

Advancing network security: a comparative research of machine learning techniques for intrusion detection

Shynggys Rysbekov¹, Abylay Aitbanov¹, Zukhra Abdiakhmetova², Amandyk Kartbayev¹

¹School of Information Technology and Engineering, Kazakh-British Technical University, Almaty, Kazakhstan

²Informatics Department, Faculty of Information Technology, Al-Farabi Kazakh National University, Almaty, Kazakhstan

Article Info

Article history:

Received Mar 1, 2024

Revised Nov 5, 2024

Accepted Nov 20, 2024

Keywords:

Anomaly detection

Hybrid model

Intrusion detection systems

Machine learning

Network security

Neural networks

Oversampling methods

ABSTRACT

In the current digital era, the advancement of network-based technologies has brought a surge in security vulnerabilities, necessitating complex and dynamic defense mechanisms. This paper explores the integration of machine learning techniques within intrusion detection systems (IDS) to tackle the intricacies of modern network threats. A detailed comparative analysis of various algorithms, including k-nearest neighbors (KNN), logistic regression, and perceptron neural networks, is conducted to evaluate their efficiency in detecting and classifying different types of network intrusions such as denial of service (DoS), probe, user to root (U2R), and remote to local (R2L). Utilizing the national software laboratory knowledge discovery and data mining (NSL-KDD) dataset, a standard in the field, the study examines the algorithms' ability to identify complex patterns and anomalies indicative of security breaches. Principal component analysis is utilized to streamline the dataset into 20 principal components for data processing efficiency. Results indicate that the neural network model is particularly effective, demonstrating exceptional performance metrics across accuracy, precision, and recall in both training and testing phases, affirming its reliability and utility in IDS. The potential for hybrid models combining different machine learning (ML) strategies is also discussed, highlighting a path towards more robust and adaptable IDS solutions.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Amandyk Kartbayev

School of Information Technology and Engineering, Kazakh-British Technical University

Tole Bi 59, Almaty, Kazakhstan

Email: a.kartbayev@gmail.com

1. INTRODUCTION

In recent years, the rapid expansion of internet of things (IoT) technologies have significantly increased both the number of internet users and the variety of applications, consequently heightening network connectivity. This development, however, has introduced numerous security vulnerabilities. Traditional security measures like firewalls, data encryption, and user authentication have been deployed to counteract these threats. Although effective against many types of attacks, these conventional methods often lack the capacity for in-depth packet analysis, leaving them vulnerable to more complex attacks. To address these limitations, intrusion prevention systems (IPS) and intrusion detection systems (IDS) have been implemented. These systems, utilizing sophisticated algorithms from machine learning, deep learning, and artificial intelligence, provide a deeper data analysis to improve attack detection. While IPS units combine detection and prevention functionalities, IDS units focus primarily on detection and traffic analysis.

The surge in internet connectivity and data transfer rates has also led to an increase in network anomalies and a corresponding rise in cyber-attacks. According to a recent vulnerability and threat report by

Skybox Security, there were 17 thousand new vulnerabilities recorded in 2019, reflecting a 3.8% increase from the previous year. In response to these growing threats, both the public and private sectors are significantly increasing their investment in cybersecurity technologies. A report by Crystal Market Research indicates that the cybersecurity market, which was valued at approximately USD 58 billion in 2012, is expected to soar to USD 173 billion by 2022. As internet usage continues to climb, cybersecurity firms are challenged to develop more sophisticated technologies and methods beyond traditional security measures. This innovation has led to the advent of proactive security technologies such as network behavior analysis, machine learning, and threat intelligence, with IDS systems increasingly central to enhancing responsiveness to cyber threats.

Recent research has focused on the application of machine learning (ML) algorithms for anomaly detection in network security [1]. These algorithms, trained on extensive datasets, are evaluated for their ability to identify potential attacks. Historical studies have predominantly examined algorithms such as support vector machine (SVM) and artificial neural network (ANN) for their efficiency in handling large datasets for network intrusion detection [2].

Among the various security technologies available, IDS stands out as a dynamic and robust solution specifically designed to detect particular network threats. An IDS continuously monitors network activities, scrutinizing them for any deviations from standard operations or abnormalities. IDS utilize two main detection methodologies: signature-based and rule-based (anomaly-based). Signature-based IDS systems work by comparing network data to known patterns of attacks stored in databases, triggering alerts when matches are found [3]. The main limitation of this approach is its failure to identify new, unknown threats, whereas rule-based IDS systems, using anomalies, create a baseline of typical network activity and detect anomalies, enabling them to identify novel attacks through adaptive capabilities [4], [5]. As a software tool, it is adept at identifying suspicious behaviors and policy violations within a network. IDS can be classified into several types: network, host, protocol-based, application protocol-based, and hybrid IDS [6], [7]. The primary detection strategies employed by IDS are misuse detection (signature-based) and anomaly detection [8].

The landscape of network security research has been significantly shaped by studies that employ machine learning algorithms to enhance IDS capabilities. Notably, research referenced in [9] employed naive Bayes, random forest, and SVM to identify the types of attacks such as denial of service. This study highlighted the superior efficacy of the random forest classifier and suggested that integrating hierarchical clustering could further improve performance. Concurrently, another investigation in [10] undertook a comparative analysis of supervised ML classifiers including random forest, SVM, gaussian naive Bayes, and logistic regression, focusing on key performance metrics as the F1-Score and accuracy. This analysis reaffirmed the dominance of the random forest classifier across various datasets and parameters. A notable approach in [11] emphasized the importance of data pre-processing for a lightweight IDS, advocating for the elimination of redundant data to ensure the reliability and accuracy of ML algorithms. Similarly, the research in [12] introduced a supervised ML-based IDS to categorize online network data as normal or anomalous, although it was restricted to detecting only denial of service (DoS) and probe attacks.

