Matrix Pearson Correlation Feature Selection and ESPRT for DDoS Anomaly Detection

Basheer Husham Ali
Computer Department, Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq
Basheer.husham@aliraqia.edu.iq (corresponding author)

Khaled Mansour Al-Rawe
College of Administration and Economics, Al-Iraqia University, Baghdad, Iraq
khaled.mansour@aliraqia.edu.iq

Ayad M. Kwad
Electrical Department, Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq
ayad@aliraqia.edu.iq

Omar Abdulkareem
College of Administration and Economics, Al-Iraqia University, Baghdad, Iraq
omar@aliraqia.edu.iq

Nasri Sulaiman
Department of Electrical and Electronic Engineering, Faculty of Engineering, UPM, Malaysia
nasri_sulaiman@upm.edu.my

Suphian Mohammed Tariq
Computer Department, Faculty of Engineering, Al-Iraqia University, Baghdad, Iraq
Suphian.Tariq@aliraqia.edu.iq

Received: 7 July 2025 | Revised: 26 July 2025 | Accepted: 2 August 2025

 DOI: https://doi.org/10.48084/etasr.13223

ABSTRACT

Many approaches have been proposed to identify malicious anomalous traffic. Statistical models are techniques that rely on the analysis and investigation of network traffic to obtain a deeper understanding. Combining the Sequential Probability Ratio Test (SPRT) and Entropy (E) is an effective technique that can be used to detect anomalies. The most common anomalies targeting servers are Distributed Denial of Service (DDoS) attacks, which are designed to prevent legitimate users from accessing services provided by a targeted server or controller. The first goal of this study is to detect malicious traffic and identify two different types of DDoS anomalies, NTP and DNS anomalies, which are commonly exploited in reflection or amplification attacks due to their stateless UDP-based nature, by implementing an Entropy and Sequential Probability Ratio Test approach (ESPRT). The second is to select relevant features to improve the detection performance by implementing a Pearson Correlation Coefficient (PCC) approach. The CIC-DDoS2019 dataset was utilized to evaluate the proposed approach. ESPRT achieved high accuracy, ranging from 97.27 to 96.23% when the number of features ranged from 5 to 55, and had a low False Positive Rate (FPR), ranging from 0.01 to 0.03.

Keywords-DDoS attack; entropy; Pearson correlation; SPRT

I. INTRODUCTION

With the expansion of technology, the maximum rate of information transferred across a given path and memory capacity has increased. More protective applications have been developed to address basic Denial of Service (DoS) attacks. The improvement in processing performance and the increase in the storage capacity of the targeted servers have led to a decrease in the effect of DDoS attacks. Consequently, attackers have adapted their tactics, leading to new forms of malicious activity [1-5].

Not only is the number of devices larger in DDoS attacks, but also the volume, size, intensity, and complexity of the traffic aimed at the victim are higher. For example, in Q2 of 2021, a 287% increase in DDoS attacks was observed, with a volume exceeding 500 Gbps, compared to Q1 of the same year [6]. DDoS attacks can be launched using botnets, which consist of a very large number of infected devices that are controlled by the main attacker. The main attacker is referred to as the botnet master, and it operates remotely [4]. Attackers follow four primary procedures to establish a botnet network for launching DDoS attacks.

Feature selection is the process of choosing the most relevant features to build an optimal intrusion detection model. The Pearson Correlation Coefficient (PCC) [7] has been used for feature selection, selecting the best features based on their correlation with the target label. This technique measures the linear relationship between two sets of data and generates values that range between 1 and -1. When the correlation values are close to 1 or -1, it indicates that the features are highly correlated with the target label. Conversely, when these values are close to 0, it suggests that the features are not correlated with the target label.

The main research question for this study was how detection techniques can identify up-to-date DDoS attacks, such as NTP and DNS, using effective and suitable features to increase accuracy, decrease False Positive Rate (FPR), and improve detection rate. This can be achieved by combining both Entropy and SPRT (ESPRT) to detect DDoS attacks and using PCC to select features. An entropy-based approach is used to provide some indication of the availability of attacks [8]. SPRT can provide quick feedback about the status of incoming traffic in the early stage. ESPRT eliminates the dependence on using a static entropy threshold by combining the entropy approach with the SPRT method to make decisions.

