International Journal on Advanced Science Engineering Information Technology

Vol.11 (2021) No. 4
ISSN: 2088-5334

Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense

Qasem Abu Al-Haija[a,*], Abdelraouf Ishtaiwi[a]

[a] *Department of Data Science & Artificial Intelligence, University of Petra, Amman 1196, Jordan*
Corresponding author: [*]qasem.abualhaija@uop.edu.jo

***Abstract*— A firewall system is a security system to ensure traffic control for incoming and outgoing packets passing through communication networks by applying specific decisions to improve cyber-defense and decide against malicious packets. The filtration process matches the traffic packets against predefined rules to preclude cyber threats from getting into the network. Accordingly, the firewall system proceeds with either to "allow," "deny," or "drop/reset" the incoming packet. This paper proposes an intelligent classification model that can be employed in the firewall systems to produce proper action for every communicated packet by analyzing packet attributes using two machine learning methods, namely, shallow neural network (SNN), and optimizable decision tree (ODT). Specifically, the proposed models have used to train and classify the Internet Firewall-2019 dataset into three classes: "allow, "deny," and "drop/reset." The experimental results exhibited our classification model's superiority, scoring an overall accuracy of 99.8%, and 98.5% for ODT, and SNN respectively. Besides, the suggested system was evaluated using many evaluation metrics, including confusion matrix parameters (TP, TN, FP, FN), true positive rate (TPR), false-negative rate (FNR), positive predictive value (PPV), false discovery rate (FDR), and the receiver operating characteristic (ROC) curves for the developed three-class classifier. Ultimately, the proposed system outpaced many existing up-to-date firewall classification systems in the same area of study.**

***Keywords*—Artificial intelligence; shallow neural network; decision tree; network security; firewalls; firewall logs; classification.**

Manuscript received 3 Feb. 2021; revised 21 May 2021; accepted 6 Jun. 2021. Date of publication 31 Aug. 2021.

I. INTRODUCTION

Data communication over the Internet is vulnerable to a wide range of potential cyber-attacks and intrusions. Once the network infrastructure is breached, hackers could distribute the data to unauthorized parties and manipulate the network data's accuracy and consistency over its entire life cycle. Consequently, various safety methods have been used in the different stages of defense to address security concerns, such as Internet Firewalls [1], Intrusion Detection/Prevention Systems (IDS/IPS) [2], and others.

Firewall devices/servers are crucial security systems to defend communication networks from external cyber-attacks [3]. They are usually installed at the networks' edges to monitor the traffic flow and protect the communication by filtering out all incoming (and sometimes the outgoing) data packets. The filtration process is typically performed by matching the network packets against predefined instructions and rules to preclude cyber threats from getting into the network. Hence, the firewall system proceeds with either to "allow," "deny," or "drop/reset" the incoming packet. Once the firewall decides what to do with the received traffic, it records all these decisions in the form of log-files. Further analysis for the firewall log-files will help improve the cyber-defense against incoming threats by integrating machine learning methods to provide automated and early classification and prediction for the traffic bypassing the firewall system. Indeed, the recent coupling of the cybersecurity field and machine learning methods produced robust and efficient security solutions [4] for diverse applications and systems.

Classification task typically employs machine learning techniques to learn several class labels of examples from the problem space [5]. For example, classifying the emails as "inbox" or "junk." Several machine learning techniques can be

employed to perform the classification/prediction tasks for the data records in the problem space [6]. Examples of classification-based machine learning techniques used to provide cyber-security solutions: Artificial Neural Networks (ANN) [7], Shallow Neural Network (SNN) [8], Convolutional Neural Network (CNN) [9], K-Nearest Neighbors (KNN) [10], Decision Tree Method (DTM) [11], Majority Voting Method (MVM) [12], Support Vector Machine (SVM) [13], and others. The use of a proper algorithm is heavily based on several factors related to the dataset nature and complexity, such as the size of a dataset, the number of features, the types of features, the structure of data, the data labeling/clustering, the data distribution [14], and IFW-2019 dataset [15].

Firewalls are essential devices to protect the communication networks by a mean of filtering out all incoming (and sometimes outgoing) traffic packets. The filtration process is performed by matching the traffic packets against predefined rules aiming to preclude cyber-threats from getting into the network. Due to the rapid increasing of security incidents and breaches [16], several recent research projects have been conducted to address the diverse issues related to firewall systems. For instance, D. Appelt *et. al.* [17] presented a machine learning and evolutionary algorithm-based approach to spontaneously identify the holes in communication networks via Web application firewalls (WAFs) due to SQL injection attacks. They implemented their model using open-source WAF tool called ModSecurity. As a result, their simulation findings show their model's effectiveness against SQL injection attacks bypassing WAFs and identifying attack patterns. Similarly, D. Ucar et al. [18] proposed a machine learning-based model for the detection of anomalies in the firewall rule repository. To do so, they have employed a dataset of firewall logs using several classification algorithms, including Naive Bayes, kNN, Decision Table and HyperPipes. As a results of their analysis, their model works with its best performance when configured with kNN recording an F-Measure of 93%. They concluded that anomalies in firewall rules can be detected by automatically analyzing large-scale log files with machine learning methods.

