variability. Conversely, false positives may arise from benign behavior mimicking attacks under certain conditions, increased noise due to high-dimensional features, and classifier uncertainty near decision boundaries.

To mitigate these errors, the following strategies are recommended:

- Employ class balancing techniques (e.g., SMOTE, class weight adjustment) to reduce FNs.
- Apply feature selection or dimensionality reduction to minimize FPs.
- Optimize classification thresholds using validation metrics.
- Evaluate model performance per attack type to identify and address specific weaknesses.

Several related works have evaluated machine learning models on the UNSW-NB15 dataset, yet their reported performance metrics vary depending on the selected models, features, and experimental setups.

For example, the study by Yakub Kayode Saheed et al. applied PCA and XGBoost to the UNSW-NB15 dataset and achieved an accuracy of 99.99%. However, the study did not report detailed F1-scores or confusion matrices, making it difficult to assess classification robustness across all classes [19].

Another study by Turukmane and Devendiran used a multi-class SVM ensemble on the CSE-CIC-IDS2018 dataset, reporting an F1-score of 99.89%. While the performance is high, the dataset used was significantly larger and more diverse, which limits the comparability to UNSW-NB15 results [20].

In contrast, our optimized LightGBM and Random Forest models, evaluated strictly on the UNSW-NB15 dataset, achieved F1-scores of 97.07% and 97.05%, respectively, along with ROC AUC scores of 0.994, which are highly competitive. Additionally, our models demonstrated strong generalization without relying on excessive feature engineering or complex ensembles.

This comparison emphasizes the effectiveness and simplicity of the proposed approach, showcasing that with proper hyperparameter optimization, standard tree-based models can achieve state-of-the-art results on modern intrusion detection datasets.

Conclusion

The performance of a variety of machine learning algorithms, such as Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost, LightGBM, and Logistic Regression, was compared in this study. All of the models that were evaluated exhibited exceptional performance in the classification of network traffic; however, certain algorithms were particularly noteworthy.

After applying Bayesian hyperparameter optimization using Optuna, both LightGBM and Random Forest models achieved significant performance improvements across all key evaluation metrics. The LightGBM model demonstrated the best overall performance, with an accuracy of 95.97%, precision of 96.09%, recall of 98.06%, and an F1-score of 97.07%, alongside a near-perfect ROC AUC score of 0.994. Random Forest model achieved nearly equivalent results, with an accuracy of 95.92%, precision of 95.81%, recall of 98.32%, F1-score of 97.05%, and the same ROC AUC score of 0.994. The high adaptability of this model to complex heterogeneous data structures, as well as its overall efficacy, are demonstrated by these figures. Gradient bousting, a high learning rate, the efficient handling of large quantities of data, and the ability to minimize errors at each iteration step are the primary benefits of LightGBM. In addition, the model is particularly well-suited for real-

time cyber threat detection systems due to its ability to perform well on categorical features and its reduced resource consumption.

Random Forest and XGBoost also demonstrated satisfactory performance, with an F1-estimation that exceeded 96.9% and an AUC-ROC value of approximately 94.5%, in addition to LightGBM. These findings substantiate the reliability of ensemble methods and their exceptional prediction accuracy in network security tasks.

The SVM model's recall was 99.86%, which was an intriguing characteristic that suggested its capacity to accurately identify all affirmative cases. Additionally, it demonstrated an exceptionally high level of sensitivity to attacks. Nevertheless, its AUC-ROC and overall accuracy were inferior to those of the ensemble models. The Logistic Regression and KNN models, despite their simplicity, also obtained satisfactory results; however, they demonstrated less robustness when confronted with more intricate types of attacks.

The research demonstrated that LightGBM is the most balanced model in terms of speed, responsiveness, and accuracy. These attributes render it an ideal choice for practical implementation in network threat detection systems. The significance of meticulously selecting a machine learning model in real-world applications is underscored by the paper, which considers the complexity of data, the availability of computational resources, and the accuracy requirements.

