A Performance Analysis Approach for Network Intrusion Detection Algorithms

Zhihao Wang, Dingde Jiang(✉), Yuqing Wang, and Junyang Zhang

School of Astronautics and Aeronautic, University of Electronic Science and Technology of China, Chengdu 611731, China
jiangdd@uestc.edu.cn

Abstract. With the development of mobile Internet and cloud computing, the amount of network traffic has been significantly increased. Security problems have drawn a lot of attention, while traditional methods are becoming increasingly unsuitable for it. In this paper, three machine learning algorithms are employed to detect network intrusion, including KNN, Random Forest, and Multilayer Perceptron. Performance evaluation and comparison between them are conducted, in terms of precision, recall, training time, etc. Simulation results on the NSL-KDD, a benchmark data set of network intrusion detection, show that the Random Forest algorithm exhibits higher detection accuracy and remarkably shorter training time.

Keywords: Network intrusion detection · Machine learning · Random forest · Multilayer Perceptron · Performance analysis

1 Introduction

With the development of the mobile Internet, many business systems are deployed on distributed cloud computing platforms. A large number of user groups generate massive amounts of network traffic. Much of network traffic is generated by malicious attacks carried out by attackers against certain servers or hosts. Some attackers act like normal users, generating data, and hiding their malicious activities under TB or even PB-level data. Due to a large amount of data or lack of network intrusion detection capabilities, hackers can invade enterprise computer systems through Trojans, backdoors, and even complex APT and "0-day" vulnerabilities, threatening the information security of the companies. When the Trojan communicates with the attacker, the generated network traffic showing obvious communication features, which can be effectively captured by intrusion detection technology [1]. However, the anomaly detection algorithms behave differently in different environments, and there is diversity between accuracy, recall, precision. Therefore, it is especially important to compare and evaluate the performance of different intrusion techniques.

Z. Li et al. propose a network intrusion detection method based on Recurrent Neural Networks and Broad Learning System to detect various known network attacks [2].

H. Song and D. Jiang (Eds.): SIMUtools 2020, LNICST 369, pp. 215–224, 2021.
https://doi.org/10.1007/978-3-030-72792-5_20

Authors study the prediction approach to end-to-end traffic in space information networks [3, 4]. I. Ahmad et al. compare the performance of support vector machine, random forest and extreme learning machine algorithm [5]. Some studies also focus on estimations to network traffic [6]. M. C. et al. study the IDS built by Snort and Suricata based on Raspberry Pi, and its performance comparison [7]. D. Jiang et al. research the behaviors and activities [8]. SAMIRA et al. designed an anomaly-based detection called Mutation Cuckoo Fuzzy for feature selection and Evolutionary Neural Network for classification [9]. Compressive sensing-based approach also can be used in [10]. Authors propose to optimize a soft computing tool widely used for intrusion detection namely Back Propagation Neural Network using a novel hybrid Framework based on improved Genetic Algorithm and Simulated Annealing Algorithm [11]. Wireless network is studied in [12–14]. The scholars use the proposed State Preserving Extreme Learning Machine algorithm [15]. Intrusion Detection for IoT network is studied in [16–18]. And industry application is studied in [19–22]. An improved convolutional neural network model is proposed in [23]. Large-Scale cyber networks are studied in [24, 25]. From the review above, we can see that the performance of the network intrusion detection algorithms still attracts a lot of attention in academia and industry.

In this paper, we study the performance comparison of three network intrusion detection algorithms. First, the general architecture of IDS (Intrusion Detection System) is illustrated. And three network intrusion algorithms are introduced, including KNN (K Nearest Neighbor), RF (Random Forest), and MLP (Multilayer Perceptron). To evaluate the performance of three network intrusion algorithms, the network intrusion dataset NSL-KDD is employed, which has been preprocessed to input to the algorithms. Besides, we present several performance comparison metrics. Evaluating simulations are carried out, which show that the Random Forest intrusion detection algorithm has better performance than the other two algorithms.

