Wireless Pers Commun (2017) 96:2891–2909
DOI 10.1007/s11277-017-4330-0

The Analysis of Firewall Policy Through Machine Learning and Data Mining

Erdem Ucar[1] · Erkan Ozhan[2]

Published online: 17 May 2017
© Springer Science+Business Media New York 2017

Abstract Firewalls are primary components for ensuring the network and information security. For this purpose, they are deployed in all commercial, governmental and military networks as well as other large-scale networks. The security policies in an institution are implemented as firewall rules. An anomaly in these rules may lead to serious security gaps. When the network is large and policies are complicated, manual cross-check may be insufficient to detect anomalies. In this paper, an automated model based on machine learning and high performance computing methods is proposed for the detection of anomalies in firewall rule repository. To achieve this, firewall logs are analysed and the extracted features are fed to a set of machine learning classification algorithms including Naive Bayes, kNN, Decision Table and HyperPipes. F-measure, which combines precision and recall, is used for performance evaluation. In the experiments, kNN has shown the best performance. Then, a model based on the F-measure distribution was envisaged. 93 firewall rules were analysed via this model. The model anticipated that 6 firewall rules cause anomaly. These problematic rules were checked against the security reports prepared by experts and each of them are verified to be an anomaly. This paper shows that anomalies in firewall rules can be detected by analysing large scale log files automatically with machine learning methods, which enables avoiding security breaches, saving dramatic amount of expert effort and timely intervention.

Keywords Firewall logs · Machine learning · Firewall rule · Computer security

✉ Erkan Ozhan
erkanozhan@gmail.com

Erdem Ucar
erdemucar@trakya.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Trakya University, 22030 Edirne, Turkey

[2] Department of Computer Engineering, Faculty of Corlu Engineering, Namik Kemal University, Silahtaraga Mah. Unv. 1.Sok., 59860 Tekirdag, Turkey

1 Introduction

Firewalls are software-based or hardware-based systems which have taken their places as indispensable elements of today's communication networks and protect the network against the internal and external threats. Also, these systems highly contribute to the companies' active and effective use of the communication technologies and the other devices available. The firewalls which can be named safety control zones of the internet serve like the security check zones of companies. Another way to reach a company's facilities is to make a demand on the internet. Firewalls arrange your demands with its own rules according to certain criteria such as your purpose of connection, which doors you can go through, which departments you are not allowed into, the files you can take into or out of the company etc. These rules called policy may vary according to company, and keeping these rules updated makes a demanding process owing to the fact that communication technologies keep changing constantly. Rule mistakes may give rise to security weakness, thus allowing unwanted events such as devices' becoming out of service and indirectly causing profit loss of a company and leak-out of a secret document to happen.

Firewalls act as the control gate for computer networks. The system administrator sets the firewalls for needs of each organization [20].

Firewalls are important components of network security. However, to manage the firewall rules have become complicated and error-prone. A firewall filtering rules must be written carefully. These rules should eliminate security vulnerabilities, and should not be conflict [1]. The firewall provides methods of secure and untrusted networks. Rule policies that determine the behaviour of the firewall. The policy sets the type access to services of trusted and non-trusted domains [18]. Setting the firewall rules is a difficult process. This is caused by without conflicting with between the rules. The rules often conflict. The order of set the rule is a critical process [14].

As for structure, the firewalls are built on two main columns: Software and Hardware. While the hardware used has a contribution to the high performance, it falls inadequate when used alone. Software factor is an important variable as well. They use a special operating system which examines the packets coming from the network traffic by opening and was prepared by each firewall producer. These systems apply the rules which were written on the permanent memory by the network administrator to the coming data packets. Initially, the rule repository in a firewall which has not been installed yet is empty. These rules are prepared and saved into the system according to the software and hardware companies use, the task they carry out, and threats they were exposed or are likely to be exposed to. Firewall controls the incoming and outgoing network traffic according to the rules written on its memory by the system administrator. These rules can be abolished or updated, or new rules can be added. It is the system administrator's duty again to find out the shortcomings and errors of the rule repository.

The administration of the firewall rules has been proven to be complicated, error-prone, costly, and ineffective for the many organization networks. In addition, server and network traffic records can be used to confirm that the firewalls rules are consistent with the network service and updated. In order to fill the gaps between what the firewall rules say and what is observed in the network, the first step is to analyse the network traffic records through the use of data mining methods. Therefore, bringing a means which will help network traffic records be analysed into light will greatly contribute to the future studies [13]. Due to various reasons, firewall policies are always prone to change, and making policy changes is the most important task of the firewall administrators. For instance, network threats like new worm may arise. So, firewall policies should be changed accordingly in order to protect the special network against the new attacks. Likewise, modern institutions constantly enlarge their connections by loading new servers, software and services, and transform their sub-networks to maintain their competitive capacity. This process is completed by updating the firewall policies as well [23].

Machine learning methods can be used to discover and reveal the difficult-to-see relationships within big data clusters [26]. Machine learning algorithms try to find out the relationship and consistency among the data clusters using various techniques.

In this study, 5,000,000 logs taken from the firewall were analysed through supervised learning, one of the machine learning methods. Firewalls end connection activities according to rules. Therefore, each firewall rule ID was used as class. Through this method logs were exposed to classification. Six machine learning algorithms which are the most well known and have the higher classification performance were chosen for this classification. Each algorithm produced a model according to cross-validation technique, which is used for model validation. Weka-parallel 3.2.3 Data Mining Software in Java, which was developed by Waikato University and rearranged by Celis and Musicant in 2002 so that it would work in a parallel way, was used as analysis software. The firewall logs were exposed to a set of operations so that they could be analyzed after they had been taken as raw data. These operations will be mentioned in the following parts of the study.

High performance computing (HPC) is the use of multi-core processors and compute nodes [32]. Parallel-working technique was employed in order that Weka software could process a great amount of data with high performance and 13 computers in which High Performance Computing operating system, developed by Microsoft, is installed were used so that the application would work in a parallel way. The amount of training data affects classification performance in supervised learning, which we used in this study. Hence the training data used in this study were given to the algorithms as inputs in forms of pieces to which 250,000 lines are added at each time. Thanks to this, ideal data size at which the classifier's performance reached the maximum level was determined. The value at which maximum learning performance is achieved is named threshold in machine learning classification problems. In the study the threshold value was found as 1.5 million tuples. As the training data size was too big, performing the analysis with one computer was not possible due to insufficient hardware. Therefore, parallel computing, one of the high performance computing techniques, was employed. The parallel computing cluster composed of 13 computers sent the training data including 1.5 million tuples to all algorithms and classification performance results were obtained one by one.

