Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

Detecting DDoS attacks using machine learning algorithms and feature selection methods

Mohammed Amin Almaiah[a,b*], Rana Alrawashdeh[c], Tayseer Alkhdour[d], Romel Al-Ali[e], Gaith Rjoub[f] and Theyazan Aldahyani[g]

[a]King Abdullah the II IT School, The University of Jordan, Amman 11942, Jordan
[b]Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan
[c]King Fahd of Petroleum and Mineral, Faculty of computer science and information system, Dhahran 31261, Saudi Arabia
[d]College of Computer Sciences and Information Technology, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[e]Associate Professor, the National Research Center for Giftedness and Creativity, King Faisal University, Saudi Arabia
[f]Faculty of Information Technology, Aqaba University of Technology, Aqaba, Jordan
[g]Applied College in Abqaiq, King Faisal University, Al-Ahsa 31982, Saudi Arabia

CHRONICLE

Article history:
Received: January 20, 2024
Received in revised format: May 18, 2024
Accepted: June 3, 2024
Available online: June 3, 2024

Keywords:
DDoS Attacks
Machine learning algorithms
Salp swarm algorithm (SSA)
PSO
GWO
SVM
KNN
ML

ABSTRACT

A Distributed Denial of Service (DDoS) attack occurs when an attacker tries to disrupt a network, service or website by flooding huge numbers of packets on the internet traffic. Detecting DDoS attacks serves the goal of spotting and addressing them promptly to reduce their effects on the network, system or service being targeted. Detecting Distributed Denial of Service (DDoS) attacks is crucial, for people, companies and network managers. The detection of DDoS attacks has ranging uses in industries such as network security safeguarding websites, managing cloud services ensuring the security of online systems and services. Detecting DDoS attacks is essential for safeguarding infrastructure upholding service availability and guaranteeing the security of online systems and services. To achieve this objective, we proposed a framework to detect DDoS attacks including six steps. In step one, we start by gathering information, which includes network activity and system records, for operations as well as instances of DDoS attacks. Step two, we identify characteristics of the data collected such as patterns in network traffic, packet details, IP addresses, types of protocols used and more. Step three, we utilize algorithms for feature selection such as Salp Swarm Algorithm (SSA), Gray Wolf Algorithm (GWA), Particle Swarm Algorithm (PSO) to pinpoint the features that can distinguish between normal activities and DDoS attack patterns. After that in step four, we divide the processed dataset into sections for training and testing purposes to develop and assess the machine learning models such as SVM (support vector machine), and KNN (K-nearest neighbor). Step five we develop a classification model using machine learning techniques like decision trees, forests, support vector machines (SVM) logistic regression models or neural networks. Finally, we assess the effectiveness of models through metrics such as accuracy rates, precision levels, recall rates, and F1 scores. The results show that the proposed models achieve high results (99.9%). In summary detecting DDoS attacks is crucial for protecting networks, systems and online services against disruptions.

1. Introduction

A Distributed Denial of Service (DDoS) attack is a cyber-assault where numerous compromised computers or devices often forming a botnet are utilized to flood a target system or network, with traffic Mohsin et al., (2021). The main aim of a DDoS attack is to deplete the target's resources like processing power or memory making the system or network incapable of handling

* Corresponding author.
E-mail address talkhdour@kfu.edu.sa m.almaiah@ju.edu.jo (M. A. Almaiah)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2024.6.001

user requests Mohsin et al. (2021) and Trab et al. (2018). During a DDoS attack the attacker commonly seizes control over a group of compromised devices by infecting them with malware or exploiting vulnerabilities as shown in Fig. 1. Once these devices, also referred to as “bots” or “zombies” are under the attacker’s command they are directed to generate a volume of traffic towards the target Anirudh et al., (2017). This excessive traffic inundates the target's infrastructure resulting in service degradation or complete unavailability, for users Lee et al. (2022). The goals of spotting DDoS attacks through choosing features and ML strategies include distinguishing between attack traffic, promptly detecting for swift responses adjusting to changing attack methods reducing the occurrence of false alarms and misses ensuring scalability and effectiveness and establishing a strong detection system Alahmadi et al., (2023) . By meeting these aims companies can proactively. Address DDoS attacks, lessen service interruptions, safeguard systems and information and uphold a safe online space for users Aljuhani (2021).

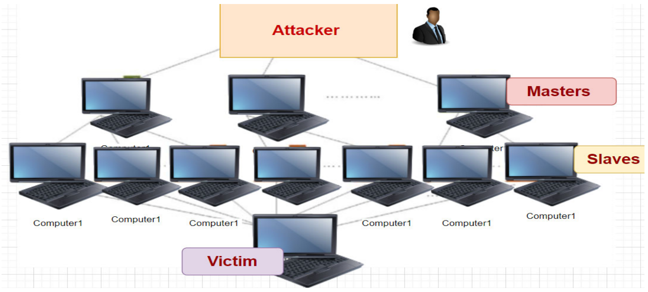


Fig. 1. DDoS Attack Architectures

Detecting Distributed Denial of Service (DDoS) attacks using feature selection and machine learning methods brings about advantages. It comes with its own set of challenges Iftikhar et al. (2023). The evolving tactics employed in these attacks present a hurdle for ML based detection approaches necessitating updates to identify emerging attack patterns. The limited availability of labeled training data impacts the precision of ML models. Striking a balance between minimizing positives (Mohmand et al., 2022). False negatives prove to be a complex task, compounded by the dynamic nature of DDoS attacks that makes selecting pertinent features for detection quite intricate Fauzi et al. (2020). Managing high volume traffic can strain scalability and performance particularly when dealing with traffic that complicates the analysis of packet level details. Moreover, resource exhaustion attacks aimed at the detection system hinder the detection process Liu (2023). Zero day attacks pose a challenge for ML models due to the absence of data on which to base detections. Overcoming these obstacles necessitates research and development efforts to enhance models, refine feature selection methodologies and explore detection strategies Ahmad et al., (2021).