The contribution of this study lies in the following:

- Detect infected traffic and identify NTP and DNS anomalies by implementing ESPRT.
- Select relevant features to improve the detection performance by leveraging the PCC.
- Improve accuracy and reduce FPR.

II. RELATED WORKS

Several DDoS attack detection techniques have been established based on different approaches, such as statistical, machine learning, and deep learning-based techniques. Statistical techniques depend on investigating and analyzing data. For instance, the study in [9] aimed to improve network availability and protect SDN controller networks from DDoS threats. The implementation combined an entropy-based method with the Packet Window Initiation (PWI) rate technique. Initially, the entropy for destination IP addresses was computed to gauge randomness and facilitate early detection of DDoS threats. The entropy value was compared with a threshold to determine the presence of an attack. However, the entropy method failed to detect DDoS attacks when attackers targeted multiple hosts. In [10], a φ-entropy technique was proposed to identify DDoS attacks in SDN networks. This method used the φ parameter to modify input parameters based on changes in network status. This adjustment improves the feature differences between malicious and benign traffic, making it easier to detect malicious activities at an early stage. This method relied on IP addresses as features. The Mininet platform was used to carry out the experiments, and a comparison was also made with the Shannon entropy. The Scapy tool and the Random function were utilized to generate the dataset. Ultimately, this method exhibited a higher detection rate compared to Shannon entropy.

In [11], a dynamic threshold method was proposed as an alternative to the static threshold approach to handle the changing behavior of traffic during DDoS attacks. This dynamic thresholding enables the detection of both high and low intensities of DDoS attacks, thus enhancing accuracy and reducing the probability of false alarms. This study introduced a technique based on rule-based detection to aid decision-making, achieved by calculating the Renyi joint entropy for source and destination IP addresses from incoming flows. The proposed method was implemented and evaluated across eight simulation scenarios that involved flow bandwidth, attack targets, and sources.

Machine learning-based techniques have also been used to identify DDoS attacks. For example, in [12], four machine learning based techniques were used: Logistic Regression (LR), ID3, Naïve Bayes (NB), and Random Forest (RF). The CICFlowMeter was used to generate 80 features from the incoming traffic. The RF regressor was used to select useful features based on their importance. This study generated a new dataset to test the proposed model. ID3 had 78%, 65%, and 69% of precision, accuracy, and F1 score, respectively, which was the best-performing compared to the other models tested. In [13], a DDoS detection method initially extracted eight entropy-based features from incoming traffic for use as input data for classification techniques. These features were fed into the next step, which involved detecting DDoS attacks and consisted of two layers. The first layer analyzed each feature and assigned a specific label using the K-Nearest Neighbors (KNN) algorithm. The second layer generated signatures for well-known attacks and detected similarities between predefined signatures and those generated from online streams during the prediction stage. This study utilized the MAWI and CICDDoS2019 datasets to evaluate their method, which achieved high performance in identifying most types of DDoS attacks in these datasets, except for SSDP and UDP attacks.

In [14], Hybrid Entropy SSAE-SVM (HESS) was introduced, which is a hybrid detection method that combines information entropy, a machine learning approach, and a deep learning approach. Initially, this method calculates the information entropy to detect suspicious traffic. A deep learning method, known as the Stacked Sparse Auto Encoder (SSAE), was used to select important features and reduce execution time. The OpenFlow table in the SDN network contains extensive information about the incoming packets. The detection phase employs the Support Vector Machine (SVM) algorithm. This model was tested on different datasets, achieving a detection accuracy of more than 98%.

In [15], three different deep learning-based classification methods were used to detect DDoS attacks, including Dense Neural Networks (DNN), Pearson Correlation Coefficient (PCC), and autoencoder algorithms. CICFlowMeter was utilized to extract features from traffic headers. The autoencoder approach was then used to select effective features before training. These three different models were used to detect DDoS attacks, and a comparison was performed using confusion matrix metrics. A DNN was selected to make decisions, and the experimental evaluation was conducted using the CICDDoS2019 dataset. In [16], a technique was proposed to identify DDoS anomalies using three different ensemble learning techniques in a smart grid environment: stacking-based, bagging-based, and boosting-based methods. CICFlowMeter features extracted from traffic were employed as inputs for these techniques. Feature selection techniques, including tree-based and PCC-based methods, were also applied to reduce the number of features. Various metrics were used to evaluate the performance of these classification techniques using the CICDDoS2019 dataset, with the stacking-based method outperforming the others.