Another noticeably related research is reported by Vartouni [19], who proposed a deep learning model for anomaly-based web application firewall. Their proposed model is constructed mainly of stacked auto-encoder (SAN), deep belief network (DBN), and one-class SVM, isolation forest, and elliptic envelope are applied as classifiers. Their experimental results showed that model demonstrated has a better performance in terms of accuracy and generalization in a reasonable time. In related research that employs a deep learning model, Ertam [20] proposed a new firewall data classification approach that uses 10 cases to obtain numerical results. The proposed approach consists of data acquisition from Firewall, feature selection and classification steps. The author evaluated their model using several classifiers including Long Short-Term Memory (LSTM), Bi-directional Long Short-Term Memory (Bi-LSTM) and Support Vector Machine (SVM). As a result, they inferred that the deep learning approach based Bi-LSTM-LSTM hybrid network is more successful than the SVM classifier scoring the highest classification accuracy of 97.38%. In conclusion, they noticed that an intelligent monitoring system is a very efficient approach for network security solutions.

Moreover, Reinforcement Learning (RL) techniques were also employed in this area such as the work conducted by J. Jeya Praise *et. al.* [21]. In this paper, the authors have developed a reinforcement learning and pattern matching (RLPM) based firewall for secured cloud infrastructure to block the malicious attacks by validating the payload signature of arriving packets. Their hybrid model provided a two-way pattern matching algorithm that validates the signature towards attaining the quick decisions. The simulation results showed that their proposed RLPM model has improved firewall response time, throughput, and malicious attack blocking by 10% less than the existing state-of-the-art methods. Furthermore, several other promising state-of-the-art research has been conducted for cybersecurity using deep neural networks [22]–[31].

Unlike aforementioned research, in this paper, we propose an intelligent self-reliant classification model that can be employed in the firewall systems to produce proper decision on every incoming traffic packet passes through the communication network Firewall system by analyzing packet attributes (i.e., through firewall logs) using SNN, and ODT techniques. Specifically, the proposed model has been trained to classify the Internet Firewall-2019 (IFW-2019) dataset into three classes, including: “allow, “deny,” and “drop/reset.” The proposed model is considered as competent contribution to this area due to the well-defined and designed model scoring a very high classification accuracy of 99.8% with low prediction overhead. In this paper, we employ two different supervised machine learning techniques to train and classify the communication traffic records provided by Internet Firewall (IFW-2019) dataset after a series of preprocessing operations. The employed learning techniques include shallow neural network (SNN), and optimizable decision tree (ODT). IFW-2019 dataset [15] comprises 65532 records from firewall logs files divided into four different categorical categories, namely, “allow,” “deny,” “drop,” and “reset-both. Based on the collected records, our ML models were trained aiming to minimize the loss function and we show that the overall accuracy of the model is superior.

In particular, the core contributions of the proposed work can be listed as follows:

- We provide a firewall prediction system that employs SNN, and ODT architectures for classifying multi-class firewall log action records in communication networks.
- We evaluate our IDS' performance on recent and important datasets for Internet firewall log files (IFW-2019 [15]), scoring a 99.8% and 98.5% classification accuracy for ODT and SNN respectively.
- We provide a detailed description of our implementation in conjunction with an extensive comparison with state-of-the-art solutions.

II. Material and Method

In this work, we concern to provide a comprehensive machine learning based framework to ensure an automated and intelligent decision-making process for the firewall system to

improve the communication network defense and security. Fig.1 demonstrates the flowchart diagram for system development method displaying the systematic stages for the proposed system starting from the initial stage of research, data gathering toward the last outcome stage, the classification stage. According to the figure, the system development comprises four modules: data gathering, data preparation, data learning, and data classification. The modules are to be discussed in the following subsections after Fig.1.

A. Data Gathering Module

Data is the vital element of any intelligent system since it allows systems and stakeholders to build the required decisions based on definite facts and records. Data (categorical, numerical, images...etc.) are usually collected into organized records in a systematic dataset [2]. Dataset can be analyzed and used to address research investigations, formulate problem statements and validate theories and outcomes. In this work, our system concerns providing automated and intelligent classification for the security actions performed by firewall devices on the network traffic, and thus, we have used a dataset [15].

IFW − 2019 [15] is a recently composed dataset from the internet traffic records on a university's firewall devices (i.e., Firat University, Turkey) and used for the automated prediction purposes of firewall actions in response to the network traffic data. IFW − 2019 accumulates 65532 firewall log files records with four different categorical labels listed for the output filed of each sample record, including: "allow", "deny", "drop", and "reset-both". The data distribution among the different classes of the files are presented in Table 1.

TABLE I
STATISTICS OF TRAFFIC DISTRIBUTION OF IFW-2019 [32].

Actions	**allow**	**deny**	**drop**	**reset-both**
No. of Records	37640	14987	12851	54
Description	Permit the data packet	Block the data packet	Drop the data packet	Send TCP reset to both the client and server devices.

IFW − 2019 records are developed with 11-features and one class-label using firewall log files. Features are carefully selected as numerical datatype to be efficiently applied to the machine learning techniques. The selected features include source port, destination port, network address translation (NAT) source port, NAT destination port, elapsed time for flow (in seconds), total bytes, bytes sent, bytes received, total packets, packets sent, and packets received [32].

Indeed, the IFW-2019 dataset is nominated for evaluation in this research since it is publicly available as .CSV filetype and comprises a sensible number of distinctive samples that prevent the classifier from being influenced by a more recurrent class. Also, this dataset covers all common security actions for firewall devices/servers on the network traffic. Moreover, it can be powerfully preprocessed and programmed to generate multi-class classification for the firewall actions in the communication networks. Finally, IFW − 2019 can be tailored, extended, updated, and stimulated.