1.1 Related Work

So far, many invaluable studies in which firewall logs have been analysed through machine learning methods have been carried out. Golnabi et al. [13] used Linux operating system firewall traffic log which contains 33,172 lines and 7 features and attempted to reduce and update the active firewall rules by examining the network traffic logs from their own frequency lines. Furthermore, Golnabi et al. [13] tried to detect dominant and decaying rules through Association Rule Mining (ARM).

In another study, Caruso et al. [6] tried to reveal the network daily check list template using a firewall record involving 76.702 lines and 8 features. The tests were carried out with K-means and EM algorithms.

In another study, Winding et al. [36] tried to detect the system anomalies looking into the firewall with machine learning methods and JRip algorithm. They have also stated that performing extra analysis might lead to better results.

Yoon et al. [39] made an effort to minimize the league of firewall rules with the aid of using network topology in their studies. As a result of their studies, they suggested that they managed this 2 or 3 times in some rules. However, they did not utilize the machine learning methods in their studies.

Breier and Brani [5], tried to propose a method for anomaly detection in log files, based on data mining techniques for dynamic rule creation.

Al-Shaer and Hamed [3], developed an algorithm in order to detect the rules which lead to anomalies in the firewall rule repository. The authors, who identified the rules which overlap or perform the same functions, made it possible to reorder and remove rules using a Java-based program called Firewall Policy Advisory. In a follow up study, Al-Shaer et al. [2] tried to detect more interrelated overlaps between the rules by developing the same software.

Tran et al. [34] developed a visualization tool which helps visualize firewall rules and ongoing operations. This tool, which also help detect rule anomalies, visualizes each firewall rule by breaking them up via Binary Decision Diagram method. The authors also managed to visualize the rule activities. They also simplified the management of firewall rules with the help of this tool, which is capable of managing rules. Similarly, Hu et al. [17] developed a tool called FAME (Firewall Anomaly Management Environment) which detects rule anomalies by visualizing them through a rule-based segmentation technique.

Frei and Rennhard [12] used server to detect anomalies in mail servers. They formed histogram matrix technique by determining these as daily activity model in form of 24 slots. The authors, who obtained the normal mail traffic as a histogram, identified a situation which is not normal according to the model as an anomaly. They showed the anomaly visually on the histogram and stated that this is simplicity for system administrators. However, they also stated that big matrix sizes created problems.

In another study, Pietraszek and Tanner [30] aimed to reduce false positive alarms by using intrusion detection . The researchers, who used JRip in machine learning algorithms, classified alerts in as false positive and true positive. The researchers, who employed classification, a machine learning technique, managed to reduce and update the rules which cause false positive alerts benefiting from JRIP algorithm, which enabled them to reduce the rules.

While average 6 features were included in other studies, 17 features in the firewall were included in our study. Therefore, more factors were taken into consideration in detecting rule anomalies. Besides, this study showed how to analyze big using parallel computing technique. When compared with other studies, performances of a great number of algorithms were comparatively measured and the performance of kNN algorithm was proved. Detecting anomalies according to the performances of machine learning classification algorithms is a method which was not employed in the previous studies. In this paper, we recommend a model and a method regarding how to achieve this.

2 Methodology

The data used in our study were processed at four stages which are data-collection, organization, transfer to the database, and finally preparation of the software and the hardware until they became analyzable.

The data to be used in the study have been extracted from a firewall. The firewall which will be used as learning data have been saved as 48 pieces with the firewall's own interface. This is due to the fact that a single log file containing high amount of data uses up the firewall's memory. The data have been downloaded in the format of ".csv". These files harboring the raw form of the training data were arranged with the help of the line and column processing software in a way that they could form tables.

The data which were distributed to the lines and columns were transferred to MySQL database and the adjustments of the machine learning classifiers were made for the purpose that they could take the training data from the database directly. The advantages of this method was mentioned in the Experimental Results section of the study. After database connection adjustments were made for WEKA, test connections were performed and it was observed that the connection was functioning properly. The attributes shown in Table 1 are available in the firewall log files transferred to the database in this study. These attributes make all of the traffic data features which can be taken from the firewall.

When the size of the data to be analyzed is considered and the results obtained from the first tests are taken into consideration, it has been found out that the computer which will start the parallel working requires a 24 GB physical memory. The computers to be used during the analysis process were connected to each other with Giga Bit LAN technology in a way that they could communicate with each other; and 4-core and 8-core processors and

Table 1 The attributes of the training data transferred to the database

Instances	Explanation
Pri	Severity levels
Subtype	Policy allowed-violation traffic
App_cat	Application category (N/A, Network.Service, Web, Update, P2P, Business, Media, IM, Proxy, eMail, Remote.Access, Game)
Dst_int	The interface through which the traffic goes out. For incoming traffic to the firewall it will be "unknown"
Dst_port	The destination port number of TCP and UDP traffic. The dst_port is 0 (zero) for other types of traffic
Dstname	The destination name or IP address
Duration	Connection Value /seconds
Proto	The protocol that applies to the session or packet
Rcvd	The total number of bytes received
Rcvd_pkt	The total number of packets received during the session
Sent	The total number of bytes sent
Sent_pkt	The total number of packets sent during the session
Service	The IP network service that applies to the session or packet. The services displayed correspond to the services configured in the firewall policies
SN	Session number
Src_port	The source port number of TCP and UDP traffic. The src_port is 0 (zero) for other types of traffic
Status	The status can be either deny or accept, depending on the applicable firewall policy
Policyid	The ID number of the firewall policy that applies to the session or packet

13 heterogen computers with physical memories of 4–8–16–24 GB were used. The system is composed of 82 cores and 172 GB RAM memory in total.

Developed by Microsoft, High Performance Computing (HPC) Server 2008 operating system, which uses clustering technology and can support parallel working, was loaded to 13 computers and the required adjustments were made. Weka 3.2.3 software, which enables parallel working was loaded to these computers as well.

3 Experimental Design

The data used in the analysis were transferred to the WEKA software, which uses machine learning techniques, in the form of 250–500–750 thousand and 1–1.5–2–2.5–3–3.5–4–4.5–5 million lines. Each of these training data consisting of 12 pieces was analysed through the use of six classifiers (Naive Bayes, kNN, DecisionTable, HyperPipes, OneR, ZeroR) . Machine learning algorithms are in various numbers. Of these algorithms which involve different approaches, 6 algorithms which yielded the best results were chosen for analyses. The outlook of the system was designed as seen in Fig. 1.

