SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
http://saucis.sakarya.edu.tr/
e-ISSN: 2636-8129
Publisher: Sakarya University
Vol. 8, No. 4, 688-700, 2025
DOI: 10.35377/saucis...1698023
RESEARCH ARTICLE

Network Traffic Factors Detection and Prediction Using Artificial Neural Networks

Muhammed Özdemir[1], Mustafa Akpınar[2*], Huseyin Eski[3]

[1]Türkiye Denizcilik İşletmeleri A.Ş, Istanbul, Turkiye
[2]Higher Colleges of Technology, Faculty of Computer and Information Science, Abu Dhabi, United Arab Emirates, ror.org/00qmy9z88
[3]Sakarya University, Computer Engineering Department, Sakarya, Turkiye, ror.org/04ttnw109

Corresponding author:
Mustafa Akpinar,
Higher Colleges of Technology
mustafakpinar@gmail.com

Article History:
Received: 12.05.2025
Revised: 18.08.2025
Accepted: 01.09.2025
Published Online: 13.10.2025

ABSTRACT
Rising technology generates an increasing data flow on the internet by the day. Specific security systems are crucial for managing this growing data traffic. One such security system is the firewall used in Turkey Maritime Enterprises Inc. (TME). All TME internet traffic is controlled by a security firewall that operates within existing rules and regulations. The security firewall produces output based on traffic categorized as "allowed" or "blocked." This study collected log records from the firewall on five different days to create datasets. Twenty-six variables were extracted from the firewall logs, excluding the output. The study aimed to identify the most significant parameters impacting the output using linear regression (LR) and principal component analysis (PCA). The analysis identified mutually influential log variables affecting the output using two methods. After, the initial dataset with 26 variables and a reduced dataset with six variables were used to predict output using the artificial neural network (ANN) for five datasets. The prediction accuracy and precision ranges were between 85% and 88% and 92% and 98%, respectively. The F1-score showed results between 89% and 92% in addition to accuracy and precision. ML and PCA methods successfully identified crucial variables for estimating output and reduced the number of variables from 26 to 5. Moreover, it was noted that the ANN could accurately determine whether firewall traffic would be blocked or allowed based on the reduced datasets. Reducing the feature space from 26 to 6 variables identified via MLR+PCA improved ANN performance across hours, indicating that compact, interpretable inputs can support accurate firewall-traffic prediction.

Keywords: Cyber security, Firewall, Log, Linear regression, Principal component analysis, Artificial neural networks

1. Introduction

With the advancement of Information Technologies, internet traffic has significantly increased due to the vast amount of data. While internet usage serves various purposes, including corporate needs, malicious actors can exploit existing systems for vulnerabilities or weaknesses in end-user security. These vulnerabilities can lead to system compromise or encryption, posing significant cybersecurity threats.

The surge in internet traffic and the continuous rise in cyber threats have necessitated the use of a security mechanism to control systems [1]. These cyber threats demand the deployment of Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS). These systems are utilized to prevent cyberattacks and vulnerabilities. However, relying solely on systems like IDS and IPS is insufficient. Additional security measures may involve the use of end-user antivirus software, as well as ensuring the security of incoming/outgoing emails through anti-spam or mail-gateway solutions, to prevent potentially harmful links or executable programs. Security products, such as Security Information and Event Management (SIEM) tools, are also employed for easy anomaly detection and to generate alerts for system administrators or end-users.

Turkey Maritime Enterprises Inc. (TME) manages all its internet traffic for end-users, and the system passes through the security firewall. TME also operates two additional locations connected to the main center. The security connections between these locations and the main center are facilitated via IPSec connections, where the internet access for these locations is routed through the security firewall at the main center. Additionally, the organization is part of KamuNet, a closed virtual network for government institutions. If provided via KamuNet by government institutions, services acquired from external sources are accessed through KamuNet. TME also offers services on KamuNet, and services obtained from the internet via KamuNet are accessed through it.

Cite as: M. Özdemir, M. Akpınar and H. Eski, "Network traffic factors detection and prediction using artificial neural networks," *Sakarya University Journal of Computer and Information Sciences*, vol. 8, no. 4, pp. 688-700, 2025. doi: 10.35377/saucis...1698023

The institution's security firewall logs are employed for all internet traffic. According to Law No. 5651, published in the Official Gazette on May 23, 2007, these logging procedures are mandatory for hosting and access providers [2]. These logs are signed with a timestamp in accordance with Law No. 5651, ensuring their unalterability and non-repudiation. Required fields within these logs include Internet Protocol (IP) and Medium Access Control (MAC) addresses.

Internet traffic is regulated through rules defined by the institution via the security firewall. These rules are customized by default according to the institution's requirements. Security profiles, such as Web Filter, Application Filter, and Domain Name System (DNS) Filter, are established, and options are selected to ensure that traffic is filtered according to these security profiles. For instance, within the institution's security rules, all traffic containing "Social Media," "Hacking," "Illegal," "Proxy," "Terrorism," or "Pornography" content from the local network to the internet is blocked, while other content is allowed. The security firewall also logs IPsec traffic between the main center and connected locations, Secure Sockets Layer Virtual Private Network (SSL VPN) traffic, and KamuNet traffic. The mentioned traffic is segmented and logged into specific sections by the firewall. Each log's traffic on the firewall is parsed into a total of 219 fields. Many of these fields are often empty. Based on the type of traffic (i.e., web, application, VPN), some fields within the logs are populated while others remain empty.

2. Literature Review

An internet security system automatically controls incoming and outgoing internet traffic through a predefined set of rules. Machine learning algorithms are used to analyze traffic on the firewall and manage internet traffic based on the results. Al-Behadili (2021) controlled the firewall traffic using a Decision Tree classification algorithm [1]. This study conducted a detailed analysis to classify daily firewall access. The results obtained were compared to assess the performance of the Decision Tree algorithm. The Decision Tree algorithm was found to be the most successful in terms of classification accuracy when compared with DVM, ANN, Multi-classifier, and other machine learning algorithms. Performance evaluation was also conducted using common metrics like classification error, Kappa Statistic, F-measure, and Mean Absolute Error. The results demonstrated that the Decision Tree algorithm outperformed all other algorithms in terms of various performance metrics for firewall activities.

