Proceedings of the 2nd International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/106/20241252

Comparative Analysis of Machine Learning Methods in the Detection of Network Intrusion

Kaixin Chen

Fudan University, 220 Handan Rd., Shanghai, China

22307130353@m.fudan.edu.cn

Abstract. In the context of increasing network security threats, traditional Intrusion Detection Systems (IDS) face challenges in detecting complex and evolving attacks. This paper presents a comparative study of three machine learning algorithms—Logistic Regression, Naive Bayes, and Multilayer Perceptron (MLP)—for network intrusion detection. Using a comprehensive dataset, the performance of these models is evaluated based on metrics such as accuracy, precision, recall, and F1 score. Results show that MLP with two hidden layers significantly outperforms other models, achieving high accuracy and robustness in detecting both normal and anomalous network traffic. The study highlights the limitations of traditional models in handling nonlinear and complex features, while also emphasizing the potential of advanced machine learning techniques to improve detection performance. Future research directions include optimizing model complexity, reducing false positives, and integrating deep learning architectures for enhanced real-time intrusion detection.

Keywords: Machine Learning, Network Intrusion, logistic regression, Naive Bayes, Multilayer Perceptron.

1. Introduction

In the digital society, the importance of network security cannot be overlooked. As the internet continues to penetrate every aspect of our daily lives, network intrusion has become an increasingly serious threat. Especially in the era of big data, the explosive growth of internet-connected devices, including IoT devices, online payment systems, and cloud computing platforms, has made the network environment more complex, thus increasing the difficulty of intrusion detection. IoT devices, which are widely used in smart homes, medical devices, and industrial control systems, are often vulnerable due to weak security protections. At the same time, cybercriminals use efficient and stealthy techniques to steal personal information, corporate data, and government secrets on a large scale, causing immeasurable economic losses and social impact.

In the field of network security, intrusion detection systems (IDS) play a critical role [1]. They are used to monitor network traffic, identify potential threats, and prevent attacks. However, traditional IDS methods largely rely on rule-based and signature-based approaches, which are designed to detect threats using predefined rules and known attack patterns. While these feature-matching techniques perform well in identifying known attacks, they struggle to handle increasingly complex and diverse attack methods. First, attackers continuously develop new techniques, such as zero-day attacks and Advanced Persistent Threats, which do not have known signatures or patterns. This makes it difficult for traditional IDS to

DOI: 10.54254/2755-2721/106/20241252

update rule sets in time to detect these new threats. Additionally, traditional IDS often lack the ability to detect abnormal behaviors, making it hard to identify network activities that appear normal but are actually malicious. In large-scale network environments, manual rule-based detection is insufficient for processing vast amounts of data in real time, reducing the accuracy and response time of intrusion detection. Traditional IDS also suffer from high false alarm rates. Due to the rigid and static nature of detection rules, these systems often mistakenly flag normal network activities as anomalies, leading to a large number of false positives. Maintaining and updating rule sets also require significant time and human resources, making it difficult to adapt to rapidly evolving attack techniques. As a result, traditional intrusion detection methods struggle to perform effectively against modern, evolving attack patterns, exposing significant gaps in network security defenses.

With the rapid development of artificial intelligence, machine learning has emerged as a powerful tool to address the challenges of network intrusion detection. Unlike traditional rule-based methods, machine learning models are data-driven. They can be trained on large historical datasets to autonomously learn normal patterns of network traffic and identify potential abnormal behaviors. In the context of big data, machine learning is capable of processing and analyzing vast amounts of network data efficiently, uncovering attack patterns hidden in noise, and thus effectively identifying unknown threats. Compared to traditional methods, machine learning-based network intrusion detection offers significant advantages. Machine learning-driven IDS systems can achieve a balance between real-time processing and accuracy, making them especially suitable for today's complex and rapidly changing network environments.

This paper aims to evaluate and analyze the application of several common machine learning methods in network intrusion detection. Through experimental comparisons of different algorithms' performance, we will explore their applicability and limitations in real-world network environments. The research presented in this paper not only provides valuable insights for academic studies in the field of network security but also offers practical, intelligent solutions for deploying intrusion detection systems.

2. Previous works

With the rapid development of information technology, network security threats are increasing, and IDS play a critical role in network defense. Traditional intrusion detection methods mainly include signature-based detection, state-based detection, and content-based detection. Each method has its strengths, but they also reveal limitations when faced with complex network environments and large-scale data processing.