The data clusters of different sizes were given to each file in 12 pieces and 72 result files were obtained. Correctly Classified Instances%, Kappa Statistic, Root Mean Squared Error ve F-measure values available in these files were extracted and converted into separate tables.

3.1 Algorithms Used in the Analyses

3.1.1 Naive Bayes Algorithm

Bayes theorem, one of the machine learning techniques, is determined with classification and probability calculation. Bayes classification is also named Naive Bayes.

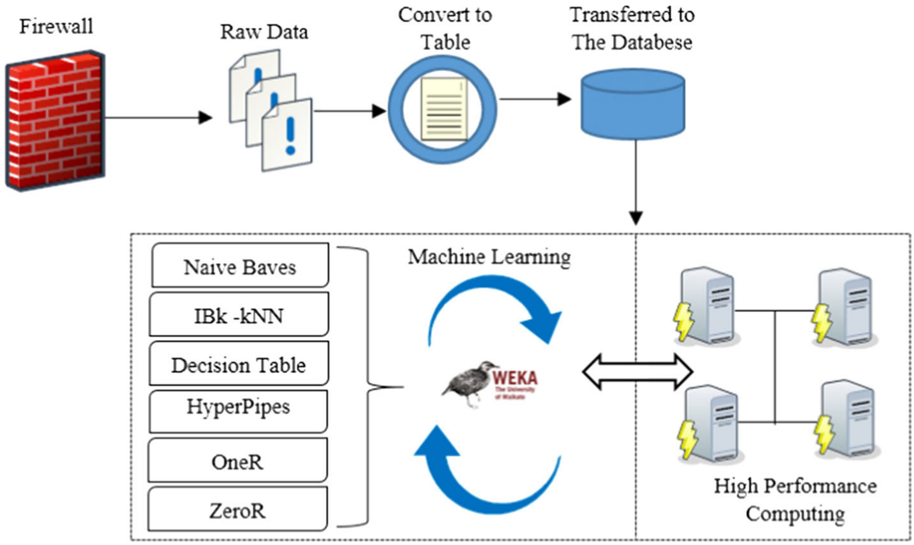

Fig. 1 The general outlook of the system

The data in Naive Bayes or simple bayes classifier are symbolized as n size feature vector. X=(x_1, x_2,..., x_n). The attribute of each class is A_1,A_2,..., A_n. Assume that the classes the data belong to are C_1, C_2,..., C_m. X, the class of which is unknown, will belong to the next highest class according to the attribute probability. Given that $P(C_i|X) > P\left(C_j|X\right)$ for $1 \leq j \leq m\, j \neq i$. $P(C_i|X)$ is named as maximum posteriori hypothesis. Thus, Bayes theorem is formulated using Eq. (1).

$$P(C_i|X) = \frac{P(X|C_i)P(C_i)}{P(X)}. \tag{1}$$

$P(X)$ is fixed for all groups. Only the multiplication of $P(X|C_i)P(C_i)$ is expected to be maximum. As the features are independent from each other, $P(x_k|C_i)$ probability could be easily obtained using Eq. (2) [15].

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|\ C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \cdots \times P(x_n|C_i) \tag{2}$$

3.1.2 kNN Algorithm

k-Nearest-Neighbor classifiers are based on learning by comparing the training lines with the given attributes [10].

The attributes of the training data tuples are shown with n in this algorithm. Each data tuple represents a point in n-sized space. In this way, training data tuples are stored in an n size pattern field. When k-nearest-neighbor classifier is given a data tuple the class of which is unknown, it looks for k training tuple pattern which is the most contiguous to the unknown data tuple. These k training tuples are the nearest neighbors of unknown data tuples. The term contiguity is defined for each tuple in terms of a metric distance unit such as euclidian distance. Euclid distance between two points or two data tuples like $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ is computed using Eq. (3) [15].

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \tag{3}$$

k-Nearest-Neighbor classifier is named IBk in WEKA software.

3.1.3 Decision Table Algorithm

This algorithm creates a decision tree according to the attributes of input-output data. Here, it is of great importance that the algorithm forms the decision table according to the attribute which is effective in the decision-making process.

A decision table is shown as $DT = <U, C \cup D, V, f>$. Here, C represents the line of discretely valued, independent variables which are not empty such as a company's financial indications. As to D, it is the line of discretely valued decision variables which are not empty and represent the classification decision. Here, U is a finite cluster, which is not empty, of N objects $\{x_1, x_2, \ldots, x_N\}$ in the enclosed space. V shows the values of the attributes and f shows the profit category. If the value of each data tuple can be defined with the composition of some conditional attribute values, the decision table is said to be deterministic. On the other hand, if the values of a set of attributes are conditionally dependent on the other attributes, the decision table is said to be non-deterministic [28].

3.1.4 HyperPipes Algorithm

HyperPipes algorithm records the value intervals observed in the training data for each attribute and category, and calculates the intervals containing the test samples attribute values by choosing the highest number of the correct intervals and the category in discrete classification problems [38]. Hyperpipes is an algorithm which performs classification operation in a short time [33]. Hyperpipes can conduct a simple and rapid classification when it is required to classify a great number of attributes. Hyperpipes saves the attribute limits for each category and then classifies each test sample according to their categories [11].

3.1.5 OneR Algorithm

OneR is an algorithm which performs classification in a rule-based way. Rule-based classifiers use if then rule cluster in data classification. The quality of the classification rules is measured by coverage and correctness rate [7]. When given a set of data, OneR creates a rule output every steps. Firstly, it creates a cluster of comparatively small candidate rules (only one rule for each attribute) , and then chooses one of these rules. These two step model is a typical situation for most of the learning systems [16]. OneR uses basic correction measure adapted by AttributeEvalOneR classifier. It may use the training data for evaluation and may also apply to nested cross-correction. All the data tuples are parameters and it accepts the minumum size of the data repository as a parameter. It learns a rule which predicts numerical and nominal class value [37].

3.1.6 ZeroR Algorithm

ZeroR is a simple algorithm which predicts majority class of the nominal data while it predicts average value of numerical test data. It applies the basic coverage algorithm for the rules [38]. ZeroR was used as one of the rule-based algorithms in WEKA [19].

