Machine Learning-Based Network Detection Research for SDNs

Jiayue Lai

School of Computer and Cyberspace Security, Fujian Normal University, Fuzhou city, China

Abstract. This research endeavors to fortify the security posture of Software-Defined Networks (SDN) through the strategic utilization of intelligent machine learning techniques, with a primary focus on mitigating detrimental Denial of Service (DoS) attacks. To accomplish this, this study constructed a rigorously designed simulated SDN environment, which served as the cornerstone for meticulously assembling a comprehensive dataset encompassing a diverse array of attack vectors, with particular emphasis on DoS. Employing a tactical blend of established and cutting-edge machine learning algorithms, including Random Forest, Logistic Regression, and Decision Tree, alongside the advanced XGBoost and LightGBM models, this study conducted an exhaustive investigation to pinpoint the most efficacious methods for swiftly and precisely identifying DoS threats. It is necessary to note that XGBoost and LightGBM demonstrate an astonishing level of multiple performance, which testifies their outstanding ability to enhance SDN security. Reasserting the idea of the critically important role of machine learning for securing SDNs against possible intrusions, these results point not only to the highly beneficial applications of machine learning for protecting SDNs against malicious intrusions but also its indispensable role in preserving network stability and optimizing performance. Moreover, it emphasizes the operational advantage of deploying multiple organic sets of machine learning algorithms, which can achieve even greater precision and efficiency than individual machine learning algorithms in practical uses, bringing it closer to developing a more robust and secure SDN environment.

1 Introduction

The goal of this investigation is to locate intruder traffic in SDNs with sophisticated machine learning algorithms. It addresses unique SDN security challenges, especially DoS attacks, which disrupt services and threaten network operations [1]. Integrating quantitative analysis with tailored machine learning methods, the research targets DoS attacks to fortify SDN resilience and establish a more secure ecosystem [2].

In the initial data collection stage, this research crafted a simulated SDN environment that closely resembled real-world traffic patterns and potential attack scenarios. By harnessing packet capturing tools, this research generated pcap files, which served as the cornerstone for its detailed analysis. Moving forward to the processing and modeling phase, it choses three

Corresponding author: 121152022048@student.fjnu.edu.cn

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

esteemed machine learning algorithms: random forest, logistic regression, and decision tree, to create attack detection models [3]. To ensure rigor, it implemented a strict training-testing division, with cross-validation techniques reinforcing the models' accuracy and stability [4]. This study highlights the groundbreaking potential of machine learning in reinforcing SDN security by enabling precise attack traffic detection.

2 Methodologies

2.1 Logistic regression

In the domain of machine learning for SDN, logistic regression is identified as a reasonable linear classification model that is especially suitable for binary classification problems [5]. Since it can differentiate between normal and attack traffic, SDN security can be enhanced with the tool [6]. The following is the step-by-step process of how the logistic regression model is constructed and fine-tuned:

2.1.1 Characterization

Removal of irrelevant or redundant features: The process involves removing features that are deemed unimportant for prediction through the utilization of methods such as correlation analysis and the chi-square test.

2.1.2 Regularization

L1 regularization: The study incorporates an L1 penalty term in the paradigm to drive the weights of a subset of features towards zero, effectively achieving feature selection.

L2 regularization: The L2 paradigm is utilized to reduce the magnitude (absolute value) of the weights, thereby helping to prevent overfitting in the study.

Elastic Net: It combines L1 and L2 regularization, leveraging the feature selection capability of L1 regularization along with the stability provided by L2 regularization.

2.1.3 Hyperparameter tuning

Grid Search: The optimal configuration is determined through exhaustive exploration of all feasible combinations of hyperparameters.

Random Search: To reduce computational requirements, a subset of hyperparameter combinations is randomly selected for evaluation.

2.1.4 Early stopping

Monitor validation set performance: Training is halted when no further improvement in performance is observed on the validation set, as a preventive measure against overfitting.

2.1.5 Data pre-processing

Normalization/Standardization: Feature values are scaled to comparable ranges to facilitate faster convergence of the model. Additionally, Bayes' theorem is leveraged to conduct a more efficient and intelligent search for optimal hyperparameters.

Dealing with missing values: Missing values are filled in to ensure that data integrity is maintained.