Detecting Distributed Denial of Service (DDoS) attacks plays a role in ensuring the security of networks, online services, internet service providers (ISPs) banks, government entities, server facilities, gaming platforms and Internet of Things (IoT) gadgets Onah et al., (2021). This practice is essential for fortifying infrastructure averting disruptions, safeguarding information, maintaining operations delivering optimal user experiences and protecting critical systems. The significance of identification and mitigation is highlighted in applications to uphold accessibility, safety and dependability in today's interconnected digital environment Ullah et al., (2023) Identifying DDoS attacks is driven by the necessity to uphold network accessibility, safeguard infrastructure assure the security of data and systems facilitate countermeasures and responses uphold user satisfaction and credibility avert monetary losses, adhere, to regulations and acquire preemptive threat insight Alzahrani and Alzahrani (2021).

Machine learning and deep learning play a role in uncovering DDoS attacks. Support Vector Machines (SVM) Random Forests, Neural Networks like MLP and CNN Long Short-Term Memory (LSTM), auto encoders, Generative Adversarial Networks (GANs) Recurrent Neural Networks (RNN) and XGBoost're employed algorithms for this purpose. These algorithms help in categorizing network traffic, recognizing patterns capturing time related dependencies and spotting activities linked to DDoS attacks Zhao (2023). The selection of the algorithm is influenced by factors such as the data computational capacities and specific system needs. Additionally, feature engineering, data preparation and model enhancement methods play a role in detecting DDoS attacks Marvi et al., (2021). In our work, we start by gathering Data; Acquire network traffic data. Then, extracting Features; Identify features, from the collected data. Next, prepare data; process the features. After that, Choosing Features using (GWO, PSO, SSA); Decide on the features. Next, splitting the dataset; Segment the data into training and testing sets. Moreover, training models. Apply machine learning algorithms such as SVM and KNN to train the model. Finally, evaluating the model. Assess the model’s performance using the test data and optimize Model and adjust model parameters for effectiveness.

When utilizing optimization techniques such, as Grey Wolf Optimizer (GWO) Particle Swarm Optimization (PSO) and Salp Swarm Algorithm (SSA) in tandem with machine learning (ML) for identifying DDoS attacks there are some research question divided into; RQ1: How can GWO, PSO and SSA be effectively employed to optimize feature selection for detecting DDoS attacks with machine learning?;RQ2: What are the best parameter settings and setups for GWO, PSO and SSA when it comes to feature selection for spotting DDoS attacks?RQ3: How do the performances of GWO, PSO and SSA stack up against each other and against feature selection methods, in the realm of detecting DDoS attacks using machine learning?RQ4: Can incorporating GWO, PSO or SSA with ML algorithms enhance the precision and effectiveness of attack detection compared to using ML alone?

When detecting DDoS attacks using feature selection methods and machine learning (ML) techniques the impacts can differ based on the approaches employed. So, the potential benefits that can arise in these contributions. Firstly, improved detection accuracy; Employing feature selection techniques assists in pinpointing the features for detecting DDoS attacks thereby enhancing the accuracy of the detection system. ML algorithms trained on these chosen features can effectively distinguish between attack traffic patterns. Secondly, reduced data complexity; Feature selection reduces data complexity by selecting a subset of features. This reduction in complexity streamlines processes and memory usage making the detection process more efficient. Thirdly, enhanced interpretability. Through feature selection, a model with enhanced interpretability can be achieved by focusing on features. This fosters a deeper comprehension of the factors influencing DDoS attacks and aids in developing countermeasures. Fourthly, scalability; Feature selection methods capable of handling datasets enable the development of DDoS detection systems. These systems can be effective. Analyze amounts of network data making them suitable for high traffic networks. Finally, comparative evaluation; Assessing feature selection methods and machine learning algorithms provides insights into their strengths and weaknesses in DDoS attack detection. Such evaluations help determine the combination of techniques for specific detection needs.

The paper is organized in this manner; In Section 2 we mention the existing research, on detecting DDoS attacks. Section 3 discusses the research subject and the hypothesis formulated for this study. Our methodology is explained in Section 4 and Section 5 describes the experiments carried out in this study. Finally, Section 6 concludes with a discussion of the results.