It is challenging for a system administrator to observe how accurately the rules established through the firewall are enforced, whether any rules are losing significance, or if there exist rules among them that might create security vulnerabilities. In reference [3], an attempt was made to determine the effectiveness of an analysis study based on classifying firewall logs using machine learning methods. In this study, 5,000,000 log lines were examined, 17 firewall log fields were extracted, and six different algorithms were utilized. The algorithms used were HyperPipes, Decision Table, ZeroR, Naive Bayes, IBk (-kNN), and OneR. Ultimately, when considering Correctly Classified Instances, Kappa, and Root Mean Squared Error values, it was observed that the IBk algorithm ranked first with a high score, followed by the Decision Table algorithm. According to the F-score results provided by the IBk algorithm, it was noted that the Policy IDs of the firewall rules required reconsideration.

Understanding how software and hardware were utilized for system and information security can be maintained and discussed by Akbas [4]. It is paramount in ensuring system security. Log records are essential components of a system, and it is highlighted that logs collected from various network devices can be analyzed in multiple fields using today's technology. Log data collected from operating systems, applications such as DHCP, security firewalls, network devices, emails, and databases can be maintained. Time information is crucial in collecting logs, hence the necessity of operational NTP services. Besides log management, the importance of Security Information Event Management (SIEM) software has also been emphasized.

Another study aimed to efficiently create security firewall rules in a system [5]. Security firewalls can be attributed to security vulnerabilities or incorrect system functioning. Solutions were proposed to address configuration-related issues such as incorrectly structured firewall rules, repetition of similar rules, or missing rules.

A recent study on classifying firewall logs aimed to identify key features that influence network traffic [6]. The dataset log was retrieved from a firewall with 1,048,576 data points from 10 days. Logs were classified and analyzed using artificial intelligence and deep learning methods. Aljabri et al. conducted two experiments in their study. The first used 11 features from firewall log fields, and the second added application and category fields, resulting in a total of 13 features. The logs' "action" results of "deny," "allow," "drop," and "Reset-Both" were utilized. Since the lowest category of the results is "Deny," the remaining categories were undersampled. Various algorithms were executed, including KKN, NB, RF, J48, and ANN. The Random Forest algorithm yielded the best results, achieving 99.11% accuracy in the first experiment and 99.64% in the second, based on the number of correctly classified instances.

The firewall actions differ for each packet it receives. Similar to [6], al Haija et al. employed similar firewall actions, such as "Allow," "Deny," or "Drop/Reset," for incoming and outgoing network traffic based on predefined rules [7]. It aimed to present an intelligent classification model that accurately classifies decisions made by the firewall using machine learning methods [7]. Shallow Neural Network (SNN) and Optimizable Decision Tree (ODT) algorithms were used. The dataset used in the study was obtained from the logs of Firat University's firewall device, which consisted of 11 result-producing log fields. The dataset was divided into 15% validation, 15% test, and 70% training. Since the "Reset-Both" category had minimal

records, it was merged with the "Drop" category. According to the classification results, SNN and ODT accuracies were 98.50% and 99.8%, respectively. Ertam and Kaya used the same dataset to estimate classification using a support vector machine (SVM) in their study [8]. They observed which kernel function yielded more effective results on classification. To reach this conclusion, they looked at F1, Precision, and Recall value results. The best recall value of 98.5% was achieved using the Sigmoid function for classification. The linear function classification observed the highest precision value of 67.75%. The best F1 score of 76.4% was obtained using the RBF function for classification. Both precision and recall values were significantly lower than the other if the polynomial kernel functions were used. The average values of precision, recall, and F1 scores showed that RBF classifications were the most optimal. Al-Tarawneh and Bani-Salameh analyzed the log records of the same dataset of Firat University's security firewall device. They used K-Nearest Neighbors (KNN), Random Forest (RF), and Deep Neural Network (DNN) classifiers [9]. Comparisons were conducted by evaluating the classifier's performance using accuracy, recall, precision, and F1-score metrics. The Random Forest algorithm selected the best features according to the analysis results.

Anomaly detection is another common topic in computer networks [10-12]. Website anomaly detection is one type of it [10], and the authors employed the Negative Selection algorithm, which belongs to the Artificial Immune System (AIS) [10]. The Yahoo Webscope S5 dataset encompassed abnormal and normal result-producing data traffic. The dataset comprised four different classes with 367 time series signal patterns. These signal patterns contained 1500 data segments, totaling 5,050,000 data points. Seven experiments were conducted, resulting in a range of 93.18% to 98.33%. The average correct classification rate for all experiments was 94.30%. It was concluded from the experiments that the negative selection algorithm of AIS could be used to detect anomalies in web traffic.

Sahin developed a security firewall for the application layer and classification algorithms to visualize the accuracy [13]. The open-source CSIC 2010 HTTP dataset comprised Post, Get, and Put requests in traffic. Their accuracy values were calculated, an attribute table was formed, and classification procedures were conducted. The dataset was divided into training and test sets with a 70:30 ratio. The C4.5 decision tree algorithm yielded the highest accuracy at 96.26%. However, the comparable results achieved by other algorithms suggest that the key to success lies more in the effective selection of attributes rather than the specific choice of algorithm.

A machine learning model was trained using Deep Neural Networks (DNN), which was trained using the dataset obtained from the internet security firewall of the UCI Machine Learning Center [14]. Eleven features were used, and a model was generalized due to the dataset's imbalance. K-fold cross-validation was applied to regularize the model, preventing overfitting [14]. The accuracy of 94.49% with test data and 95.81% with training data demonstrated better performance than similar studies.

Web Application Firewalls (WAFs) protect web applications from various types of attacks. Signature-based approaches are defended against threats by blocking malicious traffic with application-specific rules [15]. However, these approaches may become vulnerable to unexpected threats, such as zero-day attacks. Machine learning-based WAFs are more flexible and adaptable than traditional methods, producing fewer false positives and negatives. Studies in the literature have shown that this new approach is particularly effective against zero-day attacks. Therefore, employing machine learning-based approaches in WAFs might offer a more effective defense mechanism against existing and unexpected attacks. However, it is worth noting that current research is limited in determining the effectiveness of this new approach against attack types targeting web applications.

