Journal of Scientific & Industrial Research
Vol. 79, May 2020, pp. 365-368

Predictor Selection and Attack Classification using Random Forest for Intrusion Detection

Chandramohan Ambikavathi[1]* and Srinivasa Krishna Srivatsa[2]

[1]Faculty of S&H (Computer Applications), Sathyabama University, Chennai, Tamilnadu, India
[2]MIT, Anna University, Chennai, Tamilnadu, India

Received 22 April 2019; revised 26 November 2019; accepted 28 March 2020

Decision making for intrusion detection is critical in a distributed environment such as cloud or grid computing due to its' dynamic nature. Wrong or delayed decisions lead to astonishing problems. So that decision making phase is enhanced by means of selecting relevant features for prediction and trained to classify attacks. Initially the common valued features for both normal and attack behavior are removed. The random forest algorithm is used for analyzing the predictors' importance for intrusion detection. Then random forest algorithm works with the reduced and selected predictors to classify the normal user and attack behavior. Finally the classifications are used to detect intruders. Experiments are conducted and proved that classifier performance can be improved in terms of accuracy, efficiency and detection rate using random forest.

Keywords: Classification, Intrusion detection, Pcap file, Random forest

Introduction

Classification methods[1–7] for intrusion detection produce a solution for categorizing normal or intrusive behavior in a better way than statistical learning approaches. Among classification methods, random forest is the best classifier[8] and feature selector.[9,10] However, in a distributed environment, random forest lack in efficiency due to the need of more processing time for training. So that, pcap file features are filtered to segregate the features relevant for intrusion detection. In order to improve accuracy and detection rate the best M_{try} value is chosen.

Materials and Methods

Random Forest Algorithm

Random Forest is a supervised tree based classification model. It is an ensemble classification algorithm which constructs a set of decision trees and selects the best estimate. It creates multiple trees, a tree for each feature in the training data. The results of all decision trees are aggregated and the best estimator is taken into account. Randomness is another feature of random forest algorithm which is implemented by selecting a random set of best features for estimation. Among available features, the importance of each feature for classification can be calculated using VarImp() function of R. The out-of-bag (OOB) error for each M_{try} value is recorded and averaged over the forest. M_{try} is a number used for splitting each node in the tree. Random forest has been effectively used for intrusion detection in solo[8, 9, 11] or accompanied[10,12] with other classifiers.

Proposed model

According to intrusion detection, packets transmitted between VMs through network are the footprints of intruder. The present work analyses pcap files for preprocessing them and constructs classifiers using a supervised learning technique-Random Forest. The architecture of the proposed system is shown in Fig. 1.

Generally a pcap file contains vast data such as frame number, timestamp, number of bytes, protocol, source ip, destination ip, source port, destination port, error information etc. pcap files from various IDS (Intrusion Detection System) sensors of cloud network are aggregated which is an entry point for intrusion detection activity. It is done in hypervisor which is the right place for aggregation since the control of IDS is with the cloud service provider. Also monitoring both internal and external attacks are easier in hypervisor. Each VM's pcap files are collected as raw events, converted into .csv files and moved to intrusion detection system. If all pcap file fields are used as predictors for intrusion detection, it will be time consuming in training, and expensive in resource consumption which leads to poor detection

*Author for Correspondence
E-mail: ambika_vathi@yahoo.co.in

rate. Thus, pre-processing is needed to avoid these problems. So the features which have less importance for intrusion detection are removed and a reduced feature dataset is constructed. Then an optimal set with relevant features is prepared after identifying each features' importance. Then the classifier works on the pre-processed data with reduced features to construct classification model. The work flow of the proposed work is dictated as follows:

Algorithm:

1. Read the dataset, pcap file (in .CSV format) with feature set S= $\{f_1,f_2,\ldots,f_n\}$ with 'n' features.
2. For each f_i in S do
   a. Delete common valued features and construct reduced set S_1= $\{f_1,f_2,\ldots,f_m\}$ where m<n
   b. For each f_i in S_1
      i. Find feature importance using VarImp()
      ii. If VarImp() = TRUE then
      Construct optimal set S_2= $\{f_1,f_2,\ldots,f_l\}$
      where l<m
3. Partition the optimal set S_2 in the ratio 70:30 using CreateDataPartition() function.
4. Create two data frames 'training' and 'testing' with 70% and 30% partitions respectively.
5. Convert the categorical fields such as 'Protocol', 'Flag count' etc from numerical to categorical type variables.
6. Choose the best M_{try} value using tuneRF() function.
7. Train the 'training' dataset with the randomforest() classifier method.
8. Predict the 'testing' dataset using the trained model.

