Discovering Anomalous Rules In Firewall Logs Using Data Mining And Machine Learning Classifiers

Hajar Esmaeil Qasem As-Suhabni, Dr. S.D.Khamitkar

Abstract: Firewall is the main component of network security that monitors the in-out network packets according to a predetermined security rule. The security rules in companies and institutions are implemented as Firewall rules. Firewall rules in large networks have proven to be sensitive and error-prone. In addition, any improper management of these rules will cause anomalies. The aim of this study is using data mining to analyse and detect anomalies in Firewall logs. A hybrid model based on data mining and machine learning is proposed for analyzing and discovering anomalies from firewall rules. The proposed methods have shown a more superior and precise performance in terms of anomaly detection accuracy and as a result, enabling network administrators to update and optimize Firewall policy rules.

Index Terms: Machine Learning, Data Mining, log analysis, Firewall, Firewall Logs, Police Security Rules, Anomalies.

1. INTRODUCTION

All enterprises and Organizations use Firewalls to enforce their security policy. Firewall acts as a router that connects different network zones. Firewall is the main component of network security that monitors the in-out network packets according to a predetermined security rules. With the advent of internet technology, day to day work has been shifted over the internet and thus network security has become a big issue and focus in the world. For that, there is a challenge to the traditional security solutions such as Firewall and VPN to detect security breach against attacks [1]. Most network devices, such as Firewalls, generate and record vast amounts of data. This network data could become an important source for analysis, and plays a big role in network security [2]. Firewall processes high amount of internet traffic, logs suspicious activities and events and controls network access by allowing or denying network traffic based on a pre-defined rules. However, the inappropriate configuration of Firewalls could enable unauthorized users or malicious content to pass through the network [3]. Policy rules are written and managed to filter out any malicious traffic passing through the network [4]. Because of the continuous growth of security threats, writing and managing Firewall rules is a sophisticated and complex task. In addition, any inappropriate management of these rules will cause anomalies [3]. An Anomaly or outlier is an abnormal behavior in firewall logs; it has drawn a great attention by the analysts and researchers and is considered as the major feature for anomaly detection [5]. Discovering anomalies in a dataset is known as anomaly detection. Data mining has a big rule in anomaly detection, because this process involves the comparing between unexpected behaviors and expected behavior, and effectively can be used in this process [6].

- *Hajar Esmaeil As-Suhbani, Research Scholar, SRTM U., Nanded.*
- *Prof. S.D.Khamitkar, Professor & Head, SRTM U., Nanded.*

2 LITERATURE REVIEW

Decent studies have been carried out for analysing Firewall logs and detection of anomalies in Firewall rules, which varies to each other in implementation, performance and accuracy. In addition, many researchers are working on WEKA tool and they proved a higher degree of accuracy can be achieved using this tool in data mining and machine learning. Golnabi et al. [3], proposed an approach to detect the Firewall rules anomalies by using (ARM) and (FRG) algorithms based on data mining techniques for generating high level policy. Firstly, they processed 33,172 logs records extracted from Linux operating system with only the seven major attributes. They concluded that that data mining is an effective and a very practical method in Firewall log analysis. In addition, Firewall policy dominant and decaying rules have been detected. Saboori et al. [7] proposed the Apriori Algorithm to build a model for detecting novel anomaly attacks and generate real-time rules for the Firewall policy by extracting the correlation relationships among large datasets. Ucar, E. & Ozhan, E [8], proposed a model to detect anomalies in Firewall rules based on machine learning. They used Weka-Parallel-3-2-3 version of the WEKA Server to speed-up the whole process. In this experiment, around 5,000,000 data taken from a Firewall are automatically analyzed. Moreover, in this experiment KNN algorithm has shown the best performance scale in terms of all the performance metrics. Breier and Brani [9], proposed model using Apache Hadoop framework based on data mining techniques to generate the firewall rules dynamically from certain patterns for detecting anomalies. Their model has showed a better performance and result in terms of anomaly detection than traditional approaches, such as Apriori and FP-Growth algorithms. A tool implemented in Java named “Firewall Policy Advisor” by Al-Shaer et al. [10] for analyzing Firewall rules and detecting the anomalies in centralized and distributed Firewalls. In this study, a hybrid model based on data mining and some of machine learning classifiers is proposed for analyzing and discovering anomalies in firewall rules.