In [17], another DDoS detection method combined deep learning and machine learning techniques, including eXtreme Gradient Boosting (XGBoost), DNN, RF, Spiking Neural Network (SNN), and Decision Tree (DT). ANOVA, Mutual Information (MI), Extra Tree (ET), and Chi-square test were chosen for feature selection. This approach selected 45 influential features of the 82 from the CICDDoS2019 dataset. However, it is important to note that this feature selection technique requires a longer time to select relevant features. In [18], a combination of XGBoost and clustering approaches was used to detect DDoS attacks. XGBoost was used for classification. CICIDS 2017 and CICIDS 2018 datasets were used to test this method, which achieved high accuracy and precision but was computation-heavy. Table I shows the strengths and limitations of these approaches.

TABLE I. STRENGTHS AND LIMITATIONS OF EXISTING METHODS

Ref#	Strengths	Limitations
[9]	Increase network availability and detect multi-host attacks.	Using only one feature that could be bypassed by attackers.
[10]	Higher detection rate	The threshold should be changed when the network behavior changes to improve detection rates.
[11]	Uses a dynamic threshold.	Using common features that may lead to bypass detection techniques when network behavior changes.
[12]	Developed a realistic DDoS dataset and taxonomy.	Not a very high detection accuracy.
[13]	High performance in identifying most DDoS attack types.	Failed to detect some DDoS attacks such as SSDP and UDP attacks.
[14]	High performance in identifying most DDoS attack types.	Did not use feature selection methods to reduce execution time.
[15]	Can detect up-to-date attacks.	The dimension reduction part is not used and cannot work in real time.
[16]	Detect DDoS in smart grid environments.	Increased FPR.
[17]	Quickly identify DDoS anomalies in IoT environments.	Requires a longer time to select relevant features

III. DETECTION APPROACH

Features selected using the PCC [6] were fed into this stage. PCC was calculated using the pandas.DataFrame.corr() function in Python. Incoming flows and the interfaces these flows pass toward the targeted devices are gathered into groups. Each group has a fixed size known as the window size. The window size can be determined based on the number of flows or a certain time interval. In the implementation, a fixed number of flows was used to determine the window size, which varies for each data trace or dataset, which can be determined through experimentation based on detection accuracy. For example, the range of window sizes for the DARPA dataset that generated high accuracy falls within 5 to 120 flows. However, the best number for CICDDoS2019 was above 300 flows. Therefore, the best window size can be determined experimentally. For each group of flows, Shannon Entropy (E) is calculated for the selected features only as in [8]:

$$E = -\sum_0^n prob(n)\, ln\, prob\ (n) \quad (1)$$

where n is the number of unique feature values.

The results of the Entropy calculation will serve as input for the next stage, which is the SPRT technique. The SPRT makes a decision and determines whether the flows and their associated switch interfaces are normal or infected. The SPRT detection ($D\_SPRT_i^s$) monitors incoming flows based on their feature values ($FL_1, FL_2, \ldots, FL_i$). It also identifies the switch interface (s) that allows these flows to cross over to the targeted machine. The detection ($D\_SPRT_i^s$) is the likelihood ratio between these observations, whether they are compromised or normal flows injected into a compromised interface (H_2), or injected into a normal interface (H_1). Therefore, the detection can be formulated as [8]:

$$D\_SPRT_i^s = ln \frac{prob\,(FL_1^s, \ldots, FL_i^s | H_2)}{prob(FL_1^s, \ldots, FL_i^s | H_1)} \quad (2)$$

where i is the total number of flow observations. Let us assume that these observations (FL_i^s) are identically independent and distributed. Thus, the detection formula can be as in [8]:

$$D\_SPRT_i^s = \sum_{v=1}^{i} ln \frac{prob\,(FL_v^s | H_2)}{prob(FL_v^s | H_1)} \quad (3)$$

where v is each value in a group of flow observations. Because FL_i^s can be as Bernoulli random variables, the detection will be as in (4) and (5) [8]:

$$prob(0 \le FL_i^s \le 0.5 | H_1) = 1 - prob(FL_i^s > 0.5 | H_1) = \mu_1 \quad (4)$$