The rules obtained from analysis phase are formatted as condition and action taken to generate alerts. By means of Intrusion Detection Message Exchange Format (IDMEF) the generated alerts are sent to all IDS sensors in the environment for further protecting from intrusion.

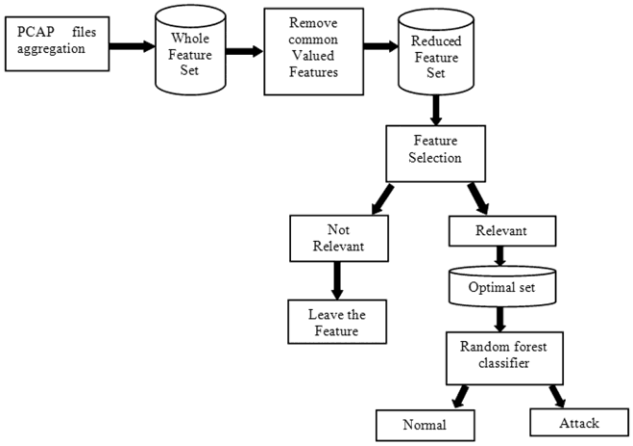


Fig.1 — Architecture of Proposed system

Implementation

A .csv file is created with attack and normal samples from the CICIDS2017 dataset.[13] This .csv file covers all attack samples: Bot, Brute force, DDoS, DoS, FTP, Infiltration, Portscan, SQL injection, SSH & XSS and normal (Nor) with 84 features. Some of the features have common values for all types of attacks and normal samples. So they are removed from the .csv file and a reduced feature set of 71 fields is obtained. Among these fields we have selected the fields which are appropriate for intrusion detection using random forest's variable importance function VarImp() to obtain the optimal set. The optimal set is split into a training set (70%) and a test set (30%). Then randomForest() function is invoked to produce a classifier with the parameters: optimal set, training partition, 20 as the number of trees and M_{try} as 7. This trained classifier is used for prediction of intruders on the test set.

Results and Discussion

Table 1 shows the list of optimal features along with the metric 'MeanDecreaseGini'. 16 features among 71 are selected by the metric. The remaining features are exempted because of negative values. Table 2a and 2b shows the confusion matrices of the random forest classifier before optimality and after optimality of predictors respectively. The OOB error estimate is 7.27% before selecting optimal predictors. The same OOB error estimate is 2.66% for optimal

Table 1 — Optimal predictors

S.No	Predictor name	MeanDecreaseGini
1.	Source IP	33.02
2.	Source port	56.89
3.	Destination port	58.65
4.	Total Backward packets	12.94
5.	Total length of Forward packets	19.01
6.	Bwd packet length max	22.86
7.	Bwd packet length std	11.93
8.	Flow packets per second	9.05
9.	Flow IAT std	6.72
10.	Fwd IAT mean	21.14
11.	Bwd IAT std	5.04
12.	Fwd header length	16.07
13.	Bwd packets per second	23.72
14.	Init_Win_bytes_forward	21.16
15.	Idle.max	0.45
16.	Idle.min	0.22

Table 2a — Confusion Matrix-Random forest (before predictor selection)

Confusion Matrix (OOB Estimate of error rate: 7.27%)	Bot	Brute force	DDoS	DoS	FTP	Infiltration	Nor	Portscan	SQL injection	SSH	XSS	Class.error
Bot	13	0	0	1	0	0	1	0	0	0	0	0.13
Brute force	0	8	0	1	0	0	0	0	3	0	5	0.52
DDoS	0	0	5	2	0	0	0	0	0	0	0	0.28
DoS	0	1	0	367	1	0	0	0	0	0	0	0.00
FTP	0	1	0	0	15	0	0	0	0	0	0	0.06
Infiltration	1	0	0	0	0	17	1	0	0	0	0	0.10
Nor	1	0	0	0	0	2	50	0	0	0	0	0.05
Portscan	0	1	0	0	0	0	0	11	1	1	1	0.26
SQL injection	0	2	0	3	0	0	0	0	8	0	3	0.50
SSH	0	0	0	0	0	0	0	0	1	17	0	0.55
XSS	0	1	0	2	1	0	0	0	1	0	12	0.36