Notable innovations in the field were discussed in [13], where a feature removal technique was applied within an SVM-based intrusion detection method, selecting the top nineteen features from the knowledge discovery and data mining cup 1999 (KDD-CUP99) dataset to boost algorithm efficiency. Additional studies, such as study [14], explored an enhanced self-adaptive Bayesian algorithm for anomaly detection, capable of processing large datasets effectively. Meanwhile, study [15] presented an innovative triangle-based k-nearest neighbor (KNN) method aimed at reducing data dimensionality, and study [16] tested an IDS employing fuzzy logic to derive fuzzy rules from definite rules using frequent items, achieving over 90% classification accuracy across all attack types. These advancements underscore a trend toward more precise and efficient methods for network intrusion detection.

Panigrahi *et al.* [17] assessed four supervised algorithms for identifying attacks such as a probe, DoS, remote to local (R2L), and user to root (U2R). They discovered that the decision tree classifier outperformed naive Bayes in prediction accuracy. Similarly, the research in [18] compared the efficacy of neural networks, and decision trees across false alarm rate, accuracy, and detection rate. They revealed that the decision tree algorithm surpassed its counterparts in efficiency.

Further exploring the integration of machine learning strategies, Akashdeep *et al.* [19] advocated for a combination of SVMs, multivariate adaptive regression splines (MARS), and ANNs to improve intrusion detection capabilities. A novel hybrid approach using SVM and radial basis function (RBF) was proposed in [20]. The significance of feature selection was addressed in [21], where a sequential search strategy was employed to evaluate the importance of attributes by their removal, enhancing algorithm performance by eliminating non-essential features. Building on this, study [22] underscored that not all dataset attributes are crucial, highlighting that the simple cart algorithm yielded superior results compared to other models.

In the context of vehicular ad hoc networks (VANETs), Meng *et al.* [23] discusses the use of SVM optimized with three intelligent algorithms-particle swarm optimization, ant colony optimization, and genetic algorithm, with latter showing superior performance. Finally, Kwon *et al.* [24] examined the security of IoT networks through quality of service indicators, proposing an IDS system based on classification using convolutional neural networks (CNNs). This approach was substantially better in the accuracy of detecting DoS attacks while reducing false positives, marking a significant advancement in the field.

Our paper aims to identify the most effective ML algorithm for anomaly detection in network environments, a vital advancement for network security solutions. It conducts a comprehensive evaluation of several machine learning algorithms using the national software laboratory knowledge discovery and data mining (NSL-KDD) dataset during the modeling phases. The NSL-KDD dataset [25], an improved version of the earlier KDD-CUP99 dataset [26], was chosen due to its more challenging set of data features, making it a suitable choice for rigorous testing of intrusion detection algorithms. The performance of these algorithms is tested across various types of network intrusions, providing a detailed comparison of their effectiveness in identifying different threat vectors. Therefore, the goal of this study is to identify the best algorithm for developing more efficient security mechanisms for network protection. This paper is organized as follows: Section 2 outlines the methodology for data analysis, section 3 discusses the study's findings with tables and illustrations, and section 4 concludes with a summary of key results.

2. METHOD

2.1. Overview of algorithms

Machine learning algorithms are getting pivotal in enhancing cybersecurity by detecting and neutralizing cyber anomalies. Anomaly detection focuses on identifying data elements, events, or observations that deviate significantly from expected patterns, which could indicate potential threats or malicious activity. Key algorithms that play a crucial role in this domain include:

- K-means clustering: this algorithm organizes data into distinct groups. It effectively identifies anomalies by isolating outliers that do not fit into any established cluster. During clustering, typical data points form cohesive groups, while anomalies, which differ significantly in feature space, remain un-clustered or loosely connected to clusters. This characteristic allows the algorithm to flag potential outliers by assessing their distance to core clusters.
- Isolated forests: based on a collection of decision trees, this unsupervised algorithm isolates anomalies efficiently by quickly segregating atypical data points. By constructing random decision trees that partition data points based on specific attributes, the algorithm can swiftly identify outliers, which makes this method effective in high-dimensional datasets and is computationally efficient.
- Support vector machines: as a supervised learning technique, SVMs classify data by creating a model that separates data points using a hyperplane. This method is particularly effective in anomaly detection because it identifies data points that are markedly distant from the rest of the dataset.
- Naive Bayes: classifiers comprise a group of classification methods based on Bayes' theorem. Usually, this collection includes various algorithms that operate under a common principle. Each method operates under the assumption that the occurrence of a specific feature within a class is independent of other features. This assumption significantly simplifies the calculation of probabilities, facilitating more efficient and streamlined classification.
- Neural networks: employing a series of interconnected nodes, these supervised learning algorithms excel in detecting anomalies by learning and recognizing patterns and correlations that deviate from typical behaviors. We used a perceptron, which is a type of neural network architecture that consists of several linear layers interconnected by nonlinear layers. It represents a foundational architecture in neural network design and is versatile enough to address various problems, including multiclass classification tasks. For the nonlinear layers, different functions can be used; the most commonly employed examples include rectified linear unit (ReLU), sigmoid, and their derivatives.
- Synthetic minority oversampling technique (SMOTE): is an upsampling algorithm that generates new synthetic samples from the minority class. It identifies several nearest neighbors for each minority class sample, selects a random subset based on the desired sample ratio, and then creates synthetic samples by choosing random points along line segments between neighbors and the original sample. Notably, SMOTE is tailored for numerical features and does not support categorical features [27].