Institutions' security policies are implemented as security firewall rules. Any abnormalities in these rules can lead to security vulnerabilities. In cases where networks are extensive and rules are complex, the manual inspection might be insufficient, making it difficult to detect abnormalities. Ucar and Ozhan proposed an automatic model based on machine learning and high-performance computing methods to detect abnormalities in security firewall rules [16]. For this purpose, security firewall records were analyzed, and the extracted features were evaluated using machine learning classification algorithms, including Naive Bayes, k-Nearest Neighbor (kNN), Decision Tree, and HyperPipes. F-Measure was used for performance evaluation. In experiments, kNN showed the best performance. Subsequently, a model based on the F-Measure distribution was predicted. This model analyzed 93 security firewall rules and predicted six rules as anomalies. It is observed that machine learning methods can automatically analyze large-scale log files and detect abnormalities in security firewall rules.

The model presented by Kulyadi et al. aimed to understand normal behaviors by learning the relationships and complex structures between messages recorded by the firewall over time [17]. A Generative Adversarial Network (GAN) model was introduced to detect anomalies in time series data. Various anomaly detection techniques were applied based on the normal behavior model that was obtained. Reconstruction errors were calculated using cosine similarity, and two different encoding techniques were presented to feed these data. Experimental results showed that using both encoding techniques, the GAN model outperformed the autoregressive integrated moving average (ARIMA) and long short-term memory (LSTM) models.

A model recommended using machine learning techniques to detect web attacks for WAFs [18], focusing on general and comprehensive features extracted from HTTP requests to address deficiencies. These features were calculated using four different datasets: CSIC 2010, HTTPParams 2015, a hybrid dataset (combining CSIC 2010 and HTTPParams), and logs from a compromised web server. Basic and extracted features were combined to classify normal and abnormal requests using

Logistic Regression, Decision Tree, and Naive Bayes algorithms. The proposed model achieved a classification accuracy of 99.6% with standard datasets and 98.8% with real web server datasets. The effectiveness of machine learning-based methods in web application firewalls has been demonstrated.

Our study utilizes a dataset from TME firewalls to identify the most influential parameters affecting network traffic, employing multiple linear regression (MLR) and principal component analysis (PCA). The study's second phase shows how it could affect the accuracy of estimations of firewall rules using evaluation metrics. We have two hypotheses in this study:

H1: Applying MLR+PCA feature selection yields higher ANN performance than training on all 26 variables.

H2: Reduced models are less prone to specificity degradation across hours than full-feature models.

3. Materials and Methods

The ability of information systems to learn from data and utilize these learnings to fulfill requested tasks or enhance problem-solving capabilities represents a branch of artificial intelligence.

In the digitized world, the utilization of concepts such as cybersecurity, the internet of things (IoT), big data, cloud computing, artificial intelligence, and machine learning has increased [19]. With the advancement of the digital world, there has also been a significant surge in data produced. Based on life experiences and acquired knowledge, humans can make specific predictions. However, in situations where humans make predictions, emotions often influence decisions, making it challenging to make genuinely objective decisions. Moreover, individuals may miss specific details in data-rich scenarios, leading to erroneous decisions [19]. Machine learning methods are employed to derive meaningful insights from acquired data.

Supervised Machine Learning reflects an algorithm's ability to generalize information derived from existing labeled or target-oriented data. In this context, the algorithm predicts new, unlabeled data [20]. The primary objective of supervised learning is for any model to analyze its data first and then generate the best-matching outputs for other data. Initially, the algorithm necessitates a labeled dataset for training. Subsequently, a model is constructed, trained, and validated with a separate dataset. Its performance is measured using the obtained results. In this study, the Artificial Neural Network algorithm was utilized.

Unsupervised machine learning is a method used to determine patterns and relationships within data. Unsupervised machine learning algorithms aim to understand the structures of datasets and explore them without the need for labels or outcome variables in the data. Unsupervised machine learning algorithms, commonly known as clustering algorithms, do not require knowledge of class labels for any data because they compute similarities among dataset elements and then partition the dataset into various segments, called clusters, based on these similarities [21].

The study used MLR and ANN, supervised learning algorithms, and PCA, an unsupervised learning algorithm. Details regarding these algorithms are provided below.

During the data preprocessing stage, as detailed in Section 4, the study, MLR, and PCA algorithms were applied to determine which parameters (variables) in the datasets had the most significant influence.

3.1. Multiple linear regression

Multiple linear regression (MLR) is a statistical method used to model the relationship between a dependent variable and one or more independent variables. The objective of MLR is to model the linear relationship between the independent variables—'x's—and the dependent variable 'y' [22].

The MLR quantitatively measures the impact of each independent variable on the dependent variables. The coefficient of each variable represents the effect of the corresponding feature on the dependent variable. Models are created using equations that express the relationships between coefficients and independent variables.

Various methods and criteria were employed to create models for MLR. The methods used include forward selection, backward elimination, and stepwise selection models. Information criteria, such as the Akaike Information Criterion (AIC), Schwarz Bayesian Criterion (SBC), and R-squared criterion, were used [23,24].

3.2. Artificial neural network

It is a mathematical model inspired by the human brain [9]. ANNs rely on their ability to adapt to changing situations by adjusting their weight and learning from experiences. Artificial neurons are organized in layers. These layers consist of a series of neurons performing similar functions. Input, output, and hidden layers are the types of these layers [25].

For an ANN to be capable of learning weights and making generalizations on data, it must first be trained. Training is achieved by the method known as backpropagation, where errors in the output are propagated backward to update the weights. This flexible structure enhances the ability of ANNs to solve complex problems and improve learning capabilities.

The classifier was implemented with scikit-learn's MLPClassifier. The network comprises two hidden layers, each with 50 units (hidden_layer_sizes=(50, 50)), and uses ReLU hidden activations. Scikit-learn applies a logistic/softmax output with

cross-entropy (log loss) internally for binary classification. Optimization uses Adam (solver='adam') with a constant learning rate schedule (learning_rate='constant') and an initial learning rate of 0.001 (learning_rate_init=0.001). We apply L2 (weight-decay) regularization = 1e-4 (alpha=0.0001). Mini-batch size follows scikit-learn's auto rule (batch_size='auto', i.e., min(200, n_samples)), and samples are shuffled each epoch (shuffle=True). Training runs for up to 200 iterations per fit (max_iter=200) with a convergence tolerance of 1e-4 (tol=1e-4). We do not use early stopping (early_stopping=False), so no validation hold-out is created; the optimization tolerance and iteration cap determine convergence. To ensure reproducibility, the main experiment uses a 70/30 train/test split (test_size=0.3) with a fixed seed (random_state=42). We also repeated training over 20 additional splits (with the same 70/30 proportion) without fixing random_state and reported summary metrics across runs to assess variability. No class reweighting or resampling was applied, and the inputs to the MLP were the selected six numeric features. Prediction thresholds are set to the scikit-learn defaults (0.5 for binary decisions). .3.3. Principal component analysis.

