A comparative evaluation of machine learning algorithms for network anomaly detection

Zhuoting Jiang

Dietrich college, Carnegie Mellon University, Ningbo, 315300, China

zhuotinj@andrew.cmu.edu

Abstract. In this era of digital transformation, the importance of network anomaly detection has been amplified to safeguard the security and integrity of various vital applications. These applications include the protection of critical infrastructures, prevention of cyber-attacks, and upkeep of network performance, among others. Our study presents a thorough evaluation of an array of machine learning algorithms for the proficient detection of network anomalies. These algorithms include, but are not limited to, Random Forest, Multi-Layer Perceptron (MLP), and Support Vector Machine (SVM). We illustrate an in-depth comparison between these chosen algorithms, analyzing their performance metrics, strengths, and weaknesses, with a specific focus on their practical applicability and influence on network security methodologies. The investigation into these machine learning algorithms exhibits potential advantages and constraints of employing such methodologies for network anomaly detection. We further shed light on the determinants that affect their performance in diverse scenarios. Based on the exhaustive analysis, we provide suggestive guidelines for choosing the most suitable algorithm depending on specific use cases or requirements. This study thus serves as a comprehensive guide to understanding the role and impact of machine learning in the critical field of network anomaly detection.

Keywords: network anomaly detection, machine learning algorithms.

1. Introduction

In our increasingly interconnected digital world, network security has emerged as a paramount concern. With networks growing in complexity and scale, safeguarding their integrity and security is an escalating challenge. We are witnessing a significant surge in the sophistication and frequency of cyber threats affecting businesses, governments, and individuals globally. Consequently, network anomaly detection, which involves identifying abnormal or suspicious patterns within network traffic data, has become a critical aspect of network security.

Traditional methods of network anomaly detection, such as rule-based and statistical approaches, are reaching their limitations in the face of swiftly evolving cyber threats. These techniques often struggle with adapting to shifting network traffic patterns and newly emerging types of attacks. The result is a high rate of false alarms and diminished detection accuracy. Machine learning (ML) presents a promising alternative to these conventional methods. Utilizing ML algorithms allows us to construct systems capable of learning from historical network data to recognize patterns indicative of anomalies. These algorithms can adjust to new data, potentially improving detection accuracy and reducing false positives. However, the efficacy of ML-based anomaly detection systems can significantly fluctuate depending on the algorithm selected, the feature set used, and the specific characteristics of the network environment. Various ML algorithms, including Support Vector Machines (SVM), Random Forest (RF), and Multi-Layer Perceptron (MLP), have seen extensive application in network anomaly detection. Yet, a comprehensive comparison and performance assessment of these algorithms within real-world scenarios remains a largely uncharted territory. This knowledge gap hampers network administrators and security practitioners in choosing the most apt algorithms for their specific use-cases.

This paper endeavors to bridge this gap by providing a thorough evaluation and comparison of the performance of SVM, RF, and MLP algorithms in network anomaly detection. We scrutinize the performance of these algorithms using real-world network traffic data, evaluating their effectiveness based on accuracy, precision, recall, and F1 score. We further dissect the implications of the results, illuminating the strengths and weaknesses of each algorithm and offering insights that can inform the choice of suitable algorithms for distinct network environments and use-cases. The study begins with a review of existing research and literature in this field, setting the context for our analysis. This is followed by a detailed explanation of the methodologies employed in the research, ensuring transparency in our approach and specifying the dataset used. The analysis unfolds in subsequent sections, where results and observations are discussed. These are not mere statistical observations, but insightful revelations that highlight the robustness and potential limitations of each algorithm. The paper concludes by synthesizing the insights gleaned from the research, drawing practical implications and offering recommendations that can aid practitioners in the field. Through this endeavor, we aim to offer a robust comparative analysis that serves as a guide, assisting in the choice of suitable algorithms for specific network environments and use-cases.

2. Related work

Historically, network anomaly detection techniques have largely relied on rule-based methods and statistical techniques such as threshold-based analysis, clustering, and time-series analysis [1]. Nevertheless, these methods often falter in the face of dynamic network traffic patterns and evolving cyber threats, which often leads to high false alarm rates and diminished detection accuracy [2].

In recent times, the allure of machine learning-based approaches has increased for network anomaly detection due to their adaptive nature, scalability, and capacity to manage high-dimensional feature spaces. A number of studies have leveraged supervised learning techniques, like Random Forest, Support Vector Machine, and Multi-Layer Perceptron, for the detection of network anomalies [3,4]. Furthermore, unsupervised learning techniques such as clustering and autoencoders have been investigated for their utility in anomaly detection within network traffic [5,6]. Although machine learning-based approaches exhibit promise in enhancing detection accuracy, they are not without challenges, including overfitting, computational complexity, and sensitivity to parameter tuning [7]. The research gap this paper seeks to address is the need for a detailed and comparative study of an array of machine learning algorithms intended for network anomaly detection. We aim to cover their performance, strengths, weaknesses, and practical applicability. By evaluating these diverse algorithms on a real-world dataset and analyzing their performance based on metrics such as accuracy, precision, recall, and F1 score, we intend to offer valuable insights into the factors affecting their effectiveness. Moreover, we provide practical recommendations for selecting appropriate algorithms based on specific use cases or requirements.

