Enhancing the Performance of Detect DRDoS DNS Attacks Based on the Machine Learning and Proactive Feature Selection (PFS) Model

Riyadh Rahef Nuiaa, *Member, IAENG,* Selvakumar Manickam, Ali Hakem Alsaeedi, and Esraa Saleh Alomari

***Abstract*—The Domain Name System (DNS), which converts domain names to IP addresses, is a critical component of the internet infrastructure. Attackers exploit the existing potential vulnerabilities in this network protocol to launch their attacks. Distributed Reflection Denial of Service (DRDoS) DN S attacks are a type of Distributed Denial of Service (DDoS) attack that uses DNS vulnerabilities to carry out their attacks. These types can rapidly deplete the resources of the targeted victim system (computational and bandwidth). With the advancement of attack methods, both quantitatively and qualitatively, traditional methods used to detect DNS threats, particularly DRDoS attacks, became ineffective. Therefore, in this paper, a new model called proactive feature selection model PFS for early detection of DRDoS attacks based on DNS responses. The PFS model is divided into two stages: features selection stage and detection DRDoS attacks stage. The PFS model was validated using the standard CICDDoS2019 dataset. The results show that the PFS model achieves a high accuracy of 91.4368% and a very low FPR while reducing the number of features from 88 to 19 in the standard CICDDoS2019 dataset.**

***Index Terms*—vDRDoS DNS attack, DNS amplification attack, DNS reflection attack, DNS misuse, DDoS DNS attacks, DNS cyberthreats attacks.**

I. Introduction

CYBERSECURITY threats have become an impressive and broad research area in recent years because they affected human lives. Cybersecurity is sometimes classified as a form of terrorism, such as cyberterrorism. Cybersecurity attempts to protect against threats that arise as a result of a company's Internet connection. The more services a business offers over and through the Internet, the bigger the cyberthreat [1][2]. Cyberattacks are easy to automate and replicate, and you can anticipate them spreading freely across untrustworthy domains [3][4]. The term "cybersecurity" refers to the process of safeguarding cyberspace against cybercriminals. Hence, it is critical to implement cybersecurity successfully in order to safeguard the Internet system and the users from various threats. Internet and network-connected devices are targets of cybercrime [5][6][7]. There are a variety of data packets that may or may not be legal to pass through the network. Therefore, monitoring network traffic is extremely difficult and unattainable, especially when the traffic is massive. Additionally, domain name attacks are common and increasing in frequency[8][9]. As an outcome, it was necessary to design our model, which has demonstrated that it is capable of detecting these particles with high accuracy. The success rate of cyberattacks on critical infrastructure, people, and financial systems continues to rise. Thus, cyberattacks have risen to prominence as the most serious problem in the digital world[10][11]. Earlier this year, the Internet Corporation for Assigned Names and Numbers (ICANN) issued a warning stating: "There is an ongoing and serious danger to core components of the Domain Name (DNS) infrastructure"[12]. The most of cyberattacks take advantage of DNS's essential role in Internet traffic facilitation. To remain ahead of detection systems and blacklists, attackers may set up dozens, hundreds, or even thousands of malicious domains. DDoS assaults, like botnets and distributed denial of service attacks, can be quite sophisticated[13][14]. On a daily basis, threat actors use this system by registering and seizing control of thousands of Internet domains. These are utilized to launch a range of different types of cyberattacks[15]. Numerous authors discuss their strategies for detecting cyberattacks via DNS traffic monitoring. For instance, [16] describes how the process of the attack detection system collects cyberattack features from networks and generates feature vectors. The strategy described in [17] is heuristic in nature and is based on data gathered concerning cyberattacks. Genetic algorithms are also used to reduce the feature set, allowing for more efficient use of system resources for detection purposes. The ability to respond to cyberattacks' presence in the network and its hosts is another benefit. Anomaly traffic patterns can be identified, for example, in [18], by analyzing the temporal-spatial patterns of DNS behaviours of various IoT systems. In order to launch their attacks, most cyberattacks take advantage of DNS's role in streamlining Internet traffic[13]. Due to the widespread use of DNS on the Internet, it has been misused in a variety of ways to perpetrate a variety of attacks[19]. However, DNS is susceptible to two vulnerabilities: DNS resolvers have confidence in all responses received following the sending of requests, and the host address mapping provided by DNS is widely trusted by users. As a result, malicious attackers can exploit these vulnerabilities by introducing a bogus IP address into the Recursive Server cache[20]. Moreover, DNS zone information may give valuable information about a system's or network's underlying infrastructure, which an attacker may use to execute direct attacks such as Distributed

Manuscript received September 1, 2021; revised February 18, 2022.

Riyadh Rahef Nuiaa is a PhD student at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Malaysia, and Senior Lecturer in the Department of Computer/College of Education for Pure Sciences, Wasit University, Iraq (email: riyadh@uowasit.edu.iq).

Selvakumar Manickam is an associate professor at National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Malaysia (corresponding author to provide e-mail: selva@usm.my)

Ali Hakem Alsaeedi is Senior Lecturer at College of Computer Science and Information Technology, Universitas of Al-Qadisiyah, Iraq (email: ali.alsaeedi@qu.edu.iq)

Esraa Saleh Alomari is an associate professor at Department of Computer/College of Education for Pure Sciences, Wasit University, Iraq (email: ealomari@uowasit.edu.iq)

Reflection Denial of Service (DRDoS) [21][22]. There is a need to identify the key cybersecurity threats (vulnerabilities) in DNS, targeted/victimized applications, mitigation techniques, and infrastructures. Therefore, the authors present their model to detect the DRDoS DNS attacks with high accuracy (detection rate) and low false positives. This article introduces a novel model for detecting DRDoS DNS attacks called "A Proactive Feature Selection Model (PFS)." The PFS model is based on metaheuristic algorithms (MH), ML, and the adaptive threshold serves as the fitness function. The PFS model's primary function is to reduce the number of features. As a consequence, the adaptive threshold (fitness function) will be updated for each search in the population in order to eliminate features that are irrelevant or redundant. The purpose of this paper is to introduce a new model to detect DRDoS attacks based on DNS responses. DNS response traffic changes when the attacks have occurred. Therefore, the PFS model focuses on those features, which will detect those attacks with high detection accuracy and lower FPR. The following table summarizes the study's significant contributions:

• The proposed feature selection algorithm used a metaheuristic optimization algorithm and an adaptive threshold to optimize detection mechanism performance by reducing the number of features.

• The new PFS model aims to detect DRDoS DNS attacks and achieve this usefulness by diagnosing vulnerabilities in the intrusion detection system that are exploited by those attacks and thus improving detection accuracy. The results demonstrated that the PFS is capable of detecting DRDoS DNS attacks with high accuracy.

• The PFS model was validated, and the results were compared to three well-known metaheuristic optimization algorithms (PSO, BA, and DE). Finally, the results and discussion section includes comparison tables.

The CICDDoS2019 dataset was used to assess the performance, reliability, and validity of the current method for detecting DRDoS DNS attacks. The results indicate that the PFS model achieves a high degree of accuracy of 91.4368% detection rate for DRDoS attacks on the DNS protocol, as well as a decrease in the false positive rate.

A. motivation

At all times, security challenges pique researchers' interest, motivating them to seek out optimal solutions. As a result, the researchers shed light on DRDoS DNS attacks that target domain names by blocking and isolating legitimate users' services. In some instances, the effects of these attacks may be directly related to human life, such as attacks on vital facilities such as medical or nuclear facilities, energy centres, and other devices.

This type of cyberattack is intriguing because of the characteristics that distinguish it from other types of cyberattacks. Therefore, the researchers concentrated their efforts on developing a solution that mitigates the risk of these attacks. As a result, a new model has been developed to detect these attacks accurately. The new method is discussed in detail in the following section (proposed system).

The results indicate that the proposed model is capable of detecting domain name attacks with a high degree of accuracy despite the high volume of network traffic. The results section discusses and details the findings.

B. Paper organization

The remaining sections of this paper are as follows: Section II explains related works; the proposed model and its methodology are presented in section III; the enhancing method of features selection by using the PFS model are presented in section IV; section V contain the experiments and results of the PFS model, the conclusion and future work of the PFS model are shown in section VI.

II. RELATED WORK

Swarm Optimization and Evolutionary Algorithms were previously used to improve detection and minimize false positives. Thus, the number of features used has a significant impact on the detection quality of DRDoS DNS. Numerous articles have been published recently that discuss cyberattack detection (DRDoS DNS attacks) using DNS-based, network traffic analysis, and Internet of Things techniques (IoT).

According to [14] proposed a distributed-based defence mechanism (DDM). To counter the threat of DNS reflection/amplification, we provide a distributed defence mechanism called DDM. It consists of a combination of defences based on intermediate networks and defences directed at specific destinations. The technique's objective is to develop a defence mechanism that significantly reduces the computational cost of reflection/amplification attacks. DDM secures DNS by requiring authentication for DNS queries. Additionally, DDM employs a classification filtering approach that is activated only in the event that bogus traffic is detected.

According to [23], the suggested method is based on the relationship between the time unit and the packet, the number of packets received, the difference between the request and response sizes of packets, and the port that obtains more packets than usual. For DRDoS attacks, the machine learning model demonstrated a high detection accuracy.