$$prob(0 \le FL_i^e \le 0.5 | H_2) = 1 - prob(FL_i^s > 0.5 | H_2) = \mu_2 \quad (5)$$

where the value of μ_2 is larger than the value of μ_1 since compromised interfaces are more likely to be injected with compromised flows to flood the targeted system with DDoS attacks. Switch interfaces are more likely to be injected with infected flows when FL_i^s is between 0 to 0.5. On the other hand, switch interfaces are more likely to have normal traffic

when the value of FL_i^s is above 0.5. Therefore, the detection of SPRT can be shown as in (6) [8]:

$$D\_SPRT_i^s = \begin{cases} D\_SPRT_{i-1}^s + ln\frac{prob\ (FL_i^s|H_2)}{prob\ (FL_i^s|H_1)},\ 0 \leq FL_i^s \leq 0.5 \\ D\_SPRT_{i-1}^s + ln\frac{prob\ (FL_i^s|H_2)}{prob\ (FL_i^s|H_1)},\ FL_i^s > 0.5 \end{cases} \quad (6)$$

By substituting (4) and (5) in (6), the detection equation can be rewritten as [8]:

$$D\_SPRT_i^s = \begin{cases} D\_SPRT_{i-1}^s + ln\frac{\mu_2}{\mu_1},\ 0 \leq FL_i^s \leq 0.5 \\ D\_SPRT_{i-1}^s + ln\frac{1-\mu_2}{1-\mu_1},\ FL_i^s > 0.5 \end{cases} \quad (7)$$

where $D\_SPRT_0^s = 0$. The detection technique of SPRT generates two types of errors that affect the accuracy of detection. These errors are the false positive error λ_1 and the false negative error λ_2. The false positive error λ_1 occurs when the detection technique mistakenly considers the normal interface H_1 as a malicious interface H_2. On the other hand, the false negative error λ_2 occurs when an infected interface is wrongly identified as a benign one. The upper bound (U) and upper bound (L) thresholds are calculated as shown in (8) to deal with these two errors [8]:

$$\begin{cases} U = log_2 \frac{\lambda_2}{(1-\lambda_1)} \\ L = log_2 \frac{(1-\lambda_2)}{\lambda_1} \end{cases} \quad (8)$$

Finally, the detection result $D\_SPRT_i^s$ for each observed flow that passes through a certain interface is checked with the upper and lower bound thresholds dynamically to make a decision. If the value of $D\_SPRT_i^s$ is larger than or equal to U, then a monitored interface with their flows is marked benign, and the test will stop. However, if the value of $D\_SPRT_i^s$ is less than or equal to L, then a monitored interface with its flow is marked compromised, and the test will stop. Finally, the detection test will continue by checking another flow observation when the above two conditions do not apply. Figure 1 shows the ESPRT detection steps.

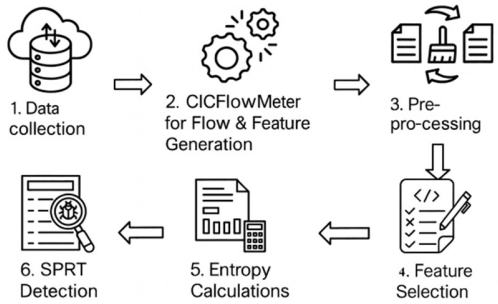

Fig. 1. ESPRT detection steps.

IV. FEATURE SELECTION RESULTS

The results were extracted from the January 12th subset that is part of the publicly available CICDDoS2019 dataset [12], further discussed in [16]. CICDDoS2019 encompasses the latest DDoS attacks. The dataset contains benign as well as actual DDoS malicious data captured and stored in pcap format, along with a set of datasets stored in csv format. Table II below shows all attack names in this dataset along with the duration of each attack.

The basis for choosing PCC among others was that this technique generated a higher value of sensitivity or True Positive Rate (TPR), specificity, accuracy, and F1-score compared to other feature selection methods. This technique also generated lower values of probability of false alarm and misses compared to others. An experiment involved six feature selection techniques, namely ANOVA, RF, Extra Tree (ET), PCC, Chi-square (Chi^2), and Mutual Information (MI), as shown in Table III. A NETBIOS dataset was chosen to run the experiment.

