Research on the Performance of Machine Learning Algorithms for Intrusion Detection System

Yue Li, Wusheng Xu, Qing Ruan

School of Computer Science and Technology, Donghua University, Shanghai, 201620, China

wusheng@mail.dhu.edu.cn

Abstract. Intrusion Detection System (IDS) is a critical approach to ensure network system security. Currently, network attacks are complicated and volatile. Moreover, hackers are also more inclined to adopt new attack techniques to obtain users' privacy. Under this circumstance, the intelligent intrusion detection system has become the primary approach to detect hackers' attacks. In this study, intelligent intrusion detection models applying the novel UNSW-NB15 data set as well as various machine learning algorithms are investigated. Furthermore, data set is pre-processed through using one-hot encoding and normalization in our experiment. Subsequently, the performance comparison of six different types of machine learning algorithms in intrusion detection tasks was implemented. The experimental results reveal that in the complex and changeable network traffic data, machine learning technology has presented desired performance.

1. Introduction

In current years, the Internet, especially the mobile Internet, has achieved explosive growth. Unfortunately, the high popularity of the Internet has led to more security problems and security threats. Accordingly, the reliability of the internet and its services is declining.

In order to defend against network threats, various security technologies can be utilized, one of which is the Intrusion Detection System (IDS). It is a system applying special analysis technology to monitor network traffic transmission or system logs with the aim to identify the behaviors that violate security policies [1]. When an intrusion is detected, an alarm will be issued or a proactive response will be taken. Different from other network security technologies, IDS is a proactive security protection technology. It can be either a software system or a hardware device. Moreover, it can collect information from various systems or network resources, and then analyze the characteristics of network traffic to cope with network attacks.

According to the first detection technology, IDS can be divided into two categories: misuse-based and anomaly-based [1]. The misuse-based detection method must establish a model on the basis of hostile and unacceptable behavior. Any behavior that conforms to this model will be determined as an intrusion. The anomaly-based detection method must establish a model of system access to normal behavior. Any behavior that does not conform to this model will be determined to be an intrusion.

Current network intrusion detection systems face the challenge of processing large volumes of data [2]. Simultaneously, with the development of artificial intelligence technology, an increasing number of researchers have begun to employ machine learning and data mining technologies for intrusion detection tasks. Intrusion detection methods based on machine learning to learn and adapt to different intrusion activities can bring higher accuracy rates as well as lower false alarm rates.
2. Dataset Analysis and Experimental Setup

2.1. Dataset Analysis

UNSW-NB15 dataset are adopted in this study. Its source files include Pcap files, BRO files, Argus files, Reports files and CSV files [3]. Specifically, the CSV files of this dataset are applied.

In CSV files, the data in this dataset has 49 features. These features are categorized into six groups as follows: Flow Features, Basic Features, Content Features, Time Features, Additional Generated Features, and Labelled Features, respectively. The Additional Generated Features are further categorized into two subgroups as follows: General Purpose Features and Connection Features [4].

In addition, a partition from this dataset is configured as a training set and a testing set [3]. Therefore, experiment is not required to divide the data set. This makes the models of all related studies different. However, the applied data is the same, which is conducive to measuring the detection performance between different models.

Nour Moustafa et al.[5] made a detailed introduction to the features of the data set. It is worth noting that their researcher established a numbering for the features. In this study, the same numbering system is followed, and # is used to identify the feature number.

Data in the 48th dimension (#48 in [5]) is based on nominal expression, which consists of one normal type and nine attack types. The nine attack types are Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms [6]. In this study, a statistical analysis of each attack type is conducted to explore the sample equilibrium.