2. Related Works

Several studies have been conducted in literature to study the detection of DDoS attacks in different domains such as IoT, mobile computing, wireless networks, etc. For instance, Mohsin et al., (2021) outlined a framework for detecting and stopping Distributed Denial of Service (DDoS) attacks using machine learning methods. The authors aim to overcome the shortcomings of techniques by suggesting a strategy that merges different machine learning algorithms with real time monitoring. Commencing with an introduction stressing the significance of DDoS detection and prevention measures, the paper conducts an examination of relevant literature to lay down the present situation in the field and pinpoint deficiencies in existing strategies. The proposed methodology delineates an encompassing approach for DDoS detection and prevention. It elucidates the process of data collection likely involving gathering network traffic data and system logs followed by preprocessing the collected data to sanitize and prepare it for analysis. To identify and thwart DDoS attacks the authors utilize machine learning models. They discuss the models and algorithms utilized which might include known techniques like Support Vector Machines (SVM) Random Forests or Neural Networks. In the discussion and analysis segment the authors scrutinize their findings. Juxtapose their approach against existing methods. They showcase the points, drawbacks and possible areas for enhancement in their strategy. The paper wraps up by outlining the benefits of the suggested framework and its importance, in the realm of detecting and preventing DDoS attacks. Ray et al., (2022) discussed a method for identifying and preventing Distributed Denial of Service (DDoS) attacks that target information in mobile healthcare (M healthcare) settings. It stresses the significance of protecting data in M healthcare systems. Acknowledges the growing threat of DDoS attacks in this area. The suggested approach comprises elements. Initially it outlines a technique for recognizing DDoS attacks on M healthcare data by examining network traffic patterns and pinpointing traffic behavior. Machine learning and anomaly detection algorithms might be utilized to improve the identification process. To mitigate the consequences of DDoS attacks the paper proposes actions. These actions might involve filtering traffic limiting rates or implementing resource allocation strategies to maintain the availability and integrity of data in M healthcare systems. Furthermore, the authors introduce a defense framework that encourages collaboration and information exchange among entities within the M healthcare community. This framework facilitates detection and coordinates responses to DDoS attacks thus enhancing security measures. The suggested approach is likely backed by an assessment, which could include simulations or real-world trials. The evaluations are conducted to measure how well the security measures safeguard M healthcare information, from DDoS attacks. Alahmadi et al., (2023) introduced the Genetic Algorithm Naive Bayes, for Anomaly Detection Model (GANBADM) to enhance the detection of activities targeting fog devices more effectively. With the growing number of devices and services in cloud fog computing can provide access to services since mobile devices are located near fog nodes. However, this proximity can lead to security concerns. Increase vulnerability due to resources in fog nodes. Hence the GANBADM model was proposed to help fog nodes better differentiate between traffic and anomalies. Specifically, GANBADM streamlines attributes to reduce the time complexity of the model while maintaining accuracy in identifying malicious activities. The model utilizes algorithms for feature selection and Naive Bayes as a classifier, for detecting network anomalies. Evaluation of the model was conducted using the NSL KDD dataset to identify DoS, Probe, R2L and U2R attacks. To validate the reliability of the suggested model this study compared the 19 features chosen using the wrapper approach GA against all the features in the dataset in terms of accuracy, precision, False Positive Rate (FPR) and execution time. The model was pitted against SVM, RF and DT classifiers. The findings indicated that GANBADM achieved an accuracy of 99.73%, precision of 99.10% FPR of 0.6% and an execution time of 0.18 seconds outperforming algorithms and classifiers and demonstrating the effectiveness of the proposed model. However, the F1 score results were lower compared to classifiers due to data quality issues within the dataset. Therefore, further evaluation with real cloud data is essential to confirm its efficacy. Moreover, as the current dataset only includes attacks it restricts fog nodes from recognizing those specific attacks while leaving them vulnerable to other attack types. Hence assessing the model with attack types is crucial, for ensuring its effectiveness. Alzahrani (2021) conducted a study on the performance of six machine learning classifiers in identifying 365 DDoS attacks using the Random Forest Regressor (RFR) as a feature selection tool for networks. The research compared the outcomes of machine learning classifiers with the CICDDoS2019 dataset, which covers types of DDoS attacks based on accuracy, precision, recall, F1 score and processing time. By utilizing RFR the feature selection was narrowed down from 80 to 24 features to detect 11 types of DDoS attacks. These selected features were then applied across all classifiers in the evaluation process. Decision Trees (DT) emerged as the classifier among others. In terms of accuracy, DT and Random Forest (RF) achieved a rate of 99%. Similarly, K Nearest Neighbors (KNN) Decision Trees (DT) Random Forest (RF) and Logistic Regression (LR) showed an accuracy rate of 99%, in precision and F1 score. For recall rates KNN, DT and RF had the ratio at 99%. On the contrary Naive Bayes (NB) KNN and DT exhibited the computation times respectively. Overall, both DT and RF yielded results in terms of accuracy, precision recall rates and F1 scores; however, DT displayed processing time at 4.53 seconds compared to RFs 84.2 seconds. Therefore, this indicates that the Decision Tree (DT) is the classifier for IoT networks. Additionally, the authors have demonstrated that their proposed system outperforms datasets when using machine learning classifiers in terms of accuracy. However, the comparison with datasets was not equitable since no feature selection algorithms were utilized in those datasets. While the proposed system shows promising results in detecting DDoS attacks it is important to note that the dataset used collected network traffic from servers with resources. Considering that networks have limited resources this makes the proposed system unsuitable for deployments. Further research is essential to assess the effectiveness of detecting DDoS attacks, for networks using IoT network traffic datasets.

Another research conducted by Marvi et al. (2021) explored DDoS attack detection using a machine learning approach, with the boosting machine (LGBM) algorithm for training. The study utilized two methods from the integrated feature selection (IFS) approach, including filter and embedded techniques to select features of identifying various types of DDoS attacks. Upon evaluating the model, the results showed an enhancement of 20% in the performance of the proposed model compared to existing models in literature concerning DDoS attack detection. On a note Norouzi et al. [17] focused their study on proposing an intrusion detection method for cyber-attacks based on a Genetic algorithm utilizing the Random Forest Model in IoMT. The research employed two datasets: NSL KDD and UNSW 2018_IoT_Botnet. Following validation of their framework it was discovered that their proposed approach demonstrated performance in terms of detecting cyber-attacks achieving a percentage accuracy rate of 99.9% along, with 100% recall and precision when compared to previous machine learning algorithms. Similarly, Seifousadati et al. (2021), the researchers have utilized a combination of machine learning and data mining techniques to identify DDoS attacks on devices. They employed the CICDDoS2019 dataset to test their model, which demonstrated 100% accuracy in detecting these attacks. Ismail and colleagues utilized machine learning methods like Random Forest and Boost to classify types of DDoS attacks achieving an accuracy of 90%. Additionally, Halim et al. Introduced an approach called GA based Feature Selection Method (GbFS) to enhance accuracy for cyber-attack detection, in networks by analyzing network traffic using machine learning algorithms to detect potential malware intruders. The research utilized three sets of data, Bot IoT, UNSW NB15 and CIRA CIC DOHBrw 2020. The results indicated an enhancement when employing the GbFS method, achieving an accuracy rate of 99.80%. Table 1 summarized the related works in terms of DDoS attack types in different layers.