Integrating these algorithms can significantly improve the precision of these systems. As digital landscapes continue to evolve, the need for robust machine learning applications in cybersecurity becomes more crucial. These algorithms are not only capable of identifying unusual patterns in network traffic or potential zero-day exploits by contrasting them against historical data but are also effective in recognizing internal threats through behavioral analysis compared to established norms [28].

The European Union Agency for Cybersecurity (ENISA) document "Cloud computing security risk assessment (CCSK)" provides an overview of existing vulnerabilities in modern information technology (IT) infrastructure. Among these vulnerabilities, the document highlights several critical areas:

- AAA vulnerabilities: these relate to authentication, authorization, and accounting, posing significant risks in managing access and tracking user activities.
- User provisioning vulnerabilities: issues here involve the management of user accounts, specifically the secure addition and removal of user access.
- Remote access: this refers to the vulnerabilities that can arise when external entities gain access to the cloud's management interfaces.
- Hypervisor vulnerabilities: as a key component in virtual environments, hypervisors present a prime target for attacks if not properly secured.
- Lack of resource isolation: these vulnerabilities lead to potential cross-tenant attacks and reputational damage due to shared resources.
- Encryption vulnerabilities: these include weak encryption of data in transit and at rest, and issues with the encryption processes themselves, such as inadequate key management or the inability to process encrypted data.

Consequently, the primary threats to information security in cloud services focus on critical components such as the hypervisor, virtual machines, network interactions, and audit mechanisms. Each of these elements requires robust security measures to mitigate the risk of compromise and ensure the integrity and availability of cloud services.

2.2. Datasets

The KDD99Cup dataset is the most widely cited dataset for classifying computer attacks, as evidenced by numerous publications. However, since it was created in 1999, its relevance for training modern traffic detection systems is increasingly questionable. Many new types of cyber-attacks have emerged since then that are not represented in this dataset, limiting its effectiveness in current applications. Another commonly referenced dataset in the field of computer attack detection is the University of New South Wales network-based 2015 (UNSW-NB15) dataset, which was compiled in 2015 and includes a broader range of contemporary cyber threats [29].

Despite its relevance, we chose not to use it for several reasons. Firstly, the dataset features a very small number of instances for each type of attack, which can lead to issues with model training and generalization. Secondly, the dataset we selected was gathered more recently, ensuring that it reflects the current threat landscape more accurately. Among more recent datasets, the adaptive wireless intrusion detection (AWID) datasets and the IoT dataset, collected in 2020, respectively, are noteworthy [30]. However, we opted not to utilize these datasets as well. They contain less common attack types that may not be relevant for broader applications, and they also have a less user-friendly data format. Incorporating these datasets would require users to convert their data into a more complex format, which could deter potential users from adopting the system we developed. Our goal is to ensure that the dataset we use is accessible, enhancing user engagement in detecting cyber threats.

This study uses the NSL-KDD dataset, an upgraded version of the KDD-CUP99 dataset that addresses some of its limitations. The improvements resolve several inherent issues that affected previous research outcomes. The KDD dataset was first introduced during "The third international knowledge discovery and data mining tools competition," aiming to develop an intrusion detection system that can differentiate within "good" and "bad" network traffic. Since then, the dataset has been widely utilized for practical applications, training, testing, and implementing machine learning technologies in the cybersecurity domain.

Over time, however, researchers have identified various problems within the dataset that can impact the results of studies and subsequent applications. As a response, the NSL-KDD dataset was proposed, incorporating necessary corrections and updates. The dataset, as shown in Table 1, contains 42 features that thoroughly describe incoming traffic. For our analysis, we will focus on the normal, R2L, and U2R classes, as these are most relevant for our learning objectives. It includes a variety of attack classes, specifically:

- DoS: attacks such as Back, Land, Neptune, Pod, Smurf, and TearDrop.
- Probe: includes attacks like Satan, Ipsweep, Nmap, and Portsweep.
- R2L: attacks such as guess password, Ftrpfake, Imap, Phf, Multihop, Warezmaster, Warezclient, and Spy.
- U2R: includes buffer overflow, Loadmodule, Rootkit, and Perl attacks.

For data preparation, we use principal component analysis (PCA), a statistical method that reduces high-dimensional data by identifying key features, retaining the most informative aspects of the dataset while reducing dimensionality. When addressing class imbalance, if the disparity is not excessive and sufficient

data is available, we may remove instances from the overrepresented class. However, it is crucial to retain enough information to avoid significantly impacting classification accuracy. Various methods can be employed for data reduction, such as random sampling from the larger class or clustering to select a fixed number of examples from each cluster. The latter approach preserves more information by ensuring that no cluster is entirely lost.