| Attack Type       | Train Dataset | Test Dataset |
|-------------------|---------------|--------------|
|                   | Records       | Proportion   | Records       | Proportion   |
| Generic           | 40000         | 22.81%       | 18871         | 22.92%       |
| Exploits          | 33393         | 19.04%       | 11132         | 13.52%       |
| Fuzzers           | 18184         | 10.37%       | 6062          | 7.36%        |
| DoS               | 12264         | 6.99%        | 4089          | 4.97%        |
| Reconnaissance    | 10491         | 5.98%        | 3496          | 4.25%        |
| Analysis          | 2000          | 1.14%        | 677           | 0.82%        |
| Backdoor          | 1746          | 1.00%        | 583           | 0.71%        |
| Shellcode         | 1133          | 0.65%        | 378           | 0.46%        |
| Worms             | 130           | 0.08%        | 44            | 0.05%        |
| Total             | 119341        | 68.06%       | 45332         | 55.06%       |

Data in the 49th dimension (#49 in [5]) is based on binary expression (0 and 1). 0 refers to normal behaviour, and 1 refers to intrusive behaviour. Similarly, the statistical analysis for this label data is conducted.

| Label | Train Dataset | Test Dataset |
|-------|---------------|--------------|
|       | Records       | Proportion   | Records       | Proportion   |
| 0     | 56000         | 31.94%       | 37000         | 44.94%       |
| 1     | 119341        | 68.06%       | 45332         | 55.06%       |
| Total | 175341        | 100%         | 82332         | 100%         |

In the experimental research, the performance of the machine learning model based on the 49th dimension label is mainly tested. Namely, the experiment is mainly carried out for binary classifications, signifying that the goal of intrusion detection system is to classify normal behaviour and abnormal behaviour in network traffic.

2.2. Experimental Setup

In this study, six machine learning algorithms are applied to establish intrusion detection models so as to evaluate the feasibility and performance of machine learning models in intrusion detection tasks. These six algorithms are K-nearest neighbor (KNN), Decision Tree, Logistic Regression, Support
Vector Machine (SVM), Random Forest, and Adaboost. In addition, Random Forest and Adaboost are ensemble models. Random Forest is based on bagging method, and Adaboost is based on boosting method [7][8]. This is beneficial to compare the detection performance of different types of algorithms.

The data sets employed in experiment are UNSW_NB15_training-set and UNSW_NB15_testing -set. The first one is applied to train the model, and the second one is applied to test the performance of model. As it is described in 2.1, the application of these two files as training and test sets respectively can provide a standard and unified data set for related research, which makes different intrusion detection models have better measurement standards. Therefore, it is advocated that the related research should apply this standard training data set and testing data set, instead of randomly customizing the training set and testing set in related studies.

As shown in Fig. 1, the experimental steps mainly include statistical analysis, data pre-processing, application of machine learning algorithms, and result analysis.

![Fig. 1 Experiment Steps](image)

The work of the statistical analysis mainly focuses on the problem of sample balance, and the results have been presented in 2.1. The key techniques mainly applied in data pre-processing are as follows:

1) One-Hot Encoding: For some feature whose data type is Nominal, such as #6, #14 in [5], it needs to apply one-hot encoding technology to process it into Integer type data. Namely, if there are N different values in a feature, then the feature can be abstracted into N different states. The one-hot encoding ensures that only one state of each sample is in the "activated state", indicating that only one of the N states has a value of 1, and the other states are 0. In this dataset, the one-hot encoding technology is utilized for the data of the Nominal type. For instance, after the one-hot encoding is performed in Table 3, it can be transformed into the form of Table 4.

| #  | service |
|----|---------|
| 1  | http    |
| 2  | ftp     |
| 3  | smtp    |

| #  | http | ftp | smtp |
|----|------|-----|------|
| 1  | 1    | 0   | 0    |
| 2  | 0    | 1   | 0    |
| 3  | 0    | 0   | 1    |

2) Min-max normalization: Min-max normalization, also known as dispersion normalization, is a linear transformation of the original data so that the resulting value is mapped between [0-1]. The normalization is calculated as follows:

\[ x^* = \frac{x - \min(x)}{\max(x) - \min(x)} \] (1)
After the data pre-processing is completed, the ML algorithms are applied to the UNSW-NB15 data set to carry out rule mining with the purpose to further distinguish normal traffic and intrusive traffic in the network.