3 PROPOSED METHODOLOGY

The proposed model discovers the anomalies in the Firewall logs to increase the network security. For this purpose, the Firewall logs have been analysed through data mining and machine learning classifiers to produce a model to for

discovering anomalies in Firewall logs repository. The outlook of the proposed model was designed as shown in Figure 1. The proposed model consists of multiple steps through which data processed to detect the anomalies in Firewall logs.

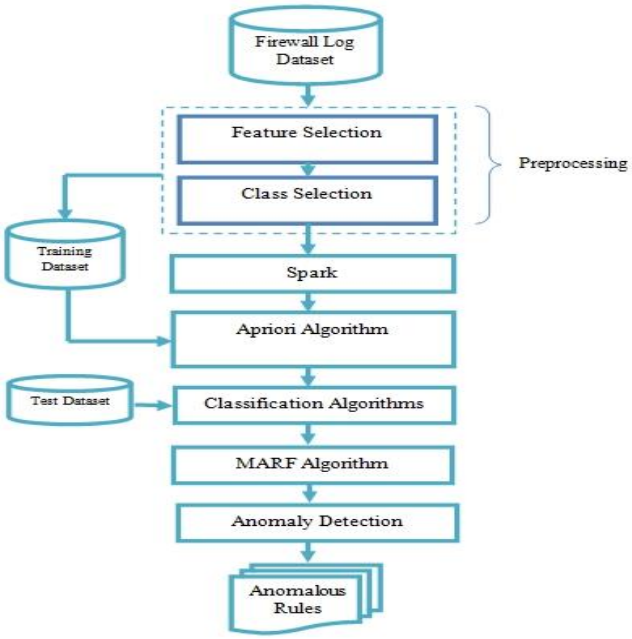


Figure 1 *The general outlook of the model*

This experiment was conducted on Windows 7 with the help of open-source Apache Spark that designed for Big Data to speed up the processing and the prediction time. Furthermore, we have used the R program and WEKA simulation tool. Firstly, the training data is divided to 6 pieces 20-50-100-200-300-500 thousands records. In the rules generating step which was implemented in our previous work [11] we used Apriori algorithm. In addition, for rule generating and aggregating step we proposed MARF Algorithm. Each of the training data was analysed through the use of 4 of the most currently used classifiers (Naive Bayes, kNN, OneR, J48). The algorithm among these four classifiers which yield the best results will be chosen.

4 EXPERIMENT ANALYSIS

The proposed model consists of multiple steps through which data processed to detect the anomalies in firewall logs. These steps are described as shown in Figure 2. In this study, we propose an approach that presents a hybrid model for anomaly detection in Firewall logs based on Big Data. To discover these anomalous rules, a new hybrid approach using data mining based on Apriori algorithm and MARF aggregation algorithm is proposed. Also, we used 4 of the most currently used classifiers of classification algorithms which improve the overall accuracy of the anomaly detection.