Table 1. Attributes of the dataset used in this study

#	Feature	#	Feature	#	Feature
1	Duration	15	Count	29	Srv diff host rate
2	Protocol type	16	Num file creations	30	Class labels
3	Flags	17	Num root	31	Same srv rate
4	Services	18	Num access files	32	Dst host count
5	Source bytes	19	Num shells	33	Dst host same srv rate
6	Destination bytes	20	Num outbound cmds	34	Dst host srv count
7	Wrong fragments	21	Is host login	35	Dst host srv diff host rate
8	Land	22	Is guest login	36	Dst host same src port rate
9	Hot	23	Su attempted	37	Dst host diff srv rate
10	Urgent	24	Srv error rate	38	Dst host rror rate
11	Logged in	25	Srv count	39	Dst host srv rror rate
12	Number of failed logins	26	Srv rror rate	40	Dst host sror rate
13	Root shell	27	Rror rate	41	Dst host srv sror rate
14	Num compromised	28	Srv sror rate	42	Diff srv rate

In our approach, we initiate PCA to reduce the dataset to 20 features, selecting components that capture the most variance while simplifying the data. The PCA model is then fitted to the feature matrix, transforming the features into a reduced space containing these 20 principal components. Subsequently, we split the dataset into training and testing sets, reserving 20% of the data for testing purpose, a standard practice for evaluating model performance.

2.3. System design

The developed system uses various methods for working with datasets to enhance the performance of models in the task of classifying computer attacks, following which we describe the implemented methods. The primary training pipeline can operate in two modes: the standard mode, where the model is trained with techniques for handling small classes, and the comparison mode, where a model is initially trained without these techniques and then with them. The accuracies of both models across different classes are calculated to compare their performance [31].

The process, as shown in Figure 1, begins with the creation of a model, which is then trained on pre-processed training data. For each sample in the minority class, several nearest neighbors from the same class are identified. From these neighbors, a random subset of the required size is selected, where the size depends on the ratio of the current number of samples in the minority class to the desired number of samples after the algorithm's application. After training, the model undergoes inference on test data, followed by calculations of class-specific and overall accuracies on this data, and the construction of a confusion matrix. Subsequently, visualization of the confusion matrix is invoked, and the constructed matrix along with the trained model's weights are saved. For each model training session, two versions of the confusion matrix are saved-one in absolute numbers and the other in relative values.

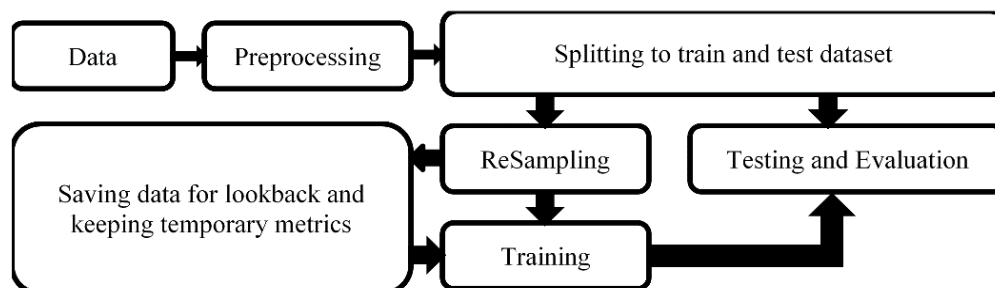


Figure 1. The general system's design and pipeline

The main file contains all the code related to methods for handling small classes. It includes an abstract class *BaseSampling* with an abstract method *fit_model*, as well as derived classes *NoUpSampling* and *BaseSmote* that implement this method. Additionally, this file contains the *create_sampling* function, which invokes the constructor for the desired model with the appropriate parameters and returns an instance of this model class. The behavior of this function is determined by parameters specified in the configuration file. There are also several small auxiliary scripts associated with this functionality.

The developed system offers multiple applications, as shown in Figure 2. Firstly, it can be utilized for inference on user data, provided that the chosen combination of model and techniques for handling small classes has been pre-trained. The second application involves the potential to reiterate the training process using one of the proposed models and methods for small classes, but with modifications to the training dataset or the model's hyperparameters. For instance, when deploying the perceptron model alongside methods for managing small classes, there is an improvement in the accuracy of detecting most types of attacks. However, an exception occurs with rare attack types, where accuracy remains low, and many attacks are mistakenly classified as benign traffic. This approach highlights the system's adaptability and its potential to refine detection capabilities under varied conditions. The primary distinction of our work from others in the field is our focus on improving classification accuracy specifically for rare classes, rather than maximizing overall classification performance. While other studies primarily track metrics such as precision, recall, and F1-score, our interest lies in understanding how classification accuracy for infrequent classes changes with different upsampling and downsampling techniques.