Table 1
DDoS attack Types

Attack Type	Description
Volume-Based Attacks	Overload the intended system or network, with an amount of traffic to exhaust its capabilities.
	Overwhelm the target network by sending an abundance of ICMP (Internet Control Message Protocol) packets. A technique known as ICMP flooding.
	To perform a UDP flood attack you flood the target with a number of UDP (User Datagram Protocol) packets.
	SYN Flood; This attack takes advantage of the TCP three way handshake by inundating the system with a volume of SYN requests to overwhelm its resources.
	Conduct a Ping Flood attack by sending ICMP Echo Request (ping) packets to the target in order to deplete its resources.
	DNS Amplification; Taking advantage of set up DNS servers to create an amount of traffic, towards the desired target.
Protocol-Based Attacks	Take advantage of weaknesses, in network protocols to interrupt the availability of services.
	The ICMP Smurf Attack involves utilizing broadcast ICMP requests to generate a deluge of ICMP replies directed towards the target.
	Ping of Death refers to the act of sending improperly formatted ping packets to disrupt or halt the functioning of the targeted system.
	Exploit the fragmentation of IP, in TCP/IP to flood network devices with packets causing them to become overwhelmed.
	The Teardrop Attack involves exploiting IP fragmentation by sending overlapping or malformed fragments to disrupt the target system.
	SYN/ACK Flood; Take advantage of the TCP handshake procedure by overwhelming the destination, with SYN/ACK packets.
Application Layer Attacks	Exploit weaknesses, in apps or services to interfere with their accessibility or drain server capacities.
	HTTP Flood is when a web server gets bombarded with a number of HTTP requests causing it to run out of resources.
	The Slowloris Attack involves manipulating the HTTP protocol by sending delayed requests in order to maintain server connections.
	DNS Flood occurs when an excessive amount of DNS requests overwhelms the servers causing them to become unresponsive.
	Conduct a SIP Flood attack, by SIP servers with an amount of SIP request traffic.
	Leveraging NTP servers to create an amount of amplified traffic directed towards the intended target.
Hybrid Attacks	Utilize methods of attack to execute well-coordinated Distributed Denial of Service (DDoS) assaults.

3. Research Methodology

In our work, detecting DDoS attacks by utilizing feature selection such as GWO, PSO, SSA along, with Support Vector Machine (SVM) and K Nearest Neighbors (KNN) algorithms involves the process of choosing features from network traffic data and using machine learning models to differentiate between attack traffic. Selecting features plays a role in this method aiming to pinpoint the informative and distinguishing features that can effectively separate normal traffic from DDoS attack traffic. Various techniques like analysis, information theory or correlation-based methods can be employed to assess the significance of features and pick a subset that captures the characteristics of the traffic. After completing the feature selection phase using GWO, PSO, and SSA the chosen features are fed into machine learning models such as SVM and KNN for classification purposes. SVM is a classification model that constructs a hyperplane to distinguish different classes in a high dimensional feature space. Its goal is to maximize the margin between classes for classification. In contrast KNN is a parametric technique that assigns a class label to a data point based on its k nearest neighbor's classes. It calculates distances or similarities between data points. Identifies the majority class among the k neighbors thereby assigning a label to the target point. To identify Distributed Denial of Service (DDoS) attacks, specific features are utilized to train Support Vector Machine (SVM) and K Nearest Neighbors (KNN) models, with labeled data that consists of attack traffic instances. These models grasp the patterns and traits of each category during training. In the detection phase the trained models are used on test data. They analyze the features of test instances. Categorize them as DDoS attack traffic based on patterns learned in training. The classification outcomes can then be utilized for examination. To activate suitable response mechanisms. The efficacy of DDoS attack detection through feature selection with SVM or KNN relies on the quality of chosen features training data and machine learning algorithms employed. Continuous refinement and optimization of feature selection procedures along with tuning parameters for SVM and KNN models can enhance accuracy and performance of the detection system.

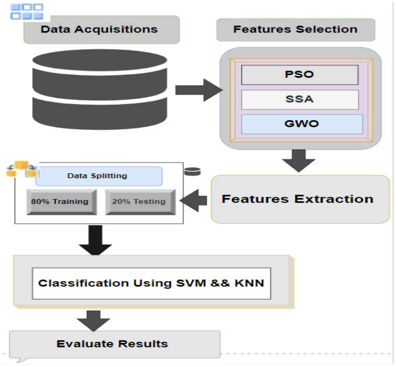


Fig. 2. Proposed Methodology Framework

3.1 Data Acquisition

When it comes to protecting against evolving network threats, Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) are tools. However, the effectiveness of anomaly based intrusion detection methods is limited due to the lack of testing and validation datasets. An analysis of eleven datasets dating back to 1998 reveals that many are outdated and unreliable, lacking diversity and failing to cover a range of known attacks or anonymize payload data. In contrast the CICIDS2017 dataset stands out for its inclusion of traffic data and realistic attack scenarios that closely resemble network activities. It provides labeled flows with timestamps, source/destination IPs, ports, protocols and types of attacks stored in CSV files. Additionally, this dataset focuses on generating background traffic using the B Profile system to mimic human interaction patterns. It examines the behaviors of 25 users across protocols while simulating attacks such as Brute Force FTP and SSH attempts, Denial of Service (DoS) attacks, Web Attacks, infiltration efforts and Botnet operations. The dataset meets eleven criteria for creating a dataset that includes network configurations and diverse sources of traffic data. During attack simulations network traffic data, like memory dumps and system call information were gathered from compromised machines. Over 80 unique network flow features were gathered through the utilization of the CICFlow Meter tool. Table 2 represents the different types of datasets.

Table 2
Intrusion Detection Datasets

Datasets	Year	Information
KDD 99 CUP	1999	41 Features represent the legitimate and attack traffic
CAID 07	2007	Containing the flooding traffic of SYN, ICMP,HTTP protocols
CAID 08	2008	Legitimate and attack traced monitored of Chicago and san Jose
NSL-KDD	2009	Refined version of KDD99dataset after removal of duplicate records
ISCX	2012	Traffic from real world physical test environment
UNSW-NB15	2015	49 features covering 9 types of attacks
CICIDS2(our dataset) 017	2017	78 features with normal traffic and attacks

3.2 Features Selection

Effective feature selection is essential in detecting DDoS attacks as it plays a role in pinpointing valuable characteristics within network traffic data. Its primary functions involve streamlining data, picking out distinguishing features minimizing interference enhancing comprehensibility and insights well as boosting effectiveness and scalability. By simplifying the models' intricacies and concentrating on attributes, feature selection enhances the precision and efficiency of the detection system. It also contributes to comprehending DDoS attack trends and behaviors facilitating improved analysis and quicker responses, in real time situations.