2.1.6 Integration methods

Bagging and Boosting: Logistic regression can be embedded as a base learner in Bagging (e.g., Random Forest) or Boosting to improve model performance.

2.2 Decision tree classifier

Decision Tree Classifier in Machine Learning with Key Goals of Reducing Complexity, Avoiding Overfitting, and Improving Efficiency [7]. The following is the process of decision tree construction is as follows:

2.2.1 Simplified establishment phase

Feature selection: The number or type of features is limited to decrease the complexity of the model.

Node Split Control: Limits tree expansion is constrained by parameters such as the maximum depth, minimum number of samples required for a split, and minimum information gain threshold, in order to prevent overfitting [8].

2.2.2 Simplified decision rules

Simple rules are utilized for splitting nodes in order to reduce the complexity of the model.

2.2.3 Pruning techniques

Pre-pruning and Post-pruning: Pre-pruning restricts the growth of a tree during the training process by imposing strict conditions, whereas post-pruning involves removing unnecessary nodes from a fully grown tree based on an evaluation using validation sets.

Costly complexity pruning: Balancing the model's complexity against its error rate is crucial in determining the optimal pruning points that minimize overfitting while maintaining good predictive performance [9].

2.2.4 Structural compression

Rule set optimization: Converting decision trees into rule sets, merging similar rules, eliminating redundancies, and optimizing the model's structure are proven strategies for enhancing its simplicity and interpretability.

Leaf node merging: To simplify the model, leaf nodes that yield similar prediction results are merged together.

Feature encoding: Compact encoding of category-based features is implemented to reduce both dimensionality and storage requirements of the model [9].

2.2.5 Reasoning optimization

Path pruning: To reduce computational load, invalid paths are excluded in advance during the reasoning process.

Parallel reasoning: Multi-core or distributed computing resources are utilized to accelerate the reasoning process.

Feature quantization: Quantization of feature values is applied to minimize storage requirements and expedite the inference process.

2.3 Random forests classifier

In the realm of machine learning applied to SDN, while SDN primarily concerns itself with network-level issues such as the abstraction of network architecture and the separation of control and forwarding, Random Forest Classifier, as a machine learning algorithm, does not directly correlate with SDN in terms of its direct application. The process of constructing and predicting with Random Forest involves the following steps:

2.3.1 Constructing decision trees

Random Forests create an ensemble of decision trees through random sampling of data and feature subset selection, thereby enhancing diversity and mitigating overfitting.

2.3.2 Integrated prediction

For the classification problem, the random forest decides the final classification result by integrating the prediction results of all decision trees using a voting mechanism. For regression problems, the average of the predictions of all decision trees is used as the final prediction [10]. For regression problems, the average of the predictions of all decision trees is used as the final prediction.

The purpose of this experiment is to evaluate the effectiveness of different machine learning classifiers in detecting DoS attacks in SDN. The experiment improves the data quality and enhances the recognition ability of the model through preprocessing steps such as data cleaning, label coding, solo hot coding and data normalization. The goal is to find out the most suitable DoS attack detection model for SDN environment and provide theoretical and practical guidance for network security.

3 Results

This experiment evaluates the performance of 15 different classification models on the KDD99 dataset. The performance of each model is evaluated by metrics such as accuracy, precision, recall, F1 score, confusion matrix and ROC curve. Here three models are selected for detailed experimental results as follows:

3.1 Logistic regression

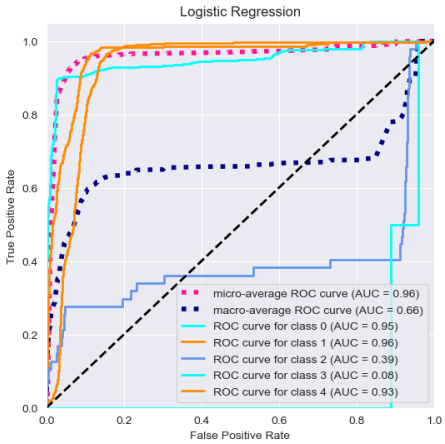

Fig. 1. Logistic Regression Performance Metrics. Below the figure.(Picture credit: Original)