3.2 Criteria Used in the Evaluation of the Analysis Results

3.2.1 Kappa Coefficient-Statistics

Kappa statistics (or Kappa coefficient) is a frequently-used method for measuring the interaction between two or more observations. Kappa was designed to show the size of the agreement between two or more observations and Kappa value is a measure changing between -1 and 1. These values indicate the agreement rate between the observations. As this value approaches 1, the agreement between the observations increases [35]. Kappa coefficients frequently measure the reliability and the validity of the agreement between categorical variables [8]. There are two applicable measurement units in the agreement between two judgements for any problem which occurs on the nominal scale. Kappa coefficient is computed using Eq. (4).

$$K = \frac{p_o - p_c}{1 - p_c} \tag{4}$$

Here, p_o indicates observed agreement. Observed agreement is the ratio of the sum of the attribute values to the general value. It can also said to be the ratio of the units where the judgments agree. p_c is the ratio of the units which are expected to agree coincidentally. Agreement test, however, is carried out according to $1 - p_c$ units for null hypothesis, which can predict disagreements between the judgments. This term serve as a denominator. As the factors function for agreement, p_o will exceed p_c; $p_o - p_c$ (their subtraction) represents the ratio of the situations which are not dependent on coincidence and this makes the numerator of the coefficient [9]. The interpretation table of Kappa coefficient according to the values it takes is shown in Table 2 [35].

3.2.2 F Measure

It is known as F-measure, F-Score or F1 Score in the literature. Performance index choice bears importance for evaluating a classifiers performance. Various methods obtained from confusion matrix (including F-measure) are recommended in machine learning and information retrieval. In the 2×2 confusion matrix given, $Accuracy(ACC) = (TP + TN)/(TP + TN + FP + FN)$ for the classifier which seperate two classes. This result is the accuracy value of the successfully classified analysis of an attribute. This value is of great variety for various data sets, thus can be misleading, and it is gathered in different TP (true positive) and TN (true negative) variables for each class. Sensivity (true positive rate veya recall) is indicated with R. $R = TP/(TP + FN)$. Precision is computed as: $P = TP/(TP + FP)$[24]. F-measure value is computed using Eq. (5).

$$F = \frac{2PR}{P + R} \tag{5}$$

3.2.3 Root Mean Squared Error (RMSE)

RMSE is a measurement of precision. It is used to determine the accuracy of conversion from one system into one other system [25]. RMSE has two important attributes. The first one is a standard measure which does not depend on measured or hidden variables. The second attribute is that it enables obtaining parametric confidence intervals, approximate

Table 2 Interpretation of Kappa

	Poor	Slight	Fair	Moderate	Substantial	Almost perfect
Kappa	0.0	0.20	0.40	0.60	0.80	1.0
Value	*Agreement*					
<0	Less than chance agreement					
0.01–0.20	Slight agreement					
0.21–0.40	Fair agreement					
0.41–0.60	Moderate agreement					
0.61–0.80	Substantial agreement					
0.81–0.99	Almost perfect agreement					

distributive attributes of which are known, and performing hypothesis tests [22]. RMSE is computed using Eq. (6).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (S_i - O_i)^2}{n}} \tag{6}$$

In this Formula, O_i is observed value, S_i is simulated value, O is mean observed value and n is the number of observations [29].

4 Experimental Results

Analysis results taken from WEKA program were saved in 72 files. Kappa statistic, Correctly Classified Instances%, Root mean squared error values were extracted from these files and were saved in the form tables. These values will be used to determine the most convenient quantity of training data and the convenient algorithm. Kappa values for 6 algorithms are shown in Fig. 2. When Kappa values of the classifiers are taken into consideration, kNN, Decision Table, Naive Bayes and HyperPipe, which are among these classifiers, can be a matter of preference in the analysis of firewall rules. As for ZeroR algorithm, it has 0 Kappa value and therefore is not a convenient algorithm for the analysis of the firewall rules. The algorithm with the highest Kappa value is kNN algorithm. As shown in Fig. 2, Kappa values of all algorithms begin to decrease once the training data exceed 1,500,000. It can be said that optimum learning was reached at 1,500,000 training data. Hence, it was thought that training datas being 1,500,000 in number would be

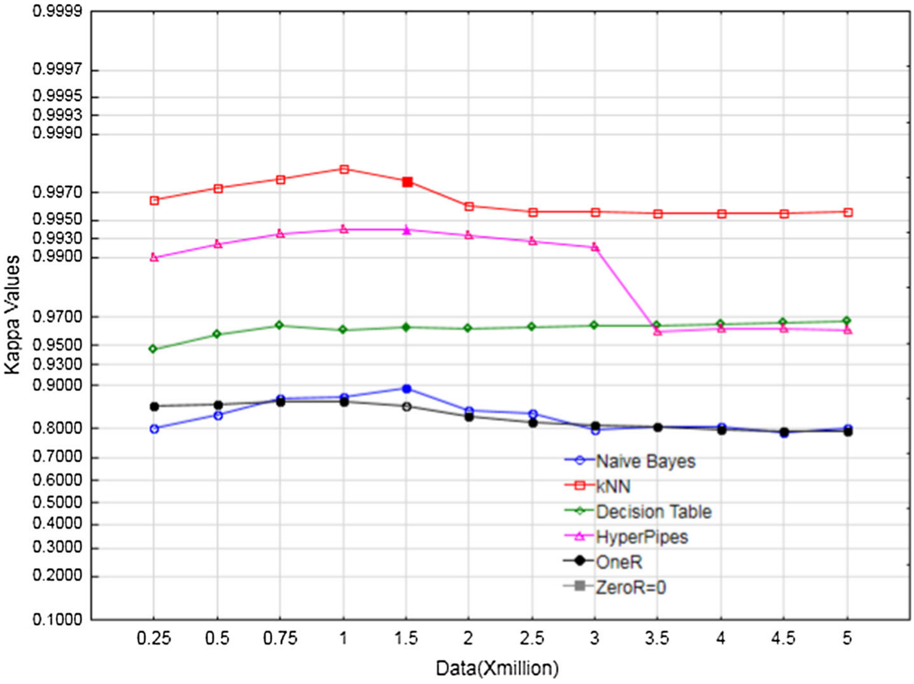

Fig. 2 Kappa values for all classification algorithms

convenient. At this stage, what the size of the training data would be was determined. As the amount of training data increases, the generalization error decreases. As the amount of training data increases, the generalization error decreases. As the complexity of the model class increases, the generalization error decreases first and then starts to increase [4]. This situation complies with the author's identify. When Correctly Classified Instances values in Fig. 3 are taken into consideration, it can be said that maximum performance was realized at the interval of 1,000,000 and 1,500,000. In parallel to Kappa values, correctly classified data number began to decrease for all classification algorithms after the training data reached 1,500,000 in number. It was observed that Correctly Classified Instances values of kNN ve HyperPipe, in particular, were so close to each other. Thus, one of the two algorithms will be chosen as a classification algorithm in the analysis of the rules considering their RMSE values. RMSE values of all classification algorithms are shown in Fig. 4. These values will prove an important indicator in determining the confidence intervals of the classifications algorithms made. According to these results, the algorithm classification confidence interval of which gave the best result is kNN algorithm. The algorithms which were used in the analyses and yielded better results than the previous evaluation results were kNN and HyperPipes algorithms. However, RMSE value of kNN algorithm yielded a better result than that of HyperPipes. Therefore, using kNN algorithm in the analysis of the firewall rules can yield the best results.