3.2.1 Particle Swarm Optimization Algorithm

The Particle Swarm Optimization (PSO) technique is used in selecting features, for detecting DDoS attacks to enhance the process and pinpoint the set of features that can boost detection accuracy. PSO involves a group of particles that navigate through the search space effectively representing feature subsets. Each particle's effectiveness is assessed based on how its feature subset performs, usually evaluated using metrics like classification accuracy. PSO represents a particle's position as a string and adjusts feature selections during optimization. By updating velocities PSO steers the search towards feature combinations. As the process unfolds particles move closer to a subset that maximizes performance revealing the selected features. PSO. Fine tunes feature selection, by exploring feature combinations and improving detection system accuracy and efficacy. The Particle Swarm Optimization (PSO) technique typically involves the following steps:

- Initialization; Begin by setting up a group of particles each representing a solution, within the search space. Assign positions to these particles within the specified search area. Set their velocities to either zero or random values.
- Fitness Assessment; Assess the fitness level of each particle by applying a fitness function to its position. This function measures how well the particle solution performs in terms of quality.
- Updating Personal Best; Adjust the position (pbest) for each particle. If the current position shows fitness than its best update pbest with the current location.
- Updating Global Best; Identify the position (gbest) from all particles indicating the position with the highest fitness value across the entire swarm.
- Adjusting Velocities; Modify particle velocities based on their velocities, personal best positions and global best position. These adjustments help particles move towards areas in the search space.
- Position Updates; Update particle positions by incorporating their velocities into their locations leading to potential solutions, within the search area.
- Termination Check: Verify if termination conditions have been met.
- Once the process can stop when it reaches a limit of iterations reaches a level of performance or meets any other predefined stopping rule.
- Result: when the stopping condition is satisfied reveal the solution discovered which indicates the overall position. This outcome signifies the nearly optimal resolution to the issue at hand.

Algorithm 1

Pseudocode of standard particle swarm optimization.

```
Initialize population
for t = 1 : maximum generation
    for i=1 : population size
        If $f(x_{i,d}(t)) < f(p_i(t))$ then $p_i = x_{i,d}(t)$
            $f(p_g(t)) = min(f(p_t(t)))$
        End
    for d =1 : dimension
        $v_{t,d}(t+1) = wv_{t,d}(t) + c_1 r_1 (p_t - x_{t,d}(t)) + c_2 r_2 (p_g - x_{t,d}(t))$
        $x_{t,d}(t+1) = x_{t,d}(t) + v_{t,d}(t+1)$
        if $v_{t,d}(t+1) > v_{max}$ then $v_{t,d}(t+1) = v_{max}$
        else if if $v_{t,d}(t+1) < v_{max}$ then $v_{t,d}(t+1) = v_{min}$
        end
        if $x_{t,d}(t+1) > x_{max}$ then $x_{t,d}(t+1) = x_{max}$
        else if if $x_{t,d}(t+1) < v_{min}$ then $x_{t,d}(t+1) = x_{min}$
            end
          end
        end
    end
```

3.2.2 Salp Swarm optimization algorithm

The Salp Swarm Optimization (SSO) technique is designed to enhance the detection of DDoS attacks by optimizing feature selection and parameter adjustments. SSO works by exploring sets of features to identify those that can effectively differentiate between network traffic and malicious attacks. It utilizes the movement of salps within the swarm to refine the search for feature subsets thereby enhancing detection precision. Moreover, SSO fine tunes parameters like thresholds and weights through movements to maximize accuracy. Minimize false alarms. By leveraging swarm intelligence, where salps interact and collectively progress towards solutions SSO aids, in identifying feature combinations and parameter setups for robust DDoS attack identification. Its ability to efficiently navigate search spaces facilitates convergence towards effective solutions ultimately boosting detection accuracy and speed. The Salp Swarm Algorithm (SSA) typically goes through these stages;

- Start; determine the size of the group (N) and the maximum number of rounds (max_iterations). Randomly. Set velocities for salps within the search area. Create a fitness function to assess each salps performance.
- Assessing Fitness; evaluate each salps fitness by applying the fitness function to its location. The fitness function gauges how well a salps solution performs.
- Updating Best Position; revise each salps position (pbest). If a current position has fitness than its best update pbest to reflect the current location.
- Determining Global Best Position; Identify the position (gbest) from all salps representing the location with the highest fitness value across the entire population.
- Adjusting Velocities; modify salps velocities based on their speeds, pbest positions and gbest position. These adjustments guide salps toward areas in the search space.
- Updating Positions; Shift salps positions by adding their velocities to their locations generating potential solutions, in the search space.
- Boundary Management; In case a salp strays, beyond the boundaries of the search space employ strategies to bring it back within the valid range.
- Conclusion Check; Verify if the end condition has been satisfied. This could be reaching the number of iterations. Attaining a satisfactory fitness level. Once the end condition is achieved, present the solution discovered which corresponds to the position. This solution signifies the close to optimal answer to the issue, at hand.