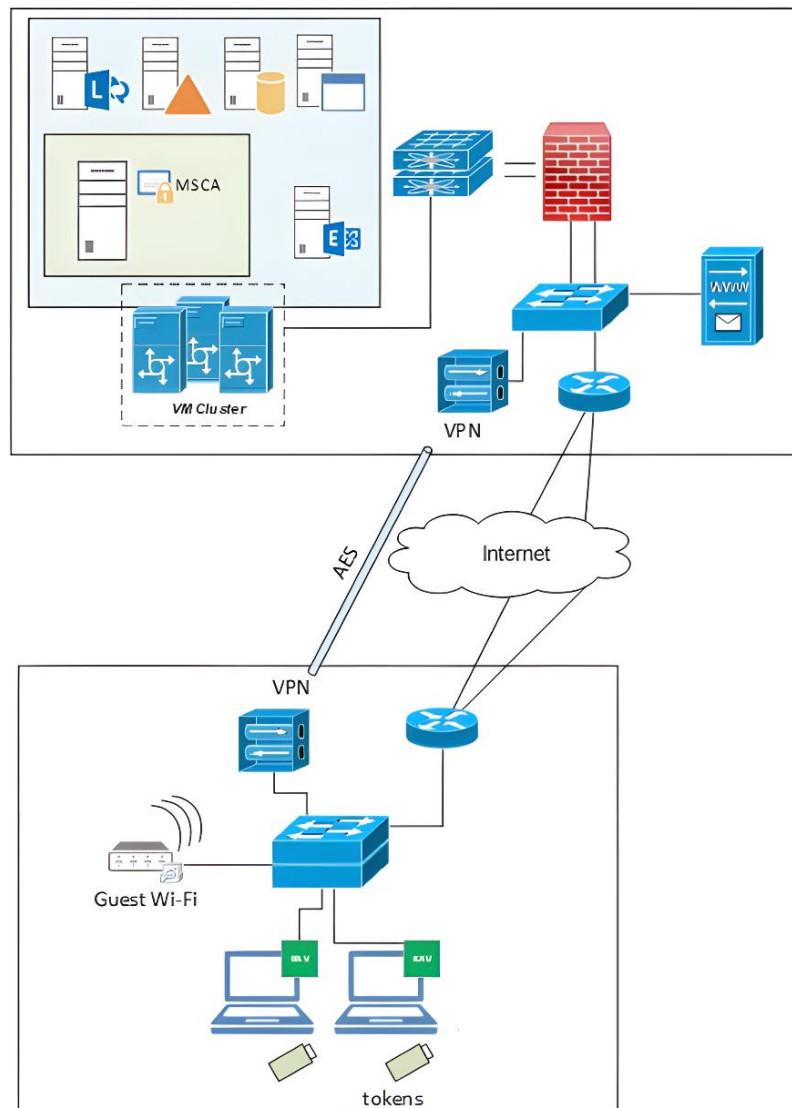


Figure 2. Overview of applications of the developed system

3. RESULTS AND DISCUSSION

We set up machine learning models that can spot traffic anomalies, experimented with these models, and compared them based on chosen metrics. Unlike linear regression, logistic regression performs well in scenarios where the classes are linearly separable or nearly separable, as it predicts the probability that an object belongs to a particular class. In its basic form, the model, like the other tested models, achieves high classification accuracy for popular attacks. For all the popular and many moderate types of attacks, the accuracy exceeds 95%. However, for rare attack classes, the accuracy drops to 0% - the model struggles to classify this type of attack, often misclassifying these attacks as benign. The performance metrics from the logistic regression model are displayed in Table 2.

Table 2. The performance metrics for the model

Metrics	Logistic regression
Training accuracy	87.97%
Test accuracy	87.62%
Training precision	83.81%
Test precision	83.56%
Training recall	91.85%

The confusion matrix as shown in Figure 3 is used to assess the performance of a classification model in this scenario, likely applied in evaluating an intrusion detection system that classifies outcomes as “normal” or “attack”. The logistic regression model demonstrates strong performance across all three metrics-precision, recall, and accuracy. The consistency observed from training to testing phases suggests that the model is robust, potentially performing well on new, unseen data. Although the model appears balanced between precision and recall, further examination of the specific business context is required to decide if the trade-off between these metrics is acceptable.

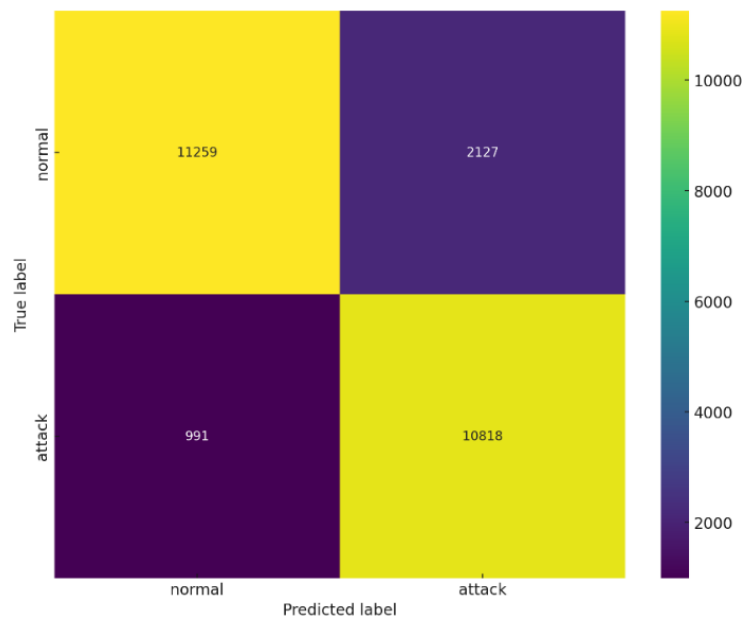


Figure 3. The confusion matrix for the logistic regression model

In the context of an intrusion detection system, a higher recall may be more critical than precision to ensure as many true threats as possible are detected, even if it results in some false alarms. However, if false positives are particularly costly or disruptive, enhancing precision becomes crucial. Especially, the k-nearest neighbor algorithm, a non-parametric, supervised learning classifier that relies on proximity for classification, is also evaluated. The performance results for the k-nearest neighbor model are shown in Table 3. This method's effectiveness hinges on its ability to classify data points based on the closest training examples in the feature space.