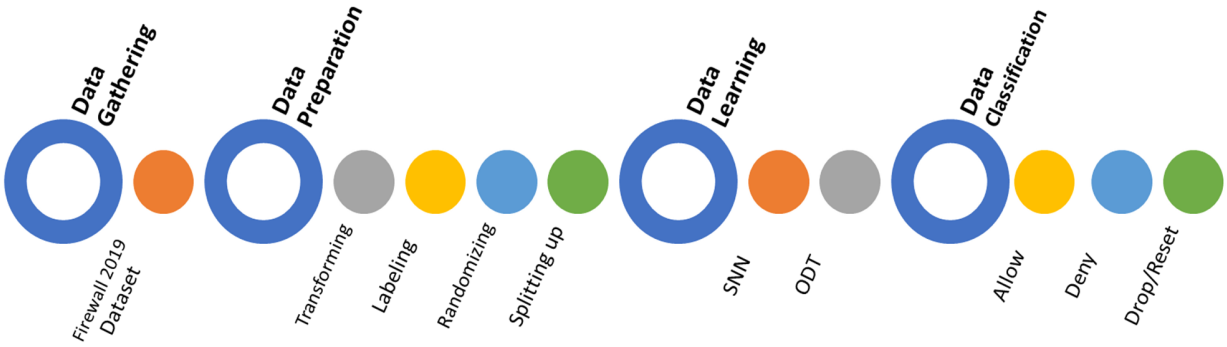


Fig. 1 Architectural diagram of the proposed classification framework.

B. Data Preparation Module

Like any machine learning-based system, the dataset undergoes a number of preprocessing operations to be prepared for use by the machine learning input layer for further processing and learning operations. In this work, our collected dataset has been processed as follows:

1) Dataset Transformation: Since the dataset records are available as .csv file with multiple rows and columns (separated by comma) where rows represent the data samples and the columns represent the features, the dataset needs to be transformed through the MATLAB system (our development platform) into a double matrix to be able for any further calculation or machine learning processing. Also, at the stage, the dataset was transformed into a matrix of features with corresponding samples ($11\ x\ 65532$) and a vector of labels ($1\ x\ 65532$).

2) Dataset Labeling: Since the dataset class feature are stored as a categorical datatype, such datatype needs to be encoded into numerical labels (labeling) as to be processed mathematically by the machine learning algorithms and calculations. Therefore, we have applied the one hot encoding techniques [17] to provide a proper labeling for the target classes as follows: Allow (100), Deny (010), Drop/Reset (001).

3) Dataset Randomization: This stage is performed to re-distribute the dataset samples in a random fashion to elude any classification preference and thus enhance the validation and testing stages by ensuring randomized dataset samples. To do so, we have used the $Shuffling$ algorithm as a data randomization policy which shuffles the data samples of the dataset through random locations.

4) Dataset Splitting Up: This stage is performed to divide the data into three datasets, namely, taring dataset, validation dataset, and testing dataset. To do so, we have used the $DivideRand$ algorithm as a dataset distribution policy that divides the targets into 3-sets using random indices. Thus, the dataset distribution Proportions are training: 70%, Validating: 15%, Testing: 15%, and the dataset distribution numbers are Training: 45,872, Validating: 9,830, Testing: 9,830.

C. Data Learning Module

In this work, we developed our inference system using a SNN and ODT to train and classify the communication traffic records provided by the IFW-2019 dataset into three classes: Allow, Deny, Drop/Reset. In the third class, we have combined both "reset-both" and "drop" actions in one class since "reset-both" has a small number of samples (i.e., only 54 samples).

1) Shallow Neural Network (SNN): In SNN, data introduced to the network goes through a single hidden layer of pattern recognition. Our SNN is composed of an input vector $(\underline{I})$ with 11-inputs (I_1; I_2; I_3; …; I_{11}) that connects the 11- features of $IFW-2109$ to the 150-neurons at the hidden layer (H_1; H_2; H_3; …; H_{150}) in a fully connected fashion. Also, every single neuron is fully connected to the 3-neurons at the output layer (O_1; O_2; O_3) producing $Softmax$ probabilities for the corresponding classes (Class_1="allow"; Class_2="deny"; Class_3="drop/reset" ;). Finally, the trainable weight vectors corresponding to each of the parametrized layers are $\underline{W}$ and $\underline{V}$ from left to right respectively. Moreover, to demonstrates the symbolic representation for the individual neurons, we consider a neuron unit with an input vector $\underline{I}$ of n elements and single output H. For every neuron, all elements of input vector $\underline{I}$ are multiplied by the corresponding weights in the weight vector $\underline{W}$ and subsequently supplied to the intersection of summation operation production the dot product of weighs and inputs ($\underline{W}^T \cdot \underline{I}$). After All, the bias b is added to the dot-product forming the net value.

2) Optimizable Decision Tree (ODT): Decision trees are powerful and popular tools for classification and prediction. Decision trees represent rules that humans can understand and use in knowledge systems such as databases. In our ODT, we have configured the tree with 11 predictors and one response variable (target variable). Also, the tree split followed split criterion of maximum deviance reduction, with a maximum number of splits of 30 splits using 30 iterations and 5-fold cross-validation.