2 System Model

In this section, we will briefly introduce three intrusion detection algorithms compared in this paper, including KNN, RF, and MLP classifiers.

1. KNN
   The basic rule of the KNN algorithm is to find the k nearest neighbors in all the N samples. When $k = 1$, KNN becomes the nearest neighbor problem. The first step of KNN is to calculate the distance between the input sample and all samples. The distance between the n-dimension vector $a(x_{11}, x_{12}, \ldots, x_{1n})$ and $b(x_{21}, x_{22}, \ldots, x_{2n})$ is calculated as (1), which is called the Euclidean Distance.

$$d_{12} = \sqrt{\sum_{k=1}^{n} (x_{1k} - x_{2k})^2} \tag{1}$$

   Then choose k nearest neighbors which have the shortest distance between the input sample. Based on the main class of these k neighbors, the classification of the input sample can be achieved.

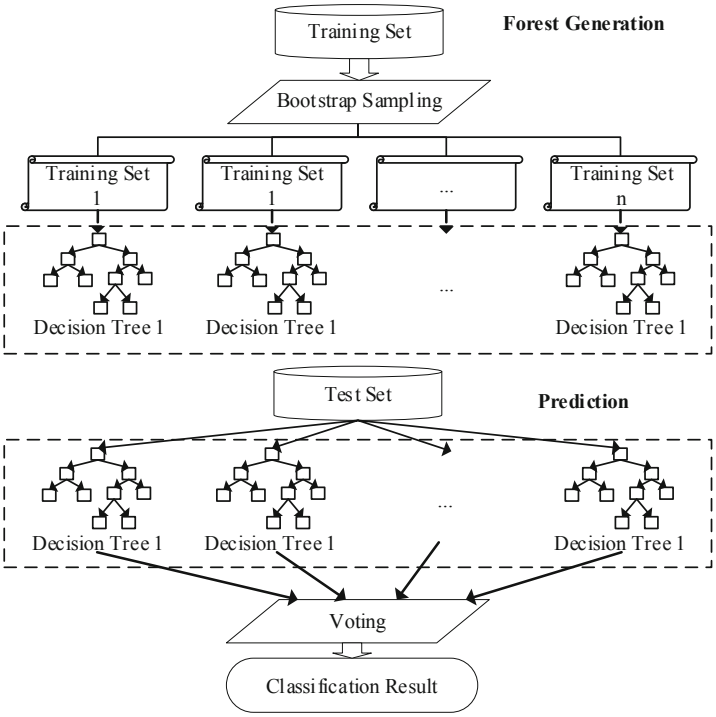

Fig. 1. Basic Architecture of Random Forest.

2. Random Forest
   Random forest algorithm is an ensemble learning algorithm with decision tree as base learner. The forest is constructed by many decision trees. There is no correlation between each decision tree of random forest. After the forest constructed, when a new input sample enters, each decision tree in the forest judges separately and gets the classification result of the sample. Finally, through the voting mechanism, combine the results of all decision trees. The one with the most classification votes belongs to this category. The basic architecture of RF is shown in Fig. 1. Figure 1 Basic architecture of random forest.
3. MLP
   MLP is also called artificial neural network (ANN). In addition to the input and output layer, there can be multiple hidden layers between the input layer and the output layer. The general MLP contains only one hidden layer, which is shown in Fig. 2. The cells of each layer are connected with all the cells of the adjacent layer. And there is no connection between the cells of the same layer. When a training sample is input to the network, the activation value of the neuron propagates from the input layer to the output layer through each middle layer. Each neuron in the output layer obtains the input response of the network. Next, according to the direction of reducing the target output and the actual error, from the output layer through the middle layer,

each connection weight is updated layer by layer, and finally back to the input layer. By cycling the above processes, a trained neural network model can be obtained.