Table 3. The performance metrics for the model

Metrics	KNN
Training accuracy	99.05%
Test accuracy	98.94%
Training precision	99.23%
Test precision	99.06%
Training recall	98.73%

Overall, the results show that the KNN excels, achieving high accuracy, precision, and recall in both training and testing phases. This performance suggests that the model has effectively learned the data patterns and can generalize well to new data, indicating a good fit with no significant signs of overfitting or underfitting. In the confusion matrix, as shown in Figure 4, the top left quadrant (13,275) indicates true positives, where the model accurately predicted the positive class. The top right quadrant (111) captures false negatives, where positive cases were incorrectly predicted as negative. The bottom left quadrant (157) shows false positives, instances where the model mistakenly predicted negative cases as positive. Finally, the bottom right quadrant (11,652) represents true negatives, correctly identified negative cases, to distinguish effectively between class labels.

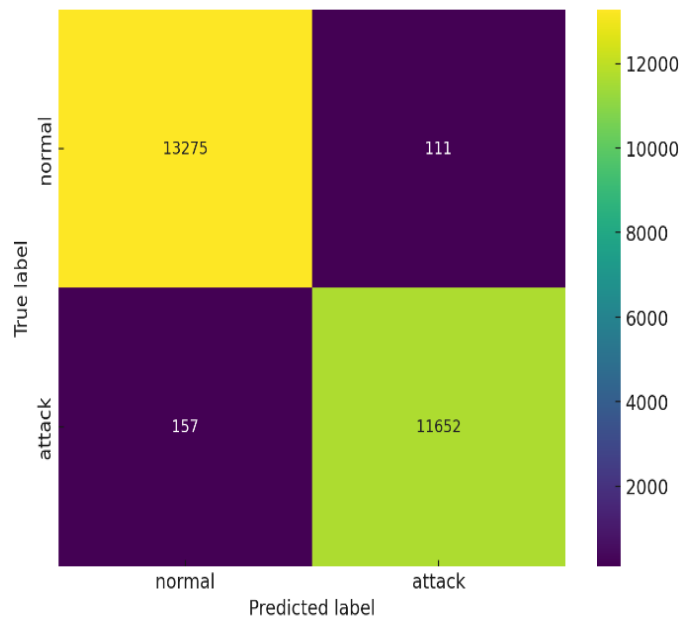


Figure 4. The confusion matrix for the KNN model

Naive Bayes classifiers comprise a family of classification methods based on Bayes' theorem. These are not singular algorithms but a suite that operates under a common principle. The main drawback of this method is the high number of false positives, which consequently reduces its usefulness for complex traffic scenarios. The decision tree is considered one of the most powerful and commonly used tools for categorization and prediction [32]. It consists of a hierarchical structure where each internal node represents a test on an attribute, each branch depicts the potential outcomes of these tests, and each leaf node assigns a class label. The performance results for the naive Bayes and decision tree models are detailed in Table 4.

Table 4. The performance metrics for the models

Metrics	Naïve Bayes	Decision tree
Training accuracy	91.80%	99.99%
Test accuracy	91.60%	99.86%
Training precision	92.62%	100%
Test precision	92.53%	99.84%
Training recall	89.47%	99.98%

The performance of the ANN model within the context of an IDS is exceptionally promising, as demonstrated by the results. The model achieves a near-perfect training accuracy of 99.994%, closely mirrored by a test accuracy. This slight discrepancy shows high effectiveness and the model is not overfitted, a common issue in machine learning. A high recall rate is crucial in an IDS as it reduces the likelihood of missing true attacks, a key factor in maintaining robust network security. The detailed results for the ANN model are displayed in Table 5.

It was discovered that reducing the feature set adversely impacted the performance of models when applied to real network traffic [33]. Based on this, the full feature set was used for detection and training, with the exception of internet protocol (IP) addresses, ports, and some other specific data. However, it is worth noting that in real infrastructure setups, keeping port information can be beneficial. Network "noise" can significantly reduce the effectiveness of experiments, which could include corrupted packets due to software errors. Since we know which services are running on specific ports, retaining this data can increase the detection accuracy.

Table 5. The performance metrics for the model

Metrics	ANN
Training accuracy	99.994%
Test accuracy	99.877%
Training precision	99.87%
Test precision	99.99%
Training recall	99.988%

4. CONCLUSION

This paper evaluates the effectiveness of various machine learning algorithms in IDS, employing techniques such as logistic regression, KNN, naive Bayes, decision tree, and neural networks. Utilizing the NSL-KDD dataset, the study offers insights into each algorithm's metrics, highlighting the need to select the appropriate algorithm based on its capability to identify various types of network attacks. Notably, the ANN model shows exceptional performance, demonstrating near-perfect metrics, which underscores the potential of neural networks in combating sophisticated cyber threats.