D. Data Classification Module

In order to calculate the probabilities for the output classes, we have used the SoftMax activation function (multi-class classifier). SoftMax is a normalized exponential formula that normalizes a vector of K real numbers ($\mathbb{R}^k$) into a probability distribution comprising of K real number-probabilities ($\mathbb{R}^k$) that are proportional to the exponentials of the input numbers [9]. To calculate the numerical probabilities for each class, we first consider the final neuron output from previous layer which are activated using Sigmoid function σ (net) as follows:

$$net^{[1]} = \left(\sum_{j=1}^{n} W_{1j} \cdot I_j\right) \underset{then}{\Longrightarrow}$$

$$H = \sigma\left(net^{[1]}\right) = \frac{1}{1+e^{-net^{[1]}}} \quad for\ i = 1, 2, 3, \dots, n \quad (1)$$

Eventually, the output layer computations via SoftMax σ: $\mathbb{R}^k \longmapsto \mathbb{R}^k$ is defined as:

$$net^{[2]} = \left(\sum_{j=1}^{n} V_{1j} \cdot H_j\right) \underset{then}{\Longrightarrow}$$

$$\underline{O} = \sigma\left(net^{[2]}\right)_i = \frac{e^{net^{[2]}{}_i}}{\sum_{j=1}^{K} e^{net^{[2]}{}_i}} \quad for\ i = 1, 2, 3, \dots, K \quad (2)$$

A sample of SoftMax classification output is provided in Table 2. According to the numerical probabilities provided in the table, the classifier will always select the label that recorded the highest probability value for each instance.

TABLE II
SAMPLE OUTPUT FROM SOFTMAX CLASSIFICATION

Class label	**C1**	**C2**	**C3**
Actions	allow	deny	Drop/reset
SoftMax Value	0.91	0.06	0.03
Selected Action		C1: Allow	

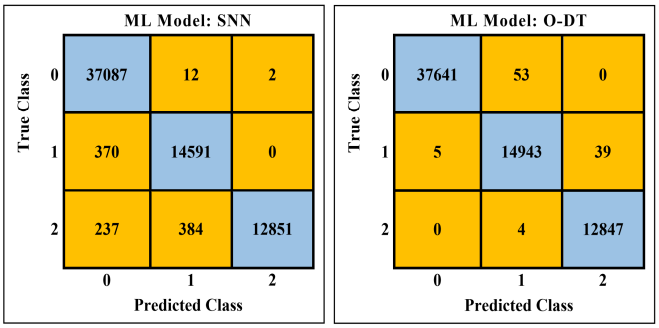
III. RESULTS AND DISCUSSION

In order to develop and evaluate the proposed IFW classification system, the training and testing phases were carried out using IFW − 2019 dataset. The predictive model is specified to differentiate between three classes: 'allow', 'deny', 'and 'drop/reset' for the network packets. The proposed predictive model is implemented using MATLAB 2020b on a commodity laptop. Also, to optimize neural network training speed and memory, MEX calculation (MATLAB executable) has been used to train and simulate the network as well as for gradient calculations. Besides, the original dataset has undergone a preprocessing stage prior the use into the machine learning techniques. The preprocessing module is responsible for the conversion of raw traffic records of IFW − 2019 into a matrix of labeled features that can be trained by the supervised learning part of the classification system. To sum-up, the specifications and configurations of the test-bench environment is shown in Table III.

TABLE III
SUMMARY OF SYSTEM DEVELOPMENT ENVIRONMENT.

Specifications	Description
Computing Platform	High performance commodity PC with: • Intel I7-8550U CPU • 4.00 GB GPU NVIDIA GF 940MX
Supervised learning techniques	W-SNN with 150-Hidden Neurons using Three-Neuron Output Layer. ODT with 30 Splits using Maximum Deviance Reduction.
Optimization Technique	Scaled Conjugate Gradient Backprop. [21] for SNN Bayesian Gradient Optimization [22] for ODT
NN Performance Analysis	Cross-Entropy Loss (LCE) Function [23] for SNN Mean Squared Error (MSE) Function [24] for ODT
Classification Learner	Linear learner algorithm Initial Learning Rate ($\alpha = 0.0001$)
Validation Frequency	6-Fold Cross Validation/ Randomly Performed at Every Run.
Number of Epochs/Iterations	387 Epochs for SNN 30 Iterations for ODT

To measure the system performance, we have evaluated the *developed* classification model in terms of several evaluation metrics [34] including the following: multi-class confusion matrix {true positives (TP), true negatives (TN), false positives (FP), false negatives (FN)}, classification accuracy (ACC), classification error percent (CEP), true positive rate (TPR), false-negative rate (FNR), positive predictive value (PPV), false discovery rate (FDR). Hence, Fig. 2 shows the confusion matrix results for our three-class classifier. Based on the values obtained for ⟨TP, TN, FP, FN⟩, we have computed the aforesaid evaluation metrics for our classification model.



	ACC%	CEP%	TPR%	FNR%	PPV%	FDR%
SNN	98.50	1.50	98.47	1.53	98.51	1.49
ODT	99.8	0.20	100.0	0.00	99.7	0.30

Fig. 2 System Evaluation: Confusion Matrix and overall evaluation metrics.

Also, since the objective of the classification model is to produce output values as close as possible to the true values, thus the trainable weights of the model are iteratively adjusted aiming to minimize the Cross-Entropy Loss (L_{CE}) value in the case of SNN model and to minimize the Cross-Entropy Mean Squared Error (MSE) value in the case of ODT model. Hence, SoftMax probability (p_i) for each predicted class (i) is compared to the true class label (t_i) and the loss (L_{CE} *or* MSE) is calculated that penalizes the probability based on how far it is form the true value [35]. A perfect model has a LCE or MSE loss of 0. The plot for mean secured error vs iteration number showing the best point hyperparameters for the ODT model, is illustrated in Fig.3 (A), while the plot for cross-entropy for training, validation, testing, and best curves is illustrated in Fig.3 (B). Moreover, Table IV provides the results obtained for Cross-Entropy (CE) and Percent Error (%E) for SNN model and Mean Squared Error (MSE) and Percent Error (%E) for the ODT model. In either model, minimizing CE or MSE results in good classification, lower values are better, and zero means no error. For %E, percent error indicates the fraction of samples that are misclassified. A value of 0 means no misclassifications while 100 indicates maximum misclassifications.