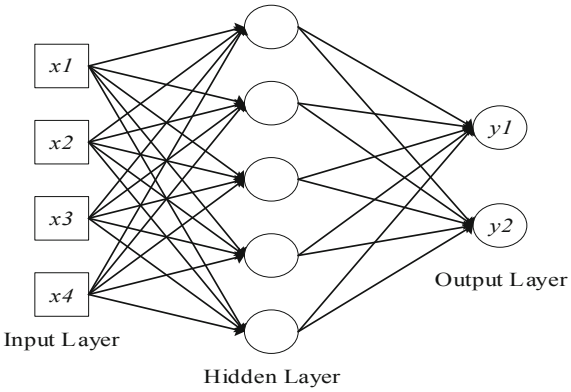

Fig. 2. General MLP architecture.

3 Research Methodology

The performance comparison is based on the standard network intrusion dataset NSL-KDD, which is the improved version of KDD-99. The data preprocessing is conducted to make the dataset more suitable for classifiers to handle. The metrics to evaluate the performance of different detection algorithms are presented in this section.

3.1 Data Preprocessing

NSL-KDD includes 39 common types of network attacks, 22 of them in the training set, and 17 in the test. There are several columns in text form. Therefore, it is necessary to convert them into the form which can be inputted into the classifier.

The raw data of the NSL-KDD dataset has some useless features and some of the features are in text form. And all of 39 types of attacks are in the data, causing it cannot be handled directly. Therefore, we carry out data preprocessing. The main steps of data preprocessing of NSL-KDD are data cleaning, data encoding, data normalization and label binarization.

1. Data Cleaning.
   The 43rd column attribute in the dataset indicates whether the sample is easy to classify, which is not essential for this paper. Thereby, it is necessary to eliminate the influence of useless features.
2. Data Encoding.
   There are three character-type features, including protocol type, service and flag. The detection algorithms based on machine learning are not capable of dealing with

characters. We conduct data encoding for the character-type features. For example, for the protocol type, there are three types, TCP, UDP and ICMP. We encode the TCP as 1, UDP as 2, and ICMP as 3. The remaining columns are also encoded in this way.

3. Data Normalization.
   When the scales of features in different dimensions of the original data are inconsistent, normalization steps are needed to preprocess the data. The normalization method we conduct to deal with NSL-KDD is the Z-Score normalization method. Z-score standardization is to scale the data to a specific range, ensuring the $\sigma = 1,\ \mu = 1$. The σ is the standard deviation of samples, and the μ is the mean value of samples. The standard deviation is defined as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{2}$$

   where N represents the total number of tested samples, x_i is the value of sample i. Z-Score transformation formula is:

$$z = \frac{x - \mu}{\sigma} \tag{3}$$

   where x is the sample before Z-Score normalization and z is the converted value.
4. Label Binarization.
   In some application scenarios, it is not necessary to distinguish different types of network attacks in detail. It only needs to detect normal and abnormal traffic. Therefore, in the data preprocessing, we encode the abnormal traffic type including Dos, Probe, R2L, U2R to 1 and the normal traffic to 0. The label binarization could effectively improve the performance of some classifiers.

3.2 Performance Comparison Metrics

The confusion matrix is a common evaluation method used to evaluate the classification performance of the intrusion detection binary classification problem. The confusion matrix used to determine the detection performance of the three systems in this paper is shown in Table 1.

Common performance metrics used to evaluate IDS performance are as follows:

- Precision
  The precision rate is an indicator of accuracy, which indicates the proportion of the number of positive cases correctly classified by the classifier to the number of positive cases. It can be expressed as

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

  where TP represents the number of abnormalities correctly detected, and the FP represents the false positive prediction of negative class.

Table 1. Confusion matrix.