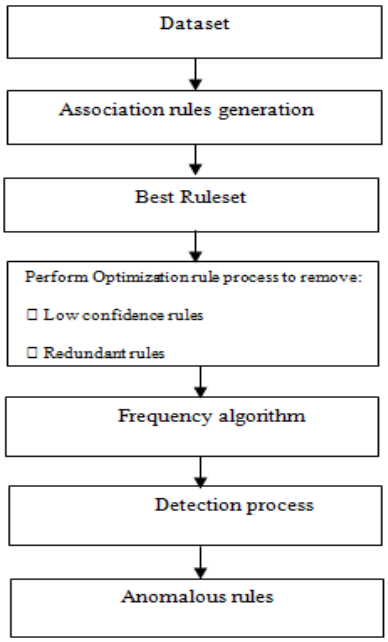


Figure 2 *Anomaly Detection Process*

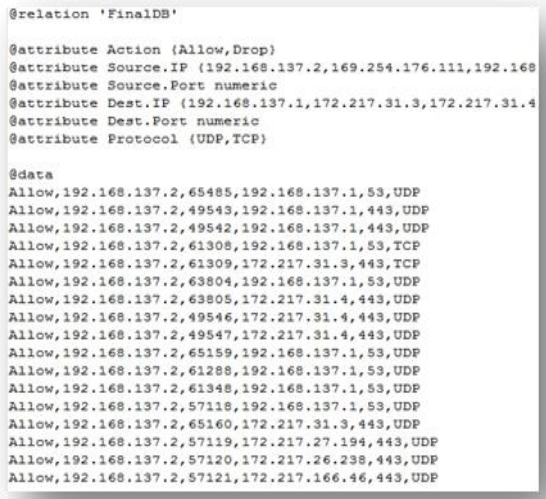
In this experiment, the training data is divided to 6 pieces 20-50-100-200-300-500 thousands records. We have compared the performance of the model based on four classification algorithms, NaiveBayes, KNN, One'R, and J48. The most convenient algorithm that provides the best performance will be used for the anomaly detection. In our previous study [11], for Apriori algorithm 0.5, 0.5 were the values of min-support and min-confidence respectively, and the number of association rules was 10 best rules. After extract the best rules of firewall log dataset using Apriori algorithm, the MARF algorithm have been used to extract the primitive rules with their frequencies. The MARF algorithm processes and reads each record in the firewall log file, then, extracts the attributes and counts its occurrence. Finally, MARF algorithm outputs the count for each unique combination of these attributes. The frequency of each discovered primitive rule in the dataset is summed up to generate the aggregated rules. Thus the initial step in the extracting the features of a packet for each log line is to generate its related unique/primitive firewall rules.

4.1 Data Collection and Preprocessing

The dataset used in our study is extracted from a firewall in a lab from Computer Science Department using Snort IDS [12] and TWIDS [13]. This dataset with initially 13 firewall rules is explained in details in our previous study [14]. In firewall log file each record indicates the information which includes in the packet header of the packet. We have chosen to work with only the 6 major attributes with the following format:

"<Action >, <Source IP>, <Source port>, <Destination IP>, <Destination port>, and <Protocol>"

Because the log data could not be used and analyzed in the form as it is stored in the log files, the preprocessing step must be implemented before any process. Preprocessing stage involves loading, cleaning and manipulating the data into a form that you can work with. The total numbers of records included in the training dataset were 520,000 instances. In addition, the Action attribute was chosen as the class attribute. Figure 3 shows a sample of an ARFF file for the Firewall logs dataset after using WEKA [15] tool.



```
@relation 'FinalDB'

@attribute Action {Allow,Drop}
@attribute Source.IP {192.168.137.2,169.254.176.111,192.168
@attribute Source.Port numeric
@attribute Dest.IP {192.168.137.1,172.217.31.3,172.217.31.4
@attribute Dest.Port numeric
@attribute Protocol {UDP,TCP}

@data
Allow,192.168.137.2,65485,192.168.137.1,53,UDP
Allow,192.168.137.2,49543,192.168.137.1,443,UDP
Allow,192.168.137.2,49542,192.168.137.1,443,UDP
Allow,192.168.137.2,61308,192.168.137.1,53,TCP
Allow,192.168.137.2,61309,172.217.31.3,443,TCP
Allow,192.168.137.2,63804,192.168.137.1,53,UDP
Allow,192.168.137.2,63805,172.217.31.4,443,UDP
Allow,192.168.137.2,49546,172.217.31.4,443,UDP
Allow,192.168.137.2,49547,172.217.31.4,443,UDP
Allow,192.168.137.2,65159,192.168.137.1,53,UDP
Allow,192.168.137.2,61288,192.168.137.1,53,UDP
Allow,192.168.137.2,61348,192.168.137.1,53,UDP
Allow,192.168.137.2,57118,192.168.137.1,53,UDP
Allow,192.168.137.2,65160,172.217.31.3,443,UDP
Allow,192.168.137.2,57119,172.217.27.194,443,UDP
Allow,192.168.137.2,57120,172.217.26.238,443,UDP
Allow,192.168.137.2,57121,172.217.166.46,443,UDP
```