5 The Analysis of Firewall Rules According to Process Model

We created Process Model shown in Fig. 5 to test firewall rules according to the information obtained from the experimental results of this study. The rules were analyzed according to this model. According to Process Model shown in Fig. 5, firstly F-measure

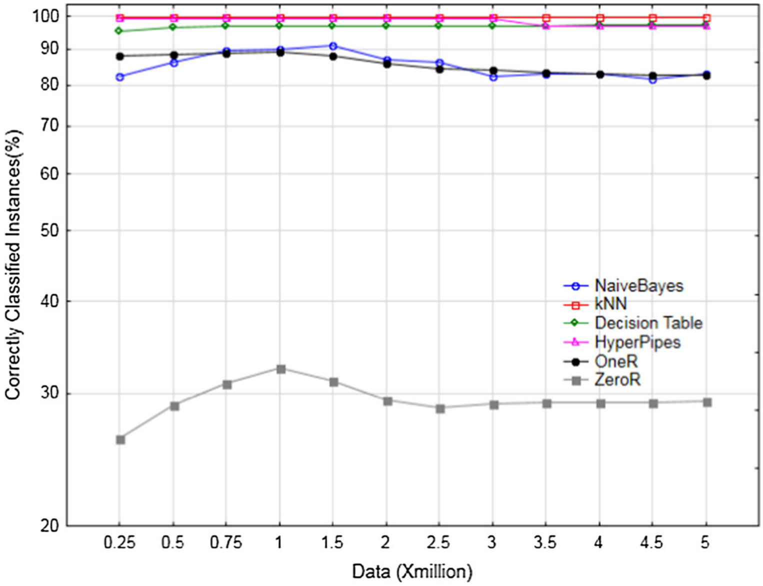

Fig. 3 Correctly Classified Instances% values obtained for all classification algorithms

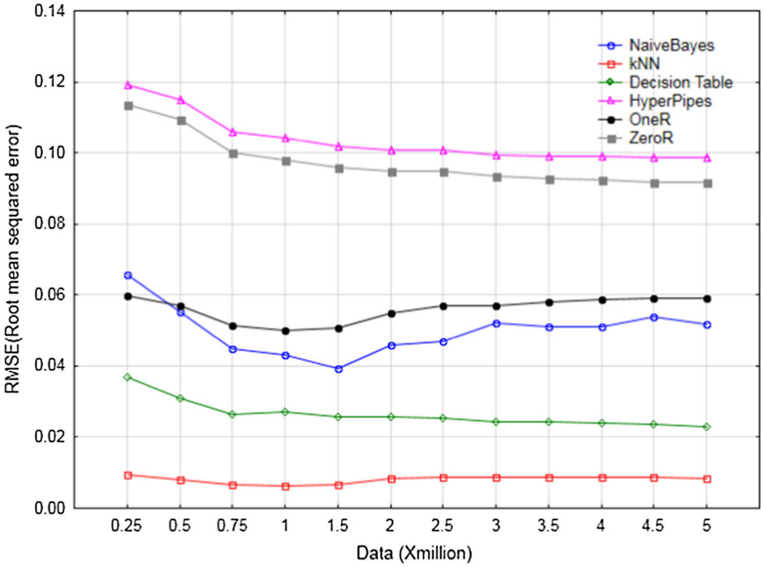

Fig. 4 RMSE values of classification algorithms

values were taken from WEKA software taking the algorithm with the highest performance into account. F-measure values obtained from kNN classifier for 93 firewall rules are shown in "Appendix". When the distribution of F-measure values shown in Fig. 6 were transferred to the graphics, the general Outlook given in Fig. 1 was obtained. When the distribution in Fig. 6 is taken into consideration, it can be seen that firewall rules 3, 21, 130, 156, 228, 230 remained out of the general distribution. Two things may have led to this situation. The first reason is that there are not sufficient data which belong to these rules in the training data. As for the second reason, the system did not manage to learn these rules due to the abnormality resulting from the fact that these rules misprocessed the network data. Firstly, the first reason was looked into. The number of data tuples belonging to the rule IDs with low F-measure values was examined from file and was shown in Table 3.

As indicated in Table 3, the quantity of training data belonging to the firewall rules with rule ID 3, 156, 228, 230 is too low. This is due to the fact that the classification algorithm failed to classify these rules correctly. An abnormality related to the rules cannot be mentioned. When the second reason why F-measure values were low was examined, it was observed that the firewall rules with rule ID 21 and 130 could not be learnt by the classification system although they had sufficient training data. For this reason, the information available in the firewall of the firewall rules was taken into examination. Upon examining the firewall of rule ID 21, it was seen that connection requests came from different positions with the use of port 3389 and they were accepted. And when the firewall belonging to rule ID 130 were examined, it was again seen that connections from different geographical positions were made through ports 22, 3389, 443, and 80. All of these connection requests were accepted by these two rules and they were not blocked. After this stage, what purpose ports 22, 3389, 443, and 80 are used for and whether they lead to security gaps or not were examined.