TABLE IV
LOSS AND ERROR VALUES FOR BOTH MODELS (SNN/ODT).

Model	CE	%E		MSE	%E
SNN	0.022	1.50	**ODT**	0.001	0.20

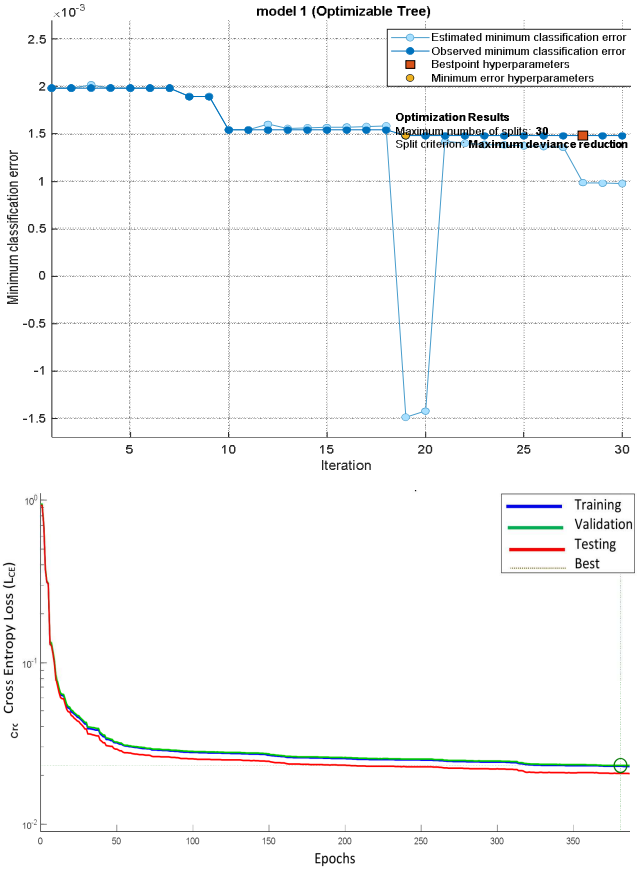


Fig. 3 Training Performance (A) ODT Model using MSE (Iteration 30, Validation Stop) and (B) SNN Model using CEL (Epoch 381, Validation Stop)

Furthermore, we have investigated the receiver operating characteristic (ROC) curve for our 3-class classifier. ROC curve represents the relative trade-offs between the true positive rate (benefits) at the y-axis against the false positive rate (costs) at the x-axis for at various threshold settings. Typically, the classification model is established based on a continuous random variable ($\boldsymbol{X}$) which is compared with a predefined threshold ($\boldsymbol{T}$), therefore, the instance is classified as "positive" if $\boldsymbol{X} > \boldsymbol{T}$, and "negative" otherwise. Accordingly, the true positive rate (TPR) and false positive rate (FPR) for a given threshold ($\boldsymbol{T}$), can be integrally computed. Consequently, $ROC\ curve$ plots parametrically $TPR(T)$ versus $FPR(T)$ with T as the varying parameter. The ROC curves of our three-classes classifier for the for SNN model and ODT Model are illustrated in Fig. 4. Since almost all experiments yield a point in the upper left corner (0,1) of the ROC space, the classifier almost provides a perfect classification case recording 99.0% and 100.0% for the area under the curve (AUS) of the SNN model and ODT Model respectively.

In addition, Fig. 5 shows a histogram of MSE errors for the training dataset, validation dataset, and testing dataset. The entire range of residuals has been divided into 20 bins. According to the figure, it can be clearly inferred that most MSE values are approaching zero. Besides, the histogram error bars seem to follow a normal distribution curve, which reflects the quality of the proposed machine learning model. Moreover, most counted errors correspond to the training dataset since it has most dataset records within the employed dataset (i.e., 70 % of the data samples belong to the training dataset, 15 % belong to the validation dataset, and 15 % belong to the testing dataset). These error cases are illustrated using color conventions where blue refers to the training error residuals, green refers to the validation error residuals, red refers to the training error residuals, and the yellow line refers to the zero-error value.