Figure 3 *A sample of an ARRF file of FinalDB firewall logs Dataset*

4.2 Apache Spark

In the next phase the firewall logs dataset is analysed and the features were inserted to machine learning classifiers including Naive Bayes, KNN, One R and J48 using Spark in Weka tool. In addition, we compare the classification performance of these algorithms in terms of measurement metrics including Accuracy, F-measure and ROC values. In general, WEKA tool only supports sequential single-node execution which considered as a major disadvantage because it take a lot of time to process a large volume of data. The major job of spark is to speed up batch processing. Spark [16] is based on the map-reduce algorithm to achieve distributed computing. Spark is a distributed framework which provides in-memory computation that permits iterative jobs to be processed 10 to 1000 times faster than MapReduce. We have used the dataset in ".CSV" format in the directory of the distributed Spark as input. The dataset is randomly shuffled and split the data into 4 partitions. In addition, the ARFF file header with added metadata attributes is computed using all the computer CPU cores. Therefore, Spark is much better to applied and used for data mining and machine learning that require iterative map reduce.

4.3 Apriori Algorithm

Apriori is considered as the most popular algorithm for association rules mining (ARM). It is one of the best association rule algorithms which proposed by Srikant and Agrawal [17]. Initially, the minimum support and the dataset are the two inputs of the Apriori algorithm, and the largest item-sets and the best rules drawn from dataset is the output. The formulas of the Support and the confidence could be calculated using Equation (1) and Equation (2) respectively. Where, the term |D| represents the total number of transactions in the database D.

$$support(X \cup Y) = (|(X \cup Y)|)/(|D|) \quad (1)$$

$$confidence(X \rightarrow Y) = (support(X \cup Y))/(support(X)) \quad (2)$$

Apriori Algorithm:

Input:

- D= Training dataset, the minimum support count threshold.

Output:

- frequent itemsets in the database

steps:

- C_k: Candidate itemset of size k
- L_k : frequent itemset of size k

- L_1 = {frequent items};
- for (k = 1; L k != Ø; k++) do begin
  - C_{k+1} = candidates generated from L k;
  - for each transaction t in database do
    - increment the count of all candidates in C_{k+1} that are contained in t
  - L_{k+1} = candidates in C_{k+1} with min_support
  - end
- return $\cup_k L_k$;

Figure 4 Apriori Algorithm

4.4 Mining Aggregating Rule by Frequency (MARF) Algorithm

The main idea of the proposed MARF algorithm is to generate unique rules. Here, the input of the MARF algorithm is the output best rules of the Apriori algorithm and the output is the count for each unique aggregation of these attributes. The MARF algorithm processes and reads each record in the log file, then, extracts the attributes and counts its occurrence. The frequency of each extracted primitive rule in the dataset is summed up to generate the aggregated rules.

MARF Algorithm:

Input:

- Collection of instances of Firewall Log Dataset after (ARM) processing.

Output:

- All instances of unique rules with their Frequencies.

Steps:

1. Record#← 0
2. **for each** *record ∈ Dataset* **do**
3. **if** *FirewallRule[i]=FirewallRule[j]*
4. Increment the frequency of record
5. else *repeat steps until no new rule is discovered*
6. End for

Figure 5 MARF Algorithm

4.5 Naive Bayes Algorithm

Naive Bayes is one of the simplest machine learning techniques based on applying Bayes theorem. It is widely used for a probabilistic classification. It is a very simple and straightforward classification algorithm because the idea of this method is really simple. Bayes' theorem solves the problem often encountered in real life. Knowing the probability of a certain condition, how to get the probability of the exchange of two events, that is, how to find P when P(A|B) is known (B|A). Here to explain what is the conditional probability.

$$p(A/B) \quad (3)$$

$$p(A/B)=P(AB)/P(B) \quad (4)$$

$$p(B/A)=P(A/B)\ P(B)/P(A) \quad (5)$$

We can easily get P(A|B) directly, P(B|A) is difficult to draw directly, but we More concerned with P(B|A). Bayes' theorem is the way for us to get P(B|A) from P(A|B). The mathematical representation of this theorem is as shown in Equation (5). Where, A and B are events and P (B) ≠0. In addition, the quality of the classifier is related to the classifier construction method, the features and characteristics of the data to be classified, and the number of training samples [18].