In Fig.1, the classification efficacy of the model is validated by the micro-averaged ROC curve (AUC 0.96) and macro-averaged ROC curve (AUC 0.66), which shows its efficiency and accuracy in the multi-classification task. It excels in identifying normal traffic (category 0, AUC 0.95) and Dos attacks (category 1, AUC 0.96). However, the model significantly underperforms when it comes to detecting R2L attacks (category 2, AUC 0.39) and U2R attacks (category 3, AUC 0.08), particularly in the latter case where its efficacy is almost negligible, highlighting a limitation in handling intricate and sophisticated attack patterns. Nevertheless, the model maintains high accuracy in recognizing the Probe attack (category 4, AUC 0.93), showing its stability in dealing with common threats.

3.2 Decision tree classifier

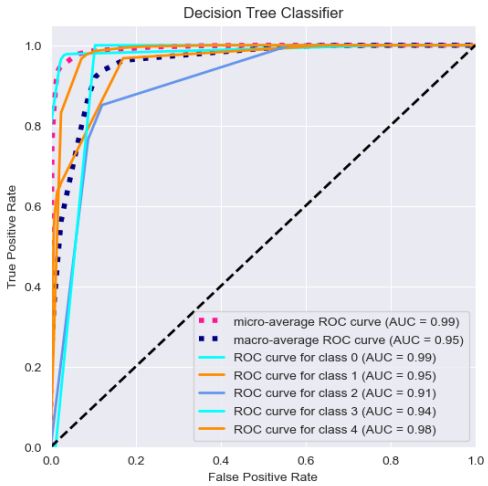

Fig. 2. Decision Tree ROC Curves. Below the figure(Picture credit: Original)

In Fig. 2, the Decision Tree Classifier demonstrates remarkable and outstanding performance in the realm of cyber attack identification, with a micro-average ROC AUC as high as 0.99, which is undoubtedly a strong proof of its high accuracy. At the same time, the macro average

ROC AUC reaches 0.95, which demonstrates that the model is able to maintain a balanced and efficient classification ability when dealing with various types of cyber-attacks, ensuring comprehensive protection. In terms of recognizing specific attack types, the decision tree classifier also shows amazing accuracy. For the identification of normal traffic, its AUC value is also as high as 0.99, which is almost perfect, which means that the model is able to accurately distinguish normal network activities and effectively filter out potential threats. When facing different types of attacks such as Dos, Probe and U2R, the model's AUC values are also stable at the high levels of 0.95, 0.94 and 0.98, respectively. These outstanding performances not only reflect the model's high sensitivity, but also highlight its strong adaptability and stability in complex network environments. Although the AUC value of the model in identifying R2L (remote-to-local) attacks is slightly lower, at 0.91, compared to other categories, this performance is still considered very good and indicative of the model's strong recognition capability and effective defense mechanism against this type of more subtle and complex attacks.

3.3 Random forest classifier

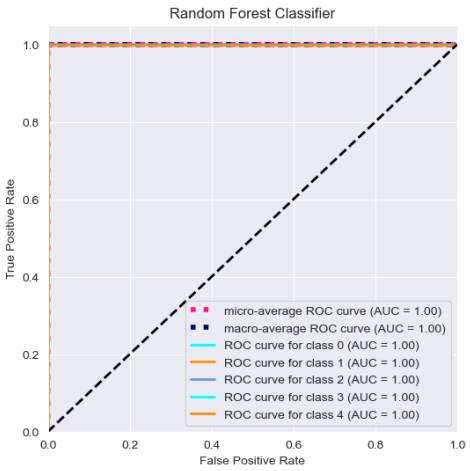

Fig. 3. Random Forest ROC Curves. Below the figure(Picture credit: Original)

The Random Forest classifier, as demonstrated in Fig. 3, displays remarkable performance on the cyber attack dataset, achieving a perfect ROC AUC score of 1.00 for both micro- and macro-averaging. This signifies its balanced and highly accurate classification across both the overall and individual attack categories, effectively distinguishing normal traffic from attack types including Dos, R2L, Probe, and U2R. The constantly high AUC values make it overwhelmingly clear that the model is exceptionally accurate and possesses remarkable skills of generalization, all of which only goes to show the critical importance of the model in enhancing the security of the network. By precisely pinpointing potential threats, this technology acts as a cornerstone in safeguarding the integrity and stability of the network environment.