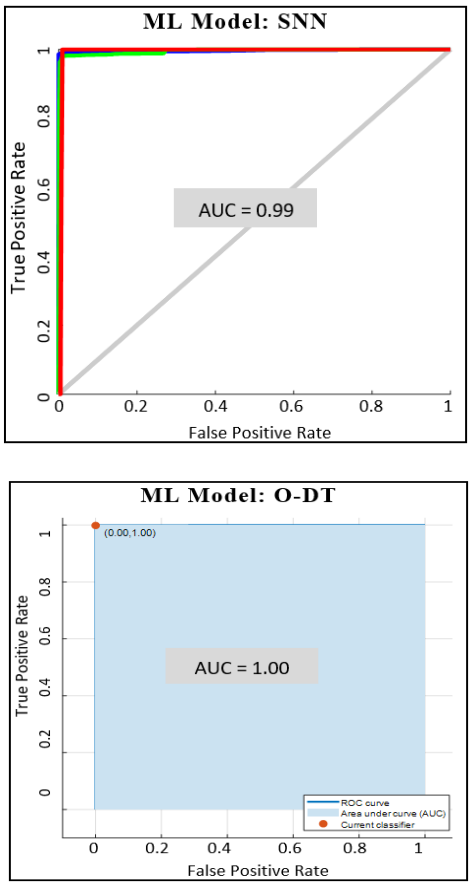


Fig. 4 Analyzing ROC Curve behavior for (A) SNN Model, (B) ODT Model

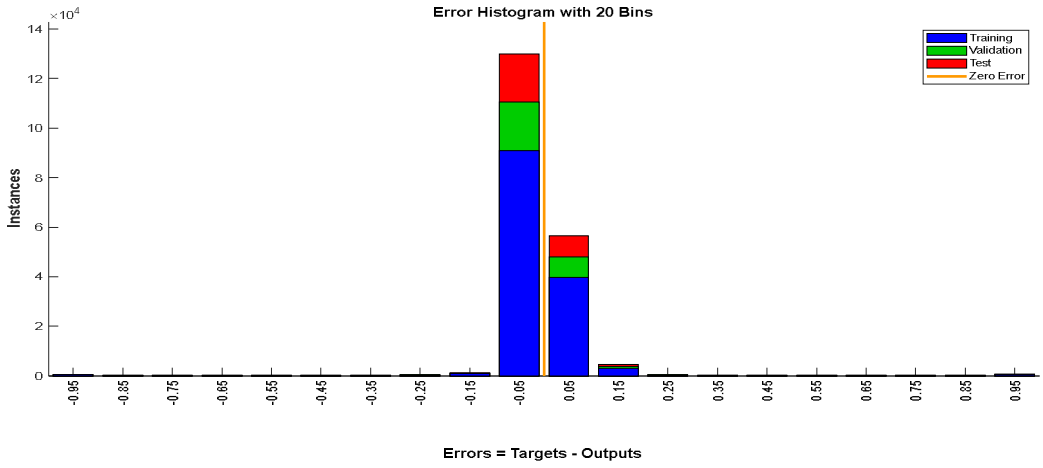


Fig. 5 NN Training Error Histogram (Epoch 124, Validation Stop)

Besides, Fig. 6 shows the neural network training state in terms of gradient analysis and validation fails during the 124 training epochs. This figure represents the current progress/status of the training at a specific time while training is in progress. In our case, six validation errors are mentioned, which means that the training will stop when the 6 validation check errors are simultaneously produced. As can be clearly seen, the validation process has been stopped after 124 epochs in which the first time the model meets 6 validation check errors from the beginning of the training process.

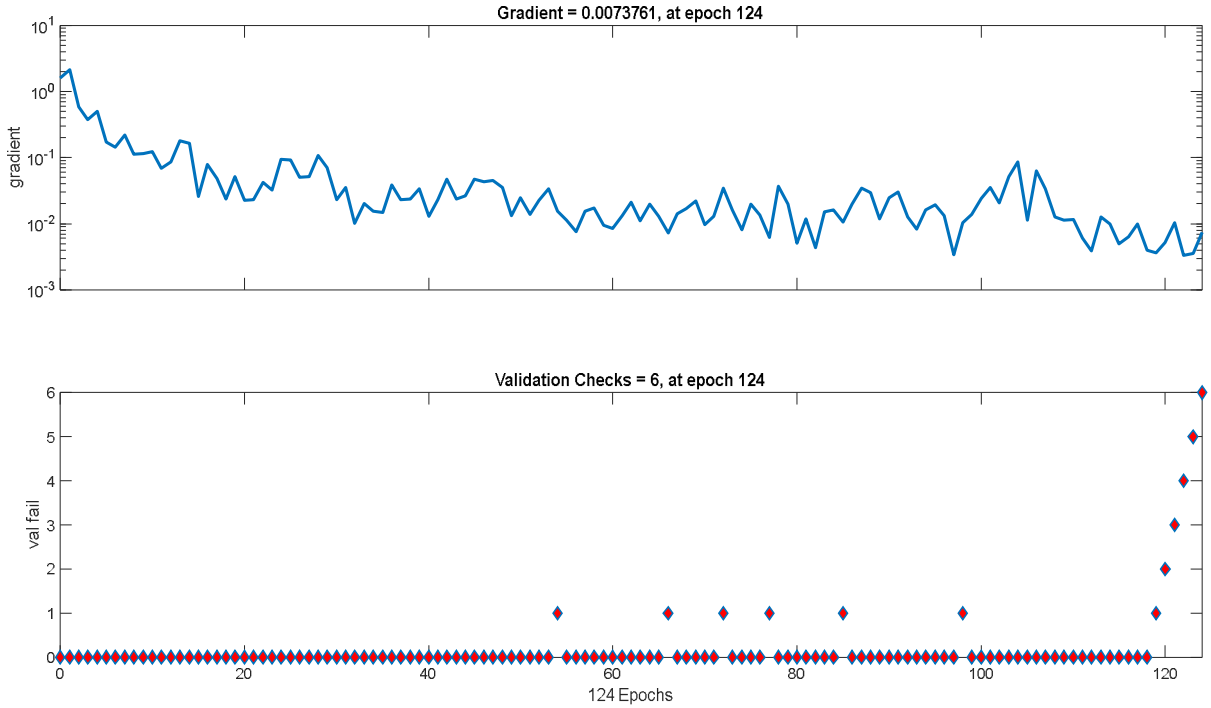


Fig. 6 NN Training State (Epoch 124, 6 Validation checks)

Finally, to gain more insight into the proposed solution's advantages, we benchmarked the IFWclassification system by comparing its performance with other state-of-the-art machine-learning-based firewall-action classification systems in terms of the classification accuracy metric. The comparisons are provided in Table V below.