Looking forward, the paper suggests several research directions to enhance IDS capabilities: integrating optimization techniques to boost real-time detection efficiency, developing hybrid models that leverage the strengths of various machine learning methods, exploring advanced deep learning architectures to detect nuanced patterns in network traffic, and applying these models in diverse real-world settings to assess practical effectiveness. Additionally, further investigation into feature selection and engineering is recommended to improve model performance. With the growth of IoT devices and edge computing, exploring anomaly detection in these new contexts is also pertinent. These efforts will significantly advance the cybersecurity field, particularly in developing more advanced, reliable, and efficient intrusion detection systems.





REFERENCES

- [1] R. Liu, J. Shi, X. Chen, and C. Lu, "Network anomaly detection and security defense technology based on machine learning: a review," *Computers and Electrical Engineering*, vol. 119, Oct. 2024, doi: 10.1016/j.compeleceng.2024.109581.
- [2] T. Sowmya and E. A. Mary Anita, "A comprehensive review of AI based intrusion detection system," *Measurement: Sensors*, vol. 28, pp. 1–13, Aug. 2023, doi: 10.1016/j.measen.2023.100827.
- [3] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, pp. 1–22, Dec. 2019, doi: 10.1186/s42400-019-0038-7.
- [4] Q. Liu, V. Hagenmeyer, and H. B. Keller, "A review of rule learning-based intrusion detection systems and their prospects in smart grids," *IEEE Access*, vol. 9, pp. 57542–57564, 2021, doi: 10.1109/ACCESS.2021.3071263.
- [5] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019, doi: 10.1109/ACCESS.2019.2895334.
- [6] K. Sivaraman, R. M. V. Krishnan, B. Sundarraj, and S. Sri Gowthem, "Network failure detection and diagnosis by analyzing syslog and SNS data: applying big data analysis to network operations," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 9S3, pp. 883–887, Aug. 2019, doi: 10.35940/ijitee.I3187.0789S319.
- [7] A. D. Dwivedi, G. Srivastava, S. Dhar, and R. Singh, "A decentralized privacy-preserving healthcare blockchain for IoT," *Sensors*, vol. 19, no. 2, pp. 1–17, Jan. 2019, doi: 10.3390/s19020326.
- [8] Z. Azam, M. M. Islam, and M. N. Huda, "Comparative analysis of intrusion detection systems and machine learning-based model analysis through decision tree," *IEEE Access*, vol. 11, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.
- [9] H. Wu, "Feature-weighted naive Bayesian classifier for wireless network intrusion detection," *Security and Communication Networks*, no. 1, pp. 1–13, Jan. 2024, doi: 10.1155/2024/7065482.
- [10] W. Li, P. Yi, Y. Wu, L. Pan, and J. Li, "A new intrusion detection system based on KNN classification algorithm in wireless sensor network," *Journal of Electrical and Computer Engineering*, no. 1, pp. 1–8, 2014, doi: 10.1155/2014/240217.
- [11] F. S. D. Pranoto and Y. Asnar, "Development of machine learning subsystem in intrusion detection system for cyber physical