3. Performance Evaluation and Discussions

3.1. Performance Measurement
Performance parameters, such as Accuracy, Precision, Recall, F1_score, and the time taken for model to detect, are applied to evaluate and compare the performance of different models. These indicators can be calculated based on the parameters in the confusion matrix. A confusion matrix is a visualization tool that acts as a basis for calculating all other parameters [2]. As shown in table 5, it possesses the following four attributes: True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) [9].

Table 5 Confusion Matrix

| Actual | Predicted | Anomaly | Anomaly |
|--------|-----------|---------|---------|
| Normal | TP        | FN      |         |
|        | FP        | TN      |         |

According to Table 5, some measures, such as Accuracy, Precision, Recall, and F1_score, could be calculated [10]. The calculation of these measures is revealed as follows:

1) **Accuracy**: It refers to the percentage of the results that the model predicts correctly in the total predictions. The optimal accuracy is 100%.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}
\]

2) **Precision**: It refers to the percentage of actual abnormal behaviour in the total samples predicted to be abnormal behaviour. The optimal precision is 100%.

\[
\text{Precision} = \frac{TP}{TP + FP} \tag{3}
\]

3) **Recall**: It refers to the proportion of abnormal behaviours correctly predicted in the total actual abnormal behaviours. The optimal recall is 100%.

\[
\text{Recall} = \frac{TP}{TP + FN} \tag{4}
\]

4) **F1_score**: In some cases, Precision and Recall are contradictory. Therefore, on the basis of these two indicators, the concept of F1 is proposed to measure the robustness of the classification model.

\[
\text{F1_score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}
\]

3.2. Evaluation
In this study, multiple iterations of training for each machine learning model have been conducted, and the optimal performance of each mode has been determined through applying a hyper-parameter learning curve. Subsequently, Accuracy, Precision, Recall, F1_score, and Cost are applied to analyze the performance of six machine learning models in intrusion detection tasks. In order to evaluate the performance of the intrusion detection model in the actual production environment, Cost is defined as
the time required for the model to classify 10,000 network traffic records. The performance analysis results are revealed in Table 6.

| Model                | Accuracy (%) | Precision (%) | Recall (%) | F1_score (%) | Cost (s) |
|----------------------|--------------|---------------|------------|--------------|---------|
| KNN                  | 85.17        | 80.36         | 96.17      | 87.78        | 138.7   |
| Decision tree (ID3)  | 92.13        | 91.25         | 94.79      | 92.99        | 0.018   |
| Logistic regression  | 81.35        | 76.49         | 95.48      | 84.93        | 0.006   |
| SVM                  | 87.05        | 82.95         | 96.27      | 89.11        | 69.11   |
| Random forest        | 90.34        | 87.17         | 96.68      | 91.68        | 0.288   |
| Adaboost             | 89.11        | 85.51         | 96.59      | 90.71        | 0.281   |

In order to evaluate the performance of the model in a more comprehensive way, the ROC curve is considered in this study. It is a graph-based way to evaluate the performance of machine learning, whose calculation method also considers the classifier's ability to classify positive and negative samples. Therefore, in the case of unbalanced samples, a reasonable evaluation of the classifier can still be made [1][2]. According to the results of the statistical analysis for the data set in the second part, the data set has an imbalance problem. Thus, the analysis result of the ROC curve is particularly significant for model evaluation in this study. In the ROC curve, an important criterion for evaluate the performance of a classifier is Area Under Curve (AUC), which is the coverage area under the curve. Generally speaking, the larger the area is, the better the performance of the classifier is. In the evaluation, AUC is given in the lower right corner of each ROC curve. The ROC curve and AUC of this experiment are displayed in Fig. 2.

3.3. Discussion

According to Table 6, the study reveals that the Decision Tree can best classify abnormal and normal network behaviors. This algorithm can enable the accuracy and precision of intrusion detection to reach 92.13% and 91.25% respectively. Moreover, its F1_score is the highest (reaching 92.99%), indicating that it has achieved good results between Precision and Recall. Meanwhile, the time consumption of the Decision Tree in the intrusion detection task is almost the smallest, which is just lower than the logistic regression, indicating that the model owns the ability to quickly respond and judge, and can quickly predict and classify network access traffic.