TABLE V
COMPARISON WITH EXISTING RELATED WORK METHODS EMPLOYING THE FIREWALL LOG FILES.

Research	Year	ML Technique	Accuracy	IM %
F. Ertam *et. al.* [15]	2019	Support Vector Machine (SVM)	79.40 %	25.69 %
A. I. Pîrîu *et. al.* [36]	2019	Deep Neural Network (DNN)	92.82%	07.52%
A. Banjongkan *et. al.* [37]	2020	Convolutional Neural Network (CNN)	76.50 %	30.46%
S. Allagi *et. al.* [38]	2020	Self-Organizing Feature Map (SOFM)	97.20 %	02.68%
Our Model	2021	Shallow Neural Network (SNN)	98.50 %	
Our Model	2021	Optimizable Decision Tree (ODT)	99.80 %	

Accordingly, it can be observed that the proposed IFW model has higher classification accuracy for the IFW – 2019 dataset compared with other existing related machine learning-based models by an improvement percent (IM%) of ≈ (2.7% – 30.5%). The improvement percent is calculated as the ration of accuracy enhancement for our model over the existing model's accuracy as follows:

$$IM\ \% = \left[\left(\frac{Our\ Model\ Accuracy}{Other\ Model\ Accuray} \times 100\right) - 100\right]\ \% \qquad (3)$$

IV. CONCLUSION

This paper has proposed and discussed a dependable automated machine-learning-based internet firewall model to classify the packet traffic for communication network systems. The proposed system uses a shallow neural network (SNN) and optimizable decision tree (ODT) using 11-attributes/predictors at the input stage and 3-classes at the output classification layer. The proposed system employs the multi-class internet firewall (IFW-2019) dataset with 70% of the records used for the training dataset, 15% used for validation data set, and 15% used for testing dataset. To evaluate the model's performance, it was adequately trained, recording a maximum accuracy of 99.8% and 98.5% achieved using ODT and SNN models, respectively, for the 3-class classifier. Besides, other machine learning metrics were evaluated to gain more insights into the system trajectory, such as positive predictive value, true positive rate, and others. Finally, based on the comparison with the existing state-of-art in the field, the achieved outcomes surpassed the existing automated classification models for the firewall actions, which contributes to this area of study.

ACKNOWLEDGMENT

The authors are grateful to the deanship of scientific research at University of Petra (UoP) for supporting the publication of this paper.

REFERENCES

[1] W. Noonan, I. Dubrawsky, "Firewall fundamentals", Pearson Education, 2006.

[2] Q. A. Al-Haija, S. Zein-Sabatto, "An Efficient Deep-Learning-Based Detection and Classification System for Cyber-Attacks in IoT Communication Networks" *Electronics,* MDPI, vol. 9, no. 12: paper no. 2152., 2020.

[3] E. Ucar, E. Ozhan, "The Analysis of Firewall Policy Through Machine Learning and Data Mining", *Wireless Personal Communication*, Springer, vol. 96, p.p. 2891–2909, 2017.

[4] G. Caspi, "Introducing Deep Learning: Boosting Cybersecurity with an Artificial Brain. Informa Tech" Dark Reading, *Analytics* http://www.darkreading.com/analytics, 2016.

[5] Q.A. Al-Haija, C.D. McCurry, S. Zein-Sabatto, "Intelligent Self-reliant Cyber-Attacks Detection and Classification System for IoT Communication Using Deep Convolutional Neural Network", Selected Papers from 12th International Networking Conference. INC 2020. *Lecture Notes in Networks and Systems*, vol.180. Springer, 2021.

[6] J. Brownlee, "4 Types of Classification Tasks in Machine Learning", Python Machine Learning, Machine Learning Mastery, 2020.

[7] S. Haykin, "Neural Networks and Learning Machines. 3rd Edition, Pearson publications, ISBN-13: 978-0-13-147139-9, 2009.

[8] C. C. Aggarwal, "Machine Learning with Shallow Neural Networks", Neural Networks and Deep Learning. Springer, 2019.

[9] Fei-Fei. CS231n: Convolutional Neural Networks for Visual Recognition. Computer Science, Stanford University. Available online: http://cs231n.stanford.edu, 2019.

[10] J. S. Meneses, Z.R. Chavez, J.G. Rodriguez, "Compressed kNN: K-Nearest Neighbors with Data Compression" *Entropy,* MDPI, vol. 21, no. 3, paper no. 234, 2019.

[11] Y.Y. Song, Y. Lu, "Decision tree methods: applications for classification and prediction. Shanghai Arch Psychiatry", *PMID: 26120265; PMCID: PMC4466856*, vol. 27, no.2, p.p.130-5, 2015.

[12] B. A. Tama, K. H. Rhee, "An extensive empirical evaluation of classifier ensembles for intrusion detection task", *International Journal Computer Systems Science and Engineering*, CRL Publishing Ltd, vol. 32, no.2, p.p.149-158, 2017.

[13] A. Ghose, "Support Vector Machine (SVM) Tutorial: Learning SVMs from examples". Medium: towards data science, 2017.

[14] R. Garg, "Types of Classification Algorithms", Analytics India Magazine, 2018.

[15] F. Ertam, M. Kaya, "Classification of firewall log files with multi-class support vector machine," *in Proc. Of 6thInternational Symposium on Digital Forensic and Security (ISDFS)*, Antalya, pp. 1-4, 2019.