TABLE II. DETAILS OF ATTACK TYPES FOR CICDDOS2019 DATASET

(January 12th) dataset		**(March 11th) dataset**	
Attack	**Attack times**	**Attack**	**Attack times**
NTP	10:35 - 10:45	PortScan	9:43 - 9:51
DNS	10:52 - 11:05	NetBIOS	10:00 - 10:09
LDAP	11:22 - 11:32	LDAP	10:21 - 10:30
MSSQL	11:36 - 11:45	MSSQL	10:33 - 10:42
NetBIOS	11:50 - 12:00	UDP	10:53 - 11:03
SSDP	12:27 - 12:37	UDP-Lag	11:14 - 11:24
UDP	12:45 - 13:09	SYN	11:28 - 17:35
UDP-Lag	13:11 - 13:15		
SYN	13:29 - 13:34		
TFTP	13:35 - 17:15		

TABLE III. TESTING CONFUSION MATRIX FOR DIFFERENT FEATURE SELECTION APPROACHES

Metrics	**Feature Selection Techniques**					
	ANOVA	**RF**	**ET**	**PCC**	**Chi^2**	**MI**
TPR	0.9949	0.9899	0.9845	0.9830	0.8350	0.8011
FPR	0.0033	0.0098	0.02013	0.0081	0.0544	0.0671
TNR	0.9966	0.9901	0.9798	0.9918	0.9455	0.9328
FNR	0.0050	0.0100	0.0154	0.0169	0.1649	0.1988
Accuracy	0.9956	0.9900	0.9825	0.9874	0.8738	0.8442
F1-Score	0.9961	0.9911	0.9844	0.9874	0.8957	0.8737

PCC serves as a feature selection technique that ranks features based on their weights, derived from their correlation with a target label. To implement this method, a correlation matrix heatmap was computed for various attack datasets on the first day of CICDDoS2019. The correlation between features and the label for the NTP attack dataset was initially calculated, as illustrated in Figure 2. This figure shows the correlation among the top-20 selected features and each feature individually. The weight values of the features most correlated with the label are displayed. For example, f17 and f57, with weights of 0.66, exhibit a strong correlation with the target label. Following suit, the third and fourth most correlated features, f5 and f26, hold weights of 0.64 and 0.6, respectively. The remaining features and their respective weights are detailed in Figure 2.

Finally, the top-10 features with the highest frequency were earmarked as inputs for the subsequent stage.