Signature-based detection is the most commonly used method for intrusion detection [2]. It identifies intrusion behaviors by matching known attack signatures. This approach is highly accurate and has a low false-positive rate, performing well against known threats. However, it relies on constant updates to the signature database, making it less effective against new types of attacks, such as zero-day attacks. Additionally, maintaining the signature database requires substantial resources. As attack techniques evolve, signature-based detection gradually fails to meet the growing demands.

State-based detection identifies attacks by monitoring changes in network connection states [3]. It is particularly effective for detecting complex attack behaviors, such as session hijacking. Compared to signature-based detection, it has advantages in recognizing more sophisticated attacks. However, the downside is its computational complexity and high resource consumption. In high-traffic networks, this method can lead to decreased system performance and demands significant hardware resources.

Content-based detection analyzes packet content to identify potential threats [3]. It is effective in detecting hidden attacks like SQL injection. However, in-depth packet analysis consumes a large amount of computing resources, and its effectiveness significantly decreases when dealing with encrypted traffic. Thus, in high-traffic or encrypted network environments, the limitations of this method become evident.

Overall, while these traditional methods are effective in certain scenarios, their limitations become more apparent as attack techniques diversify and network environments become more complex. As a

DOI: 10.54254/2755-2721/106/20241252

result, machine learning methods have emerged as a critical direction for intrusion detection, offering more flexible and intelligent solutions.

3. Dataset and Preprocessing

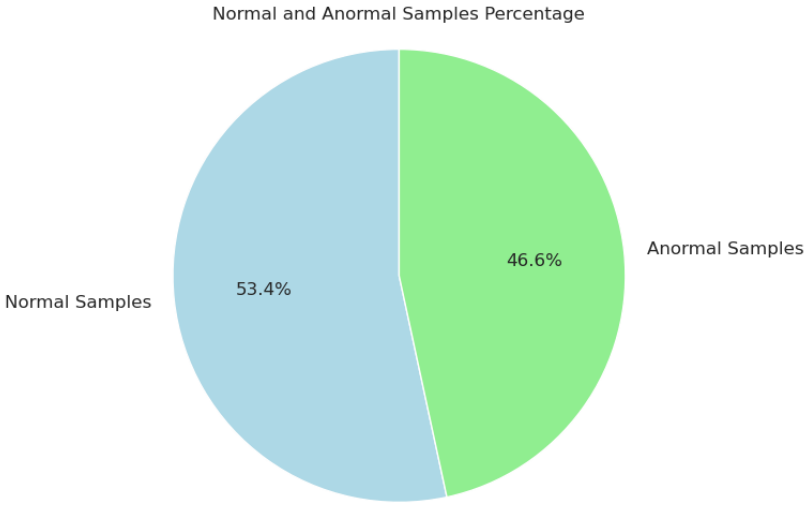

Figure 1. The proportion of samples.

The dataset used in this study is a comprehensive intrusion detection dataset generated in a simulated military network environment (Network Intrusion Detection Dataset in Kaggle). This simulated environment closely resembles real operational scenarios and includes various types of attacks to produce diverse intrusion detection data. There are 25192 records, including 53.4% normal records and 46.6% anormal records. Each connection is defined as a series of TCP packets transmitted from a source IP address to a target IP address over a specific protocol during a certain time period. Each connection is clearly labeled as either "normal" or "anomalous," with anomalous connections further categorized by specific attack types. Each connection record consists of approximately 100 bytes of data. For each TCP/IP connection, the dataset provides 41 features, including 3 qualitative and 38 quantitative features, used to describe both normal and attack behaviors. The target variable is divided into two classes: normal and anomalous.

In the data processing steps, we first performed comprehensive data cleaning, removing duplicate packets that could cause model overfitting. Missing values were handled using interpolation or mean imputation to ensure data integrity. Next, we extracted key features from each packet, such as source IP address, destination IP address, port number, and protocol type. For feature processing, we standardized the numerical features by applying min-max scaling, which normalized the data to the [0,1] range. Categorical features were converted into numerical form using one-hot encoding. To improve computational efficiency and reduce model complexity, we applied feature selection techniques. Through correlation analysis and dimensionality reduction methods, such as PCA, we retained the features most relevant to network attack behavior. Finally, the dataset was split into training and testing sets, with 70% used for training and 30% for testing. The processed data was then serialized and saved in a format suitable for machine learning model input.

4. Comparative Methods

In the context of Network Intrusion Detection Systems, selecting an appropriate machine learning algorithm is crucial as it directly impacts the system's accuracy, detection efficiency, and false alarm rate. This study provides a detailed comparison of three classic machine learning algorithms—Logistic

DOI: 10.54254/2755-2721/106/20241252

Regression, Naive Bayes, and Multilayer Perceptron (MLP)—aiming to offer algorithm selection and optimization recommendations tailored to specific network security threat scenarios.