According to [24], the well-known reflected attacks can be prevented by employing a reliable approach of packet filtering based on NAT. Only outward requests will be routed through NAT if the system of DRDoS is available. While DRDoS attacks are ongoing, the response that must be filtered is the illegitimate response.

According to [25] proposed traffic throttling using reinforcement learning RL, and the RL agent permits the traffic throttling technique by receiving traffic data. This model would be dynamically deployed at the primary router. The objective is to filter out the attack's traffic while maintaining as much legal traffic as possible by filtering the router's traffic to prevent amplification. Thus, it is more intelligent and efficient than the usual technique of traffic throttling based on port location.

According to [26] have proposed a novel method for detecting and defending against AR-DDoS attacks. DIDA is a distributed in-network defence architecture that is entirely distributed and operates on the data plane. As a result, it can be used to monitor and track individual user connections without causing confusion for network controllers.

According to [27], a new system aims to distinguish and mitigate DNS amplification, a type of distributed denial-of-service attack. The proposed system separates the stage of

attack detection from the stage of attack mitigation. It is based on SDN and consists of two stages: detection and mitigation of attacks. When the upper threshold is exceeded, the mitigation process proceeds to the second stage.

The method proposed by [28] includes a collection of geographically distributed routers known as the Barrier of Routers (BoR). The network that needs to defend itself should redirect all inbound and outbound traffic through the BoR before dropping the attack traffic at the amplification attacks. According to [29] proposed model by saving the history of DNS queries, Software-Defined-Networking (SDN) can be used to distinguish legitimate DNS replies from attack packets and use them as evidence. To mitigate this type of attack, the proposed method employs strict one-to-one mapping to distinguish orphan DNS replies from legitimate DNS responses.

According to [30] used K-means clustering and expectation-maximization techniques to distinguish DRDoS attack operations. Based on the mechanism's analysis, three attack scenarios have been revealed. First, open DNS resolvers received innocuous DNS queries via the Internet. Second, according to the researchers, this response amplification can lead to the largest amplification attack at DNS from small groups of machines.

According to [31], provide a strategy to identify and prevent DRDoS DNS attacks utilizing SDN and the SVM algorithm in this study. According to the research, the enormous size of the packets in a short period of time may indicate a reflection attack. It takes less time to stop an attack when both queries and answers are displayed.

According to [32], detect DNS DRDoS attacks with the help of Chukwa and Hadoop Single Node Cluster. MapReduce would then process the cluster's data. While monitoring workstations, responses to orphan DNS would be analyzed to detect DNS attacks that flood the victim with massive traffic in order to consume its resources. Simultaneously, no request for these responses has been made by the victim.

According to [33], the Brain Chain model is divided into two phases. The first phase is a model for detecting DDoS using machine learning, with the goal of revealing illegal flows (illegal DNS demands) in real-time. The second phase aims to mitigate the effects of illegal streams, which were implemented in order to recover the network as soon as possible. The rate of detection was used to assess the performance of the BF machine learning algorithm.

The primary disadvantage of SDN is that it is used when the switch's memory is full, which results in communication delays. This is thus one of the method's disadvantages.

III. PROPOSED SYSTEM

In previous work [34], we proposed a method to detect several types of DRDoS attacks but this paper focus on only detecting DRDoS DNS attacks. The proposed model is designed to detect DRDoS DNS attacks in the midst of massive network traffic. One of them is through the analysis of DNS traffic behaviour, a process referred to as DNS based DRDoS attack detection. As a result, network traffic can be monitored and analyzed for the purpose of detecting DRDoS DNS attacks. However, different types of packets traffic that pass through the network must be analyzed in order to achieve the model's objective. As a result, the proposed PFS model was evaluated using the benchmark dataset in this study [35] called CICDDoS2019, which contains 88 features. Figure 1 depicts the three stages of the proposed model: Data Pre-Processing is in charge of receiving network traffic and filtering DNS traffic rather than other packet traffic.

It has two levels: traffic filtering and data normalization. The second stage, termed DNS Features Selection, aims to select a DNS feature generated as a result of feature extraction. Our fitness function is an adaptive threshold, with ML classifiers and MH algorithms used to extract the most compelling features from these. Adaptive threshold is a technique that can be used to improve the search method in metaheuristic algorithms. As demonstrated in algorithm 1, the optimization technique (fitness function) has been validated against three well-known metaheuristic algorithms. All of the results indicate that the proposed model is more effective and superior to prior models. The proposed model is designed to detect DRDoS DNS attacks by analyzing the DNS traffic from the dataset used; Figure 1 shows the proposed model's general design. DRDoS DNS attacks use DNS responses to launch attacks, so our focus is on DNS responses and ignore other packets in the network. Therefore, the first step is to distinguish DNS packets from other packets, and Figure 2 shows how to separate DNS packets from other packets.

Algorithm 1 A Proactive Features Selection Model (PFS)

Input: Initialization Parameters
Output: Optimal Features
$\theta \leftarrow 0.5$ ▷ initially set the threshold ($\theta = 0.5$)
$\sigma \leftarrow 0$ ▷ sigma is a stagnation sensitive parameter
while $iter \leq Max - iteration$ **do**
    update population according to optimization method
    $Local_{best} \leftarrow$ find(best)
    if $Global_{best} < Local_{best}$ **then**
        $Global_{best} \leftarrow$Local$_{best}$
        $\sigma \leftarrow 0$
    else
        $\sigma \leftarrow \sigma + 1$
    if $\sigma \leq 2 * Population\ size$ **then**
        $\sigma \leftarrow 0$
        $\theta \leftarrow select\ randomly$ ▷ max-min Eq.
    return Optimal Features

A. Data Preparation Stage

Unnecessary data such as redundant, incomplete, noise, irrelevant, and many more can occasionally be found in packets. As a consequence, Figure 3 represents the data (packets) entering the Pre-Processing stage. The Min_Max data normalization mechanism is used to normalize data, and it converts or scales the data values of each function into a proportional set. The dataset used was normalized to the range [0, 1] [36] according to (1).

Furthermore, the data normalization process addresses missing values within datasets, clarifies outlier values, and resolves discrepancies. Normalizing the data is an important step in removing the dataset's biased features. The minimum and/or maximum values of the unnormalized data are used for rescaling.

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

Where min is the minimum value in feature, max is the maximum value in feature.

IV. ENHANCE THE FEATURE SELECTION BY USING THE PFS MODEL

The selection of features is a difficult issue. When the dimensionality of a feature is high, as in DNS packets, selecting the appropriate features is critical. Metaheuristic algorithms are the best fit. The proposed model yielded three subsets based on the PSO, BA, and DE algorithms. The proposed system used dynamic behaviour to set the value of θ that selected only features that ranked higher than the threshold to select an optimal value for the θ. Equation number (2) calculates the (θ) for wrapper models.

$$\triangle\theta = \theta Max - \theta Min \quad (2)$$

Where θ Max and θ Min can be determined by the user.

$$\overline{\theta} = \theta Min - \triangle\theta \times (1 - \frac{CurrnetIter}{MaxIter})^2 \times \lambda \quad (3)$$

Where θ Min and $\triangle\theta$ can be computed from Equation (2), CurrnetIter,MaxIter, and λ is a random variable in the interval [-1,1].

During the initial stage of the search process, the system should apply as high of θ as possible to restrict to features with high-rank values only. As the search progresses, the value of θ decreases. Figure 4 depicts the probability of θ during search iterations.

Optimal feature selection is influenced by the appropriate θ value chosen based on the initial value of θ. Therefore, the upper and lower bound of the period range of θ is not constant and can be changed by the user based on the resulting quality of the model. Therefore, the user can specify the range or period within which the θ is located. For the reason stated above, we used the adaptive threshold because the initial θ represents the threshold. We established an adaptive threshold with an initial value to distinguish between normal and abnormal behaviour. If the results do not satisfy and setting an acceptable threshold is not straightforward, the user can change this adaptive threshold value based on the result and expand the search space. The adaptive threshold is more useful than other thresholds because its ability to adapt to changes in network traffic during an attack and set an initial value of the threshold has become difficult.

V. EXPERIMENT

This section describes the Data Collection used in conjunction with the Data Pre-Processing Protocol in detail. We also provide the performance metrics that were used in our experiments. In addition, we show the architecture of our model. Finally, we compare our model to that of other classifiers. All experiments were carried out on a 2.90 GHz Intel Core i7 computer with 16 GB of RAM and the Windows 10 Pro - 64 bit operating system. PyCharm IDE Python and Python 3.8 are used to run our model. When the researchers created the PFS model, they focused on improving detection accuracy and reducing the number of features; these two factors are critical when developing a new approach for the intrusion detection system. IDS efficiency is measured using metrics based on its ability to categorize network traffic into the appropriate categories.

Table I shows the design of the matrix of confusion that contains potential classification cases like the predicted class and the actual class.