The second optimal performance is the Random forest algorithm. It is an ensemble learning method based on bagging learning strategy, whose accuracy reaches 90.34%. It is worth noting that it possesses a recall rate of 96.68%, which is the highest among all models, indicating that it can correctly classify most abnormal network behaviors and improve the security performance of intrusion detection systems.

The next one is the Adaboost algorithm, which is an ensemble algorithm based on the boosting learning strategy. It owns an accuracy of 89.11%. Furthermore, it possesses almost the same performance as the Random forest in terms of recall, F1_score, and cost. In addition, among the remaining three algorithms, Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) are very potential intrusion detection models. However, they all involve abundant calculations and consume overmuch time. Furthermore, although the time consumption of logistic regression is the smallest among all models, other aspects of performance have not reached a high level.

On the other hand, according to Fig. 2, it is found that the AUC value of the Random Forest ROC Curve is 98.07%, which is the largest among all models. With the continuous increase of the False Positive Rate (FPR), the value of True Positive Rate (TPR) is relatively small but growing rapidly when FPR between (0, 0.1). In the interval of (0.1, 1), the value of TPR is stable between (0.9, 1). The following ones are Decision Trees and Adaboost. Their ROC curves are similar to random forest, with AUC values of 97.41% and 96.57%, respectively. In the interval of (0, 0.1), the curve tends to be flat after the TPR value rises rapidly. The ROC curves of the other three intrusion detection models have many fluctuations, and the AUC value is relatively small. Combining the data in Table 6 and Fig. 2, it can be seen that these three models perform poorly.
4. Conclusion and Future work
The combination of network intrusion detection and machine learning technology is in the process of continuous development. This study focuses on applying six most popular machine learning techniques in network intrusion detection to evaluate the performance of machine learning techniques in intrusion detection tasks. The experimental results reveal that Decision Tree, Random Forest, and Adaboost have the optimal detection performance. These three models are all tree-based models, Therefore, the tree-based model is worthy of further research in intrusion detection tasks. K-nearest neighbors, Logistic Regression, and Support Vector Machines have their own obvious weaknesses. For instance, the detection rate of Logistic Regression is low, and the time consumption of K-nearest neighbors and Support Vector Machines is relatively large.
In the future, in order to reduce the amount of calculation of the model, the feature reduction of the data set will be applied in experiments. Moreover, it is hoped that sample equalization techniques will be applied in the research to solve the imbalance of various attack samples, so as to carry out multi-attack classification research on intrusion detection system.

References
[1] Buczak A L, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection[J]. IEEE Communications surveys & tutorials, 2015, 18(2): 1153-1176.
[2] Malhotra H, Sharma P. Intrusion Detection using Machine Learning and Feature Selection[J]. International Journal of Computer Network & Information Security, 2019, 11(4).
[3] Australian Centre for Cyber Security (ACCS).The UNSW-NB15 Dataset Description[Online].https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/
[4] Janarthanan T, Zargari S. Feature selection in UNSW-NB15 and KDDCUP'99 datasets[C]/2017 IEEE 26th international symposium on industrial electronics (ISIE). IEEE, 2017: 1881-1886.
[5] Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)[C]/2015 military communications and information systems conference (MilCIS). IEEE, 2015: 1-6.
[6] Moustafa N, Slay J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set[J]. Information Security Journal: A Global Perspective, 2016, 25(1-3): 18-31.
[7] Moustafa N, Turnbull B, Choo K K R. An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things[J]. IEEE Internet of Things Journal, 2018, 6(3): 4815-4830.
[8] Sagi O, Rokach L. Ensemble learning: A survey[J]. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018, 8(4): e1249.
[9] Susmaga R. Confusion matrix visualization[M]/Intelligent Information Processing and Web Mining. Springer, Berlin, Heidelberg, 2004: 107-116.
[10] Mohamed H, Hefny H, Alsawy A. Intrusion detection system using machine learning approaches[J]. Egypt. Comput. Sci. J, 2018, 42(3): 1-13.