REFERENCES

1 Zhexebay, D., Skabylov, A., Ibraimov, M., Khokhlov, S., Agishev, A., Kudaibergenova, G., Orazakova, A., & Agishev, A. Deep Learning for Early Earthquake Detection: Application of Convolutional Neural Networks for P-Wave Detection. Applied Sciences, 15(7), 3864 (2025). https://doi.org/10.3390/app15073864.

2 Moulaei, K., Shanbehzadeh, M., Mohammadi-Taghiabad, Z., Mousavi, S. F., & Jafari, S. Comparing machine learning algorithms for predicting COVID-19 mortality. BMC Medical Informatics and Decision Making, 22(1), 2 (2022). https://doi.org/10.1186/s12911-021-01742-0.

3 Seydi, S. T., Kanani-Sadat, Y., Hasanlou, M., Sahraei, R., Chanussot, J., & Amani, M. Comparison of Machine Learning Algorithms for Flood Susceptibility Mapping. Remote Sensing, 15(1), 192 (2023). https://doi.org/10.3390/rs15010192.

4 Zhao, Z., Islam, F., Waseem, L. A., Tariq, A., Nawaz, M., Islam, I. U., Bibi, T., Rehman, N. U., Ahmad, W., Aslam, R. W., Raza, D., & Hatamleh, W. A. Comparison of three machine learning algorithms using Google Earth Engine for Land Use Land Cover classification. Rangeland Ecology & Management, 92, 129–137 (2024). https://doi.org/10.1016/j.rama.2023.10.007.

5 Ijaz, M., Durad, M. H., & Ismail, M. Static and dynamic malware analysis using machine learning. In 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST) (pp. 687–691). IEEE (2019). https://doi.org/10.1109/IBCAST.2019.8667136.

6 Akhtar, M. S., & Feng, T. Malware Analysis and Detection Using Machine Learning Algorithms. Symmetry, 14(11), 2304 (2022). https://doi.org/10.3390/sym14112304.

7 Baker del Aguila, R., Contreras Pérez, C. D., Silva-Trujillo, A. G., Cuevas-Tello, J. C., & Nunez-Varela, J. Static Malware Analysis Using Low-Parameter Machine Learning Models. Computers, 13(3), 59 (2024). https://doi.org/10.3390/computers13030059.

8 Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., & Yang, A. Comparative research on network intrusion detection methods based on machine learning. Computers & Security, 121, 102861 (2022). https://doi.org/10.1016/j.cose.2022.102861.

9 Ahmad, Z., Khan, A. S., Shiang, C. W., Abdullah, J., & Ahmad, F. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. Transactions on Emerging Telecommunications Technologies, 32(1), e4150 (2021). https://doi.org/10.1002/ett.4150.

10 Asif, M., Abbas, S., Khan, M. A., Fatima, A., Khan, M. A., & Lee, S.-W. MapReduce based intelligent model for intrusion detection using machine learning technique. Journal of King Saud University – Computer and Information Sciences, 34(10, Part B), 9723–9731 (2022). https://doi.org/10.1016/j.jksuci.2021.12.008.

11 Musleh, D., Alotaibi, M., Alhaidari, F., Rahman, A., & Mohammad, R.M. Intrusion detection system using feature extraction with machine learning algorithms in IoT. Journal of Sensor and Actuator Networks, 12(2), 29 (2023). https://doi.org/10.3390/jsan12020029.

12 Attou, H., Guezzaz, A., Benkirane, S., Azrour, M., & Farhaoui, Y. Cloud-based intrusion detection approach using machine learning techniques. Big Data Mining and Analytics, 6(3), 311–320 (2023). https://doi.org/10.26599/BDMA.2022.9020038.

13 Saheed, Y.K., Abiodun, A.I., Misra, S., Holone, M.K., & Colomo-Palacios, R. A machine learning-based intrusion detection for detecting Internet of Things network attacks. Alexandria Engineering Journal, 61(12), 9395–9409 (2022). https://doi.org/10.1016/j.aej.2022.02.063.