Table 2b — Confusion Matrix – Random forest (after predictor selection)

Confusion Matrix (OOB Estimate of error rate: 2.66%)	Bot	Brute force	DDoS	DoS	FTP	Infiltration	Nor	Portscan	SQL injection	SSH	XSS	Class.error
Bot	15	0	0	0	0	0	1	0	0	0	0	0.00
Brute force	0	15	0	0	0	0	0	0	1	0	0	0.11
DDoS	0	0	5	2	0	0	0	0	0	0	0	0.28
DoS	0	0	0	369	0	0	0	0	0	0	0	0.00
FTP	0	1	0	0	15	0	0	0	0	0	0	0.06
Infiltration	0	0	0	0	0	18	1	0	0	0	0	0.05
Nor	1	0	0	0	0	1	51	0	0	0	0	0.03
Portscan	0	1	0	1	0	0	0	13	0	0	0	0.13
SQL injection	0	0	0	0	0	0	0	2	14	0	0	0.12
SSH	0	0	0	0	0	0	0	0	0	18	0	0.00
XSS	0	1	0	2	0	0	0	0	0	0	16	0.15

set. Bot, DoS and SSH attacks are classified with 100% accuracy (0.0 class error rate as in Table 2b). Infiltration attack has an error rate of 0.1. Also the other attacks have less error rate in optimal set compared to whole set. M_{try} parameter of randomForest() function is selected using tuneRF() method, in order to get the best M_{try} value. The relationship between M_{try} value and OOB error rate is shown in Fig. 2. The function depicts that the error rate is low when M_{try} value is 6 or 7. The performance evaluation of the classification algorithm is done by the three metrics accuracy, precision and recall. Accuracy (1) is a measure that indicates the overall persistence of the system. It is a proposition of correctly detected events to all events (normal and attack) occurred. Precision (2) is a measure of positive predictions or detection rate. It gives the proposition of attacks correctly detected to all predicted attacks. Recall (3) is a measure of sensitivity or true positive rate. It is the proposition of attacks correctly detected

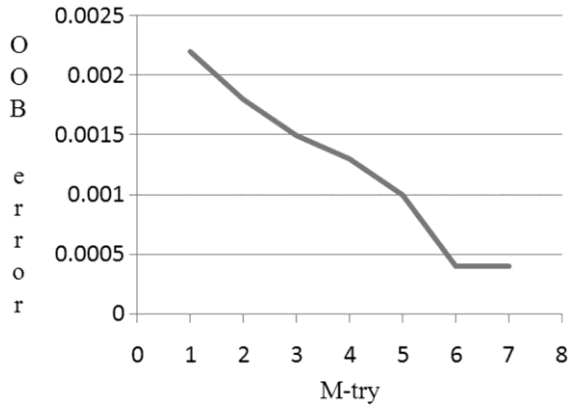
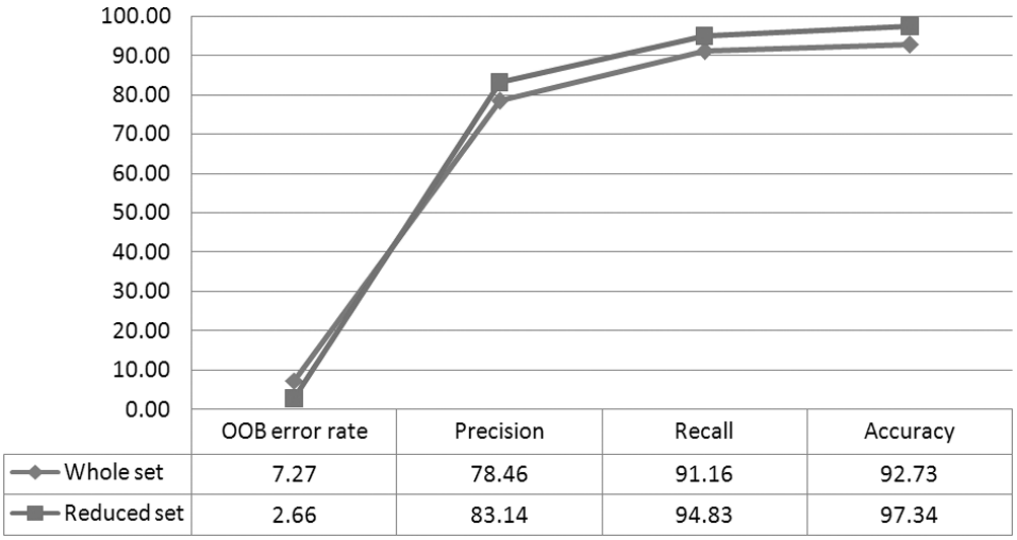