Naive Bayes is a simple probabilistic classifier based on Bayes' theorem, which assumes that features are conditionally independent given the class [4]. Although this assumption simplifies the computation, it is also a major theoretical limitation of Naive Bayes, as features in real-world data are often interdependent. The algorithm performs well on low-dimensional datasets but may suffer in scenarios where high-dimensional features are correlated. Despite this, Naive Bayes is widely used in real-time intrusion detection due to its simplicity and computational efficiency. Theoretically, Naive Bayes is suitable for large-scale data processing, especially when features are relatively independent. However, its sensitivity to class imbalance may pose challenges in practical applications, particularly when attack samples are scarce in the dataset.

MLP is a complex feedforward neural network capable of learning intricate data patterns through multiple layers of nonlinear transformations [5]. The structure of an MLP consists of an input layer, one or more hidden layers, and an output layer, with each layer comprising multiple neurons. Each neuron transforms its input signal using an activation function such as ReLU or Sigmoid. The theoretical foundation of MLP allows it to increase model complexity by adding depth and width to the hidden layers, enhancing its ability to learn nonlinear relationships in the data. However, MLP's performance is highly dependent on the configuration of the network architecture and hyperparameters, such as learning rate and batch size, which need to be optimized through methods like cross-validation. Furthermore, MLP may require substantial computational resources during training, especially on large datasets, leading to longer training times and increased resource consumption.

Logistic Regression is a statistical classification method that seeks to find a linear decision boundary in the feature space to distinguish between different classes. It predicts a probability by applying the Sigmoid function to map the output of linear regression to the [0,1] range. Theoretically, Logistic Regression performs best when there is a linear relationship between the features and the target outcome. Its main advantages are simplicity and high interpretability, making it particularly suitable for scenarios where quick implementation is necessary. Parameter estimation in Logistic Regression typically uses the maximum likelihood estimation method. Although this method can be computationally intensive, it provides stable classification results. However, the limitations of Logistic Regression in handling nonlinear data are evident, which can be a drawback in complex network security environments.

5. Results

We conducted experiments comparing the performance of four models—Logistic Regression, Naive Bayes, Multilayer Perceptron 1 (MLP1, with one hidden layer containing 10 neurons), and Multilayer Perceptron 2 (MLP2, with two hidden layers, each containing 30 neurons)—across several key performance metrics, including accuracy, F1 score, precision, and recall. The results are summarized in Table 1 and Figure 2.

Table 1. Metrics for different methods.

Model	Accuracy	F1-Score	Precision	Recall
LGR	0.95	0.96	0.95	0.97
Naïve Bayes	0.90	0.92	0.88	0.95
MLP1	0.96	0.96	0.94	0.99
MLP2	0.98	0.99	0.99	0.99

Logistic regression demonstrated balanced performance across all evaluation metrics, with an accuracy of 0.95, an F1 score of 0.96, and precision and recall of 0.95 and 0.97, respectively. This indicates that the logistic regression model can achieve high precision while maintaining a high recall rate, effectively distinguishing between normal and abnormal traffic. However, as a linear model, its

DOI: 10.54254/2755-2721/106/20241252

ability to handle complex nonlinear features is limited, which may restrict its performance in certain data patterns.

The Naive Bayes model achieved an accuracy of 0.90 in this test, slightly lower than that of Logistic Regression, but with an F1 score of 0.92, indicating strong classification capabilities. Notably, the recall rate was as high as 0.95, suggesting that it effectively captured most network intrusion events. However, due to the assumption of feature independence, Naive Bayes may not perform well when dealing with complex or highly correlated features, as reflected in its precision of 0.88 and relatively high false positive rate.