14 Moustafa, N., & Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), 1–6 (2015). https://doi.org/10.1109/MilCIS.2015.7348942.

15 Turukmane, A.V., & Devendiran, R. M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. Computers & Security, 137, 103587 (2024). https://doi.org/10.1016/j.cose.2023.103587.

16 Ajagbe, S.A., & Alabi, O.O. Comparative Study of Machine Learning Models Using UNSW Datasets. In The 45th Annual Conference of the South African Institute of Computer Scientists and Information Technologists (p. 69). URL: https://saicsit2024.mandela.ac.za/saicsit2024/media/Store/documents/SAICSIT_PGS_v1-3.pdf#page=72.

17 Hussain, A., Khatoon, A., Aslam, A., & Khosa, M. A comparative performance analysis of machine learning models for intrusion detection classification. Journal of Cybersecurity, 6,1 (2024). https://www.proquest.com/openview/08a4c605b57abbca71467cce40765b15/1?pq-origsite=gscholar&cbl=4585457.

18 Mishra, N., & Mishra, S. A Review of Machine Learning-based Intrusion Detection System. EAI Endorsed Transactions on Internet of Things, 10 (2024). https://doi.org/10.4108/eetiot.5332.

19 Tahri, R., Benslimane, Y., Rifi, M., & Maqnaoui, M. Intrusion detection system using machine learning algorithms. ITM Web of Conferences, 46, 02003 (2022). https://doi.org/10.1051/itmconf/20224602003.

20 Saranya, T., Sridevi, S., Deisy, C., Chung, T. D., & Ahamed Khan, M.K. Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Computer Science, 171, 1251–1260 (2020). https://doi.org/10.1016/j.procs.2020.04.133.

[1]Кикбаев Н.Е.,
магистрант, ORCID ID: 0009-0000-0145-5718,
e-mail: kikbaevnurbek@gmail.com
[1]Жексебай Д.М.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com
[2]Синь Ю.,
профессор, ORCID ID: 0000-0001-6169-0795,
e-mail: qyuxiao@purdue.edu
[3]Тынымбаев С.Т.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: s.tynym@gmail.com
[3]Айтмагамбетов А.З.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: a.aitmagambetov@iitu.edu.kz
[1]Абдижалилова Л.Б.,
магистрант, ORCID ID: 0009-0000-5965-7195,
e-mail: abdijalil.lazzat@bk.ru
[1]*Сқабылов А.А.,
PhD, ORCID ID: 0000-0002-5196-8252,
*e-mail: Alisher.skabylov@kaznu.edu.kz

[1]Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы қ., Қазақстан
[2]Солтүстік-Батыс политехникалық университеті, Сиан қ., Қытай
[3]Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан

ЖЕЛІЛІК ТРАФИКТЕГІ АНОМАЛИЯЛАРДЫ АНЫҚТАУ ҮШІН МАШИНАЛЫҚ ОҚЫТУ ӘДІСТЕРІН САЛЫСТЫРМАЛЫ ЗЕРТТЕУ

Аңдатпа

Киберқауіптердің қарқынды өсуіне және соның салдарынан желілік трафиктің артуына байланысты белгілі және жаңа шабуыл түрлерін жылдам анықтай алатын шабуылдарды анықтау жүйелеріне (IDS) сұраныс артып келеді. Желілік пакеттердің әрекетін автономды түрде талдау және оларды қалыпты немесе зиянды деп жіктеу үшін машиналық оқыту әдістерін пайдалану – бұл мәселені шешудің перспективалы тәсілі. Бұл зерттеудің мақсаты – иллюстрация ретінде желі деректерін талдауды қолдана отырып, желілік қауіпсіздік мәселелерін шешуде әртүрлі машиналық оқыту алгоритмдерінің тиімділігін бағалау. Зерттеу барысында UNSW-NB15 деректер жинағы негізінде желіге заңсыз кіруді анықтаудағы машиналық оқыту үлгілерінің өнімділігі тексерілді. Негізгі назар Random Forest, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), XGBoost, LightGBM және Logistic Regression модельдеріне аударылды. Талдау нәтижесінде барлық модельдер жоғары классификация дәлдігін көрсетті; алайда LightGBM моделі ең үздік нәтижелерге қол жеткізді. Атап айтқанда, ол дәлдік бойынша – 95,86%, нақтылық бойынша – 96,02% және F1-өлшем бойынша – 96,99% нәтижелерін көрсетті. Бұл оның күрделі әрі біртекті емес деректерді тиімді басқару мүмкіндігін растады. Жалпы алғанда, зерттеу желілік қауіпсіздік жүйелерінде қолданылатын үлгілерді таңдаудың маңыздылығын айқындады. Нәтижелер IDS жүйелерін жобалауда нақты мақсаттар мен деректердің ерекшеліктеріне сәйкес ең қолайлы машиналық оқыту үлгісін таңдаудың тиімділігін дәлелдейді.