	Predicted value	
Observed value	TP (True Positive)	FN (False Negative)
	FP (False Positive)	TN (True Negative)

- True Positive Rate (TPR)/Recall
  The *TPR* is defined as the ratio of the number of correctly predicted network anomalies and the total number of network anomalies. *TPR* is also called Recall or sensitivity. *TPR* can be represented as

$$TPR = \frac{TP}{TP + FN} = Recall = Sensitivity \tag{5}$$

  where FN represents the number of normal conditions that are erroneously detected.
- False Positive Rate (FPR)
  The false positive rate is defined as the proportion of normal conditions incorrectly classified as a intrusion and all normal conditions, which can be represented as

$$FPR = \frac{FP}{FP + TN} \tag{6}$$

  where TN represents the number of correctly detected normal conditions.
- F1-Score
  F-measure is the weighted harmonic average of precision and recall, which is quite effective for the imbalanced classification problem. We use the F1-Score in this paper, which can be expressed as:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{7}$$

4 Simulation and Result Analysis

4.1 Evaluation Strategy

The simulation experiment in this paper is based on the NSL-KDD intrusion detection data set. 80% of the dataset is taken as training set and 20% is taken as the testing set. The officially provided test set is used as the validation set. We compared the training results of the classifiers with the original data, the normalized and binary data, as well as the training time and performance on the validation set. The metrics in 3.3 are used to evaluate the performance of the KNN, RF, and MLP classifier. And the ROC Curve is drawn to compare the performance of the two-class classification algorithm.

4.2 Simulation Results

Comparison on precision, recall, F1-Score and training time between KNN, RF and MLP is shown in Table 2. RF has better performance when processing the raw data of NSL-KDD dataset. The precision, recall and F1-Score are higher than the other two classifiers, while the training time is significantly shorter than them. As KNN is based on the distance calculation between the samples, the training time is much longer than the other classifiers, which will be even higher when the samples are normalized.

Table 2. Classification result on raw dataset.

	Precision	Recall	F1-Score	Time
KNN	0.71627	0.72529	0.67444	9.49819
RF	0.80297	0.73771	0.68995	0.83753
MLP	0.64075	0.70232	0.65793	4.54791

The detection precision on each attack type is shown in Fig. 3. If the detection accuracy is less than 50%, the classifier is almost unavailable, because the accuracy is less than that of the random guess classifier. Therefore, in the comparing experiment, we only illustrate the detection result higher than 50%. The detection precision of the three algorithms on the training data is higher than 95%. However, MLP cannot detect the R2L and U2R attacks effectively. There is not much difference in the detection precision between KNN and RF on the train data. As detecting on the test data, because there are unknown attack types, detection precision decrease obviously. And the Random Forest classifier has the best performance.

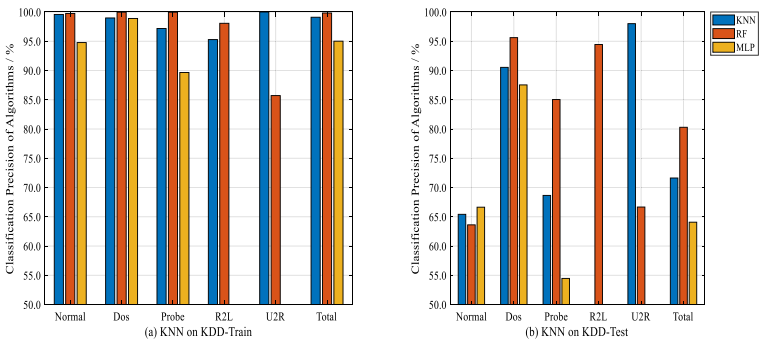

Fig. 3. Precision comparison on NSL-KDD.

The detection results after data normalization are shown in Fig. 4. After normalization, the detection precision of KNN and MLP increases, while the Random Forest decrease than before. Because KNN is based on the distance calculation of the samples, and MLP can converge better and faster after standardization. But the training time of

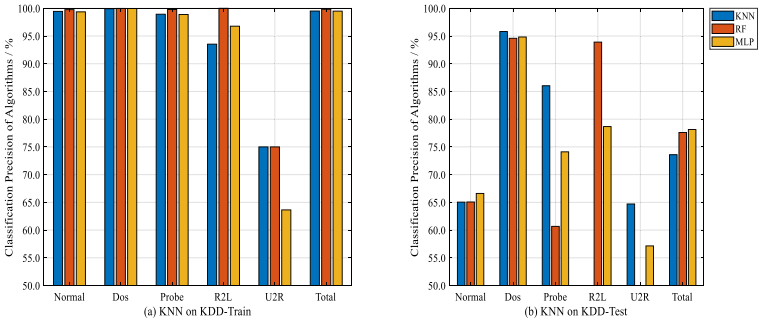

Fig. 4. Precision comparison on normalized NSL-KDD.