4.6 IBK (KNN) Algorithm

The K-Nearest Neighbour (KNN) algorithm is a simplest classifier and easy to implement algorithm for solving both classification and regression problems. It often produces competitive results and has more advantages over several other data mining methods. In general, KNN classification algorithm is based on the learning by comparing the training lines with the given attributes [19]. KNN is learnt by comparing a specific test instance with a set of training instances that are analogues to it. Therefore, it could be described as the learning by similarity. The classification by KNN based on the class of their closest neighbours, where more than one neighbour is taken into consideration [21]. KNN classifier is named IBk in WEKA tool [8]. Basically, the KNN algorithm uses Euclidean distance. The Euclidian distance between two data tuples or two points like x and y is given by the formula shown in Equation (6).

$$dist(x,y)= \sqrt{\sum_{i=1}^{n}(x_i - y_i} \quad (6)$$

4.7 One R Algorithm

One R algorithm is a very effective and a straightforward classification algorithm. It learns a rule which predicts nominal and numerical class value. The quality of the classification rules is measured by the correctness rate and coverage [20]. One R creates a rule output every step for a given data.

One R Algorithm:

For each attribute A,

a) For each value VA of the attribute, make a rule as follows:

Count how often each class appears

to Find the most frequent class Cf

Create a rule when A=VA

Class attribute value = Cf

b) Calculate the error rate of all rules

Chose the rule with the smallest error rate.

Figure 6 *One R Algorithm*

4.7 J48 Algorithm

J48 is a simple statistical classification method. It based on creating a decision tree from the training data using the concept of information entropy. The greedy top-down construction technique is applied to produce the decision trees for classification.

J48 Algorithm:

Input:

- Training Dataset = D

Output:

- Decision Tree = T

Steps:

- DTBUILD (*D)
- T=φ;
- T= Create root node and label with splitting attribute;
- T= Add arc to root node for each split predicate and label;
- For each arc D= Database created by applying splitting predicate to D;
  - If stopping point reached for this path, then T'= create leaf node and label with appropriate class;
  - Else T'= DTBUILD(D);
  - T= add T' to arc;

Figure 6 *J48 Algorithm*

Internal nodes identify the different features; the branches between nodes provide us with the possible value these attributes can have in the experiential samples, while the terminal nodes provide us with the final value of the dependent variable. J48 classification algorithm is an extension of ID3. The traditional and general characteristics of J48 are accounting for missing values, derivation of rules, continuous attribute value ranges, decision trees pruning, etc. In the WEKA tool, J48 algorithm is C4.5 algorithm [21].

5 EXPERIMENTAL RESULTS

The results using the four proposed classification algorithms are presented in Table 1. To evaluate the proposed model it compares some of classification machine learning algorithms based on the Performance Metrics such as: Accuracy% and Kappa statistic values which extracted and saved in tables. These values will be used to determine the best and most convenient algorithm. The robustness or correctness of our proposed model is measured using different Performance Metrics.

The proposed approach compares 4 famous machine learning algorithms namely:

- Native Bayes
- KNN
- One R
- J48 decision tree.

Table 1 *Classification algorithms analysis results*

Performance Metrics	Accuracy %	Incorrectly Classified Instances %	Kappa statistic	Root mean squared error	Time (sec)
Algorithm / Data(Thousands)	Naive Bayes				
20	95.126	4.874	0.8361	0.0991	0.001
50	96.001	3.999	0.8567	0.0985	0.002