PCA, introduced by Karl Pearson in 1901, is an effective statistical method [18]. Large datasets pose a problem due to the high computational costs they can incur. Therefore, reducing the dimensionality of datasets is often desired to enhance system performance and retain as much information as possible from the original data [26].

Let $X \in R^{n \times p}$ be the data matrix and Z its column-centered version (optionally standardized). PCA finds directions v_k that maximize variance by solving the eigenproblem of the sample covariance $S=(Z^TZ)/(n-1)$ [27, 28]:

$$S\mathbf{v}_k = \lambda_k \mathbf{v}_k \, , \lambda_1, \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0 \quad (1)$$

Scores (principal components) are the projections of the centered data onto these directions:

$$\mathbf{t}_k = Z\mathbf{v}_k \;\; and \; T = ZV \quad (2)$$

Components are ordered by λ_k (variance captured); we report their proportions in the text and use the leading components to guide variable reduction.

PCA is primarily an unsupervised machine learning technique widely used to reduce the dimensionality of datasets and understand the relationships between variables in these datasets. It is one of the simplest and most commonly used dimensionality reduction methods [26].

PCA aims to reduce the number of variables in high-dimensional datasets and comprehend the complexity among these variables. The study employed PCA to explore the relationships among the data in five different datasets, each with 26 parameters, to identify the most effective data.

3.4. Proposed model evaluation metrics

Accuracy, Precision, Recall, and F1-score metrics measure the performance of classification and categorization estimations. For each metric, the higher the metric value, the higher the classifier's performance [29]. The binary classification was employed for all five datasets. In binary classifications, when tested, we obtain four different counts: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) [29]. The accuracy, precision, recall, F1-score, and specificity values are calculated using Equations 3, 4, 5, 6, and 7.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + FP + TN + FN}} \quad (3)$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}} \quad (4)$$

$$\boldsymbol{Recall} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} \quad (5)$$

$$\boldsymbol{F1\ Score} = \boldsymbol{2} * \frac{\boldsymbol{Precision} * \boldsymbol{Recall}}{\boldsymbol{Precision + Recall}} \quad (6)$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN + FP}} \quad (7)$$

4. Data Preparation

The Directorate General of TME has a FortiGate 101F security firewall manufactured by Fortinet. The logs from this security firewall are forwarded via syslog redirection to the FortiLogger software, where they are parsed and transformed into a readable format. The datasets have been collected from the FortiGate firewall device. Five log files were generated to create the dataset. The dates and time intervals of these log files are shown in Table 1 below.

The output obtained from the firewall was saved as CSV files. They were transferred to Microsoft SQL Server to manage the log files in a common environment, and specific programming tasks were performed.

Data preprocessing involves organizing, modifying, and filling in missing data, as well as performing other operations on the datasets. The raw data obtained from the log fields were subjected to data preprocessing to make them suitable for analysis. The parsed log fields generated by the security firewall are detailed in Table 2. The first step in the preprocessing was removing empty variables. The remaining 29 log variables were converted to numerical values in the second step. The data obtained from log files comes in different types, including numerical and textual data. For instance, fields like 'srcport' and 'dstport' are numerical, while 'srcip,' 'dstip,' 'apprisk,' and similar log fields are textual. To ascertain the significance of the data on the final results in subsequent stages and to ensure more accurate outcomes with the algorithms used for prediction, all data in the dataset has been converted into a numerical format. Several data points, both before and after preprocessing, are presented in Figure 1. Approximately 10,000 data points were cleaned in each log during the preprocessing step. Textual/categorical fields used in modeling were encoded as follows: levels, proto, appcat, profiles, fn_direction → one-hot encoding (binary indicators per category) without imposing ordinal structure; apprisk → ordinal integers 1–5 reflecting the device's native risk scale. IP-like identifiers (e.g., srcip, dstip) were not used directly as features. All numeric fields were min–max scaled within each log to [0,1] after train/test splitting to avoid leakage.

Table 1. Collected the log information

Dataset Name	Date	Hour
Log 1	15-08-2022	11:00 – 12:00
Log 2	24-08-2022	15:00 – 16:00
Log 3	08-09-2022	10:00 – 11:00
Log 4	20-09-2022	14:00 – 15:00
Log 5	09-11-2022	11:00 – 12:00

Table 2. Variables prepared from firewall logs and their descriptions

Variable Name	Long Version	Description
SrcIP	Source IP	Indicates the source's IP address.
SrcPort	Source Port	Indicates the source port.
DstIP	Destination IP	Indicates the destination IP address.
DstPort	Destination Port	Indicates the destination port.
Appcat	Application Category	Indicates the category of the used application.
SrcIntf	Source Interface	Indicates the source interface on the firewall.
DstIntf	Destination Interface	Target interface on the firewall.
SubType	Sub-Type	Subcategory of the traffic
Level	Level	Indicates the security level.
PolicyId	Policy-ID	ID value representing firewall rules.
Proto	Protocol	It represents the used protocol.
AppList	Application List	The field indicates which security rule generated by the application control was passed through.
Profile	Profile	The field indicates which security rules it passed through for web-based (WebFilter, DNS, System) traffic.
PolicyName	Policy Name	The field indicates the name of the conducted traffic.
App	Application	The field indicates which application the log traffic was made with.
TrandIsp	Nat Translation	Indicates the type of NAT performed.
SentByte	Sent Byte	The number of bytes sent by the firewall.
TransIP	Translation IP	The IP address to which the traffic is being NAT.
RcvdByte	Received Byte	The number of bytes received by the firewall.
AppRisk	Application Risk	The risk level under application control.
TransPort	Translation Port	The port where the NAT operation is performed.
Action	Action	The firewall provides the outcome as a result of the traffic.
SrcIntfRole	Source Interface Roles	Shows which role the Source Interface belongs to
User	User	The name of the user generating the traffic found by the FSSO Agent.
DstIntfRole	Destination Interface Roles	The role associated with the destination interface.
SrcCountry	Source Country	The location where the source is located.
PolicyType	Policy Type	Indicates whether any policy is applied or not.
Cat	Category ID	It specifies the ID of the category in web traffic.
CatDesc	Category Description	It indicates the description of the category in web traffic.