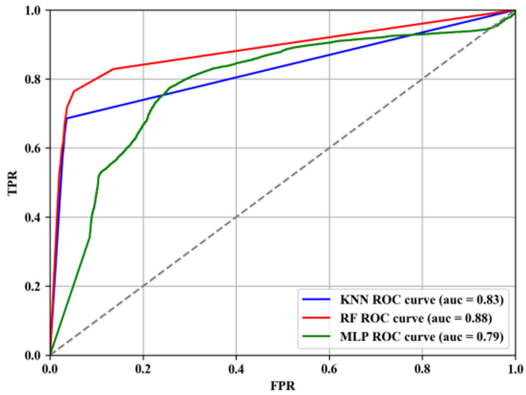

Fig. 5. ROC curve of two-class classifier.

KNN increases to 106.38 s, and the MLP increases to 16.5 s. The precision of MLP exceeds RF and KNN, while it is still lower than the RF on the raw data.

After label binarization, the three classifiers turn into the two-class classifier. Figure 5 shows the ROC (receiver operating characteristic) Curve of them. The larger the area occupied by the ROC curve, the better the performance of the classifier. Obviously, the Random Forest Classifier still has better performance than KNN and MLP.

5 Conclusion

This paper investigates and compares the performance of different machine learning algorithms in network intrusion detection, including KNN, Random Forest and MLP. The NSL-KDD data is employed in the comparison, with data preprocessing. Speaking of detecting precision, recall and F1-Score, the Random Forest algorithm outperforms the other two algorithms. Except the MLP behaves better slightly after normalization. Besides, the Random Forest has the best training efficiency, with remarkably short training time. Therefore, as an ensemble learning method, the Random Forest is suitable

for network intrusion detection in this paper. For future work, more intrusion detection algorithms and feature transformation techniques will be investigated.

Acknowledgement. This work was supported in part by the National Natural Science Foundation of China (No. 61571104), the Sichuan Science and Technology Program (No. 2018JY0539), the Key projects of the Sichuan Provincial Education Department (No. 18ZA0219), the Fundamental Research Funds for the Central Universities (No. ZYGX2017KYQD170), the CERNET Innovation Project (No. NGII20190111), the Fund Project (Nos. 61403110405, 315075802), and the Innovation Funding (No. 2018510007000134). The authors wish to thank the reviewers for their helpful comments.

References

1. Xie, J., Li, S., Zhang, Y., et al.: A method based on hierarchical spatiotemporal features for trojan traffic detection. In: 2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC), pp. 1–8 (2019)
2. Li, Z., Batta, P., Trajkovic, L.: Comparison of machine learning algorithms for detection of network intrusions. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4248–4253 (2018)
3. Qi, S., Jiang, D., Huo, L.: A prediction approach to end-to-end traffic in space information networks. Mob. Netw. Appl. (2019). https://doi.org/10.1007/s11036-019-01424-2, online available
4. Wang, Y., Jiang, D., Huo, L., Zhao, Y.: A new traffic prediction algorithm to software defined networking. Mob. Netw. Appl. (2019). https://doi.org/10.1007/s11036-019-01423-3. online available
5. Ahmad, I., Basheri, M., Iqbal, M.J., et al.: Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access **6**, 33789–33795 (2018)
6. Jiang, D., Huo, L., Li, Y.: Fine-granularity inference and estimations to network traffic for SDN. PLoS ONE **13**(5), 1–23 (2018)
7. Cosar, M., Kiran, H.E.: Performance comparison of open source IDSs via Raspberry Pi. In: 2018 International Conference on Artificial Intelligence and Data Processing (IDAP), pp. 1–5 (2018)
8. Jiang, D., Huo, L., Song, H.: Rethinking behaviors and activities of base stations in mobile cellular networks based on big data analysis. IEEE Trans. Netw. Sci. Eng. **7**(1), 80–90 (2020)
9. Sarvari, S., Sani, N.F.M., Hanapi, Z.M., et al.: An efficient anomaly intrusion detection method with feature selection and evolutionary neural network. IEEE Access **8**, 70651–70663 (2020)
10. Jiang, D., Wang, W., Shi, L., Song, H.: A compressive sensing-based approach to end-to-end network traffic reconstruction. IEEE Trans. Netw. Sci. Eng. **7**(1), 507–519 (2020)
11. Chiba, Z., Abghour, N., Moussaid, K., et al.: A hybrid optimization framework based on genetic algorithm and simulated annealing algorithm to enhance performance of anomaly network intrusion detection system based on BP neural network. In: 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), pp. 1–6 (2018)
12. Jiang, D., Li, W., Lv, H.: An energy-efficient cooperative multicast routing in multi-hop wireless networks for smart medical applications. Neurocomputing **2017**(220), 160–169 (2017)