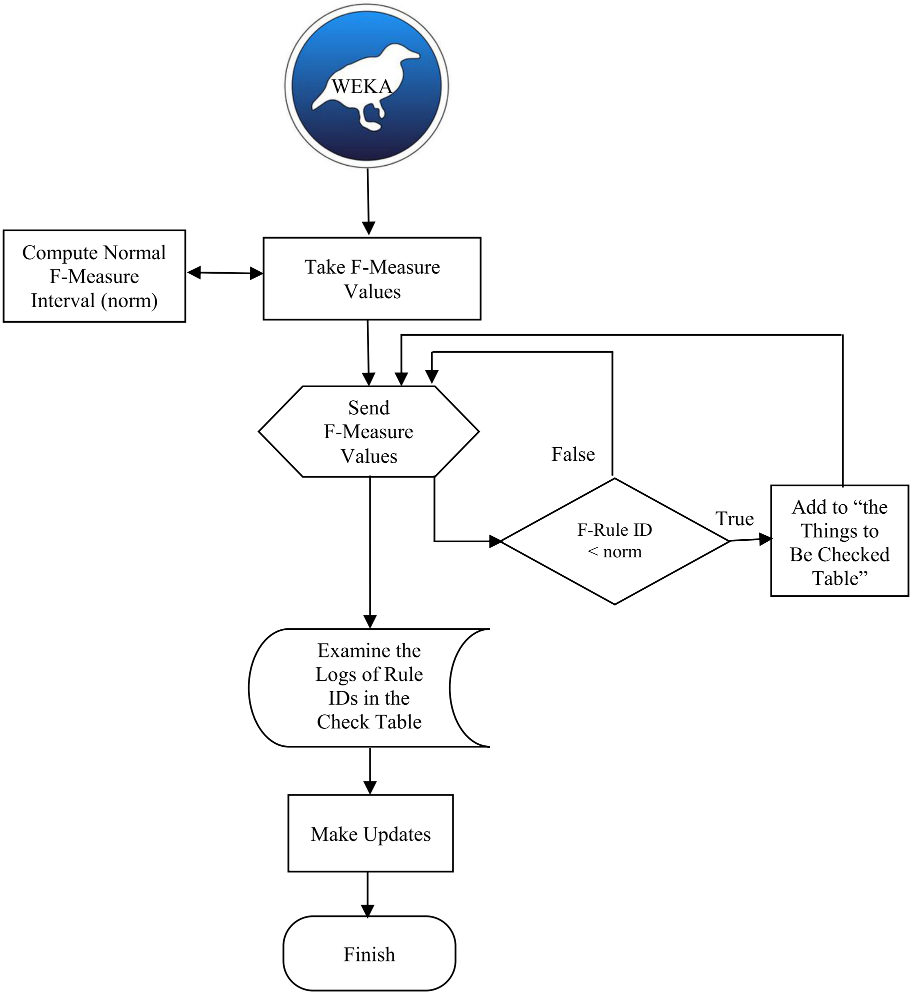
WEKA
Compute Normal F-Measure Interval (norm)
Take F-Measure Values
Send F-Measure Values
False
F-Rule ID < norm
True
Add to "the Things to Be Checked Table"
Examine the Logs of Rule IDs in the Check Table
Make Updates
Finish

Fig. 5 Process model for analysis of firewall rules

5.1 Risk Analysis According to Process Model Results

Application firewall is a special type of attack-blocking systems. These firewalls generally aim to protect web-based applications against the attacks coming from ports 80 and 443 [21].

Windows NT ve Windows 2000 terminal servers allow an attacker from distance to make countless remote desktop connection requests using port 3389. This situation causes memory overflow and service denial [27].This method is a frequently-applied service stopping strategy at present. Therefore, it is obvious that it gives rise to security breaches.

Port 22 is used as a default for SSH (secure shell) connection. Unix systems uses SSH connection type instead of FTP and Telnet over SSL (secure socket layer) mostly in order to encode network connections. One of the DoS attacks is that it causes the system to become unstable operating codes under SSH credential [31].

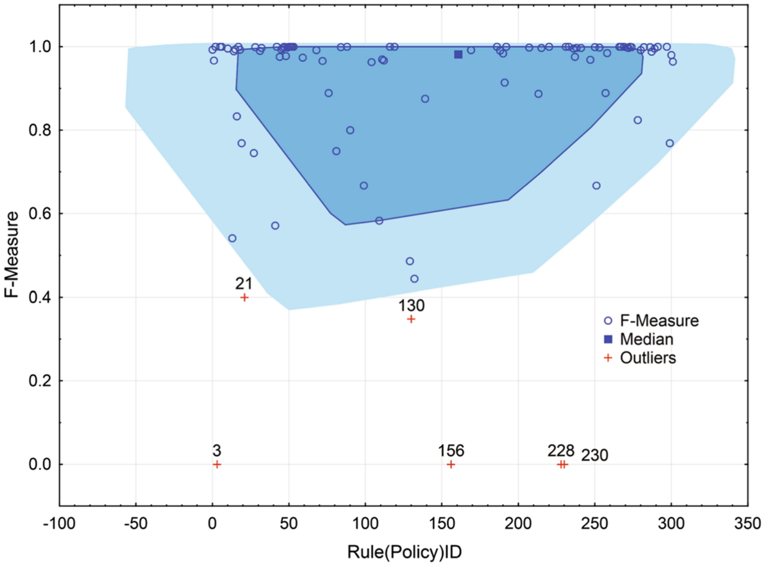

Fig. 6 F-measure distribution of firewall rules according to kNN algorithm

Table 3 F-measure and Instances values of firewall rules remaining out of the general distribution of F-measure values

Rule ID	F-measure	Instances
3	0	2
21	0.444	39
130	0.348	20
156	0	4
228	0	3
230	0	2

It would be convenient that Firewall rule IDs 21 and 130, which are out of the normal distribution and may lead to security breaches, should be checked by the system administrator.

6 Conclusions and Future Works

It can be said that machine learning methods give good results in detecting abnormalities in the firewall rules. The number of the connections where the rules in the firewalls work and are brought to a decision are great in number. As a consequence, it may take a long time to check. A more effective and rapid updating administration can be enabled by decreasing the number of rules to be checked with Process Model, which brings machine learning and high performance methods, which are recommended in this study, together.

Another result which is observed during the application stage of this study is the advantages the use of database brings. Thanks to the database use, data sizes can be analyzed gradually. Therefore, it can allow detecting maximum performance level of the number of training data.

It was observed that kNN algorithm yielded the highest performance in the analysis of the firewall . This may result from the fact that kNN by nature can reach the extereme points of the training data.

It was seen in all the algorithms that they achieved the maximum learning performance when they reached the training data value, composed of 1,500,000 data tuples, and their performance level, however, began to fall after this point.

The researchers who will carry out their studies about this subject are bound to encounter vast amounts of data. Because of this, they may prefer high performance computing methods in order that they can shorten the duration of analysis.

In the studies to come, whether machine learning methods can be adjusted so that they can operate on a firewall at real-time can be studied. What the quality of the hardware requirements for this will be can be a problem researchers may experience owing to the fact that processing a great amount of data will call for high capacity hardware.

Different types of parallel computing technique, which was used in the study, can be tried for vast quantities of data. 1,500,000 optimum learning threshold value, shown as a result by machine learning classifiers, can be analysed using different firewall.

Appendix

F-measure values of firewall rules according to kNN classifier.