TABLE I
CONFUSION MATRIX FOR EACH ATTACK CLASS

		Predicted Class	
		Positive class	Negative class
Actual Class	Negative class	TN	FP
	Positive class	FN	TP

The researchers have been calculating the Accuracy, False-Positive (FP) and other accuracy metrics like Precision, Recall, and F1scor by the following equations:

$$TP = \frac{TP}{TP + FN} \quad (4)$$

$$TN = \frac{TN}{TN + FP} \quad (5)$$

$$FN = \frac{FN}{TP + FN} \quad (6)$$

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

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (8)$$

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1scor = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} \quad (11)$$

Table II displays the results of running the K-Nearest Neighbours KNN only, followed by PSO KNN without PFS, BA KNN without PFS, and DE-KNN without PFS. We use the three previous models to implement the PFS model. The results show that both PSO and BA with PFS achieve higher accuracy than other models other than PFS. Furthermore, when using PFS, the number of features decreased. The details have also revealed that the PFS-PSO-KNN has 19 features, the PFS-BA-KNN has 34 features, and the PFS-DE-KNN has 45 features. Thus, in terms of accuracy and number of features, the PFS-PSO-KNN and the PFS-BA-KNN outperform the PFS-DE-KNN.

Table III shows that when running the Random Forests, RF is used first, followed by PSO-RF without PFS, BA-RF without PFS, and DE-RF without PFS. The PFS model is then implemented in conjunction with the three previous models. The results show that both PSO and DE with PFS achieve greater accuracy than other models when compared

to PFS. When using PFS, the number of features is reduced, and the details are also displayed. The PFS-PSO-RF number of features is 49, the PFS-BA-RF number of features is 41, and the PFS-DE-RF number of features is 53. In terms of accuracy and other accuracy metrics such as precision, recall, and f1scor, the PFS-PSO-RF and PFS-DE-RF outperform the PFS-BA-RF.

Table IV demonstrates that when running the Support Vector Machine SVM only and then PSO-SVM without PFS, BA-SVM without PFS, and DE SVM without PFS, the PFS model is then implemented in conjunction with the three previous models. The results show that both PSO and DE with PFS achieve greater accuracy than other models when compared to PFS. The number of features reduced using PFS is also demonstrated; the details also demonstrate that the PFS-PSO-SVM number of features is 48, the PFS-BA-SVM number of features is 30, and the PFS-DE-SVM number of features is 45. In terms of accuracy and other accuracy metrics such as precision, recall, and f1scor, the PFS-PSO-SVM and PFS-DE-SVM outperform the PFS-BA-SVM.

Fig.5 shows that the accuracy line curve, while PSO-KNN without PFS appears to perform better in terms of detection accuracy in the early iterations, PSO-KNN with PFS yields a higher detection accuracy rate after iteration 68. The final accuracy rate achieved by PSO-KNN with PFS is 91.43%, compared to 89.02% without PFS. Furthermore, when using PFS, the number of features decreased from 88 to 19.

Fig. 6 depicts the accuracy line curve, despite the fact that in the early iterations, BA-KNN without PFS appears to perform better in terms of detection accuracy. After iteration 176, BA-KNN with PFS achieves a higher detection accuracy rate. BA-KNN with PFS achieves a final accuracy rate of 91.27%, compared to 89.1% without PFS. Furthermore, when using PFS, the number of features decreased from 88 to 34.

Fig. 7 shows that the accuracy line curve, although in the early iterations, DE-KNN without PFS, seems to perform better in terms of detection accuracy from the first iteration and yields a higher detection accuracy rate. The final accuracy rate achieved by DE-KNN without PFS is 93.55%, while it is 91.80% with PFS. Furthermore, when using PFS, the number of features decreased from 88 to 45.

Fig. 8 represents the accuracy line curve, despite the fact that, in the early iterations, PSO-RF without PFS appears to perform better in terms of detection accuracy. PSO-RF with PFS produces a higher detection accuracy rate after iteration 5. The final accuracy rate achieved by PSO-RF with PFS is 86.69%, compared to 84.74% without PFS. Furthermore, when using PFS, the number of features decreased from 88 to 49.

Fig. 9 shows that the accuracy line curve, although in the early iterations, BA-RF without PFS, seems to perform better in terms of detection accuracy from the first iteration and yields a higher detection accuracy rate. The final accuracy rate achieved by BA-RF without PFS is 86.73%, while it is 86.65% with PFS. Furthermore, when using PFS, the number of features decreased from 88 to 41.

Fig. 10 shows that the accuracy line curve, although in the early iterations, DE-RF with PFS seems to be performing better in terms of detection accuracy from the first iteration, and it yields a higher detection accuracy rate. The final accuracy rate achieved by DE-RF with PFS is 86.65%, compared to 81.03% without PFS. Furthermore, when using PFS, the number of features decreased from 88 to 53.

Fig. 11 shows that the accuracy line curve, although in the early iterations, PSO-SVM with PFS, seems to perform better in terms of detection accuracy from the first iteration and yields a higher detection accuracy rate. PSO-SVM with PFS achieves a final accuracy rate of 85.39%, whereas PSO-SVM without PFS achieves a rate of 83.13%. When PFS was used, the number of features dropped from 88 to 48.

Fig. 12 shows that the accuracy line curve, although in the early iterations, BA-SVM without PFS, seems to perform better in detection accuracy from the first iteration and yields a higher detection accuracy rate. The final accuracy rate achieved by BA-SVM without PFS is 85.40%, while it is 85.35% with PFS. Furthermore, when using PFS, the number of features decreased from 88 to 30.

Fig. 13 shows that the accuracy line curve, despite the fact that in the early iterations, DE SVM without PFS appears to perform better in terms of detection accuracy. After iteration 179, DE-SVM with PFS achieves a higher detection accuracy rate. The final accuracy rate achieved by DE-SVM with PFS is 85.405%, compared to 85.402% without PFS. Furthermore, when using PFS, the number of features decreased from 88 to 45.

VI. CONCLUSION

The massive growth in network traffic volume results in the discovery of some anomalies in traffic, which may represent cyberattacks. Verification of the massive volume of traffic with an immense number of features takes time and may result in inaccurate results, which may impact the correctness of the findings. As a result, the proposed PFS model has been trained on the CICDDoS2019 standard dataset containing mixed DNS traffic consisting of anomalous data traffic (DDoS DNS attacks) and benign data traffic. The PFS model passes through several iterations and attempts to achieve high detection accuracy and low false positive rate based on features selected, which played a crucial role in the detection strategy. The results show that the PFS model is better than other models in detecting DRDoS DNS attacks. The suggested model will be implemented in a real-world context to verify its correctness and to focus on the time required to trigger an alarm when an assault is detected.