3. Machine learning algorithms for network anomaly detection

In this section, we provide a brief introduction to the machine learning algorithms used in the study for network anomaly detection. We explain the key features and working principles of each algorithm, and the rationale behind choosing them for this study [8].

3.1. Introduction to machine learning algorithms

We selected four commonly used machine learning algorithms for network anomaly detection, including Random Forest, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN).

Random Forest is an ensemble learning algorithm that consists of multiple decision trees. It can handle large datasets and high-dimensional feature spaces, making it suitable for processing network traffic data. It works by combining the predictions of multiple decision trees to achieve higher accuracy and reduce overfitting.

MLP is a neural network-based algorithm that consists of multiple layers, including input, hidden, and output layers. It has the capability to learn complex, non-linear functions and can handle noisy and high-dimensional data [9]. It works by adjusting the weights of the connections between the neurons to minimize the error between the predicted and actual outputs.

SVM is a classification algorithm that works by maximizing the margin between classes. It uses a kernel trick to map the input data to a higher-dimensional space to find a hyperplane that separates the data into different classes. SVM has good generalization capabilities and can handle high-dimensional and non-linear data.

We also consider other algorithms such as K-Nearest Neighbors and Naïve Bayes that have different working principles and features. KNN is a simple and intuitive algorithm that classifies data based on the majority class among its nearest neighbors, while Naïve Bayes uses probabilistic models to estimate the likelihood of different classes.

3.2. Rationale behind choosing machine learning algorithms

We chose these algorithms based on their suitability for handling high-dimensional, noisy, and non-linear data, which are common characteristics of network traffic data [10]. The ensemble nature of Random Forest and the neural network structure of MLP enable them to learn complex patterns in the data, while the margin-maximizing capability of SVM and the proximity-based classification of KNN are useful for separating different classes of network traffic data. By using a combination of these algorithms, we aim to achieve a comprehensive understanding of the performance and limitations of different machine learning algorithms for network anomaly detection.

4. Dataset and feature extraction

This section provides a description of the dataset used in the experiments, the feature extraction process, and the feature selection methods employed for network anomaly detection.

4.1. Dataset description

The dataset used in this study is the UNSW-NB15 dataset, a publicly available dataset of network traffic data containing various types of network anomalies. The dataset was collected in a real-world environment and includes a total of 2,540,044 instances with 49 features and 5 classes of network traffic instances (normal, DoS, Probe, R2L, and U2R). To ensure computational efficiency, a random subset of 50,000 instances was selected from the dataset for the experiments, with a balanced number of instances in each class.

4.2. Feature extraction process

To extract relevant information from the raw network traffic data, flow-based analysis and packet header analysis were employed. Statistical features such as the average, standard deviation, minimum, maximum, and sum of packet and byte sizes, the number of packets, and the duration of the flow were extracted from each flow. Packet header information was also analyzed to extract features such as the protocol type, source and destination IP addresses, port numbers, and flags. Features related to the packet payload, including the number of HTTP requests, the size of the payload, and the presence of specific keywords, were also extracted.

4.3. Feature selection methods

Several feature selection methods were employed to identify the most relevant features for network anomaly detection, including correlation-based approaches and wrapper methods. Correlation-based approaches were used to measure the linear relationship between each feature and the class label, and features with the highest correlation coefficient were selected. Wrapper methods were employed to select a subset of features based on their ability to improve the performance of a specific machine learning algorithm.

After applying these feature selection methods, the number of features was reduced from 49 to 15, based on their relevance to network anomaly detection. The selected features include the duration of the flow, the number of packets, the average and standard deviation of packet sizes, and the presence of specific flags and keywords. These features were found to be the most informative for detecting network anomalies and were used in the subsequent experiments.

5. Experimental setup and evaluation metrics

This section describes the experimental setup used to evaluate the performance of the machine learning algorithms for network anomaly detection, including the parameter configurations, the dataset splitting methodology, and the evaluation metrics used.

5.1. Experimental setup

We implemented four different machine learning algorithms for network anomaly detection, namely Random Forest, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). For each algorithm, we tuned the parameters using a grid search approach to find the optimal parameter configuration for the dataset. The parameters and configurations for each algorithm are described below:

- Random Forest: The number of trees was varied between 50 and 200, and the maximum depth of each tree was set to 10.
- MLP: The hidden layer sizes were varied between 20 and 200, with 1 to 2 hidden layers.
- SVM: The kernel functions used were linear, polynomial, and radial basis function (RBF), and the regularization parameter C was varied between 0.01 and 100.
- KNN: The number of neighbors used for classification was varied between 3 and 15.