Rule ID	TP rate	FP rate	Precision	Recall	F-measure
5	1	0	1	1	1
186	1	0	1	1	1
6	1	0	0.999	1	1
45	0.994	0	0.994	0.994	0.994
50	1	0	1	1	1
236	0.996	0.001	0.995	0.996	0.996
53	1	0	1	1	1
220	1	0	1	1	1
46	1	0	0.993	1	0.997
213	0.886	0	0.888	0.886	0.887
47	1	0	1	1	1
273	1	0	1	1	1
10	0.993	0	1	0.993	0.996
32	0.996	0	0.999	0.996	0.997
231	1	0	1	1	1
49	0.996	0	0.999	0.996	0.998
14	0.984	0	0.994	0.984	0.989
84	0.998	0	0.999	0.998	0.999
207	0.998	0	0.998	0.998	0.998
250	0.999	0	1	0.999	0.999
15	0.992	0	0.995	0.992	0.994
190	0.997	0	0.972	0.997	0.984
48	0.971	0	0.985	0.971	0.978

Rule ID	TP rate	FP rate	Precision	Recall	F-measure
241	0.995	0	0.999	0.995	0.997
51	1	0	1	1	1
52	1	0	1	1	1
237	0.974	0	0.979	0.974	0.976
31	0.993	0	0.987	0.993	0.99
18	0.989	0	0.996	0.989	0.993
88	1	0	1	1	1
191	0.941	0	0.889	0.941	0.914
238	0.997	0	0.997	0.997	0.997
0	0.986	0	1	0.986	0.993
300	0.972	0	0.988	0.972	0.98
272	0.997	0	0.997	0.997	0.997
215	0.996	0	0.998	0.996	0.997
253	0.998	0	0.998	0.998	0.998
112	0.957	0	0.978	0.957	0.967
301	0.959	0	0.97	0.959	0.964
287	0.99	0	0.986	0.99	0.988
116	1	0	1	1	1
2	1	0	1	1	1
233	1	0	1	1	1
169	1	0	0.983	1	0.992
271	0.996	0	1	0.996	0.998
109	0.609	0	0.56	0.609	0.583
258	0.97	0	1	0.97	0.985
266	1	0	1	1	1
289	0.996	0	0.995	0.996	0.996
282	0.999	0	0.999	0.999	0.999
129	0.474	0	0.5	0.474	0.486
72	1	0	0.933	1	0.966
68	0.984	0	1	0.984	0.992
280	0.997	0	0.987	0.997	0.992
111	0.96	0	0.98	0.96	0.97
13	0.625	0	0.476	0.625	0.541
41	0.667	0	0.5	0.667	0.571
139	0.836	0	0.918	0.836	0.875
17	1	0	1	1	1
247	0.94	0	1	0.94	0.969
27	0.636	0	0.897	0.636	0.745
286	0.999	0	0.999	0.999	0.999
44	0.984	0	0.968	0.984	0.976
104	0.992	0	0.936	0.992	0.963
132	0.364	0	0.571	0.364	0.444
130	0.333	0	0.364	0.333	0.348
19	0.833	0	0.714	0.833	0.769

Rule ID	TP rate	FP rate	Precision	Recall	F-measure
119	1	0	1	1	1
90	0.667	0	1	0.667	0.8
278	0.84	0	0.808	0.84	0.824
81	0.75	0	0.75	0.75	0.75
1	1	0	0.935	1	0.967
16	0.833	0	0.833	0.833	0.833
76	0.8	0	1	0.8	0.889
21	0.444	0	0.364	0.444	0.4
28	0.999	0	0.999	0.999	0.999
257	1	0	0.8	1	0.889
59	0.949	0	1	0.949	0.974
42	1	0	1	1	1
299	0.833	0	0.714	0.833	0.769
251	0.667	0	0.667	0.667	0.667
269	1	0	1	1	1
99	0.667	0	0.667	0.667	0.667
228	0	0	0	0	0
230	0	0	0	0	0
192	1	0	1	1	1
3	0	0	0	0	0
156	0	0	0	0	0
291	1	0	1	1	1
188	0.994	0	0.988	0.994	0.991
267	1	0	1	1	1
297	1	0	1	1	1
274	0.999	0	0.999	0.999	0.999

References

1. Al-Shaer, E. (2004). Managing firewall and network-edge security policies. In *2004 IEEE/IFIP Network Operations and Management Symposium* (Vol. 1, p. 926). Seoul: IEEE. doi:10.1109/NOMS.2004.1317810.
2. 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. doi:10.1109/JSAC.2005.854119.
3. Al-Shaer, E. S., & Hamed, H. H. (2003). Firewall policy advisor for anomaly discovery and rule editing. In G. Goldszmidt & J. Schnwlder (Eds.), *Integrated network management VIII: Managing it all* (p. 1730). Boston, MA: Springer. doi:10.1007/978-0-387-35674-7.
4. Alpaydın, E. (2010). *Introduction to machine learning* (2nd ed.). Cambridge, MA, London: MIT Press.
5. Breier, J., & Branišová, J. (2015). A dynamic rule creation based anomaly detection method for identifying security breaches in log records. *Wireless Personal Communications*,. doi:10.1007/s11277-015-3128-1.
6. Caruso, C., Malerba, D., & Papagni, D. (2005). Learning the daily model of network traffic. In *Foundations of Intelligent Systems*(pp. 131–141). Saratoga Springs, NY. http://link.springer.com/chapter/10.1007/11425274_14.
7. 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.

