Chapter 53
Intrusion Detection: A Machine Learning Approach

Vipul Borhade, Aparna Nayak and R. Dakshayani

1 Introduction

Intrusion generally refers to malicious activities directed at computer network system to compromise its integrity, availability, and confidentiality. Network security is important because modern information technology relies on it to drive businesses and services. Security can be enforced on the network through intrusion detection systems. These are security devices or software usually implemented by large and medium organizations to enforce security policies and monitor network perimeter against security threats and malicious activities. Other associated systems include firewall and intrusion prevention system. Essentially, intrusion detection device or application scrutinizes every incoming or outgoing network traffic and analyzes packets for known and unknown events. Detected known events and violations are logged usually in a central security information and event management system. Malicious activities or unknown events may be set up to alert system administrator or the related packets dropped depending on the configurations enabled on the intrusion detection system. Prevention of security breaches cannot be completely avoided. Hence, effective intrusion detection becomes important for organizations to proactively deal with security threats in their networks. However, many existing intrusion detection systems are rule-based [3] and are not quite effective in detecting a new intrusion event that has not been encoded in the existing rules. Besides, intrusion detection rules development is time-consuming and it is limited to knowledge of known intrusions

V. Borhade (✉) · A. Nayak · R. Dakshayani
Department of Computer Engineering, FCRIT, Mumbai University,
Vashi, Navi Mumbai 400703, India
e-mail: vborhade75@gmail.com

A. Nayak
e-mail: naparnanayak@gmail.com

R. Dakshayani
e-mail: dakshayani.r@fcrit.ac.in

© Springer Nature Singapore Pte Ltd. 2020

H. Vasudevan et al. (eds.), *Advanced Computing Technologies and Applications*, Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-15-3242-9_53

Table 1 Comparative study of different algorithms

	Cross Validation (10 Folds)	
	Correctly classified	Incorrectly classified
Naive bayes	58866 (76.19%)	18245 (23.84%)
Decision tree	76141 (98.51%)	1150 (1.49%)
KNN	76474 (98.94%)	817 (1.06%)
Random forest	76829 (99.40%)	462 (0.60%)

only. Data mining techniques, on the other hand, through supervised and unsupervised learning algorithms have been shown to be effective in identifying and differentiating known and new intrusions from network event records or data [5]. It is, therefore, worthwhile to explore the application of data mining techniques as an effective alternative approach to detect known and potential network intrusions. The proposed method aims to find a suitable machine learning algorithm which can predict the type of network attack with the highest accuracy and then develop a system which uses this algorithm to detect network intrusion. Table 1 also shows a comparative study on raw data in Weka tool based on Naive Bayes, decision tree, K-nearest neighbor, and random forest algorithms. Random forest algorithm shows the highest accuracy compared to all other algorithms; hence, the method proposed in this paper is based on random forest algorithm. The dataset used for model training is NSL-KDD dataset. NSL-KDD is dataset introduced to solve a problem like experimental validation of data, possibility of dropped packets,no identification of exact definition of attack, and duplication of records in the KDDCup99 dataset. The total number of records in NSL-KDD dataset is 1,152,281 which is helpful for training and testing of model. The evaluation of the results of different projects will be consistent and easily comparable. The major reasons to chose the NSL-KDD dataset over the KDD 99 dataset were because there are no duplicate records in the NSL-KDD datasets; therefore, data will not suffer the problem of overfitting and the performance of the learners is not biased by the methods which have better detection rates on the frequent records and the classification rates of distinct machine learning methods vary in a wider range, which makes it possible to have an accurate evaluation of different learning techniques.

2 Literature Survey

Intrusion detection system is important because it helps in identifying suspicious activities in your network and prevent further damage. In recent years, many machine learning algorithms are proposed which helps in the improvement of intrusion detection systems. Chang [2] model with the help of KDD-1999 Cup dataset their model is based on feature extraction with the help of random forest algorithm and train model using support vector machine. They have claimed that their accuracy is improved from 90 to 95% when they selected 14 features instead of total 41 features available

in the dataset. But it also increases their false alarm detection rate by 2 to 3% which is not accepted in case of intrusion detection.

Primartha [4] proposed model for IoT-based systems where they used the random forest algorithm on different datasets like NSL-KDD, UNSW-NB15, and GPRS considerably different number of trees and evaluate results based on accuracy and false alarm rate. In terms of accuracy, the NSL-KDD dataset performs better with an accuracy of 99%, unlike UNSW-NB15 which has an accuracy of 95% and GPRS whose accuracy ranges from 89 to 92% depending on a number of trees.