[16] Q. A. Al-Haija, L. Tawalbeh, "Autoregressive Modeling and Prediction of Annual Worldwide Cybercrimes for Cloud Environments," *in Proc. Of* 10th International Conference on Information and Communication Systems (ICICS), 2019, pp. 47-51.

[17] D. Appelt, C. D. Nguyen, A. Panichella, L. C. Briand, "A Machine-Learning-Driven Evolutionary Approach for Testing Web Application Firewalls," *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 733-757, 2018, doi: 10.1109/TR.2018.2805763.

[18] E. Ucar, E. Ozhan, "The Analysis of Firewall Policy Through Machine Learning and Data Mining", *Wireless Personal Communication, Springer,* vol. 96, p.p. 2891–2909, 2017.

[19] A. M. Vartouni, M. Teshnehlab, S. S. Kashi, "Leveraging deep neural networks for anomaly-based web application firewall", *IET Information Security*, vol. 13, p.p. 352-361, 2019.

[20] F. Ertam, "An efficient hybrid deep learning approach for internet security", *Physica A: Statistical Mechanics and its Applications*, Elsevier, vol. 535, 2019

[21] J.J. Praise, R.J Raj, J.V. Benifa, "Development of Reinforcement Learning and Pattern Matching (RLPM) Based Firewall for Secured Cloud Infrastructure", *Wireless Personal Communication, Springer,* vol.115, p.p. 993–1018, 2020.

[22] G. Bendiab, S. Shiaeles, A. Alruban, N. Kolokotronis, "IoT Malware Network Traffic Classification using Visual Representation and Deep Learning", *in Proc. Of 6th IEEE Conference on Network Softwarization (NetSoft)*, Ghent, Belgium, 29 June–3 July 2020; pp. 444–449.

[23] R. Shire, S. Shiaeles, K. Bendiab, B. Ghita, N. Kolokotronis, "Malware Squid: A Novel IoT Malware Traffic Analysis Framework Using Convolutional Neural Network and Binary Visualization", *in Proc. Of Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. *Lecture Notes in Computer Science*; Springer, vol.11660, 2019

[24] I. Baptista, S. Shiaeles, N. Kolokotronis, "A Novel Malware Detection System Based On Machine Learning and Binary Visualization", *in Proc. Of IEEE International Conference on Communications (IEEE ICC)*, China, pp. 1–6, 2019.

[25] K.A. Taher, B.M. Jisan, M.M Rahman, "Network Intrusion Detection using Supervised Machine Learning Technique with Feature Selection", *in Proc. Of International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST*), South Asia, 10–12 January 2019; pp. 643–646.

[26] X. Gao, C. Shan, C. Hu, Z. Niu, Z. Liu, "An Adaptive Ensemble Machine Learning Model for Intrusion Detection"*, IEEE Access,* vol. 7, p.p. 82512–82521, 2019.

[27] Q. A. Al-Haija, M. Alkhatib, A. B. Jaafar, "Choices on Designing Gf (P) Elliptic Curve Coprocessor Benefiting from Mapping Homogeneous Curves in Parallel Multiplications", *International Journal on Computer Science and Engineering (IJCSE)*, ISSN: 0975-3397, vol. 3 no. 2, 2011.

[28] S. Sapre, P. Ahmadi, K. Islam, "A Robust Comparison of the KDDCup99 and NSL-KDD IoT Network Intrusion Detection Datasets through Various Machine Learning Algorithms"*, arXiv:1912.13204v1*, 2019.

[29] M.M. Chowdhury, F. Hammond, G. Konowicz, C. Xin, H. Wu, J. Li, "A few-shot deep learning approach for improved intrusion detection", 2017 *in Proc. Of IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON),* NY, USA, 19–21 October 2017; pp. 456–462.

[30] A. Javaid, Q. Niyaz, W. Sun, M. Alam, "A Deep Learning Approach for Network Intrusion Detection System", *in Proc. Of 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, NY, USA, 24 May 2016; pp. 21–26.

[31] Y. Imamverdiyev, L. Sukhostat, "Anomaly detection in network traffic using extreme learning machine", *in Proc. Of IEEE 10th International Conference on Application of Information and Communication Technologies (AICT),* Azerbaijan, 12–14 October 2016; pp. 1–4.

[32] UCI: Machine Learning Repository, "Internet Firewall Data Set", Center for Machine Learning and Intelligent Systems, 2019.

[33] A. Wang, "Encode Smarter: How to Easily Integrate Categorical Encoding into Your Machine Learning Pipeline", Feature Labs. https://blog.featurelabs.com, 2019.

[34] Q. A. Al-Haija, M. Smadi, S. Zein-Sabatto, "Multi-Class Weather Classification Using ResNet-18 CNN for Autonomous IoT and CPS Applications" *in Proc. Of IEEE 7th Annual Conference on Computational Science & Computational Intelligence (CSCI'20),* Las Vegas, USA, 2020.

[35] K.E. Koech, "Cross-Entropy Loss Function", Medium: towards data science, 2020.

[36] A. I. Pîrîu, M. Leonte, N. Postolachi and D. T. Gavrilut, "Optimizing Cleanset Growth by Using Multi-Class Neural Networks," *in Proc. Of 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC),* Romania, pp. 425-429, 2018.

[37] A. Banjongkan, et. al., "A Comparative Study of Learning Techniques with Convolutional Neural Network Based on HPC-Workload Dataset" *Inter. Journal of Machine Learning and Computing*, vol. 10, no.1, 2020.

[38] S. Allagi and R. Rachh, "Analysis of Network log data using Machine Learning," *in Proc. Of IEEE 5th International Conference for Convergence in Technology*, India, pp. 1-3, 2019.