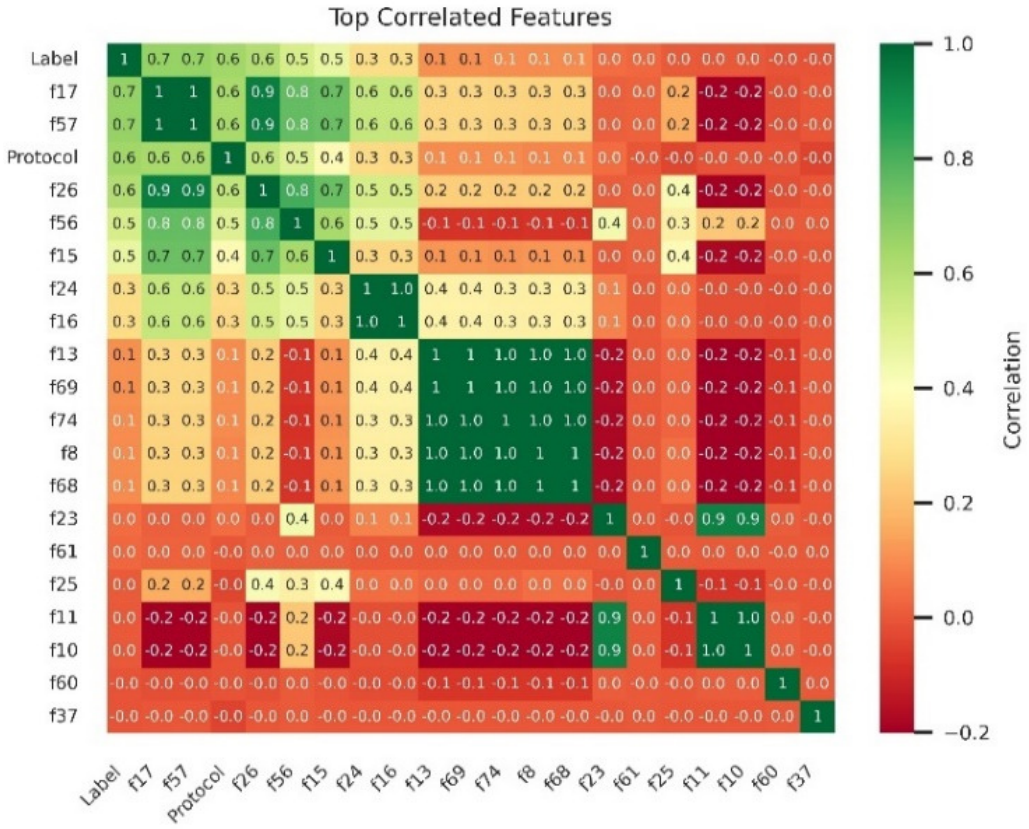

Fig. 2. Top-20 features of NTP dataset for the first day of CICDDoS2019 using a PCC heatmap.

V. CONFUSION MATRIX METRICS

A. *Sensitivity vs. Probability of False Alarm*

Figure 3 presents the relationship between sensitivity and the probability of false alarm for all features of the ESPRT detection technique across NTP attacks in CICDDoS2019.

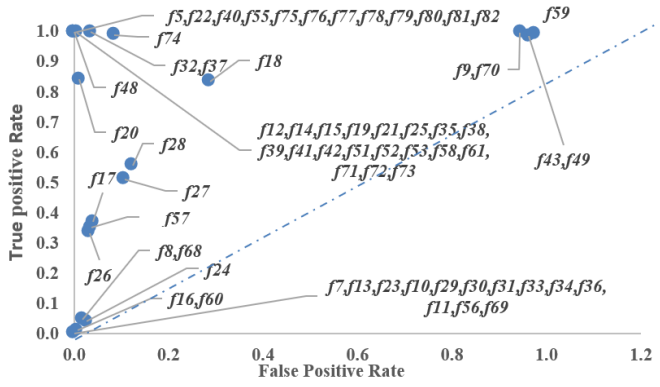

Fig. 3. TPR vs FPR for all features of ESPRT using NTP attacks on CICDDoS2019 dataset.

Sensitivity is the proportion of correctly identified malicious flow traffic among all actual flow traffic. False alarm is the average of normal flow traffic that was incorrectly predicted by the detection algorithm as compromised traffic. Features located in the upper-left side of each graph indicate the effectiveness of these features in the ESPRT detection process. For example, this figure shows sensitivity and probability of false alarm values for all features of ESPRT for the NTP attack dataset. Some features have sensitivity close to 1 and a probability of false alarm close to 0, as shown in this figure.

B. *Specificity vs. Miss Rate*

Figure 4 presents the connection between specificity and miss rate metrics of all features in the ESPRT identification method for NTP attacks in CICDDoS2019. Specificity is the average of normal flow traffic that was predicted correctly as normal flow, while miss rate is the rate of compromised traffic that was incorrectly predicted as normal traffic. These features are located in the upper left side of each graph, meaning that they are effective in the detection process. For example, most features in the NTP attack dataset have a specificity value close to 1. However, around 16 features have low NTP values close to 0. Additionally, most features in the NTP attack dataset have FPR values less than 0.65, while some features have a miss rate value close to 1, such as *f8*, *f13*, *f16*, *f24*, *f56*, *f60*, *f68*, and *f69*.

C. *Comparison Based on Feature Selection Techniques*

The accuracy of ESPRT fell in the range of 97.27 to 96.23% when the number of features ranged from 5 to 55. For example, the accuracy in [16] for bagging-, boosting-, and stacking-based methods, with 21 selected features, was 93%, 92.2%, and 93.4% respectively, all lower than the accuracy of

the ESPRT. Moreover, as shown in Table IV, the accuracy values in [12] were 65%, 56%, 11%, and 2% respectively, much lower than the accuracy of ESPRT. The accuracy in [17] was between 91.29% to 96.77%, still lower or on par with the accuracy of the ESPRT method. The standard deviation for the accuracy values of ESPRT was (±0.39), which indicates that accuracy is stable across different runs. Therefore, the accuracy of ESPRT using PCC feature selection was higher than most others. Furthermore, the TPR for ESPRT was also better or close to most approaches, as shown in Table IV. The FPR for ESPRT fell in the range of 0.1 to 0.3, which is lower than or similar to the FPR values of other approaches used in the comparison. Additionally, ESPRT required less execution time or complexity, leading to reduced time complexity or cost and improved performance of the detection approach.