Ahmad [1] gives a comparison of support vector machine, random forest, and extreme learning algorithms. They use NSL-KDD dataset by considering only numeric features. Machine learning algorithm is applied for full samples, half samples, and 1/4 samples by taking 80% of total available data as training samples and 20% is taken as testing samples. The accuracy of support vector machine is around 99%. For random forest, it shows accuracy from 97 to 97.5% and for extreme learning ranges from 97.5 to 99% depending upon a number of samples taken.

3 Proposed Method

As a preprocessing, data is cleansed with one-hot encoding. All nonnumerical features will be converted into binary format. Furthermore, only training data samples are scaled by removing mean to cluster the samples. Figure 1 describes the block diagram of the proposed method. Feature selected by considering all the samples of training and testing data. The cross-validation method is used which helps to get better model. The description of the proposed method is as follows.

3.1 Data Preprocessing

All features are made numerical using one-hot encoding. This technique will transform each categorical feature with m possible inputs to n binary features, with one active at the time only. The features are standardized by scaling to unit variance to avoid the influence of large values. Each feature will have zero average with standard deviation of one after feature scaling.

3.2 Feature Selection

Eliminate redundant and irrelevant data by selecting a subset of relevant features that fully represent the given problem. ANOVA F-test is useful to determine contribution of each feature with respect to labels; univariate feature selection is done using ANOVA F-test method. Percentile of highest score is determined using percentile method (sklearn.featureselection). Recursive feature elimination (RFE) is applied on the subset to select the most contributing features in the system.

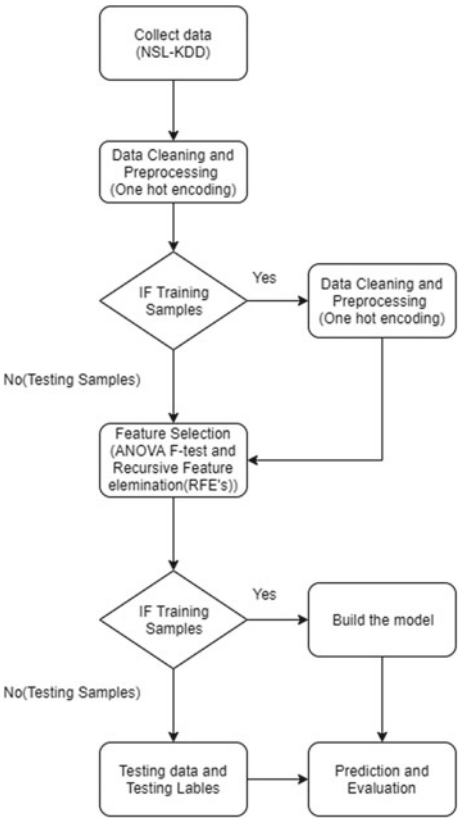

Fig. 1 Proposed method

3.3 Build the Model

A large number of individual decision trees that operate ensemble in random forest. Random forest tree model is built in Python using Colab notebook and Keras. By leaving one-third of the cases from sample, training set for the current tree is drawn. Sampling with replacement method is used. As trees are added to the forest, it is possible to run unbiased estimate of the classification because of out-of-bag (OOB) data. It is also used to get estimates of variable importance. Proximities are computed for each pair of cases as each tree is built and all of the data run down the tree. Proximity is increased by one if two cases occupy the same terminal node. By dividing the number of trees, proximities are normalized at the end of each run. To illuminate low-dimensional views of the data, the replacement of missing data and to locate outlier proximities could be used. Outstanding accuracy among all other existing algorithms can be achieved by this method. The same method could be used on large datasets. Classification of new data from input vector is possible by attaching input vector down each of the trees in the forest. Classification given by each tree is known as "votes" for that class. The classification having the most votes is chosen by forest.

3.4 Prediction and Evaluation (Validation)

Data is divided into two parts for testing and training purposes. Based on test data, prediction model is built. Multiple measures such as accuracy score, f-measure, recall, and confusion matrix are considered based on tenfold cross-validation.

4 Result Analysis

Figures 2 and 3 show the accuracy of cross-validation model using recursive feature elimination and cross-validation (RFECV) graph. A slight fluctuation in accuracy is visible if the total number of features considered to build model is changed.