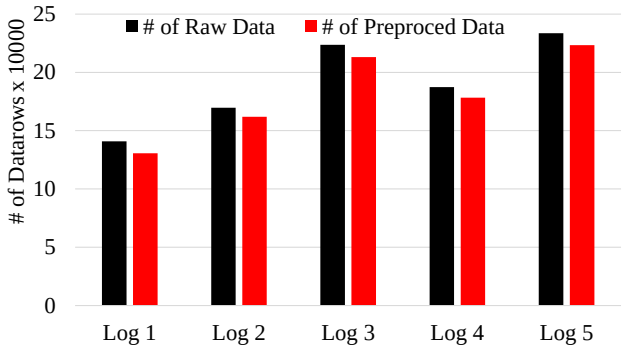

Figure 1. Raw and processed data numbers

4. Results

In this section, the selection results of the most influential variables will be presented, and the accept-reject status of traffic on the network will be estimated using an ANN. MLR and PCA were employed to examine the parameters significantly impacting the output in the datasets.

MLR was initially utilized in the analysis process using the Statistical Analysis Software (SAS) web platform. A 95% confidence interval was chosen for the operations. MLR models were constructed using three different approaches: forward selection, backward elimination, and stepwise. All these methods were employed to calculate the impact of each dataset's variables on the logs. Three criteria were used to find the best MLR model using the three approaches. These were AIC, SBC, and R^2. Consequently, processing the datasets involved obtaining results across all possible combinations.

The t-value was used as a reference to assess the influence of log fields (variables) on the output in the datasets. Absolute values of the t-value were considered in descending order to prioritize the effects (Table 3). Following this ranking, it was determined that the highest t-value holds the most significant impact.

For each dataset, rankings were established based on models and criteria, and the average values of these rankings were computed. Subsequently, a new ranking was generated based on these average values.

Following the analyses, attention was given to the rankings derived from the average values. In this context, it was determined that, according to the linear regression method, 'levels,' 'proto,' 'appcat,' 'apprisk,' 'profiles,' 'dstinnfrole,' 'fn_direction,' and 'trandisp' parameters have a more significant impact on the output.

Table 3. MLR t-values result.

Variable	T-Value	Rank
levels	177.51	1
proto	101.19	2
appcat	-76.7	3
apprisk	64.58	4
dstintfrole	42.8	5
profiles	29.14	6
app	26.71	7
srccountry	-19.91	8
fn_direction	-19.51	9
applist	-18.57	10
users	18.41	11
srcintf	-17.89	12
trandisp	16.42	13
Transip	-15.47	14
cat	15.26	15
srcintfrole	-13.11	16
dstip	12.24	17
policyid	12.15	18
policyname	-8.87	19
srcip	4.89	20
srcport	3.41	21
subtype	-2.9	22
dstport	-1.68	23

A similar process was performed with PCA after identifying parameters with a more significant impact on the results using MLR. All variables except the output were selected for analysis. The analysis involved 26 different principal components. Table 4 presents three principal components and the results of 12 variables as an example. Each variable has a coefficient associated with each principal component; the values of the variables are multiplied by their respective coefficients based on the components.

Table 4. The first three principal component values of PCA

Variable	Prin1	Prin2	Prin3
srcip	0.013137	-0.10555	-0.00474
srcport	-0.00596	0.031809	0.045008
dstip	0.000151	-0.03505	0.241091
dstport	0.041506	-0.01788	-0.14743
srcintf	0.000535	-0.06307	0.177739
srcintfrole	0.001584	-0.08541	0.476522
dstintf	-0.00487	-0.03088	0.278789
dstintfrole	0.003235	0.109204	-0.40813
appcat	0.27566	0.197122	0.003896
subtype	-0.37778	0.064988	0.069135
levels	-0.11948	0.431319	0.052395
policyid	0.078101	-0.21846	-0.29393

Each principal component has an impact on the PCA model. In most cases, the first three principal components account for 50% of the dataset's explanation. Proportion values of the first three components are given in Table 5. The impact ranking was calculated for PCA. The proportions of the components were sorted in descending order (Table 5).

Table 5. Proportion value for components

Component	Proportion
1	0.2582
2	0.1318
3	0.1206

When calculating the impact ranking, each parameter in the relevant component is multiplied by the proportion value to calculate the Result1 value (Equation 8).

$$\boldsymbol{Result1 = Impact\ Ranking * Proportion} \quad (8)$$

The component value is multiplied by the proportion value to calculate the Result2 value for each component (Equation 9).

$$\boldsymbol{Result2 = Component\ Value * Proportion} \quad (9)$$

These mathematical operations are conducted for 26 components, producing results for each component. The results fall into three different categories: for each dataset, the average ranking based on all parameters' component values, the 'Result1' value representing the average of the multiplication of variables' impact ranking by the proportion value, and the 'Result2' value representing the average of the multiplication of component value by the proportion value.

According to the resulting values, a lower value in the "Impact Ranking" and "Result1" average signifies a higher impact degree for the variables. Meanwhile, a higher value in the "Result2" average indicates a higher degree of impact for the variables.

Table 6. Mutual parameters affecting the result

MLR	PCA
levels	policyid
proto	**apprisk**
appcat	**proto**
apprisk	applist
profiles	**appcat**
dstinnfrole	**levels**
fn_direction	cat
trandisp	transip
	profiles
	fn_direction
	policyname

Three outcomes were derived after the analyses were conducted on five different datasets. As observed in the PCA analysis, specific parameters have a greater impact on the output. These parameters include 'policyid,' 'apprisk,' 'proto,' 'applist,' 'appcat,' 'levels,' 'cat,' 'transip,' 'profiles,' 'fn_direction', and 'policyname', totaling 11 parameters that have a more significant effect on the output.

The analyses conducted using MLR also revealed that the parameters having the most significant impact on the output were identified as 'levels,' 'proto,' 'appcat,' 'apprisk,' 'profiles,' 'dstinnfrole,' 'fn_direction,' and 'trandisp,' as indicated above. In the PCA analyses, it was determined that the parameters with the most impact on the output were 'policyid,' 'apprisk,' 'proto,' 'applist,' 'appcat,' 'levels,' 'cat,' 'transip,' 'profiles,' 'fn_direction,' and 'policyname'. The comparison of the most significant variables among the methods is presented in Table 6. We merged overlapping statements in Table 6 and clarified that mutual variables across MLR and PCA ("levels," "proto," "appcat," "apprisk," "profiles," "fn_direction") form the candidate set for reduction.