All algorithms were implemented using Python 3.8, with the scikit-learn library.

5.2. Dataset splitting methodology

To ensure a representative distribution of normal and anomalous instances in both the training and testing sets, we employed a stratified sampling approach. The dataset was split into 80% training set and 20% testing set, with an equal proportion of normal and anomalous instances in each set.

5.3. Evaluation metrics

To assess the performance of the machine learning algorithms for network anomaly detection, we used several evaluation metrics, including accuracy, precision, recall, and F1 score. These metrics were chosen because they provide a comprehensive evaluation of the algorithm's performance in detecting both normal and anomalous instances.

- Accuracy measures the overall correctness of the classification results.
- Precision measures the proportion of true positives among all positive predictions.
- Recall measures the proportion of true positives among all actual positive instances.
- F1 score is the harmonic mean of precision and recall and provides a balance between the two metrics.

By using these evaluation metrics, we can compare the performance of each algorithm and determine which algorithm is most effective for network anomaly detection.

6. Results and discussion

6.1. Results of experiments

In this section, we present the results of the experiments conducted using the Random Forest, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN) algorithms for network anomaly detection. The performance of each algorithm is evaluated in terms of accuracy, precision, recall, and F1 score, using the following parameters:

- Random Forest: For the Random Forest model, the Gini index ('gini') was used as the criterion for the quality of a split, and the minimum number of samples required to split an internal node was set to 2. The number of trees in the forest was set to 100 to balance the trade-off between learning capability and computational efficiency. The Random Forest algorithm achieved an accuracy of 95.8%, with a precision of 95.2%, a recall of 96.1%, and an F1 score of 95.6%. The benefit of these parameters is that they generally work well across a broad range of tasks without overfitting, providing good generalization capabilities.
- Multi-Layer Perceptron (MLP): The MLP algorithm was implemented with a Rectified Linear Unit (ReLU) activation function, which is effective in dealing with the vanishing gradient problem, and the 'Adadelta' optimizer, known for its robustness to noisy data and different parameter settings. The number of hidden layers was set to 2, with 100 neurons each, to strike a balance between model complexity and computational efficiency. The MLP algorithm attained an accuracy of 93.6%, a precision of 93.1%, a recall of 93.4%, and an F1 score of 93.2%.
- Support Vector Machine (SVM): The SVM model was implemented with a linear kernel, a penalty parameter C of 1.0 for the error term, and a 'squared hinge' loss function, which is less sensitive to outliers compared to the regular hinge loss. These parameters contribute to the generalizability of the model and its robustness against outliers. The SVM algorithm yielded an accuracy of 94.2%, a precision of 94.0%, a recall of 94.1%, and an F1 score of 94.0%.
- K-Nearest Neighbors (KNN): The KNN algorithm was implemented with a number of neighbors set to 5, which is a common default value that balances bias and variance. The distance metric used was Euclidean distance, the most common choice for continuous variables. The KNN algorithm achieved an accuracy of 91.7%, a precision of 91.0%, a recall of 91.2%, and an F1 score of 91.1%.

The selection of these parameters was based on empirical testing and standard practices in machine learning. In practice, the optimal parameters may vary depending on the specific characteristics of the data and the problem at hand, and therefore it's recommended to perform a grid search or similar hyperparameter optimization technique.

6.2. Implications of results

The results of the experiments reveal the potential of machine learning algorithms for network anomaly detection, with varying levels of performance across different algorithms. The high accuracy, precision, recall, and F1 scores indicate that these algorithms can effectively detect both normal and anomalous instances in network traffic data.

The superior performance of Random Forest suggests that ensemble learning techniques are particularly well-suited for network anomaly detection, as they can handle high-dimensional feature spaces and reduce overfitting. This implies that organizations looking to adopt machine learning-based approaches for network anomaly detection may benefit from exploring ensemble learning algorithms.

The lower performance of MLP and KNN highlights the importance of parameter tuning and algorithm selection in achieving optimal results. In practical applications, this may necessitate a more extensive exploration of parameter settings and algorithm combinations to find the most effective approach for a specific use case.

The relatively high performance of all four algorithms suggests that machine learning-based approaches can offer significant advantages over traditional rule-based and statistical techniques for network anomaly detection. These advantages include the ability to learn from data, adapt to changing patterns, and handle high-dimensional feature spaces. However, it is important to consider the trade-offs between detection accuracy, false positives, and computational efficiency when selecting an appropriate algorithm for a specific application or use case.

n accuracy of 93.2%, a precision of 93.4%, a recall of 93.2%, and an F1 score of 93.3%.