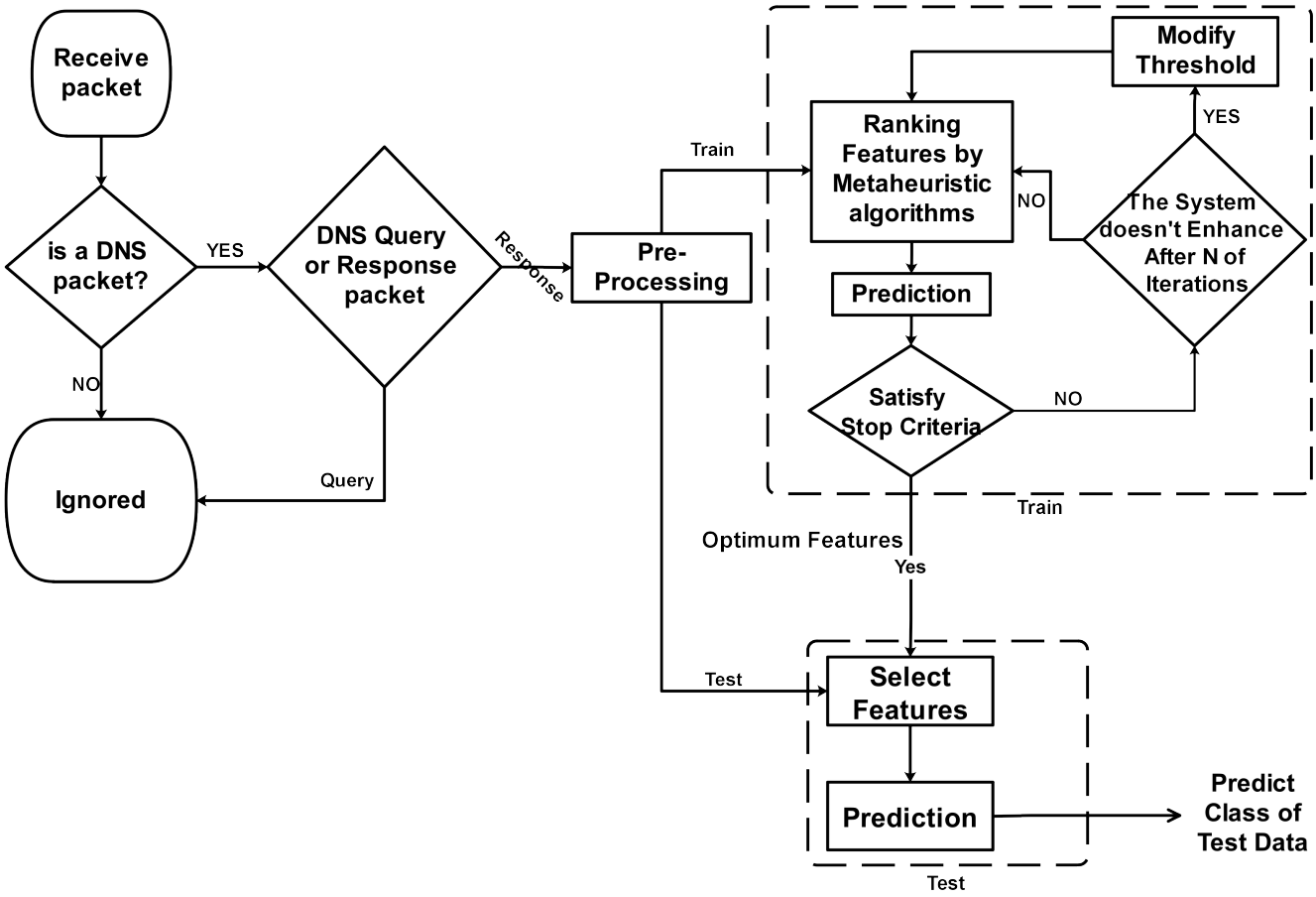

Fig. 1. The flowchart of the proposed model

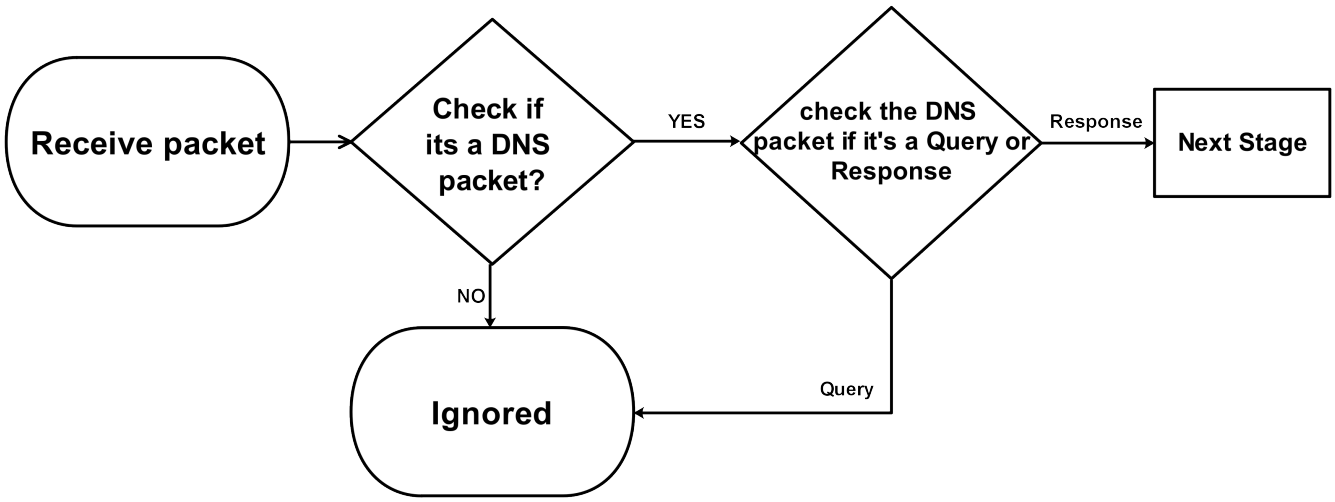

Fig. 2. Filtering the network traffic of dataset

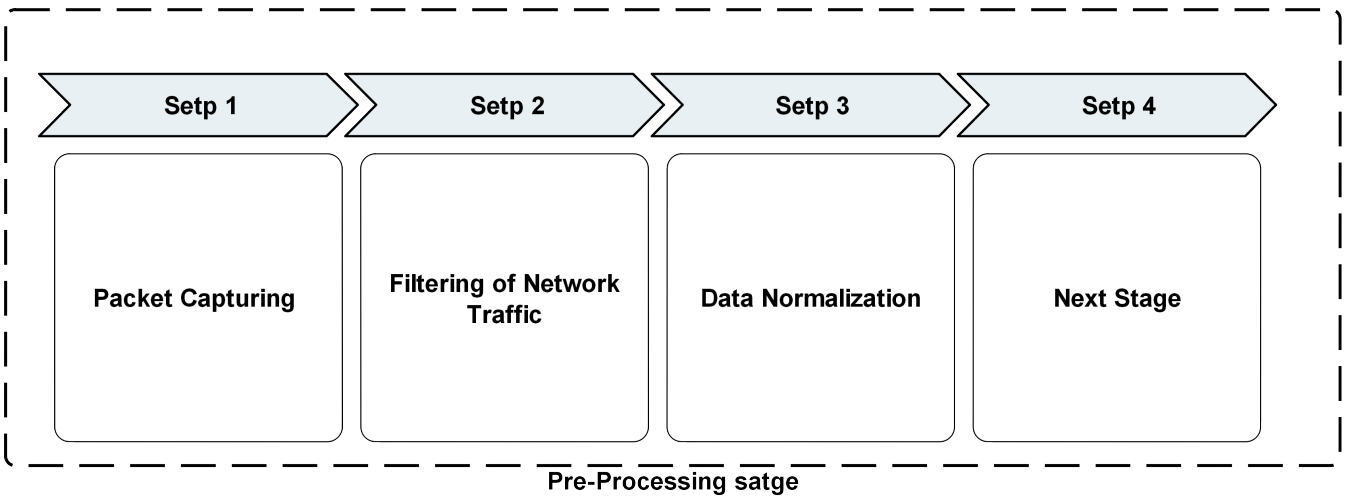

Fig. 3. Data Pre-Processing stage

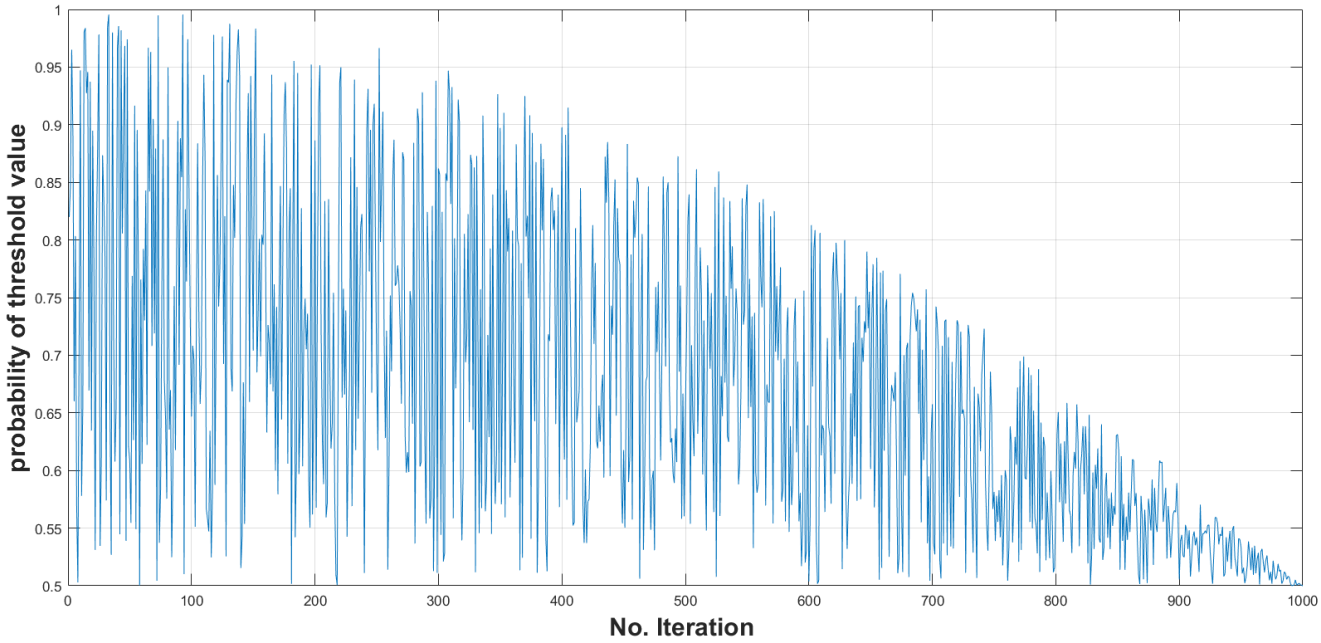

Fig. 4. the probability of theta θ

TABLE II
PFS MODEL PERFORMANCE AND KNN WITH THE THREE MH ALGORITHMS RELATIVE TO THE DRDOS DNS ATTACK

Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1scor	No. of features
KNN	88.4053	91.1772	8.8228	11.5947	89.7913	72.6454	88.4053	79.7542	88
PSO-KNN without PFS	84.3854	93.6603	6.3397	15.6146	89.0228	77.9141	84.3854	81.0207	45
PFS-PSO-KNN	86.2264	96.6473	3.3527	13.7736	91.4368	87.2105	86.2264	86.7156	19
BA-KNN without PFS	86.9435	91.2565	8.7435	13.0565	89.1	72.4931	86.9435	79.0634	37
PFS-BA-KNN	84.0796	98.4776	1.5224	15.9204	91.2786	93.6115	84.0796	88.5899	34
DE-KNN without PFS	88.1395	98.9698	1.0302	11.8605	93.5547	95.7762	88.1395	91.7993	65
PFS-DE-KNN	87.251	96.3657	3.6343	12.749	91.8084	86.4189	87.251	86.833	45

TABLE III
PFS MODEL PERFORMANCE AND RF WITH THE THREE MH ALGORITHMS RELATIVE TO THE DRDOS DNS ATTACK.

Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1scor	No. of features
RF	65.0431	98.7413	1.2587	34.9569	81.8922	93.2099	65.0431	76.6198	88
PSO-RF without PFS	70.1942	99.2956	0.7044	29.8058	84.7449	96.3235	70.1942	81.2088	46
PFS-PSO-RF	73.4396	99.956	0.044	26.5604	86.6978	99.7745	73.4396	84.6051	49
BA-RF without PFS	73.4884	99.9736	0.0264	26.5116	86.731	99.8646	73.4884	84.6699	37
PFS-BA-RF	73.3488	99.9648	0.0352	26.6512	86.6568	99.8193	73.3488	84.5609	41
DE-RF without PFS	63.3562	98.7061	1.2939	36.6438	81.0311	92.639	63.3562	75.2491	59
PFS-DE-RF	73.3732	99.9384	0.0616	26.6268	86.6558	99.6843	73.3732	84.5286	53

TABLE IV
PFS MODEL PERFORMANCE AND SVM WITH THE THREE MH ALGORITHMS RELATIVE TO THE DRDOS DNS ATTACK.

Model	TP	TN	FP	FN	Accuracy	Precision	Recall	F1scor	No. of features
SVM	61.5487	98.5562	1.4438	38.4513	80.0524	91.8651	61.5487	73.7114	88
PSO-SVM without PFS	66.8376	99.428	0.572	33.1624	83.1328	96.9781	66.8376	79.1351	31
PFS-PSO-SVM	70.8596	99.9384	0.0616	29.1404	85.399	99.6732	70.8596	82.8322	48
BA-SVM without PFS	70.8306	99.9824	0.0176	29.1694	85.4065	99.9063	70.8306	82.8927	35
PFS-BA-SVM	70.7269	99.9736	0.0264	29.2731	85.3502	99.8594	70.7269	82.8055	30
DE-SVM without PFS	70.8306	99.9736	0.0264	29.1694	85.4021	99.8595	70.8306	82.8766	46
PFS-DE-SVM	70.8735	99.9384	0.0616	29.1265	85.4059	99.673	70.8735	82.8416	45

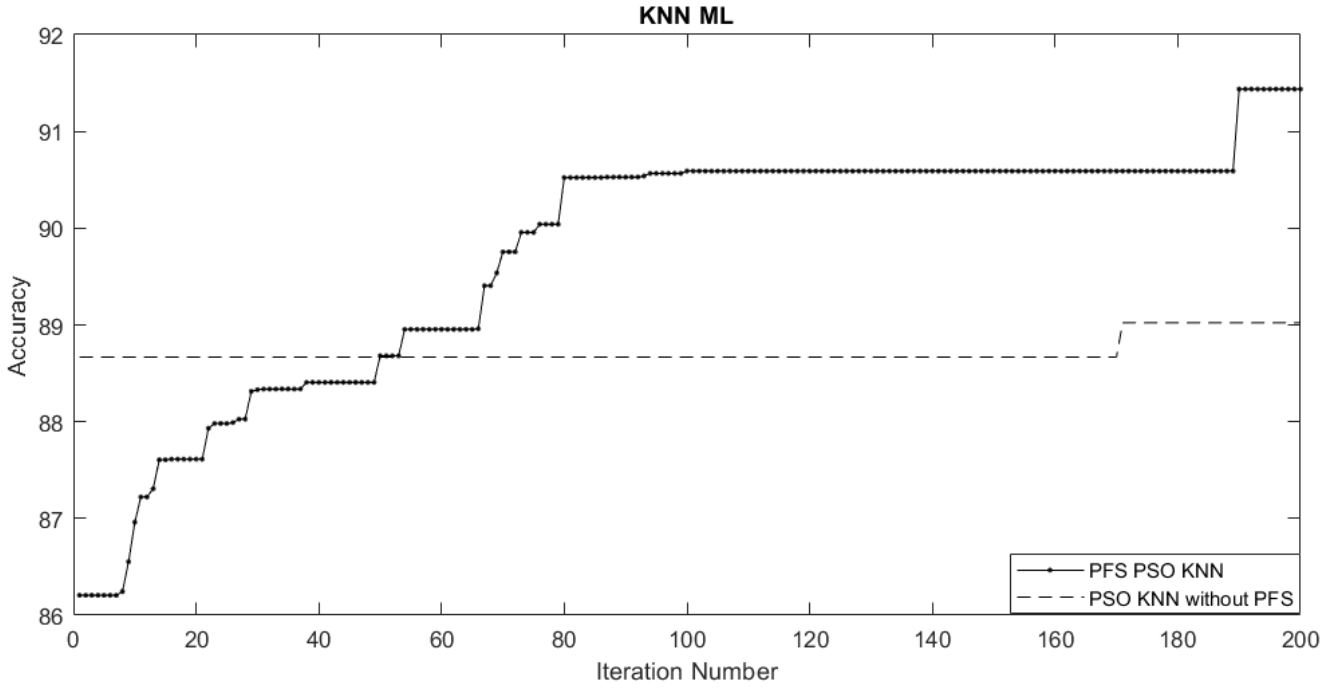

Fig. 5. Accuracy line curve of the PSO algorithm based on KNN

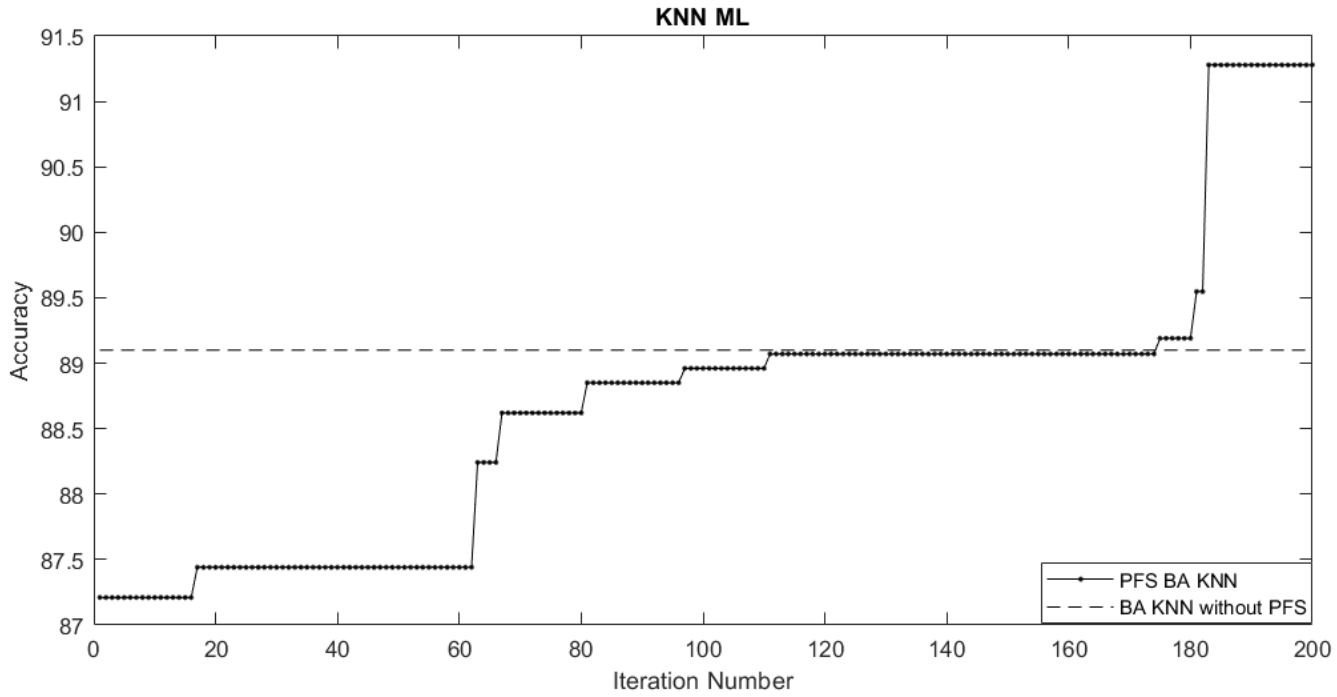

Fig. 6. Accuracy line curve of the BA algorithm based on KNN.