In the data arrangement section, following the examination and organization of log records obtained from the firewall traffic, 27 log variables were organized into five datasets, along with the outputs. MLR and PCA algorithms were used to identify which log fields had a significant impact on the output. Through these algorithms, six different log fields that had the most impact on the outcome were identified. New datasets were then formed using these log fields.

5.1. Estimation of firewall traffic permissions

Two different datasets were prepared, one comprising 26 parameters and the other 6. These datasets contain 70% training data and 30% test data. Initially, datasets containing 27 variables, including the output, were employed. Subsequently, datasets with only 7 variables, including the output, were employed.

The ANN models consist of two hidden layers, each with 50 neurons. Ten ANN models were prepared and trained. The number of neurons in the input layer depends on the number of variables. Six variables include models constructed with six neurons, and 26 variables include models constructed with 26 neurons. There are two output values, zero and one. Models were trained with the training datasets.

The predictions were conducted using an ANN on five different test datasets, and the estimation results are presented in Table 7. Following the predictions using ANN on Log 1, Log 2, Log 3, Log 4, and Log 5 datasets. When examining the accuracy value across the entire dataset series, it was observed that the accuracy of all datasets with six variables was higher than that of their respective scenarios with 26 variables. This suggests that reducing the number of variables in the dataset has a positive impact on the model, increasing its flexibility. The increased predictive power with fewer variables demonstrates the significance of the identified variables affecting the model. Furthermore, noticeable improvements are seen in accuracy, recall, precision, specificity, and F1-score values.

Table 7. Results based on the artificial neural network algorithm with 26 and 6 parameters in the datasets.

Dataset	# of variables	F1-Score	Recall	Precision	Specificity	Accuracy
Log 1	26	0.78732	0.65218	**0.99311**	**0.98825**	0.74559
	6	**0.90020**	**0.82975**	0.98374	0.96438	**0.86717**
Log 2	26	0.86600	**0.99434**	0.76701	0.23552	0.77944
	6	**0.89652**	0.85819	**0.93843**	**0.85750**	**0.85799**
Log 3	26	0.81727	0.69327	**0.99528**	**0.99141**	0.77572
	6	**0.91383**	**0.88690**	0.94243	0.85830	**0.87899**
Log 4	26	0.79498	0.67304	**0.97088**	**0.95258**	0.75650
	6	**0.90303**	**0.87921**	0.92819	0.84019	**0.86756**
Log 5	26	0.89782	0.86386	0.93456	0.80924	0.85071
	6	**0.92112**	**0.89543**	**0.94833**	**0.84614**	**0.88357**

When the average results of all datasets are obtained, the F1-Score, Recall, Precision, Specificity, and Accuracy values for models with 26 variables are 0.83268, 0.77534, 0.93217, 0.7954, and 0.7816, respectively. On the other hand, for models with six variables, these values increased by 8.92%, 12.2%, 1.72%, 9.79%, and 11.45%, respectively, resulting in values of 0.90694, 0.8699, 0.94822, 0.8733, and 0.87106, respectively.

Across all datasets, it is evident that reducing the number of parameters from 26 to 6 generally leads to better performance metrics. Specifically, F1-Score, Recall, Precision, and Accuracy consistently improved with fewer parameters. This suggests that the neural network algorithm benefits from a more streamlined parameter set, possibly due to reduced overfitting and enhanced generalization. Specificity showed significant variations, especially in Log 2, indicating potential issues with false positive rates in some datasets when more parameters are used. Overall, the results highlight the importance of parameter selection in improving the performance of machine learning models.

Figure 2 presents TP, TN, FP, and FN values for each dataset and model. In all datasets, the number of logs allowing traffic is greater than that of logs denying traffic. In this context, TP in Figure 2 represents the number of logs that allow traffic. Precision, indicating the accuracy of predicting permitted traffic, was higher in 6-variable datasets than in 26-variable datasets, except for the Log 1 dataset. In the Log 1 dataset, the 6-variable dataset showed approximately a 1% lower success rate.

The value shown as TN in Figure 2 indicates the accuracy with which the blocked traffic in the datasets was predicted. In this context, the specificity value in Table 7 demonstrates the effectiveness of predicting blocked traffic in the analysis. Upon examining Table 7, it can be observed that the specificity value in Log 1, Log 3, and Log 4 datasets resulted in lower successful outputs in the 6-variable datasets compared to the 26-variable datasets, while in Log 2 and Log 5 datasets, it performed better. The notably lower specificity for Log 2 under the 26-variable configuration is consistent with (i) a more imbalanced "allow"/"deny" distribution in that hour and (ii) the susceptibility of high-dimensional inputs to spurious correlations, which inflate FP rates. After reduction to six variables, specificity improves markedly, indicating the reduced set curtails noise while preserving signal.

According to the results presented in Table 7, the F1-score values in the 6-variable datasets yielded better outcomes compared to those in the 26-variable datasets. When reviewing the Recall values, except for the Log 2 dataset, all datasets in the 6-variable format yielded better results than the 26-variable datasets.

When evaluating the prediction results obtained through an ANN to identify variables with MLR and PCA, it becomes evident that the correct variables were identified within the dataset. We compared No selection and MLR+PCA using the same ANN and splits. MLR+PCA provided the most consistent improvements in F1 and recall across all logs, indicating complementary benefits of coefficient-based ranking

Analyzing the counts of TP, TN, FP, and FN across datasets, it is evident that reducing the number of parameters from 26 to 6 leads to varied impacts on the performance metrics. Across all logs, the number of TPs increases significantly with fewer parameters, indicating an improvement in correctly identifying positive cases. There is a mixed impact on TN, with some logs showing a decrease (e.g., Log 3) and others showing an increase (e.g., Log 2), suggesting variability in the model's ability to identify negative cases correctly. The count of FP generally increases with fewer parameters, as observed in Logs 1 and 2, indicating more instances where the model incorrectly identified positive cases. The number of FN decreases across all logs with fewer parameters, showing a substantial improvement in minimizing missed positive cases.