Finally, the mean accuracy for the proposed approach was 96.97%. The Confidence Interval (CI), which is a range of values that estimates where the true average accuracy likely falls 95% of the time, lies between 96.56% and 97.37%. The CI 95% for [16] lies in the range 91.35% and 94.38%, while the CI for [12] lies in the range of 16.79% and 83.79% which shows clearly that its performance is unreliable. However, the method in [17] shows decent performance, but ESPRT is more reliable. On the other hand, the P-value is an indicator showing whether the difference between methods is significant. The P-value of this approach was 0.0022 and 0.0277 compared to [16, 12], which means that the difference is statistically significant.

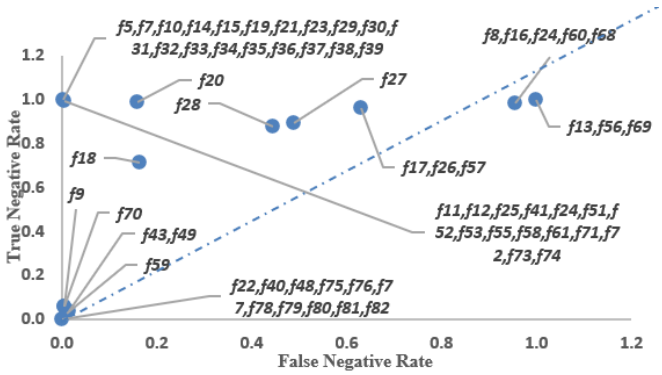

Fig. 4. Specificity (TNR) vs. Miss Rate (FNR) for all features of ESPRT using NTP attacks based on the CICDDoS2019 dataset.

TABLE IV. PEARSON CORRELATION OF ESPRT VS. STATE-OF-THE-ART APPROACHES USING CICDDOS2019

Ref	Approach name	Feature selection method	Features #	Accuracy	TPR or Recall	FPR	Time cost (s)
This study	ESPRT	PCC	5	97.27	88.45	0.02	0.189
			15	96.96	95.89	0.02	0.831
			25	97.23	96.01	0.02	1.712
			35	96.23	95.59	0.02	2.584
			45	97.20	96.32	0.01	3.456
			55	96.90	97.33	0.03	4.328
[16]	Bagging	PCC and tree-based.	21	93	94.8	9.5	-
	Boosting			92.2	94.04	9.3	-
	Stacking			93.4	96	8.9	-
[12]	ID3	RF	5	65	-	-	-
	RF			56	-	-	-
	Naïve Bayes			11	-	-	-
	Logistic Regress-ion			2	-	-	-
[17]	RF	MI	45	96.77	86	0	750
	XGBoos			96.67	85	0	1166
	DNN			96.17	58	0.03	590
	SNN			95.14	56	0.02	438
	DT			91.29	84	0.01	64

VI. CONCLUSION

DDoS attacks pose a significant cybersecurity threat. The ESPRT approach was implemented to detect infected traffic and identify anomalies in NTP and DNS. The PCC approach was implemented to select relevant features to improve detection performance. ESPRT achieved high accuracy, ranging from 97.27 to 96.23% when the number of features ranged from 5 to 55. ESPRT also has a very low FPR, ranging from 0.01 to 0.03. Furthermore, ESPRT generated high TPR and required less execution time or complexity, leading to reduced time complexity or cost and improved detection performance.

However, this experiment was conducted on one offline dataset. Another significant target for DDoS attacks is Software-Defined Networks (SDN), which represent a new infrastructure that enables network administrators to program network devices and servers, thereby enhancing network performance. Finally, this infrastructure has also become a target for DDoS threats, making its implementation a potential focus for future work.

REFERENCES

[1] A. Verma, R. Saha, N. Kumar, G. Kumar, and Tai-Hoon-Kim, "A detailed survey of denial of service for IoT and multimedia systems: Past, present and futuristic development," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19879–19944, Jun. 2022, https://doi.org/10.1007/s11042-021-11859-z.