- system,” in *2023 International Conference on Electrical Engineering and Informatics (ICEEI)*, Oct. 2023, pp. 1–6, doi: 10.1109/ICEEI59426.2023.10346649.
- [12] S. B. Saidin and S. B. I. Hisham, “A survey on supervised machine learning in intrusion detection systems for Internet of Things,” in *2023 IEEE 8th International Conference on Software Engineering and Computer Systems (ICSECS)*, Aug. 2023, pp. 419–423, doi: 10.1109/ICSECS58457.2023.10256275.
- [13] A. Ponnalar and V. Dhanakoti, “An intrusion detection approach using ensemble support vector machine based chaos game optimization algorithm in big data platform,” *Applied Soft Computing*, vol. 116, Feb. 2022, doi: 10.1016/j.asoc.2021.108295.
- [14] M. Amru *et al.*, “Network intrusion detection system by applying ensemble model for smart home,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3485–3494, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3485-3494.
- [15] M. C. Thrun, J. Märte, and Q. Stier, “Analyzing quality measurements for dimensionality reduction,” *Machine Learning and Knowledge Extraction*, vol. 5, no. 3, pp. 1076–1118, Aug. 2023, doi: 10.3390/make5030056.
- [16] P. S. Bhattacharjee, A. K. Md Fujail, and S. A. Begum, “A comparison of intrusion detection by k-means and fuzzy c-means clustering algorithm over the NSL-KDD dataset,” in *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec. 2017, pp. 1–6, doi: 10.1109/ICCIC.2017.8524401.
- [17] R. Panigrahi *et al.*, “A consolidated decision tree-based intrusion detection system for binary and multiclass imbalanced datasets,” *Mathematics*, vol. 9, no. 7, pp. 1–35, Mar. 2021, doi: 10.3390/math9070751.
- [18] M. Al-Zewairi, S. Almajali, and A. Awajan, “Experimental evaluation of a multi-layer feed-forward artificial neural network classifier for network intrusion detection system,” in *2017 International Conference on New Trends in Computing Sciences (ICTCS)*, Oct. 2017, pp. 167–172, doi: 10.1109/ICTCS.2017.29.
- [19] Akashdeep, I. Manzoor, and N. Kumar, “A feature reduced intrusion detection system using ANN classifier,” *Expert Systems with Applications*, vol. 88, pp. 249–257, Dec. 2017, doi: 10.1016/j.eswa.2017.07.005.
- [20] K. Wang, A. Zhang, H. Sun, and B. Wang, “Analysis of recent deep-learning-based intrusion detection methods for in-vehicle network,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1843–1854, 2022, doi: 10.1109/TITS.2022.3222486.
- [21] A. Kanade *et al.*, “Analysis of wireless network security in internet of things and its applications,” *Indian Journal of Engineering*, vol. 21, no. 55, pp. 1–12, Apr. 2024, doi: 10.54905/diss.v21i55.e1ijel675.
- [22] F. ur Rehman and C. Izurieta, “Statistical metamorphic testing of neural network based intrusion detection systems,” in *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, Jul. 2021, pp. 20–26, doi: 10.1109/CSR51186.2021.9527993.
- [23] S. Thian Meng, S. Yogarayan, S. Fatimah Abdul Razak, S. Kannan, and A. Azman, “Advances of vehicular ad hoc network using machine learning approach,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 32, no. 3, pp. 1426–1433, Dec. 2023, doi: 10.11591/ijeecs.v32.i3.pp1426-1433.
- [24] D. Kwon, K. Natarajan, S. C. Suh, H. Kim, and J. Kim, “An empirical study on network anomaly detection using convolutional neural networks,” in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, Jul. 2018, pp. 1595–1598, doi: 10.1109/ICDCS.2018.00178.
- [25] M. Tavallae, E. Bagheri, W. Lu, and A. A. Ghorbani, “A detailed analysis of the KDD CUP 99 data set,” in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Jul. 2009, pp. 1–6, doi: 10.1109/CISDA.2009.5356528.
- [26] S. Choudhary and N. Kesswani, “Analysis of KDD-Cup’99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT,” *Procedia Computer Science*, vol. 167, pp. 1561–1573, 2020, doi: 10.1016/j.procs.2020.03.367.
- [27] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002.
- [28] M. Heigl, E. Weigelt, D. Fiala, and M. Schramm, “Unsupervised feature selection for outlier detection on streaming data to enhance network security,” *Applied Sciences*, vol. 11, no. 24, pp. 1–30, Dec. 2021, doi: 10.3390/app112412073.
- [29] N. Moustafa and J. Slay, “UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),” in *2015 Military Communications and Information Systems Conference (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.
- [30] E. Chatzoglou, G. Kambourakis, and C. Koliass, “Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset,” *IEEE Access*, vol. 9, pp. 34188–34205, 2021, doi: 10.1109/ACCESS.2021.3061609.
- [31] F. Nussipova, S. Rysbekov, Z. Abdiakhmetova, and A. Kartbayev, “Optimizing loss functions for improved energy demand prediction in smart power grids,” *International Journal of Electrical and Computer Engineering*, vol. 14, no. 3, pp. 3415–3426, Jun. 2024, doi: 10.11591/ijece.v14i3.pp3415-3426.
- [32] N. Assymkhan and A. Kartbayev, “Advanced IoT-enabled indoor thermal comfort prediction using SVM and random forest models,” *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 8, pp. 1040–1050, 2024, doi: 10.14569/IJACSA.2024.01508102.
- [33] D. Preuveneers and W. Joosen, “Sharing machine learning models as indicators of compromise for cyber threat intelligence,” *Journal of Cybersecurity and Privacy*, vol. 1, no. 1, pp. 140–163, Feb. 2021, doi: 10.3390/jcp1010008.

BIOGRAPHIES OF AUTHORS






Shynggys Rysbekov     is a motivated student pursuing a bachelor's degree in information systems at the International University of Information Technologies and continuing his education with a master's in data science at the Kazakh-British Technical University. He shows initiative in his studies and has independently learned programming languages, including HTML, CSS, JavaScript, and Python. Shynggys is active in the educational process, participates in online courses, and is committed to self-improvement. His passion for information technology and continuous quest for knowledge highlight his potential for future professional growth in the field of IT. He can be contacted at email: obershyngys@gmail.com.






Abylay Aitbanov    is a dedicated software engineer with a bachelor's degree in computer science from Kazakh-British Technical University and currently pursuing a master's in software engineering at the same institution. With over a year of experience in software development, Abylai has a strong background in designing, implementing, and delivering high-quality web and backend applications. Abylai's professional journey includes roles as a Python Developer at Halyk Bank and KMF, where he honed his skills in backend development, data management, and software optimization. Currently, he works as a software engineer at Arbuz.kz, where he continues to grow his technical expertise. He can be contacted at email: abylai.aitbanov@gmail.com.



Zukhra Abdiakhmetova    is an associate professor and vice dean at the Faculty of Information Technology, Al-Farabi Kazakh National University, specializing in computer science. She has a remarkable academic background, having earned all her degrees in computer science from Al-Farabi Kazakh National University. Her research spans various aspects of machine learning and wavelet transform, notably in medical applications like ECG analysis for heart fibrillation prediction. She has also worked on projects such as homomorphic encryption on microcontrollers and machine learning for heart disease study. She can be contacted at email: zuhra.abdyahmetova@kaznu.edu.kz.



Amandyk Kartbayev    is an associate professor at Kazakh-British Technical University, Almaty, Kazakhstan. He obtained his bachelor's degree in math from Satbayev University, Master of Science, and PhD in information systems from Al-Farabi Kazakh National University. He has also worked as a senior IT engineer at KMG Engineering and had a research internship at Fondazione Bruno Kessler in Italy. Kartbayev's research contributions include papers on machine translation and data models, reflecting his expertise in computational linguistics and data engineering. He can be contacted at email: a.kartbayev@gmail.com.