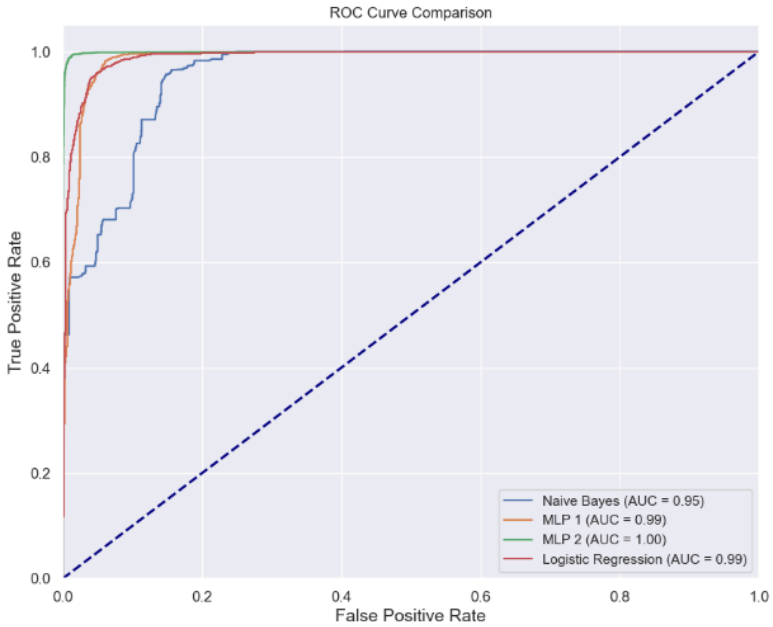

Figure 2. ROC for different methods.

MLP1 excelled in recall, achieving an almost perfect score of 0.99, meaning it detected nearly all attack behaviors. With an accuracy of 0.96, MLP1 outperformed Logistic Regression, though its precision of 0.94 could point to a higher false positive rate in some cases. Nonetheless, MLP1, with its nonlinear modeling capabilities, effectively handled the complexity of network data. MLP2 outperformed all other models, with an accuracy of 0.98 and F1 score, precision, and recall all reaching 0.99. This exceptional performance indicates that MLP2 is highly robust and excels at classifying large and complex datasets, accurately identifying both normal and abnormal traffic.

In summary, MLP2 significantly outperformed the other models in this experiment, thanks to its complex multilayer structure and nonlinear processing capabilities. While Naive Bayes performed well in recall, it suffered from a higher false positive rate, and Logistic Regression exhibited balanced performance in most scenarios. The MLP models, particularly MLP2, showed greater potential due to their ability to capture nonlinear patterns in complex data, achieving near-perfect balance between precision and recall. Additionally, hyperparameter tuning and model complexity were critical factors affecting performance, especially for the MLP models.

6. Conclusion

In this study, we conducted a detailed comparative analysis of Naive Bayes, Logistic Regression, and MLP models to assess their effectiveness in detecting network attacks. Through a comprehensive evaluation of these models, we identified several key findings.

First, there were significant performance differences among the models. The MLP2 model performed best across all metrics, particularly in handling complex data features and large datasets. It achieved

DOI: 10.54254/2755-2721/106/20241252

exceptionally high accuracy, F1 score, precision, and recall, nearly reaching perfect classification. This superior performance is due to its complex network structure, which effectively captures nonlinear relationships in the data, making it particularly suited for analyzing multidimensional features and detecting abnormal behavior patterns in network intrusion detection. In contrast, while Logistic Regression is a linear model, it still provides high classification performance in simpler or linearly separable tasks, making it ideal for scenarios where resources are limited and rapid deployment is required. Naive Bayes, though efficient in computation, is limited by its independence assumption, which affects its performance when dealing with complex or interdependent features. While it achieved a high recall rate, its higher false positive rate reduced its overall effectiveness.

Secondly, we observed a positive correlation between model complexity and performance. MLP2, the most complex model in this study, consistently outperformed others in all testing scenarios, demonstrating its superior ability to handle complex and nonlinear data. Simpler models like Naive Bayes and Logistic Regression struggled with this type of data. Lastly, hyperparameter tuning played a critical role in improving model performance. For the MLP2 model, adjusting the learning rate and the number of hidden layers significantly enhanced its adaptability to data features and its generalization ability, while also effectively reducing overfitting.

Although this study provides valuable insights into the application of different machine learning models for network attack detection, we also identified several key limitations that need to be addressed in future research. First, while the MLP2 model showed outstanding performance, its high computational complexity led to longer training times and higher computational costs, which may limit its practicality in resource-constrained environments. Thus, reducing the computational complexity and training time without sacrificing performance is an important area for future research. Second, the dataset used in this study was limited in size. Although the models performed well on this dataset, the results may not generalize to larger real-world network environments. The use of static datasets may not fully capture the dynamic nature of network traffic, so future research should include larger and more dynamic datasets to better validate the models' practicality and robustness.

Additionally, while MLP models exhibited strong detection capabilities, controlling the false positive rate remains a challenge. The relatively poor precision of the Naive Bayes and Logistic Regression models could reduce system efficiency in real-world applications. Future research should focus on optimizing model structures or introducing new false positive control strategies to mitigate the impact of false positives on system performance. Finally, this study primarily trained and tested models using batch processing, whereas real-world network intrusion detection systems require real-time detection capabilities. Despite the superior performance of deep learning models like MLP, their real-time efficiency has not yet been fully tested. Future research should focus on improving the real-time performance of these models, optimizing their response speed and detection efficiency to meet the real-time demands of network environments.