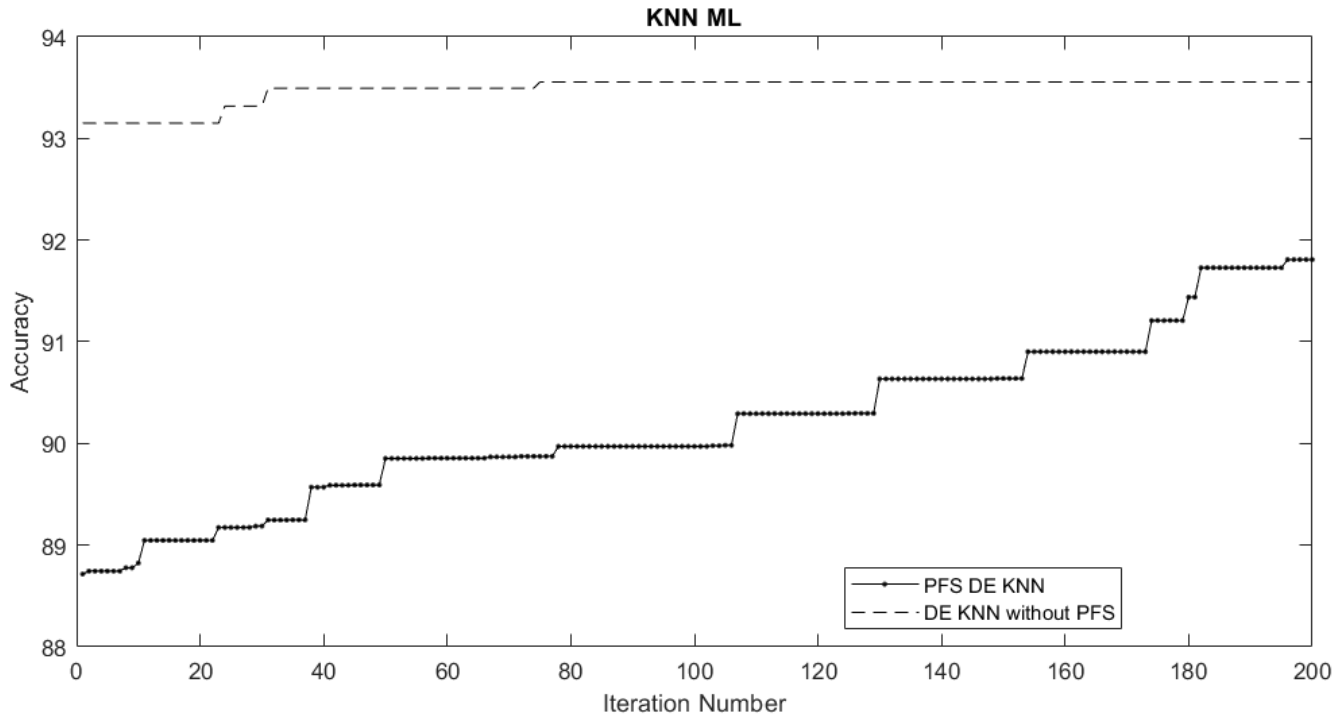

Fig. 7. Accuracy line curve of the DE algorithm based on KNN.

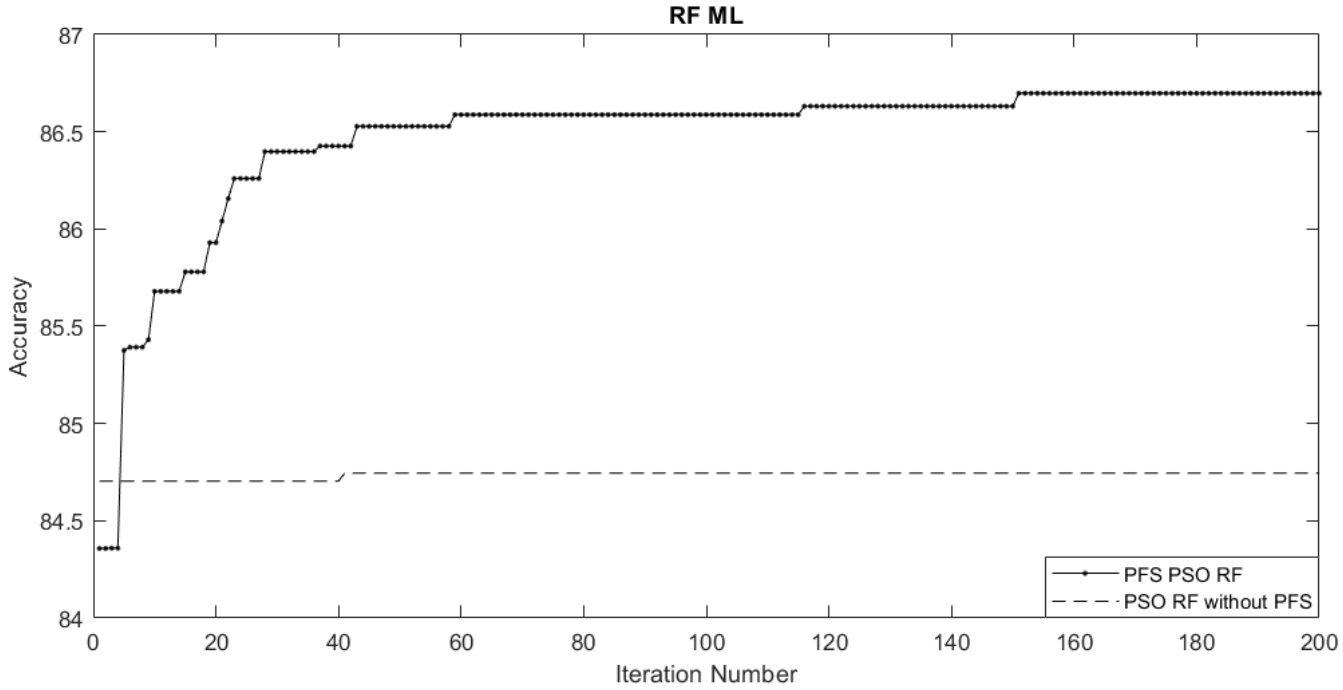

Fig. 8. Accuracy line curve of the PSO algorithm based on RF.

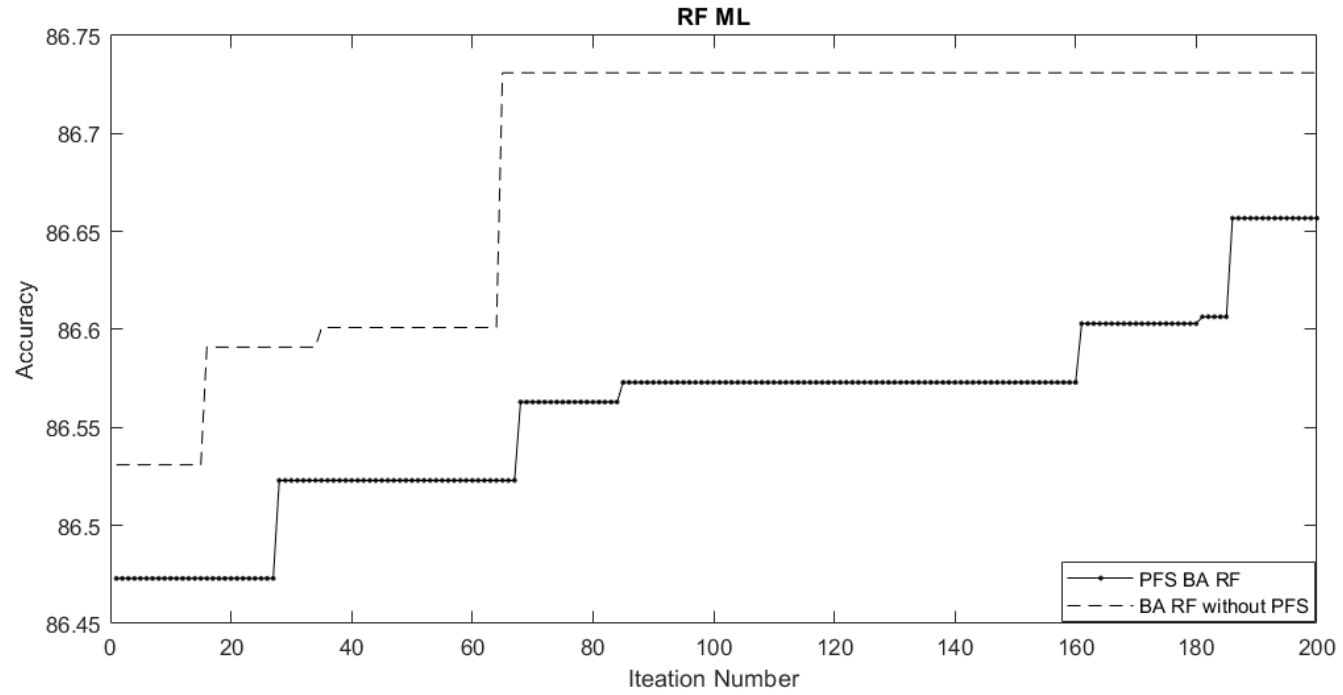

Fig. 9. Accuracy line curve of the BA algorithm based on RF.