Logistic Regression, Decision Trees, and Random Forests are three best models for cyber attack detection. Logistic Regression works well on large data and effectively is able to filter out normal traffic from anomalous traffic. Nevertheless, for better understanding of intricate and subtle attack patterns further modifications are required. On the other hand, Decision Trees quickly sort data putting User-to-Root (U2R) attacks from normal traffic. However, it can sometimes have some difficulty in providing classifications that are more precise for

Denial-of-Service (DoS) attacks bags. However, it is important to continue being cautious of cases of over fitting so as to achieve the best results.

4 Conclusion

In this comprehensive research study, the performance of various machine learning algorithms is systematically assessed towards building the capability of identifying Denial-of-Service (DoS) attacks in a complex Software-Defined Networking (SDN) architecture. In order to protect the peak of precision during model training, sensitive and time-consuming preprocessing procedures are carried out including data cleaning accompanying data encoding and normalization steps all based on the KDD Cup 99 dataset. This very involved preprocessing approach is not only about fine polishing of the data and purging it from the defilements but also about transforming it to a form that prepares it in the best ways possible for deposition into algorithms known to the machines.

In this research, it compares several algorithms depending on the algorithm's complexity and simplicity to differentiate normal traffic from DoS attack in SDN. It is designed to enrich the readers with the desired knowledge concerning their performance as well as their drawbacks. The findings show that XGBoost and LightGBM perform better in identifying DoS attacks, especially minority class attacks, thus providing useful information and concrete recommendations on improving DoS attack detection in SDN systems. Furthermore, it underscores the promising potential of integrated learning and gradient boosting techniques in bolstering network security.

In the future, this research plans to further optimize these models through strategies such as hyper-parameter tuning, feature selection, and model fusion to reduce false alarms and missed alarms and improve the security of SDNs. Also, this research looks forward to applying these methods to other network attack detection to support enhancement of network security.

References

1. N. Sultana, N. Chilamkurti, W. Peng, R. Alhadad, Survey on SDN based network intrusion detection system using machine learning approaches. Peer-to-Peer Networking and Applications. **12**, 493-501 (2019)
2. A. Abubakar, B. Pranggono, Machine learning based intrusion detection system for software defined networks, 2017 seventh international conference on emerging security technologies (EST), (2017), 138-143
3. S. Nanda, F. Zafari, C. DeCusatis, E. Wedaa, B. Yang, Predicting network attack patterns in SDN using machine learning approach, 2016 IEEE Conference on Network Function Virtualization and Software Defined Networks, IEEE, (2016), 167-172
4. N. Ahmed, A. B. Ngadi, J. M. Sharif, S. Hussain, M. Uddin, M. S. Rathore, F. T. Zuhra, Network threat detection using machine/deep learning in SDN-based platforms: a comprehensive analysis of state-of-the-art solutions, discussion, challenges, and future research directions. Sensors. **22**, 7896 (2022)
5. A. Ahmad, E. Harjula, M. Ylianttila, I. Ahmad, Evaluation of machine learning techniques for security in SDN, in 2020 IEEE Globecom Workshops (GC Wkshps), (2020), 1-6
6. A. S. Jose, L. R. Nair, V. Paul, Towards detecting flooding DDoS attacks over software defined networks using machine learning techniques. Rev. Geintec-Gestao Inovacao Tecnol. **11**, 3837-3865 (2021)

7. K. S. Sahoo, A. Iqbal, P. Maiti, B. Sahoo, A machine learning approach for predicting DDoS traffic in software defined networks, 2018 International Conference on Information Technology (ICIT), (2018), 199-203
8. J. A. Perez-Diaz, I. A. Valdovinos, K. K. R. Choo, D. Zhu, A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. IEEE Access. **8**, 155859-155872 (2020)
9. A. O. Sangodoyin, M. O. Akinsolu, P. Pillai, V. Grout, Detection and classification of DDoS flooding attacks on software-defined networks: a case study for the application of machine learning. IEEE Access. **9**, 122495-122508 (2021)
10. A. Ahmad, Evaluation of machine learning techniques for intrusion detection in software defined networking. Master's thesis. **1**, (2020)