Fig. 2 — M_{try} parameter selection

to all attacks. The decrease in OOB estimate of error rate and increase in all performance metrics for optimized set is shown in Fig. 3. The accuracy rate gained 4.61% more after feature selection.

$$Accuracy = (TP + TN)/(TP + FP + TN + FN) \quad \ldots (1)$$



	OOB error rate	Precision	Recall	Accuracy
Whole set	7.27	78.46	91.16	92.73
Reduced set	2.66	83.14	94.83	97.34

Fig. 3 — Comparison of all features and optimal features

$$Precision = TP/(TP + FP) \quad \ldots (2)$$

$$Recall = TP/(TP + FN) \quad \ldots (3)$$

The drawbacks of classification methods are complexity, time and resource consumption, which are resolved using proposed method. The traditional IDS is tuned to work in a distributed environment.

Conclusions

Efficient method for intrusion detection is presented in this paper. Generic features of pcap file consume more time which affect efficiency and generate more false alarms which affect accuracy. So intrusion characteristic specific features are taken into account for detection. Instead of using all predictors for intrusion detection, the optimized 16 predictors are used for predicting intruders. It is done by considering the features' importance. The large volume of pcap files are moved to preprocessor for extracting needed features and classification algorithm is used for generating classifiers. It is useful to monitor the network and prevent it from further contiguous attacks. Along with the accuracy enhancement, efficiency is also improved in terms of resource and time consumption.

References

1 Aburomman A A & Reaz M B I, A novel SVM-KNN-PSO ensemble method for intrusion detection system, *Applied Soft Comput*, **38** (2016) 360–372.

2 Aslahi-Shahri B, Rahmani R, Chizari M, Maralani A, Eslami M, Golkar M & Ebrahimi A, A hybrid method consisting of GA and SVM for intrusion detection system, *Neural Comput Appl*, **27** (2016) 1669–1676.

3 Feng W, Zhang Q, Hu G & Huang J X, Mining network data for intrusion detection through combining SVMS with ant colony networks, *Future Gener Comput Syst*, **37** (2014) 127–140.

4 Kuang F, Xu W & Zhang S, A novel hybrid KPCA and SVM with GA model for intrusion detection, *Applied Soft Comput*, **37** (2014) 178–184.

5 Kuang F, Zhang S, Jin Z & Xu W, A novel SVM by combining kernel principal component analysis and improved chaotic particle swarm optimization for intrusion detection, *Soft Comput*, **19** (2015) 1187–1199.

6 Teng S, Wu N, Zhu H, Teng L & Zhang W, SVM-DT-based adaptive and collaborative intrusion detection, *IEEE/CAA J Automat Sinica*, **5** (2018) 108–118.

7 Golmah, V, An efficient hybrid intrusion detection system based on C5. 0 and SVM, *Int J Database Theory Appl*, **7** (2014) 59–70.

8 Farnaaz N & Jabbar M, Random forest modeling for network intrusion detection system, *Procedia Comput Sci*, **89** (2016) 213–217.

9 Hasan M A M, Nasser M, Ahmad S & Molla K I, Feature selection for intrusion detection using random forest, *J Inf secur*, **7** (2016) 129.

10 Dhanalakshmi R & Khaire U M, Feature selection and classification of microarray data for cancer prediction using MapReduce implementation of random forest algorithm, *J Sci Ind Res*, **78** (2019) 158–161.

11 Singh K, Guntuku S C, Thakur A & Hota C, Big data analytics framework for peer-to-peer botnet detection using random forests, *Inf Sci*, **278** (2014) 488– 497.

12 Hasan M A M, Nasser M, Pal B & Ahmad S, Support vector machine and random forest modeling for intrusion detection system, *J Intell Learn Syst Appl*, **6** (2014) 45.

13 Iman S, Arash H L & Ali A G, Toward generating a new intrusion detection dataset and intrusion traffic characterization, *4th Int Conf on Inf Sys Sec and Priv* (Portugal) January 2018.