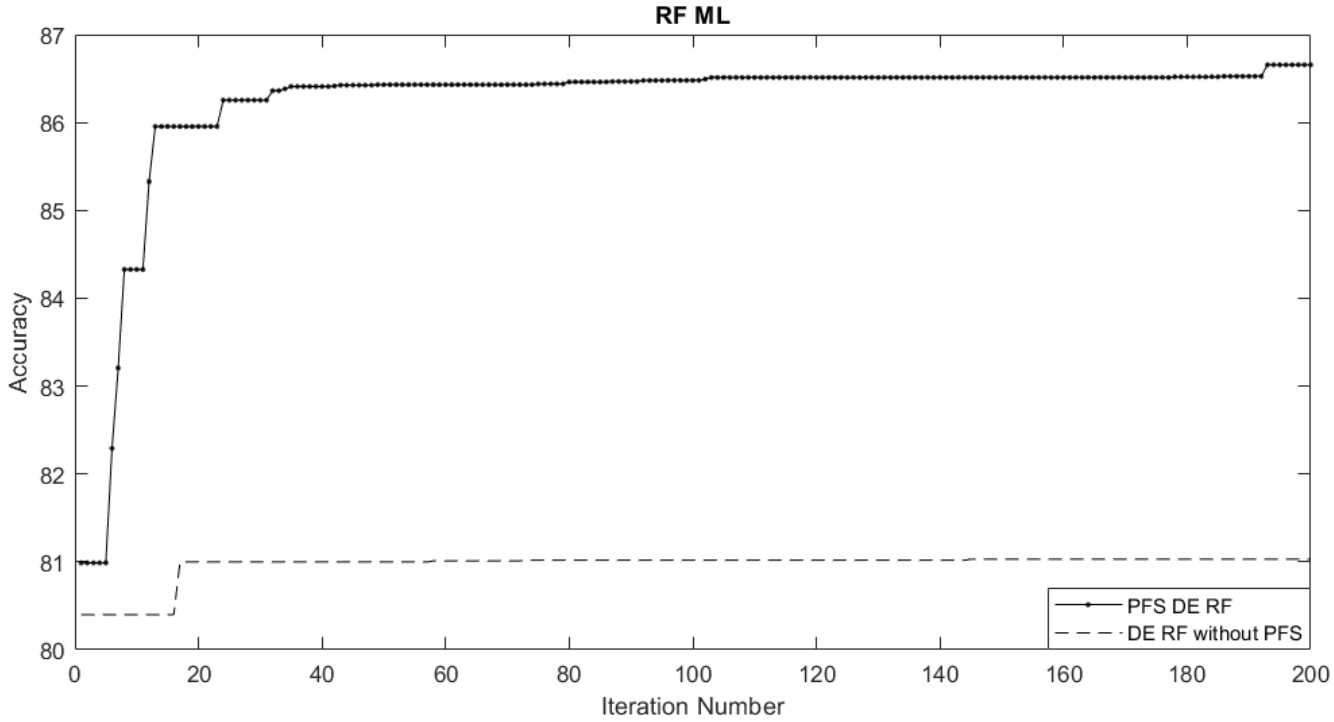

Fig. 10. Accuracy line curve of the DE algorithm based on RF.