Тірек сөздер: машиналық оқыту, желілік трафик, LightGBM, киберқауіпсіздік, IDS, деректерді талдау.

[1]Кикбаев Н.Е.,
магистрант, ORCID ID: 0009-0000-0145-5718,
e-mail: kikbaevnurbek@gmail.com
[1]Жексебай Д.М.,
PhD, ORCID ID: 0009-0008-1884-4662,
e-mail: zhexebay92@gmail.com
[2]Синь Ю.,
профессор, ORCID ID: 0000-0001-6169-0795,
e-mail: qyuxiao@purdue.edu
[3]Тынымбаев С.Т.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: s.tynym@gmail.com
[3]Айтмагамбетов А.З.,
профессор, ORCID ID: 0000-0002-9326-9476,
e-mail: a.aitmagambetov@iitu.edu.kz
[1]Абдижалилова Л.Б.,
магистрант, ORCID ID: 0009-0000-5965-7195,
e-mail: abdijalil.lazzat@bk.ru
[1*]Сқабылов А.А.
PhD, ORCID ID: 0000-0002-5196-8252,
e-mail: Alisher.skabylov@kaznu.edu.kz

[1]Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан
[2]Северо-Западный политехнический университет, г. Сиань, Китай
[3]Международный университет информационных технологий, г. Алматы, Казахстан

СРАВНИТЕЛЬНОЕ ИССЛЕДОВАНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ВЫЯВЛЕНИЯ АНОМАЛИЙ В СЕТЕВОМ ТРАФИКЕ

Аннотация

Спрос на системы обнаружения вторжений (IDS), которые могут оперативно определять как известные, так и новые типы атак, растет из-за быстрого расширения киберугроз и последующего увеличения сетевого

трафика. Использование методов машинного обучения для автономного анализа поведения сетевых пакетов и классификации их как нормальных или вредоносных является многообещающим способом решения этой проблемы. Целью данного исследования является оценка пригодности различных алгоритмов машинного обучения для решения проблем сетевой безопасности путем использования анализа сетевых данных в качестве иллюстрации. В данном исследовании оценивается эффективность моделей машинного обучения при обнаружении сетевых вторжений с использованием набора данных UNSW-NB15. Основная цель этого исследования – оценить эффективность различных моделей машинного обучения, включая случайный лес, метод K-ближайших соседей (KNN), опорную векторную машину (SVM), XGBoost, LightGBM и логистическую регрессию, в приложениях сетевой безопасности. Согласно анализу, все модели продемонстрировали высокую точность классификации; однако модель LightGBM достигла самых значительных результатов. Эта модель продемонстрировала самые высокие значения точности (95,86%), точности (96,02%) и F1-меры (96,99%), что подтверждает ее способность эффективно управлять сложными и неоднородными данными. В целом исследование подчеркивает важность выбора наиболее подходящей модели на основе целей системы безопасности и специфики данных.

Ключевые слова: машинное обучение, сетевой трафик, LightGBM, кибербезопасность, IDS, анализ данных.

Article submission date: 29.06.2025