100	98.9002	1.0998	0.9219	0.0986	0.03
200	99.1179	0.8821	0.9367	0.0917	0.06
300	99.2564	0.7436	0.9462	0.0862	0.09
500	99.2582	0.7418	0.9553	0.0857	0.12
	IBK (KNN)				
20	98.5355	1.4645	0.9173	0.0451	0.01
50	98.8891	1.1109	0.9231	0.0492	0.01
100	99.8625	0.1375	0.9897	0.0345	0.02
200	99.8662	0.1338	0.9899	0.0338	0.03
300	99.8731	0.1269	0.9901	0.0331	0.04
500	99.8812	0.1188	0.9921	0.0241	0.05
	One R				
20	96.981	3.019	0.8774	0.081	0.01
50	97.483	2.517	0.8991	0.072	0.02
100	99.5238	0.4762	0.9536	0.069	0.03
200	99.5174	0.4826	0.9629	0.0695	0.04
300	99.5201	0.4799	0.9635	0.0697	0.05
500	99.5111	0.4891	0.9619	0.0699	0.07
	J48				
20	97.267	2.733	0.8914	0.0661	0.01
50	98.336	1.664	0.8992	0.0625	0.02
100	99.5251	0.4749	0.9593	0.0515	0.03
200	99.6600	0.34	0.9699	0.0354	0.10
300	99.6991	0.3009	0.9801	0.0341	0.44
500	99.7736	0.2264	0.9812	0.0344	0.20

According to the obtained results listed in Table 1, the accuracy of the proposed model is around 100%. Therefore, we can conclude that the result of our integrated model is perfect in terms of accuracy based on four Machine Learning classification algorithms.

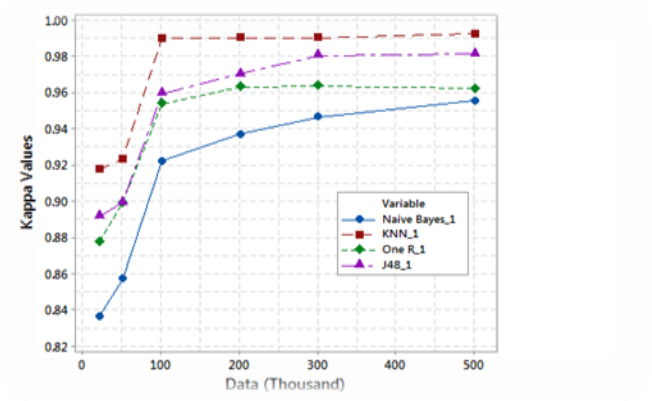


Figure 8 *Kappa values obtained for all classification algorithms with different volumes of data*

As a result, it shown that the firewall logs can be efficiently analysed with machine learning methods. In addition, it is shown that the 500,000 records training data would be convenient for our experiment. As the amount of training data increases, the overall error such as Root mean squared error decreases. On the other hand, Correctly Classified Instances% values are observed. This value is accepted by the classifier and shows the amount of training data that is subject to classification.

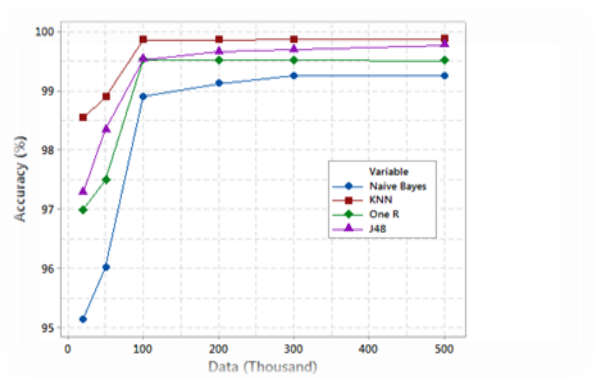


Figure 9 *Classifier accuracy % for all classification algorithms with different volumes of data*

According to Figure (8) and Figure (9) KNN algorithm has the highest Kappa values and highest accuracy. However, the performance of the remaining classification algorithms is not to be underestimated. Another result of this analysis is the Navie Byes algorithm has the lowest performance in terms of Correctly Classified Instances, Kappa and Root Mean Squared Error. Root Mean Squared Error is known as the square root of the mean error. Referring to Figure 10, the Root Mean Squared Error for the KNN algorithm value is the lowest.