[2] A. Sanmorino, L. Marnisah, and H. D. Kesuma, "Detection of DDoS Attacks using Fine-Tuned Multi-Layer Perceptron Models," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 16444–16449, Oct. 2024, https://doi.org/10.48084/etasr.8362.

[3] B. H. Ali *et al.*, "Shannon entropy based DDoS attacks detection using combination of machine learning based feature importance techniques,"

presented at the International Research Conference of Engineering and Applied Sciences 2023: IRCEAS2023, Baghdad, Iraq, 2025, Art. no. 030019, https://doi.org/10.1063/5.0257765.

[4] B. H. Ali, N. Sulaiman, S. A. R. Al-Haddad, R. Atan, S. L. M. Hassan, and M. Alghrairi, "Identification of Distributed Denial of Services Anomalies by Using Combination of Entropy and Sequential Probabilities Ratio Test Methods," *Sensors*, vol. 21, no. 19, Jan. 2021, Art. no. 6453, https://doi.org/10.3390/s21196453.

[5] R. Efendi, "Optimizing Neural Network Architecture for Detecting DDOS Attacks using ANN and XGBoost in Imbalanced Networks," *Engineering, Technology & Applied Science Research*, vol. 15, no. 3, pp. 22518–22526, Jun. 2025, https://doi.org/10.48084/etasr.9909.

[6] "DDoS Threat Landscape Report Q2 2022," *Resource Library*. https://www.imperva.com/resources/resource-library/reports/ddos-threat-landscape-report-q2-2022/.

[7] P. Chen, F. Li, and C. Wu, "Research on Intrusion Detection Method Based on Pearson Correlation Coefficient Feature Selection Algorithm," *Journal of Physics: Conference Series*, vol. 1757, no. 1, Jan. 2021, Art. no. 012054, https://doi.org/10.1088/1742-6596/1757/1/012054.

[8] P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," in *2016 IEEE International Conference on Communications (ICC)*, Kuala Lumpur, Malaysia, May 2016, pp. 1–6, https://doi.org/10.1109/ICC.2016.7510992.

[9] P. Valizadeh and A. Taghinezhad-Niar, "DDoS Attacks Detection in Multi-Controller Based Software Defined Network," in *2022 8th International Conference on Web Research (ICWR)*, Tehran, Iran, Islamic Republic of, May 2022, pp. 34–39, https://doi.org/10.1109/ICWR54782.2022.9786246.

[10] R. Li and B. Wu, "Early detection of DDoS based on φ-entropy in SDN networks," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Chongqing, China, Jun. 2020, pp. 731–735, https://doi.org/10.1109/ITNEC48623.2020.9084885.

[11] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, and S. Al-Sarawi, "Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates," *Applied Sciences*, vol. 12, no. 12, Jun. 2022, Art. no. 6127, https://doi.org/10.3390/app12126127.

[12] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, Oct. 2019, pp. 1–8, https://doi.org/10.1109/CCST.2019.8888419.

[13] M. H. Nguyen, Y. K. Lai, and K. P. Chang, "An Entropy-based DDoS attack Detection and Classification with Hierarchical Temporal Memory," in *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Sep. 2021, pp. 1942–1948.

[14] Z. Long and W. Jinsong, "A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN," *Computers & Security*, vol. 115, Apr. 2022, Art. no. 102604, https://doi.org/10.1016/j.cose.2022.102604.

[15] J. Li, "Detection of DDoS Attacks based on Dense Neural Networks, Autoencoders and Pearson Correlation Coefficient," M.S. Thesis, Dalhousie University, Canada, 2020.

[16] T. T. Khoei, G. Aissou, W. C. Hu, and N. Kaabouch, "Ensemble Learning Methods for Anomaly Intrusion Detection System in Smart Grid," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, Mt. Pleasant, MI, USA, May 2021, pp. 129–135, https://doi.org/10.1109/EIT51626.2021.9491891.

[17] V. Gaur and R. Kumar, "Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 1353–1374, Feb. 2022, https://doi.org/10.1007/s13369-021-05947-3.

[18] Z. S. Dhahir, "A Hybrid Approach for Efficient DDoS Detection in Network Traffic Using CBLOF-Based Feature Engineering and XGBoost," *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 2, pp. 174–190, Sep. 2024, https://doi.org/10.62411/faith.2024-33.