8. Chmura Kraemer, H., Periyakoil, V. S., & Noda, A. (2002). Kappa coefficients in medical research. *Statistics in Medicine*, *21*(14), 2109–2129. doi:10.1002/sim.1180.
9. Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *20*, 37–46. doi:10.1177/001316446002000104.
10. Cover, T., & Hart, P. (1967). Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 2127. doi:10.1109/TIT.1967.1053964.
11. Eisenstein, J., & Davis, R. (2004). Visual and linguistic information in gesture classification. In *Proceedings of the 6th International Conference on Multimodal Interfaces—ICMI04*, (p. 113). New York, NY: ACM Press. doi:10.1145/1027933.1027954.
12. Frei, A., & Rennhard, M. (2008). Histogram matrix: Log file visualization for anomaly detection. In *ARES 2008—3rd International Conference on Availability, Security, and Reliability, Proceedings* (pp. 610–617). doi:10.1109/ARES.2008.148.
13. Golnabi, K., Min, R. K., Khan, L., & Al-Shaer, E. (2006). Analysis of firewall policy rules using data mining techniques. In *10th IEEE/IFIP Network Operations and Management Symposium NOMS 2006* (Vol. 5, pp. 305–315). IEEE. doi:10.1109/NOMS.2006.1687561.
14. Gouda, M. G., & Liu, A. X. (2007). Structured firewall design. *Computer Networks*, *51*(4), 1106–1120. doi:10.1016/j.comnet.2006.06.015.
15. Han, J., & Kamber, M. (2006). Data mining concepts and techniques. In J. Gray (Ed.), *Data mining: Concepts and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers.
16. Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, *11*(1), 63–91.
17. Hu, H., Ahn, G. J., & Kulkarni, K. (2012). Detecting and resolving firewall policy anomalies. *IEEE Transactions on Dependable and Secure Computing*, *9*(3), 318–331. doi:10.1109/TDSC.2012.20.
18. Hunt, R. (1998). Internet/intranet firewall security-policy, architecture and transaction services. *Computer Communications*, *21*(13), 1107–1123. doi:10.1016/S0140-3664(98)00173-X.
19. Kerdegari, H., Samsudin, K., Ramli, A. R., & Mokaram, S. (2012). Evaluation of fall detection classification approaches. In *2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)* (Vol. 1, pp. 131–136). Kuala Lumpur: IEEE. doi:10.1109/ICIAS.2012.6306174.
20. Khan, B., Khan, M. K., Mahmud, M., & Alghathbar, K. S. (2010). Security analysis of firewall rule sets in computer networks. In *2010 Fourth International Conference on Emerging Security Information, Systems and Technologies* (pp. 51–56). Venice: IEEE. doi:10.1109/SECURWARE.2010.16.
21. Kowalski, K., & Beheshti, M. (2006). Analysis of log files intersections for security enhancement. In *Third International Conference on Information Technology: New Generations (ITNG06)* (pp. 452–457). Las Vegas: IEEE. doi:10.1109/ITNG.2006.32
22. Lai, K., & Kelley, K. (2011). Accuracy in parameter estimation for targeted effects in structural equation modeling: Sample size planning for narrow confidence intervals. *Psychological Methods*, *16*(2), 127–148. doi:10.1037/a0021764.
23. Liu, A. X. (2012). Firewall policy change-impact analysis. *ACM Transactions on Internet Technology*, *11*(4), 1–24. doi:10.1145/2109211.2109212.
24. Maratea, A., Petrosino, A., & Manzo, M. (2014). Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, *257*, 331–341. doi:10.1016/j.ins.2013.04.016.
25. Moses, K. P., & Devadas, M. D. (2012). An approach to reduce root mean square error in toposheets. *European Journal of Scientific Researach*, *91*(2), 268–274.
26. Nilsson, N. J. (1998). *Introduction to Machine Learning*. Stanford, CA. Retrieved from http://robotics.stanford.edu/people/nilsson/mlbook.html.
27. NIST. (2016). *National Vulnerability Database*. Technical report, National Institute of Standarts and Information Technology Laboratory, Gaithersburg, MD. https://nvd.nist.gov/home.cfm.
28. Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*(1st edn.). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-76917-0.
29. Parker, A., de Cortázar-Atauri, I. G., Chuine, I., Barbeau, G., Bois, B., Boursiquot, J. M., et al. (2013). Classification of varieties for their timing of flowering and veraison using a modelling approach: A case study for the grapevine species Vitis vinifera L. *Agricultural and Forest Meteorology*, *180*, 249–264. doi:10.1016/j.agrformet.2013.06.005.
30. Pietraszek, T., & Tanner, A. (2005). Data mining and machine learning towards reducing false positives in intrusion detection. *Information Security Technical Report*, *10*(3), 169–183. doi:10.1016/j.istr.2005.07.001.
31. Shinder, T. W., Amon, C., Shimonski, R. J., & Shinder, D. L. (2003). *The best damn firewall book period*. Rockland, MA: Syngress Publishing. doi:10.1016/B978-193183690-6/50046-7.

32. Smith, M. C., & Peterson, G. D. (2005). Parallel application performance on shared high performance reconfigurable computing resources. *Performance Evaluation*, *60*(1–4), 107–125. doi:10.1016/j.peva.2004.10.004.
33. Smusz, S., Kurczab, R., & Bojarski, A. J. (2013). A multidimensional analysis of machine learning methods performance in the classification of bioactive compounds. *Chemometrics and Intelligent Laboratory Systems*, *128*, 89–100. doi:10.1016/j.chemolab.2013.08.003.
34. Tran, T., Al-Shaer, E. S., & Boutaba, R. (2007). 055 PolicyVis: Firewall security policy visualization and inspection. In *Proceedings of the 21st conference on Large Installation System Administration Conference USENIX Association* (Vol. 7, pp. 1–16). http://usenix.org/event/lisa07/tech/full_papers/tran/tran.pdf.
35. Viera, A. J., & Garrett, J. M. (2005). Understanding inter observer agreement: The kappa statistic. *Family Medicine*, *37*(5), 360–363, http://www.ncbi.nlm.nih.gov/pubmed/15883903.
36. Winding, R., Wright, T., & Chapple, M. (2006). System anomaly detection: Mining firewall logs. In *2006 Securecomm and Workshops* (pp. 1–5). Baltimore, MD: IEEE. doi:10.1109/SECCOMW.2006.359572.
37. Witten, I. H., & Frank, E. (2005). *Data mining practical machine learning tools and techniques* (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers Inc.
38. Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining practical machine learning tools and techniques* (3rd ed.). Burlington, MA: Elsevier Inc.
39. Yoon, M., Chen, S., & Zhang, Z. (2010). Minimizing the maximum firewall rule set in a network with multiple firewalls. *IEEE Transactions on Computers*, *59*(2), 218–230. doi:10.1109/TC.2009.172.

Erdem Ucar received her Bachelor degree in Physics at the Trakya University, Faculty of Arts and Science in 1990. He has received her Master in computer engineering from the same university in 1993. He received his Ph.D. degree at Trakya University in Edirne, Faculty of Engineering in Computer Software. Currently he works as a Assoc.-Prof.Dr. at Trakya University in Turkey.

Erkan Ozhan received his Bachelor degree in computer education at the Firat University, Faculty of Computer Education in 2000. He continued with his studies at the Trakya University in Edirne, Faculty of Engineering, where he received his Master degree in computer engineering. He received his Ph.D. degree at Trakya University in Edirne, Faculty of Engineering in Computer Software (machine learning and data mining). Currently he works as a Assist.Prof.Dr. at Namık Kemal University in Turkey, in the Computer Engineering. His research interests are in the area of security controls and attack detection, data mining and high performance computing.