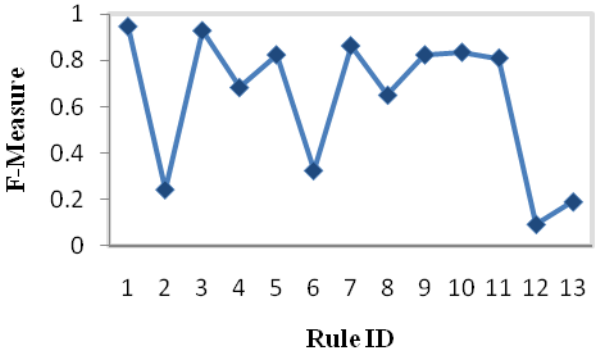


Figure 10 *KNN Firewall rules Corresponding values of F-measure*

Hence, KNN algorithm has showed better successful analysis results than the other 4 classification algorithms. Therefore, the Firewall rules were analyzed according to this algorithm. The F-measure values of KNN algorithm shown in Figure 10 are taken into consideration. Clearly as Table 2 shows, it can be noticed that firewall rules 2, 6, 12, 13 are anomalous rules. This is because they may be redundant to other rules or obsolete rules. These anomalous rules should be retailored and checked again to ensure that Firewall works efficiently and to obtain most convenient, corrected and anomaly-free Firewall rules.

Table 2 *Performance Measures of KNN classifier for the aggregated Firewall rules*

Rule ID	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
1	0.807	0.263	0.532	0.807	0.947	0.491	0.817	0.555
2	0.353	0.07	0.387	0.511	0.244	0.389	0.759	0.331
3	0.573	0.004	0.545	0.853	0.929	0.432	0.908	0.373

4	0.393	0.011	0.629	0.393	0.684	0.478	0.857	0.454
5	0.625	0.003	0.625	0.625	0.825	0.622	0.885	0.358
6	0.269	0.009	0.412	0.269	0.326	0.321	0.904	0.315
7	0.444	0.008	0.308	0.444	0.864	0.664	0.992	0.546
8	1	0.002	0.6	1	0.651	0.774	0.999	0.7
9	0.071	0.001	0.5	0.071	0.825	0.186	0.932	0.427
10	0.364	0.009	0.545	0.364	0.836	0.733	0.864	0.314
11	0.273	0.001	0.75	0.273	0.811	0.745	0.858	0.269
12	0.692	0.115	0.521	0.692	0.094	0.516	0.876	0.455
13	0.506	0.06	0.712	0.506	0.191	0.508	0.86	0.699

6 CONCLUSION

Firewall is the main component of network security that monitors the in-out network packets according to a predetermined security rules. Firewall controls network access by allowing or denying network traffic according on a set of user-defined policy rules. Hence, Firewall rules have proven to be error-prone, complicated, time consuming, and any inappropriate management of these rules will cause anomalies. Therefore, we continuously need to monitor, and update firewall rules to enhance and optimize these security policy. In this experiment, we have proposed and implement a hybrid model based on data mining and machine learning techniques analyze Firewall logs and detect anomalies in Firewall rules repository. The training data have been divided to 6 pieces 20-50-100-200-300-500 thousands records. We have compared the performance of the model based on four classification algorithms, NaiveBayes, KNN, One'R, and J48. It was noticed that, KNN algorithm has showed better successful performance results than the other 4 classification algorithms. Therefore, the Firewall rules were analyzed according to this algorithm. In addition, according to the F-measure values of KNN algorithm we observed that the Firewall rules 2, 6, 12, 13 are anomalous rules. These anomalous rules should be retailored and checked again to ensure that Firewall works efficiently and to obtain most convenient, corrected and anomaly-free Firewall rules. The proposed methods have shown a more precise performance in terms of accuracy and anomaly detection as a result enabling network administrators to update and optimize Firewall policy rules.

REFERENCES

[1] M. Xue and C. Zhu, "Applied Research on Data Mining Algorithm in Network Intrusion Detection," 2009 International Joint Conference on Artificial Intelligence, Hainan Island, 2009, pp. 275-277. doi: 10.1109/JCAI.2009.25.

[2] Roesch, M. (1999, November). Snort: Lightweight intrusion detection for networks. In Lisa (Vol. 99, No. 1, pp. 229-238).

[3] K. Golnabi, R. K. Min, L. Khan and E. Al-Shaer, "Analysis of Firewall Policy Rules Using Data Mining Techniques," 2006 IEEE/IFIP Network Operations and Management Symposium NOMS 2006, Vancouver, BC, 2006, pp. 305-315. doi: 10.1109/NOMS.2006.1687561.