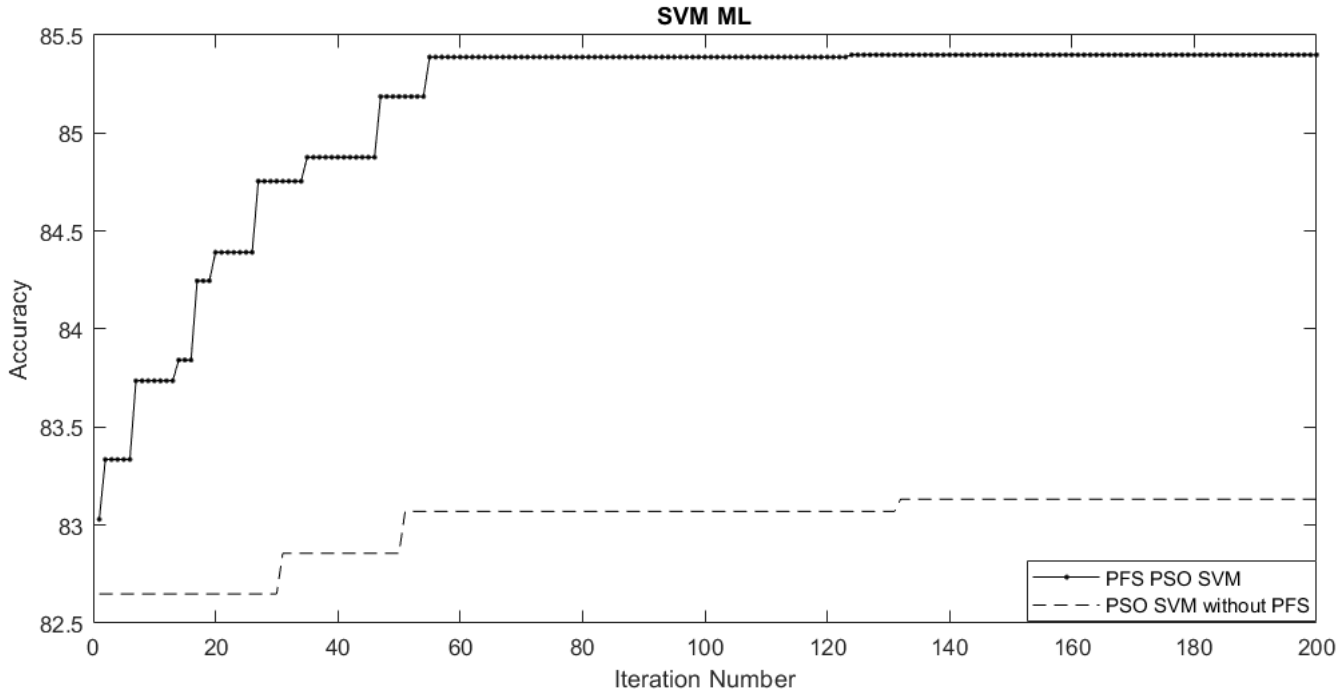

Fig. 11. Accuracy line curve of the PSO algorithm based on SVM

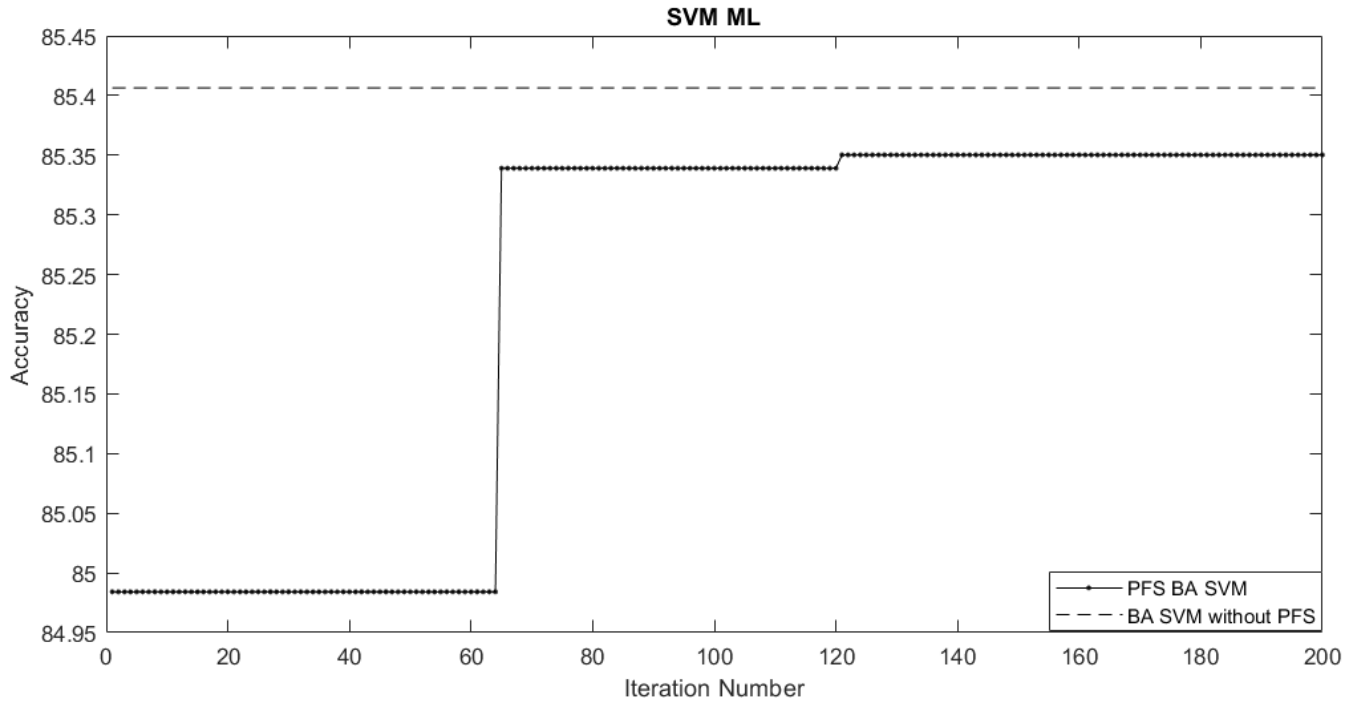

Fig. 12. Accuracy line curve of the BA algorithm based on SVM

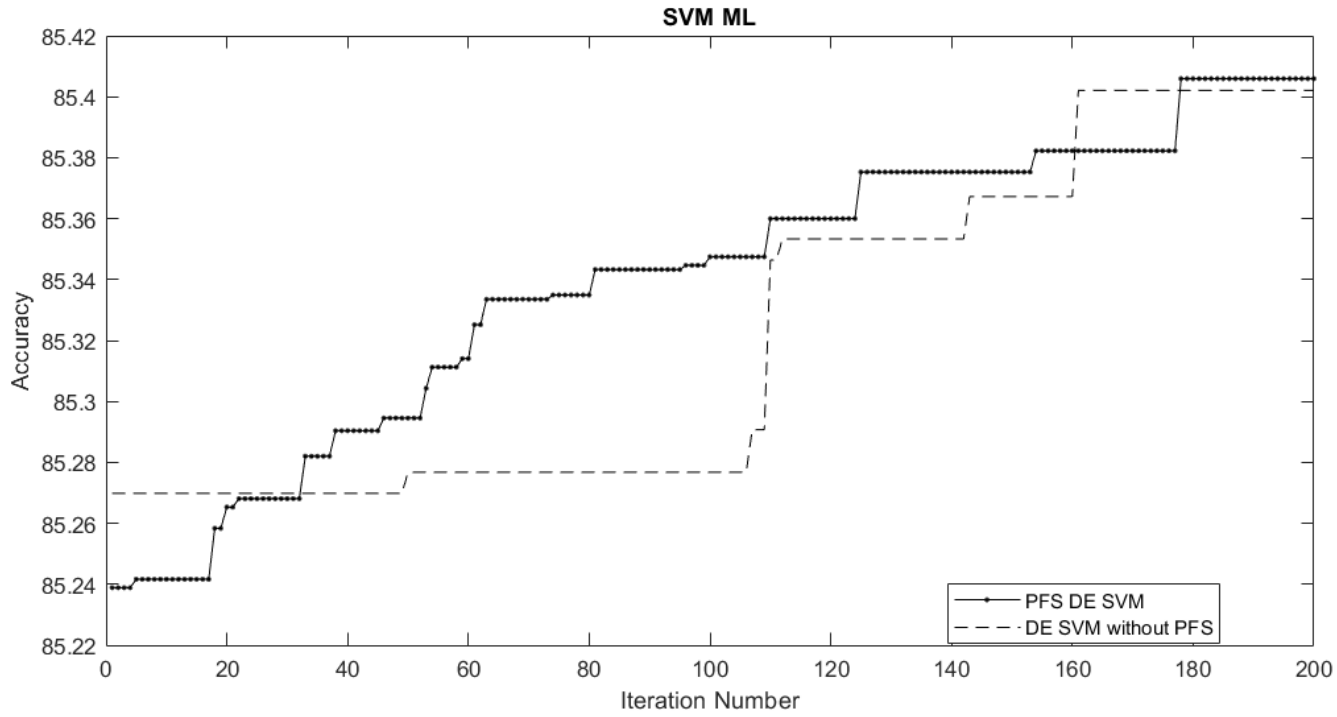

Fig. 13. Accuracy line curve of the DE algorithm based on SVM

REFERENCES

[1] B. von Solms and R. von Solms, "Cybersecurity and information security–what goes where?," Inf. Comput. Secur., 2018.

[2] H. Suryotrisongko and Y. Musashi, "Review of Cybersecurity Research Topics, Taxonomy and Challenges: Interdisciplinary Perspective," in 2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), 2019, pp. 162–167.

[3] R. Alguliyev, Y. Imamverdiyev, and L. Sukhostat, "Cyber-physical systems and their security issues," Comput. Ind., vol. 100, pp. 212–223, 2018.

[4] M. Z. Gunduz and R. Das, "Cyber-security on smart grid: Threats and potential solutions," Comput. networks, vol. 169, p. 107094, 2020.

[5] H. S. Brar and G. Kumar, "Cybercrimes: A proposed taxonomy and challenges," J. Comput. Networks Commun., vol. 2018, 2018.

[6] S. Lysenko, K. Bobrovnikova, O. Savenko, and A. Kryshchuk, "BotGRABBER: SVM-based self-adaptive system for the network resilience against the botnets' cyberattacks," in International Conference on Computer Networks, 2019, pp. 127–143.

[7] S. G. Qureshi and S. K. Shandilya, "Advances in Cyber Security Paradigm: A Review," in International Conference on Hybrid Intelligent Systems, 2019, pp. 268–276.

[8] S. Y. Enoch, Z. Huang, C. Y. Moon, D. Lee, M. K. Ahn, and D. S. Kim, "HARMer: Cyber-Attacks Automation and Evaluation," IEEE Access, vol. 8, pp. 129397–129414, 2020.

[9] M. Humayun, M. Niazi, N. Z. Jhanjhi, M. Alshayeb, and S. Mahmood, "Cyber security threats and vulnerabilities: a systematic mapping study," Arab. J. Sci. Eng., vol. 45, no. 4, pp. 3171–3189, 2020.

[10] "Check Point Research. The 2020 Cyber Security Report." https://research.checkpoint.com/2020/the-2020-cyber-security-report/ (accessed Jun. 22, 2021).

[11] FBI, "Cyber Crime — FBI," 2018. https://www.fbi.gov/investigate/cyber (accessed Jun. 22, 2021).

[12] A. Chikada, "Cyber security and the brand," Comput. Fraud Secur., vol. 2019, no. 9, pp. 6–9, 2019.

[13] C. Anderson and T. Saleh, "Investigating cyber attacks using domain and DNS data," Netw. Secur., vol. 2021, no. 3, pp. 6–8, 2021.

[14] R. R. H. Amin, D. Hassan, and M. Hussin, "Preventing DNS misuse for Reflection/Amplification attacks with minimal computational overhead on the Internet," Kurdistan J. Appl. Res., pp. 60–70, 2020.

[15] E. Paz and E. Gudes, "Comparison of DNS Based Methods for Detecting Malicious Domains," in International Symposium on Cyber Security Cryptography and Machine Learning, 2020, pp. 219–236.

[16] K. Bobrovnikova, S. Lysenko, and P. Gaj, "Technique for IoT Cyberattacks Detection Based on DNS Traffic Analysis," CERU, vol. 2623, p. 19, 2020.

[17] S. Lysenko, K. Bobrovnikova, R. Shchuka, and O. Savenko, "A Cyberattacks Detection Technique Based on Evolutionary Algorithms," in 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2020, pp. 127–132.

[18] K. Xu, F. Wang, S. Jimenez, A. Lamontagne, J. Cummings, and M. Hoikka, "Characterizing DNS Behaviors of Internet of Things in Edge Networks," IEEE Internet Things J., vol. 7, no. 9, pp. 7991–7998, 2020.

[19] D. Lazar, K. Cohen, A. Freund, A. Bartik, and A. Ron, "IMDoC: Identification of Malicious Domain Campaigns via DNS and Communicating Files," IEEE Access, vol. 9, pp. 45242–45258, 2021.

[20] Z. Yu, D. Xue, J. Fan, and C. Guo, "Dnstsm: DNS cache resources trusted sharing model based on consortium blockchain," IEEE Access, vol. 8, pp. 13640–13650, 2020.

[21] M. Skwarek, M. Korczynski, W. Mazurczyk, and A. Duda, "Characterizing Vulnerability of DNS AXFR Transfers with Global-Scale Scanning," in 2019 IEEE Security and Privacy Workshops (SPW), 2019, pp. 193–198.

[22] R. R. Nuiaa, S. Manickam, and A. H. Alsaeedi, "Distributed reflection denial of service attack: A critical review.," Int. J. Electr. Comput. Eng., vol. 11, no. 6, 2021.

[23] Y. Gao, Y. Feng, J. Kawamoto, and K. Sakurai, "A Machine Learning Based Approach for Detecting DRDoS Attacks and Its Performance Evaluation," in 2016 11th Asia Joint Conference on Information Security (AsiaJCIS), Aug. 2016, pp. 80–86, doi: 10.1109/AsiaJCIS.2016.24.

[24] T. Lukaseder, K. Stölzle, S. Kleber, B. Erb, and F. Kargl, "An SDN-based Approach For Defending Against Reflective DDoS Attacks," in 2018 IEEE 43rd Conference on Local Computer Networks (LCN), 2018, pp. 299–302.

[25] Y. Zhang and Y. Cheng, "An Amplification DDoS Attack Defence Mechanism using Reinforcement Learning," in 2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 634–639.

[26] X. Z. Khooi, L. Csikor, D. M. Divakaran, and M. S. Kang, "DIDA: Distributed In-Network Defense Architecture Against Amplified Reflection DDoS Attacks," in 2020 6th IEEE Conference on Network Softwarization (NetSoft), 2020, pp. 277–281.

[27] K. Özdinçer and H. A. Mantar, "SDN-based Detection and Mitigation System for DNS Amplification Attacks," in 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2019, pp. 1–7.

[28] V. Gupta and E. Sharma, "Mitigating DNS Amplification Attacks Using a Set of Geographically Distributed SDN Routers," in 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 392–400.

[29] S. Kim, S. Lee, G. Cho, M. E. Ahmed, J. P. Jeong, and H. Kim, "Preventing DNS amplification attacks using the history of DNS queries with SDN," in European Symposium on Research in Computer Security, 2017, pp. 135–152.

[30] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," Comput. Commun., vol. 62, pp. 59–71, 2015.

[31] C.-C. Chen, Y.-R. Chen, W.-C. Lu, S.-C. Tsai, and M.-C. Yang, "Detecting amplification attacks with software defined networking," in 2017 IEEE conference on dependable and secure computing, 2017, pp. 195–201.

[32] A. S. Jose and A. Binu, "Automatic detection and rectification of dns reflection amplification attacks with hadoop mapreduce and chukwa," in 2014 Fourth International Conference on Advances in Computing and Communications, 2014, pp. 195–198.

[33] Z. Abou El Houda, A. Hafid, and L. Khoukhi, "BrainChain-A Machine learning Approach for protecting Blockchain applications using SDN," 2020.

[34] R. R. Nuiaa, S. Manickam, A. H. Alsaeedi, and E. S. Alomari, "A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks.," Int. J. Electr. Comput. Eng., vol. 12, no. 2, 2022.

[35] 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), Oct. 2019, pp. 1–8, doi: 10.1109/CCST.2019.8888419.

[36] D. Singh and B. Singh, "Investigating the impact of data normalization on classification performance," Appl. Soft Comput., vol. 97, p. 105524, 2020.