Future network intrusion detection research will focus on leveraging advanced machine learning techniques to address increasingly complex attack patterns and data imbalance issues. Specifically, Generative Adversarial Networks (GANs) [6], Transformer models [7], as well as BERT [8] and GPT technologies [9], will play a central role. GANs have already shown significant promise in anomaly detection and generating malicious traffic, with their ability to enhance the adaptability of IDS through the generation of realistic adversarial samples. Notably, Wasserstein GAN (WGAN) [10], by introducing the Earth Mover’s Distance, improves the model's stability in handling complex data distributions and enhances its ability to detect zero-day and unknown attacks. Conditional GAN (cGAN) [11] effectively addresses the data imbalance problem, proving advantageous in generating small-sample attack data. Future research will further optimize this architecture to enhance its performance. Transformer models, due to their self-attention mechanism and efficiency in processing sequential data, have shown great potential in network intrusion detection, significantly reducing training time and improving model convergence. Future studies will explore optimizing Transformer’s multi-head attention layers and feedforward neural network layers to enhance detection of complex network attack behaviors. Additionally, time-aware Transformers or hierarchical Transformers could be developed to

DOI: 10.54254/2755-2721/106/20241252

address specific cybersecurity challenges. Moreover, the pre-training and fine-tuning mechanisms of BERT and GPT will be introduced into the field of network intrusion detection. These models capture contextual information from input sequences through bidirectional encoders, improving the detection of complex attack patterns. GPT's generative and reasoning capabilities can not only be used to generate anomalous traffic samples but also to automatically generate intrusion detection rules or policies, further enhancing detection of attacks requiring deep reasoning or long-distance dependencies.

By integrating these advanced deep learning techniques, future network intrusion detection systems will become more intelligent and adaptive, effectively countering increasingly complex and diverse attack patterns, thus providing stronger protection for cybersecurity.

References

[1] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecur, vol. 2, no. 1, p. 20, Jul. 2019, doi: 10.1186/s42400-019-0038-7.

[2] B. Nawaal, U. Haider, I. U. Khan, and M. Fayaz, "Signature-Based Intrusion Detection System for IoT," in Cyber Security for Next-Generation Computing Technologies, CRC Press, 2024.

[3] N. T. N and D. Pramod, "Insider Intrusion Detection Techniques: A State-of-the-Art Review," Journal of Computer Information Systems, Jan. 2024, Accessed: Sep. 23, 2024. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/08874417.2023.2175337

[4] B. Ravinder, S. K. Seeni, V. S. Prabhu, P. Asha, S. P. Maniraj, and C. Srinivasan, "Web Data Mining with Organized Contents Using Naive Bayes Algorithm," in 2024 2nd International Conference on Computer, Communication and Control (IC4), Feb. 2024, pp. 1–6. doi: 10.1109/IC457434.2024.10486403.

[5] D. Chauhan, A. Yadav, and F. Neri, "A multi-agent optimization algorithm and its application to training multilayer perceptron models," Evolving Systems, vol. 15, no. 3, pp. 849–879, Jun. 2024, doi: 10.1007/s12530-023-09518-9.

[6] W. Lim, K. S. C. Yong, B. T. Lau, and C. C. L. Tan, "Future of generative adversarial networks (GAN) for anomaly detection in network security: A review," Computers & Security, vol. 139, p. 103733, Apr. 2024, doi: 10.1016/j.cose.2024.103733.

[7] H. Kheddar, "Transformers and Large Language Models for Efficient Intrusion Detection Systems: A Comprehensive Survey," Aug. 14, 2024, arXiv: arXiv:2408.07583. doi: 10.48550/arXiv.2408.07583.

[8] A. Alferaidi et al., "A Novel Hybrid, BERT and Deep Learning Model Network Intrusion Detection System for Healthcare Electronics," IEEE Transactions on Consumer Electronics, pp. 1–1, 2024, doi: 10.1109/TCE.2024.3412199.

[9] "GPT and Interpolation-Based Data Augmentation for Multiclass Intrusion Detection in IIoT | IEEE Journals & Magazine | IEEE Xplore." Accessed: Sep. 23, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10418592

[10] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN, " Dec. 06, 2017, arXiv: arXiv:1701.07875. Accessed: May 28, 2022. [Online]. Available: http://arxiv.org/abs/1701.07875

[11] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets, " Nov. 06, 2014, arXiv: arXiv:1411.1784. Accessed: May 28, 2022. [Online]. Available: http://arxiv.org/abs/1411.1784