[4] Agrawal, S., & Agrawal, J. (2015). Survey on anomaly detection using data mining techniques. Procedia Computer Science, 60, 708-713.

[5] Al-Shaer, E. S., & Hamed, H. H. (2004, March). Discovery of policy anomalies in distributed firewalls. In Ieee Infocom 2004(Vol. 4, pp. 2605-2616). IEEE.

[6] Chandala, V., Banerjee, A., & Kumar, V. (2009). Anomaly Detection: A Survey, ACM Computing Surveys. University of Minnesota.

[7] Saboori, E., Parsazad, S., & Sanatkhani, Y. (2010, August). Automatic firewall rules generator for anomaly detection systems with Apriori algorithm. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 6, pp. V6-57). IEEE.

[8] Ucar, E., Ozhan, E.: The analysis of firewall policy through machine learning and data mining. Wirel. Pers. Commun. 96, 2891 (2017). https://doi.org/10.1007/s11277-017-4330-0.

[9] Breier, J., & Branišová, J. (2017). A dynamic rule creation based anomaly detection method for identifying security breaches in log records. Wireless Personal Communications, 94(3), 497-511.

[10] Al-Shaer, E., Hamed, H., Boutaba, R., & Hasan, M. (2005). Conflict classification and analysis of distributed firewall policies. IEEE journal on selected areas in communications, 23(10), 2069-2084.

[11] Caruso, C., Malerba, D., & Papagni, D. (2005, May). Learning the daily model of network traffic. In International Symposium on Methodologies for Intelligent Systems (pp. 131-141). Springer, Berlin, Heidelberg.

[12] H. E. As-Suhbani and S. D. Khamitkar (2019) Mining Frequent Patterns in Firewall Logs Using Apriori Algorithm with WEKA. Recent Trends in Image Processing and Pattern Recognition. RTIP2R 2018. In Computer and Information Science (pp. 978-981-13-9186-6). Springer, Singapore. https://doi.org/10.1007/978-981-13-9187-3_50.

[13] Snort. An open source network intrusion detection system. http://www.Snort.org/.

[14] TWIDS Tool: TWIDS. http://twids.cute.edu.tw/en.

[15] As-Suhbani, H., Khamitkar, S.D. (2017): Enhancing snort IDS performance using TWIDS for collecting network logs dataset. Int. J. Res. Adv. Eng. Technol. 42–45 (2017). https://doi.org/10.22271/engineering

[16] URL download WEKA: http://www.cs.waikato.ac.nz/ml/weka/

[17] Spark, A. Spark Homepage: http://spark.apache.org.

[18] Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the: Webb. G.I, Association Rules (1993). In Handbook

[19] Kotsiantis, S., Kanellopoulos, D.: Association rules mining: a recent overview. GESTS Int. Trans. Comput. Sci. Eng. 32(1), 71–82 (2006)

[20] Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the: Webb. G.I, Association Rules (1993). In Handbook

[21] Alexandre Balon-Perin, "Ensemble-based methods for intrusion detection", Master thesis for the degree of Master in Computer Engineering Academic year 2011-2012. Norwegian University of Science and

Technology (NTNU).
[22] Cover, T., & Hart, P. (1967). Nearest neighbour pattern classification. IEEE Transactions on Information Theory, 13(1), 2127. doi:10.1109/TIT.1967.1053964.
[23] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.
[24] Chen, N., Shou, G., Hu, Y., & Guo, Z. (2009). An experimental research of traffic identification algorithms in broadband network. In 2009 International Symposium on Computer Network and Multimedia Technology(pp. 1–4). Wuhan: IEEE. doi:10.1109/CNMT.2009.5374758.
[25] Bhargava, N., Sharma, G., Bhargava, R., & Mathuria, M. (2013). Decision tree analysis on j48 algorithm for data mining. Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering, 3(6).
[26] Kaur, G., & Chhabra, A. (2014). Improved J48 classification algorithm for the prediction of diabetes. International Journal of Computer Applications, 98(22).