Algorithm 2
Pseudocode of the SSA algorithm.

```
Initialize the Salp population $x_i$ $(I = 1,2,\ldots\ldots,n)$ considering $u_b$, and $I_b$,
While (end condition is not satisfied)
    Calculate the fitness of each search agent (salp)
     $F$= the best search agent
      Update $c_1$ by equation (2)
        For each Salp ($x_i$)
            1f ($i==1$ )
                    Update the position of the leading Salp by equation
        Else
                    Update the position of the follower Salp by equation (4)
        End
   End
    Amend the Salps based on the upper and lower bounds of variabłes
End
Return $F$
```

3.2.3 Gray Wolf Optimization Algorithm

The Gray Wolf Optimization (GWO) method proves its effectiveness in identifying DDoS attacks by tuning the parameters and setups of the detection system. GWO excels at refining factors like thresholds and weights to boost the efficiency of the detection process. Drawing inspiration from the structure and hunting instincts of wolves, GWO adjusts the positions of search agents (wolves) to move towards improved parameter configurations. Moreover, GWO can facilitate feature selection by delving into feature subsets within the search space and assessing their ability to differentiate between traffic and attack patterns. This approach adeptly navigates search spaces leading to convergence towards optimal solutions for precise and timely detection. Furthermore, GWOs capability to address multidimensional and multimodal optimization tasks positions it as a choice for enhancing detection performance through optimization of multiple parameters or features. The Gray Wolf Optimization (GWO) algorithm typically goes through these steps:

- Start; Determine the size of the population (N) and the maximum number of iterations (max_iterations). Randomly position the search agents (wolves) within the search space. Define a fitness function to assess each wolfs performance.
- Fitness Assessment; Evaluate each wolfs fitness by applying the fitness function to its location. This function gauges how well the wolfs solution performs
- Adjust Alpha, Beta and Delta Positions; Identify the three wolves with the fitness, in the population.
- Update Omega Positions;
- Update the locations of the remaining wolves (excluding alpha, beta or delta) using this formula;

$$= (alpha\_position + beta\_position + delta\_position) / 3$$

- Dealing with Boundaries; If a wolf strays, beyond the specified search space limits apply techniques to bring it back within the acceptable range.
- Balancing Exploration and Exploitation; Adjust the wolves' positions to strike a balance between exploring areas and exploiting ones. The alpha, beta and delta wolves focus on exploitation by moving towards regions while the omega wolves explore parts of the search space by moving towards central positions.
- Ending Criteria; Verify if the ending criteria have been met. This can involve reaching the number of iterations or achieving a level of fitness. Upon meeting the ending criteria reveal the solution discovered which corresponds to where the alpha wolf's positioned. This solution signifies nearly optimal resolution to the issue, at hand.

Algorithm 3

Pseudocode of the Grey Wolf Optimization algorithm

```
Input: Problem Size, Population Size
Output: Pg_ best
Star
    Initialization of the population of grey wolves Xi (i= 1 ,2, ... n)
    Initialization of a, A, and C
    Calculation of the fitness values of search agents and grading of agents.
    (Xα= the best solution in the search agent, Xβ= the second best solution
      in the search agent, and X the third best solution in the search agent)
    t = 0
    While ( t < Maximum number of iterations)
     For each search agent
         Updating the position of the current search agent by Equation
     End for
    Updating of a, A, and C
    Calculation of the fitness values of all search agents and grading them
    Updating the positions of Xα, Xβ, and Xδ̄
    t = t +1
End while
End
```

Table 2

Compare between the Proposed Features Selection Algorithms

Algorithm	Advantages	Disadvantages	Limitations
Gray Wolf Optimization (GWO)	-It adeptly navigates intricate search scenarios -Conducts both focused searches - Proves to be beneficial, for optimizing across multiple modes and dimensions	- Sometimes it's possible to get stuck in optima during the optimization process. -This can happen when the initial group of solutions is not well varied leading to a convergence, towards the solution.	- fine-tuning of parameters is needed - slow convergence
Particle Swarm Optimization(PSO)	-It effectively navigates through search areas -Conducts both specific searches at the same time - Easy to implement with a straightforward concept Suitable, for optimizing across multiple modes and dimensions	- Can get trapped in local optima - Sensitive to the initial population - slow convergence - fine-tuning of parameters is needed	- Requires fine-tuning of parameters - May have slow convergence
Salp Swarm Algorithm (SSA)	-It efficiently navigates through search areas -Conducts both local searches, at the same time -Suitable, for optimizing in various modes and dimensions	- Can get trapped in local optima - Sensitive to the initial population - slow convergence	- fine-tuning of parameters is needed - slow convergence

Table 3

The proposed Features Selection Parameters Setting

Algorithm	Parameters	Default Values
Gray Wolf Optimization (GWO)	Population Size (N)	10-50
	Maximum Number of Iterations (max_iterations)	100-1000
	Coefficient (A)	2
	Search Space Boundaries	Defined by the problem
Salp Swarm Algorithm (SSA)	Population Size (N)	10-50
	Maximum Number of Iterations (max_iterations)	100-1000
	Step Size (c)	0.1-0.9
	Search Space Boundaries	Defined by the problem
Particle Swarm Optimization (PSO)	Population Size (N)	10-50
	Maximum Number of Iterations	100-1000
	Inertia Weight (w)	0.4-0.9
	Cognitive Parameter (c1)	1-2
	Social Parameter (c2)	1-2
	Search Space Boundaries	Defined by the problem

3.2.4 Features Classification Using ML (SVM&&KNN)

SVM and KNN stand out as choices when it comes to machine learning algorithms for classifying data. SVM focuses on determining the hyperplane to classes by utilizing support vectors located close to the decision boundary. It can work with both linear data using kernel functions such as linear, polynomial and RBF. SVM proves efficient in handling data, managing outliers effectively and showcasing good generalization performance. However, it may become computationally demanding when dealing with datasets. On the other hand KNN classifies data by looking at the similarity of features. It examines the k neighbors within the training dataset. Assigns a class label based on majority voting. KNN is known for its simplicity, flexibility in not assuming data distribution patterns and adaptability to changes over time. While it performs well with to sized datasets it can be influenced by factors like the choice of k value and high dimensional feature spaces. The decision between using SVM and KNN hinges on factors such as characteristics, complexity of the problem, at hand available computational resources and interpretability of results. Through experimentation and comparing performance metrics one can determine which algorithm suits best for a given classification task.

Table 4
A comparison between SVM and KNN machine learning algorithms.