13. Yang, H., Wang, F.: Wireless network intrusion detection based on improved convolutional neural network. IEEE Access **7**, 64366–64374 (2019)
14. Jiang, D., Zhang, P., Lv, Z., et al.: Energy-efficient multi-constraint routing algorithm with load balancing for smart city applications. IEEE Internet of Things J. **3**(6), 1437–1447 (2016)
15. Singh, K., Mathai, K.J.: Performance comparison of intrusion detection system between deep belief network (DBN) algorithm and state preserving extreme learning machine (SPELM) algorithm. In: 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), pp. 1–7 (2019)
16. Jiang, D., Wang, Y., Lv, Z., Wang, W., Wang, H.: An energy-efficient networking approach in cloud services for IIoT networks. IEEE J. Sel. Areas Commun. **38**(5), 928–941 (2020)
17. Zhang, Y., Li, P., Wang, X.: Intrusion detection for IoT based on improved genetic algorithm and deep belief network. IEEE Access **7**, 31711–31722 (2019)
18. Wang, F., Jiang, D., Qi, S.: An adaptive routing algorithm for integrated information networks. China Commun. **7**(1), 196–207 (2019)
19. Liu, W., Liu, X., Di, X., et al.: A novel network intrusion detection algorithm based on fast fourier transformation. In: 2019 1st International Conference on Industrial Artificial Intelligence (IAI), pp. 1–6 (2019)
20. Jiang, D., Wang, Y., Lv, Z., Qi, S., Singh, S.: Big data analysis based network behavior insight of cellular networks for Industry 4.0 applications. IEEE Trans. Ind. Inf. **16**(2), 1310–1320 (2020)
21. Liang, W., Li, K., Long, J., et al.: An industrial network intrusion detection algorithm based on multifeature data clustering optimization model. IEEE Trans. Industr. Inf. **16**(3), 2063–2071 (2020)
22. Jiang, D., Huo, L., Lv, Z., Song, H., Qin, W.: A joint multi-criteria utility-based network selection approach for vehicle-to-infrastructure networking. IEEE Trans. Intell. Transp. Syst. **19**(10), 3305–3319 (2018)
23. Khan, R.U., Zhang, X., Alazab, M., et al.: An improved convolutional neural network model for intrusion detection in networks. In: 2019 Cybersecurity and Cyberforensics Conference (CCC), pp. 74–77 (2019)
24. Huo, L., Jiang, D., Qi, S., et al.: An AI-based adaptive cognitive modeling and measurement method of network traffic for EIS. Mob. Netw. Appl. (2019). https://doi.org/10.1007/s11036-019-01419-z. online available
25. Miehling, E., Rasouli, M., Teneketzis, D.: A POMDP approach to the dynamic defense of large-scale cyber networks. IEEE Trans. Inf. Forensics Secur. **13**(10), 2490–2505 (2018)