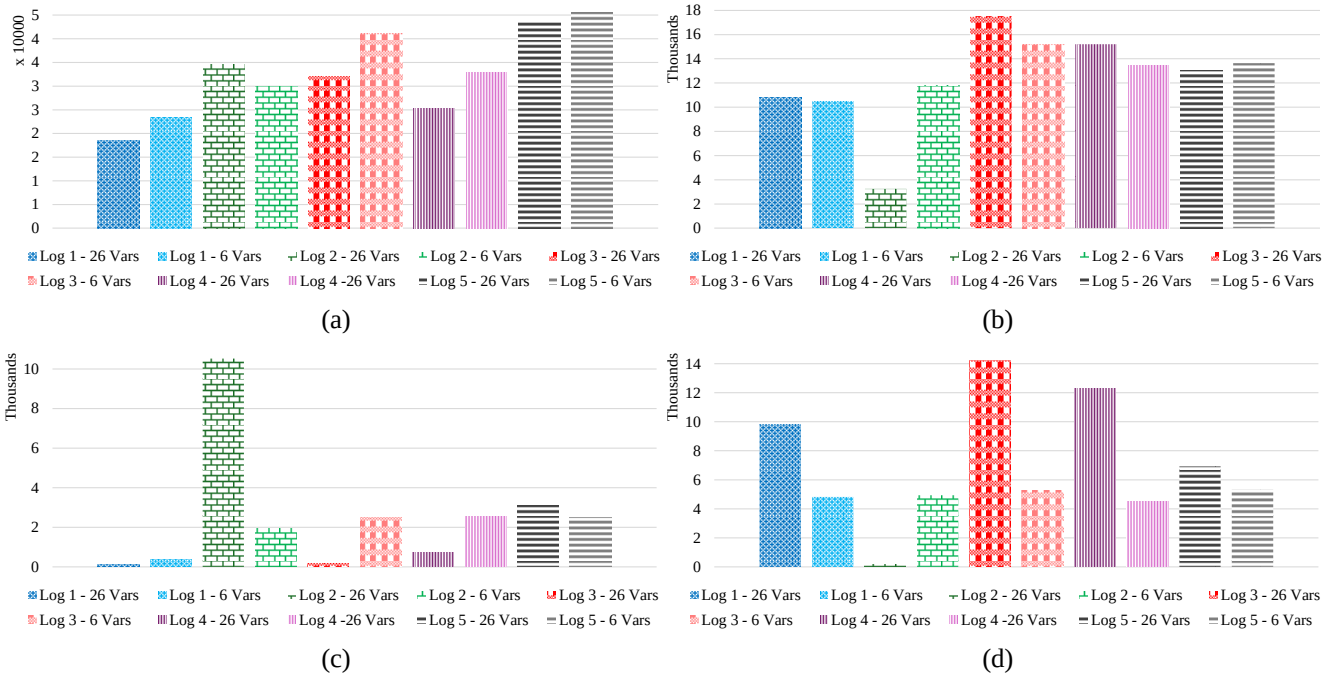

Figure 2. The evaluation metrics are in datasets with 26 and 6 parameters. Numbers of (a) TP, (b) TN, (c) FP, and (d) FN

Reducing the number of parameters from 26 to 6 enhances the model's ability to correctly identify positive cases (TP) and reduce the number of false negatives (FN). However, this comes with increased FP in some datasets, which suggests a trade-off between sensitivity and specificity. These results underscore the importance of parameter selection in optimizing model performance and balancing the identification of true cases while minimizing incorrect identifications.

6. Conclusions and Discussion

This study collected logs from the FortiGate brand firewall used in TME over five days and times, specifically during office hours. These logs were processed through various software and programs to make them usable. Five different datasets were generated, each comprising 26 variables that produced output.

Our datasets derive from a single FortiGate deployment and five one-hour office-time windows, which may limit generalization to other vendors, policies, and traffic mixes (e.g., off-hours or peak loads). Future work will expand temporal coverage (full multi-week windows), include controlled attack traffic and stress periods, and validate across multiple firewall brands to assess portability of the selected variables.

The aim was to determine whether it was possible to predict results using fewer variables by identifying the variables that had the most impact on the output using MLR and PCA methods. After performing operations with MLR and PCA, 6 variables were identified as having a significant impact on the output: “levels,” “proto,” “applicant,” “app risk,” “profiles,” and “fn_direction. "

When these significant variables were tested using the ANN to estimate output, the 6-parameter datasets yielded more successful results than the 26-parameter datasets, based on accuracy, recall, and F1-Score metrics, in most cases. However, in some datasets, the 26-variable datasets performed better in terms of specificity and precision metrics than the 6-variable datasets.

The successful reduction of datasets using the methods employed to identify influential parameters is evident. With the newly obtained datasets, the ANN demonstrated an accuracy range of 85% to 88% and a precision range of 92% to 98% when predicting with the datasets. Furthermore, the F1-score ranged between 89% and 92% upon examination.

For on-premises inference, a single CPU core processes ~thousands of events/second with sub-millisecond per-record preprocessing (one-hot expansion over $\leq O(100)$ categories) and sub-millisecond ANN forward pass. Batch scoring amortizes overheads. The reduced 6-variable representation minimizes memory and cache pressure. Data privacy is maintained by processing on a secure infrastructure; no payload contents are required—only metadata already logged by the firewall.

In conclusion, adequate data could be identified using MLR and PCA methods. Furthermore, depending on the dataset, the ANN demonstrated a high accuracy level in determining whether firewall traffic would be blocked or permitted.

References

[1] H. N. K. Al-Bedahili, “Decision Tree for Multiclass of Firewall Access,” *in International Journal of Intelligent Engineering and Systems*, vol. 14, no 3, pp. 294-302, 2021.

[2] İnternet Ortamında Yapılan Yayınların Düzenlenmes ve Bu Yayınlar Yoluyla İşlenen Suçlarla Mücadele Edilmesi Hakkında Kanun, 23 Mayıs 2007 26530 Sayılı Karar, Resmi Gazete, 2007

[3] E. Özhan, “Güvenlik Duvarı Günlüklerinin Makine Öğrenmesi Yöntemleri ile Analizi ve Bir Model Çıkartılması,” PhD Dissertation, Trakya University, Turkey, 2013.

[4] E. Akbaş, “Bilgi Güvenliği ve Log Yönetimi Sistemlerinin Analizi,”, Web Sayfası: http://www. academia. edu/9203287/Bilgi_G% C3% BCvenli% C4% 9Fi _ve_Log_Y% C3% B6netimi_Sistemlerinin_Analizi, 2017

[5] A. E. Mohammed, “Güvenlik Duvarı Kurallarına Ait Anomalilerin Tespiti ve Optimizasyonu,” M.S. Dissertation, Sakarya University, Turkey, 2018.