	SVM (Support Vector Machines)	KNN (k-Nearest Neighbors)
Advantages	-High efficiency for dimensional data -High efficiency in handling nonlinear decision boundaries - High efficiency in handling outliers - Good generalization performance	-It considered simple Algorithm - Doesn't assume distributed data High efficiency in adapting to any changes in the data -High efficiency in handling small-medium size datasets
Disadvantages	-Expensive for large dataset - Can suffer from the curse of dimensionality in high-dimensional feature spaces	-sensitive to the value of K
Limitations	-Careful selection is needed for kernel and parameters -sensitive to overlapping -sensitive to noise	- Lack of interpretability and difficulty in explaining predictions -High training time for large dataset

4. Analysis and Results

Assessment tools are used to measure the performance of machine learning models or algorithms. In classification tasks used evaluation metrics include; Accuracy. This measure assesses how accurate a classifier's predictions are, across fields. It computes the ratio of predicted instances (including positives and true negatives) to the instances considered. While accuracy provides insight into a model's performance it can be misleading when there is a distribution of classes in the dataset. Sensitivity (also known as recall or true positive rate). This metric indicates the percentage of instances correctly identified by the classifier. It is calculated by dividing the number of positives by the sum of positives and false negatives. Sensitivity is particularly valuable when minimizing negatives is essential such as, in detecting all cases of a disease. Specifically, this metric measures how well negative instances are identified by the classifier. It determines the percentage of negatives out of the sum of negatives and false positives. Specificity becomes crucial when accurately identifying instances like distinguishing individuals without a disease. Precision (or positive predictive value); Precision evaluates how many predicted instances are actually positive. When calculating precision, it involves determining the ratio of positives to the sum of positives and false positives. Understanding Precision offers insights into how the classifier can minimize positives in situations where the cost of inaccuracies is significant. The F measure also referred to as the F1 score combines Precision and Recall forming a metric that balances both aspects of performance. This metric calculates the average of Precision and Recall offering a measure to evaluate the effectiveness of the classifier. The F measure is advantageous in scenarios with class distributions. When there is a necessity to reduce both false positives and false negatives simultaneously. The formulas, for each of our metrics are as follows:

$$Accuracy = (TP + TN) / (TP + TN + FP + FN)$$

$$Sensitivity = (TP) / (TP + FN)$$

$$Specificity = (TN) / (TN + FP)$$

$$Precision = TP / (TP + FP)$$

$$F1\ score = 2 \times (Precision \times Recall) / (Precision + Recall)$$

In detecting DDoS attacks TP stands for identified attacks TN stands for classified non attacks FP represents non attacks mistakenly labeled as attacks and FN signifies missed detections of real attacks. By examining TP, TN, FP and FN we can assess the systems performance. The goal is to reduce positives and false negatives while increasing positives and true negatives. This requires choosing algorithms tuning detection thresholds and consistently enhancing the system through performance assessment and feedback.

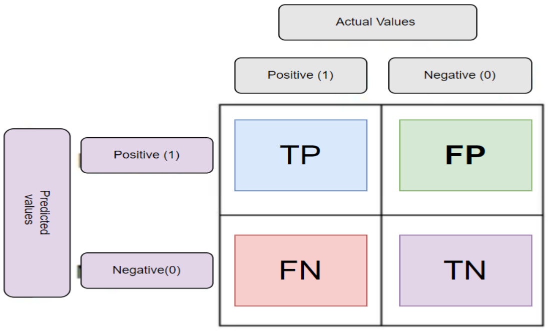


Fig. 3. Confusion Matrix

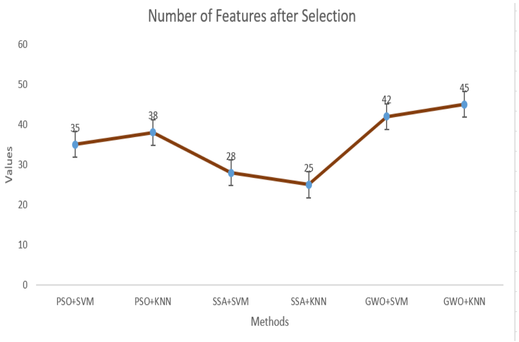


Fig. 4. Number of features after the selection

In Fig. 4, PSO that combined with SVM algorithm chooses 35 features through the feature selection process. For PSO combined with KNN it also selects 38 features after the selection process. The algorithm that merges SSA with SVM picks 28 features in total. Similarly, when combining SSA with KNN it ends up selecting 25 features. GWO combined with SVM opts for 42 features post selection. In comparison GWO paired with KNN chooses 45 features after the feature selection phase. Upon analyzing the number of selected features we notice some variations across the algorithms. PSO+SVM and PSO+KNN have a count of selected characteristics while SSA+SVM and SSA+KNN exhibit selected attributes. Notably GWO+SVM and GWO+KNN stand out for having several chosen features.

Table 5

Evaluation results of the machine learning algorithms.

Algorithm	Accuracy	Sensitivity	Specificity	Precision	Recall	F1-score
PSO+SVM	1.0	1.0	1.0	0.99	0.999	0.999
PSO+KNN	0.999	0.999	0.999	0.999	0.999	0.999
SSA+SVM	0.999	0.999	0.999	0.999	0.999	0.999
SSA+KNN	0.978	0.994	0.991	0.968	0.991	0.98
GWO+SVM	0.998	0.998	0.998	0.999	0.998	0.998
GWO+KNN	0.999	0.999	0.999	0.999	0.999	0.999