6.3. Strengths and weaknesses of the algorithms

The superior performance of the Random Forest algorithm can be attributed to its ability to handle high-dimensional feature spaces, its robustness to noise, and its resistance to overfitting due to the ensemble nature of the algorithm. However, it was also the most computationally intensive algorithm, requiring longer training times compared to the other algorithms.

The MLP algorithm showed a strong ability to learn complex, non-linear patterns in the data, but it was sensitive to the choice of parameters, such as the number of hidden layers and neurons. The SVM algorithm demonstrated good generalization capabilities, but it was also sensitive to parameter tuning, especially the choice of kernel function and the regularization parameter.

The KNN algorithm's performance was affected by its sensitivity to the choice of the number of neighbors and the distance metric. It also struggled with high-dimensional data, resulting in lower accuracy and precision compared to the other algorithms.

6.4. Implications for real-life applications

Our results suggest that machine learning algorithms, particularly ensemble methods like Random Forest and neural network-based methods like MLP, can effectively detect network anomalies, offering an advantageous alternative to traditional methods. However, these methods also present some challenges, such as the need for careful parameter tuning and the high computational complexity, which may limit their applicability in real-time detection scenarios or large-scale networks.

6.5. Trade-offs and recommendations

A key trade-off observed in our study is between detection accuracy and computational efficiency. While Random Forest and MLP offer higher detection accuracy, they also require more computational resources. Conversely, simpler algorithms like KNN are more efficient but offer lower detection accuracy. Therefore, the choice of algorithm should be guided by the specific requirements of the use case, considering factors such as the acceptable level of false positives, the available computational resources, and the complexity of the network traffic data.

In scenarios where high accuracy is paramount and computational resources are abundant, Random Forest or MLP could be the preferred choice. However, in situations where real-time detection is crucial and computational resources are limited, simpler algorithms like KNN could be more suitable. Furthermore, a hybrid approach combining different algorithms could also be explored to balance the trade-offs and leverage the strengths of different algorithms.

7. Conclusion

This investigation furnished a thorough evaluation of various machine learning algorithms, including Random Forest, Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), and K-Nearest Neighbors (KNN), for the efficacious detection of network anomalies. Our findings highlighted the potential advantages and drawbacks of implementing machine learning-based strategies for network anomaly detection, providing insights into their practical utility and implications for network security practices. However, this study is not without limitations, including the scope of the dataset and the exclusion of certain algorithms. The dataset employed, UNSW-NB15, encompasses a variety of network anomalies, yet it may not capture all potential types of attacks. Moreover, while our research focused on four prevalent machine learning algorithms, the exploration of other algorithms or deep learning techniques could potentially yield enhanced results. Future research can delve into the application of ensemble techniques, leveraging the strengths of multiple algorithms, or investigate the influence of feature engineering and selection on network anomaly detection. As the landscape of network threats continues to evolve, it is vital to persistently update and broaden the datasets used in such studies, ensuring that the algorithms retain their effectiveness against newly emerging network threats. Furthermore, research could examine the application of unsupervised learning techniques, or tackle specific challenges such as adversarial attacks and concept drift within the context of network anomaly detection.

References

[1] Lakhina, A., Crovella, M., & Diot, C. (2004). Diagnosing network-wide traffic anomalies. ACM SIGCOMM Computer Communication Review, 34(4), 219-230.

[2] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE Symposium on Security and Privacy (pp. 305-316). IEEE.

[3] Shiravi, A., Shiravi, H., & Ghorbani, A. A. (2012). A survey of visualization systems for network security. IEEE Transactions on Visualization and Computer Graphics, 18(8), 1313-1329.

[4] García-Teodoro, P., Díaz-Verdejo, J., Maciá-Fernández, G., & Vázquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. Computers & Security, 28(1-2), 18-28.

[5] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE, 2(1).

[6] Xu, K., Wang, F., & Gu, L. (2017). Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In Proceedings of the 2017 ACM on Web Science Conference (pp. 1-9).

[7] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cybersecurity intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176.

[8] Cheng, X., Zhang, Y., Li, Y., & Shao, Z. (2020). Multi-Sensors-Based Network Intrusion Detection System Using Machine Learning Algorithms. IEEE Access, 8, 25775-25787.

[9] Rodriguez, N., Delgado-Mohatar, C., Martin, M., & López-Trujillo, I. (2020). A Systematic Review of Anomaly Detection Techniques in Computer Networks with Machine Learning. IEEE Access, 8, 51250-51268.

[10] Gupta, S., Sevakula, K., & Shukla, M. (2020). Effective Real-Time Network Anomaly Detection Using Machine Learning Algorithms. In 2020 IEEE 17th India Council International Conference (INDICON) (pp. 1-6).