Figure 4 shows the confusion matrix for the various types of attacks such as normal, probe, and DoS. Tables 2 and 3 show the confusion matrix for probe and DoS attacks, respectively. It is evident from the matrix that actual normal and probe attacks and predicted normal and probe attacks were identified more accurately both

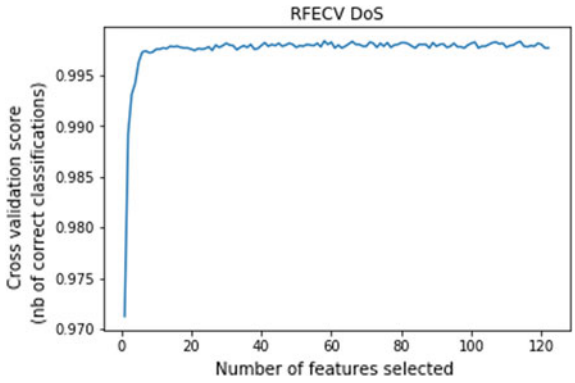

Fig. 2 Feature selection: DoS using REFCV

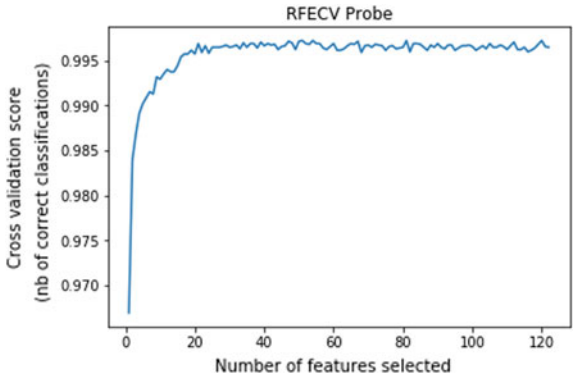

Fig. 3 Feature selection: Probe using REFCV

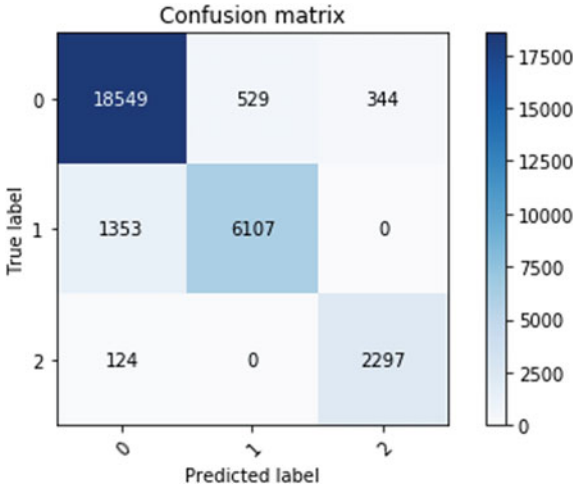

Fig. 4 Confusion matrix of three various attack

Table 2 Confusion matrix of probe attack

Predicted	Actual	
	Normal	Probe
Normal	9367	344
Probe	124	2297

Table 3 Confusion matrix of DoS attack

Predicted	Actual	
	Normal	DoS
Normal	9182	529
DoS	1353	6107

probe and DoS attacks. Accuracy of probe attack is 96.14%. For DoS attack, accuracy is 89%. Since the proposed method considered all features of the available dataset, it is found that accuracy is reduced. In [1], the author has used a very complicated procedure and not used the complete dataset to compute the accuracy.

5 Conclusion and Future Work

The proposed method reviewed the basis of intrusion detection system and application of machine learning algorithms like random forest, SVM, and Naive Bayes in intrusion detection systems. Algorithm is built for network-based intrusion detection system.

In future, it is expected that current model is further developed to detect various other types of attacks.

References

1. Ahmad I, Basheri M, Iqbal MJ, Rahim A (2018) Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 6:33789–33795
2. Chang Y, Li W, Yang Z (2017) Network intrusion detection based on random forest and support vector machine. In: 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), vol 1. IEEE, pp 635–638
3. Kshirsagar VK, Tidke SM, Vishnu S (2019) Intrusion detection system using genetic algorithm and data mining: an overview
4. Primartha R, Tama BA (2017) Anomaly detection using random forest: a performance revisited. In: 2017 international conference on data and software engineering (ICoDSE). IEEE, pp 1–6
5. Shah SAR, Issac B (2018) Performance comparison of intrusion detection systems and application of machine learning to snort system. Future Gener Comput Syst 80:157–170. https://doi.org/10.1016/j.future.2017.10.016, http://www.sciencedirect.com/science/article/pii/S0167739X17323178