[6] Aljabri, M., Alahmadi, A. A., Mohammad, R. M. A., Aboulnour, M., Alomari, D. M., & Almotiri, S. H., “Classification of firewall log data using multiclass machine learning models,” *in Electronics* vol. 11, No.12, 1851, 2022.

[7] Q. A. Al-Haija, A. Ishtaiwi, “Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense,” *International Journal on Advanced Science, Engineering and Information Technology*, vol. 11, No. 4, pp. 1688-1695, 2021.

[8] F. Ertam, M. Kaya, “Classification of Firewall Log Files With Multiclass Support Vector Machine,” *in International Symposium on Digital Forensic and Security (ISDFS)*, Antalya, Turkey, 2018, pp. 1-4.

[9] B. A. Al-Tarawneh, H. Bani-Salameh, “Classification of Firewall Logs Actions Using Machine Learning Techniques and Deep Neural Network*,” in AIP Conference Proceedings*, Jordan, Amman, 2023, vol. 2979, No. 1.

[10] K. İlhan, “Web Trafik Verilerinde Yapau Bağışıklık algoritmaları ile Anomali Tespiti,” M.S. Dissertation, Bilecik Seyh Edebali University, Turkey, 2019.

[11] K. Ovaz Akpinar, I. Ozcelik, “Anomaly detection on EtherCAT based water level control automation,” *in 2020 5th International Conference on Computer Science and Engineering (UBMK),* Diyarbakir, Turkey, 2020, pp. 79-82.

[12] K. Ovaz Akpinar, I. Ozcelik, "Methodology to determine the device-level periodicity for anomaly detection in EtherCAT-based industrial control network," *IEEE Transactions on Network and Service Management, 18*, 2, 2308-2319, 10.2020.

[13] M. Şahin, "Uygulama Katmanı için Güvenlik Duvarı Geliştirilmesi," M.S. Dissertation, Gebze Teknik University, Turkey, 2017.

[14] C. Lillmond, G. Suddul, "A Deep Neural Network Approach for Analysis of Firewall Log Data," in International Conference on Advances in Computing and Technology (ICACT–2021) Faculty of Computing and Technology (FCT), University of Kelaniya, Sri Lanka, 2021, pp. 42-46.

[15] S. Applebaum, T. Gaber, A. Ahmed, "Signature-based and Machine Learning-based Web Application Firewalls: A Short Survey" *in Procedia Computer Science*, vol. 189, pp. 359-367, 2021.

[16] E. Uçar, E. Özhan, "The Analysis of Firewall Policy Through Machine Learning and Data Mining" *in Wireless Personel Communications*, vol. 96, pp. 2891-2909, 2017.

[17] S. P. Kulyadi, p. Mohandas, S. K. S. Kumar, M. J. S. Raman, V. S. Vasan, "Anomaly Detection Using Generative Adversarial Networks on Firewall Log Message Data," *in 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI),* Pitesti, Romania, 2021, pp. 1-6.

[18] A. Shaheed, M. H. D. Bassam Kurdy, "Werb Application Firewall Using Machine Learning and Feature Engineering," *in Security and Communication Networks*, vol. 2022, 2021.

[19] E. Tosunoğlu, R. Yılmaz, E. Özeren, Z. Sağlam, "Eğitimde Makine Öğrenmesi: Araştırmalardaki Güncel Eğilimler Üzerine İnceleme", *in Ahmet Keleşoğlu Eğitim Fakültesi Dergisi,* vol. 2. No., 2., 2021.

[20] M. W. Berry, A. Mohamed, B. W. Yap, "Supervised and Unsupervised Learning for Data Science," *in Springer Nature*, Cham, Switzerland, 2020.

[21] A. Şenol, Y. Canbay, M. Kaya, "Trends in Outbreak Detection in Early Stage by Using Machine Learning Approaches," *in Bilişim Teknolojileri Dergisi*, vol. 14. No. 4, pp. 355-366, 2021.

[22] D. H. Maulud, A. M. Abdulazeez, "A Review on Linear Regression Comprehensive in Machine Learning", *in Journal of Applied Science and Technology Trends*. vol. 1, No. 4, pp. 140-147, 2020.

[23] K.P. Burnham, Multimodel Inference: Understanding AIC and BIC in Model Selection. Sociological Methods & Research, 33, 2, 261–304, 2004.

[24] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, Forecasting: Methods and Applications, NY, USA, John Wiley: New York, 2008.

[25] İ. M. Nasser, M. O. Al-Shawwa, S. S. Abu-Naser, "A Proposed Artifical Neural Network for Predicting Movies Rates Category" *in Interrnational Journal of Academic Engineering Research (IJAER),* vol. 3. No. 2, pp. 21-25, 2019.

[26] M. Kayakuş, M. Terzioğlu, "Yapay Sinir Ağları ve Çoklu Doğrusal Regresyon Kullanarak Emeklilik Fonu Net Varlık Değerlerinin Tahmin Edilmesi", *Bilişim Teknolojileri Dergisi*, vol. 14, No. 1, pp. 95-103, 2021.

[27] I. T. Jolliffe, Principal component analysis: With 28 illustrations. New York, USA: Springer, 2002.

[28] T. Hastie, J. Friedman, R. Tisbshirani, The elements of Statistical Learning: Data Mining, Inference, and prediction. New York: Springer, 2017.

[29] A. Akhtar, S. Akhtar, B. Bakhtawar, A. A. Kashif, N. Aziz, M. S. Javeid, "Covid-19 Detection From CBC Using Machine Learning Techniques*," in International Journal of Technology, Innovation and Management (IJTIM),* vol. 1, No. 2, pp. 65-78, 2021.

Article Information Form

Authors Contributions

The first author contributed to design, data collection, data analysis, writing, technical support, and literature review. The second author contributed to concept/design, data analysis, technical support, and critical review of content. The third author contributed to technical/material support and literature review.

Conflict of Interest Notice

The authors declare that there is no conflict of interest regarding the publication of this paper.

Ethical Approval

Scientific and ethical principles were adhered to during the preparation of this study, and the bibliography lists all the studies that benefited from them. The organization authorized the use of fully anonymized firewall logs for research. No human-subjects data were collected; therefore, IRB/ethics committee approval was not required. Data handling complied with Law No. 5651 (timestamped, integrity-preserving logs) and internal security policies; no IP addresses or user identifiers were reported.

Artificial Intelligence Statement

No artificial intelligence tools were used while writing this article.

Plagiarism Statement

This article has been scanned by iThenticate™.