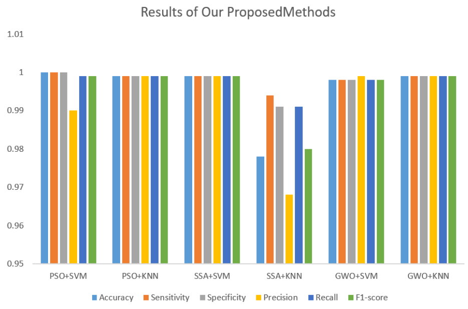


Fig. 5. Proposed Methods Results

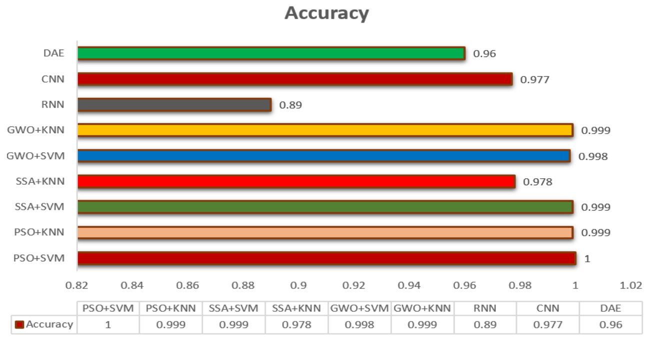
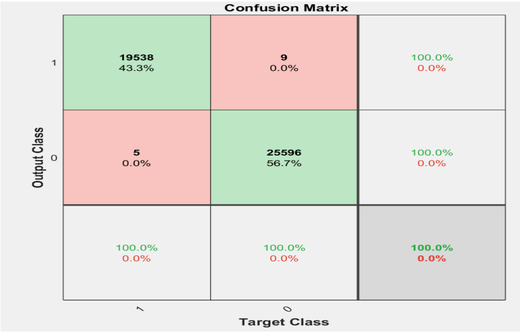


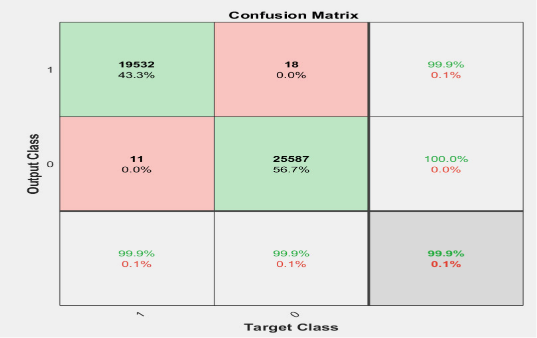
Fig. 6. Comparing our Results with Previous Results

The analysis of the algorithms reveals that they excel in categorizing data. Combinations, like PSO+SVM, PSO+KNN, SSA+SVM, GWO+SVM and GWO+KNN showcase accuracy, sensitivity, specificity, precision, recall and F1 scores. These algorithms consistently display performance across measures showcasing their proficiency in classification tasks. SSA+KNN shows performance when compared to the other combinations especially in terms of precision and recall. Nevertheless, it still exhibits sensitivity and specificity levels indicating its competence, in identifying positive and negative instances. In Figs. 10-15, algorithms can be divided into two groups based on their accuracy levels. The first group, which includes PSO+SVM, PSO+KNN, SSA+SVM, GWO+SVM and GWO+KNN demonstrates accuracy, in classifying instances with scores that're near perfection. These algorithms can classify the majority of instances in the dataset showcasing their efficiency in detecting DDoS attacks. On the other hand the second group comprises SSA+KNN, RNN, CNN and DAE algorithms that show lower accuracy scores compared to the first group. While these algorithms still perform well, they have a margin of error in classification tasks. Further examination using performance metrics would be valuable in assessing their effectiveness, in detecting DDoS attacks.



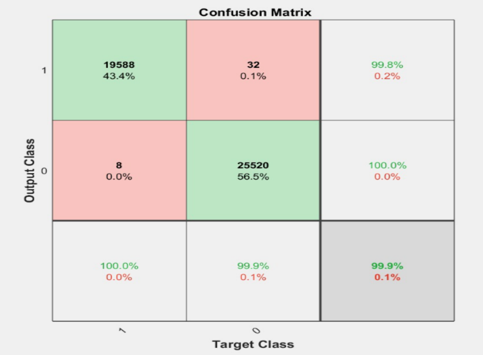
Output Class	1	0	
1	19538 43.3%	9 0.0%	100.0% 0.0%
0	5 0.0%	25596 56.7%	100.0% 0.0%
	100.0% 0.0%	100.0% 0.0%	100.0% 0.0%

Fig. 7. Evaluation results for the PSO+KNN algorithms



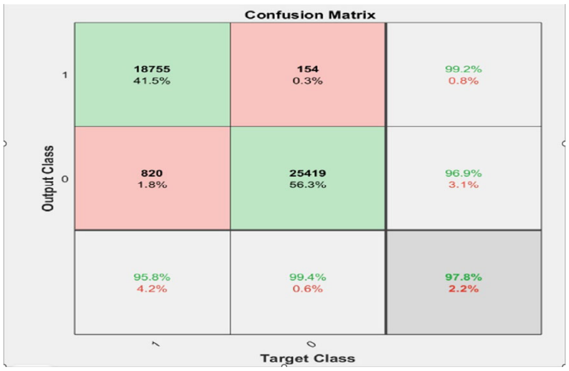
Output Class	1	0	
1	19532 43.3%	18 0.0%	99.9% 0.1%
0	11 0.0%	25587 56.7%	100.0% 0.0%
	99.9% 0.1%	99.9% 0.1%	99.9% 0.1%

Fig. 8. Evaluation results for the PSO+SVM algorithms



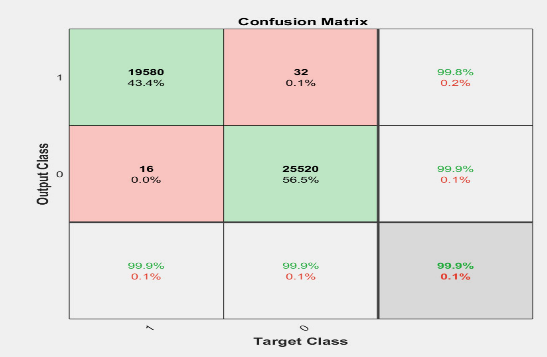
Output Class	1	0	
1	19588 43.4%	32 0.1%	99.8% 0.2%
0	8 0.0%	25520 56.5%	100.0% 0.0%
	100.0% 0.0%	99.9% 0.1%	99.9% 0.1%

Fig. 9. Evaluation results for the SSA+SVM algorithms



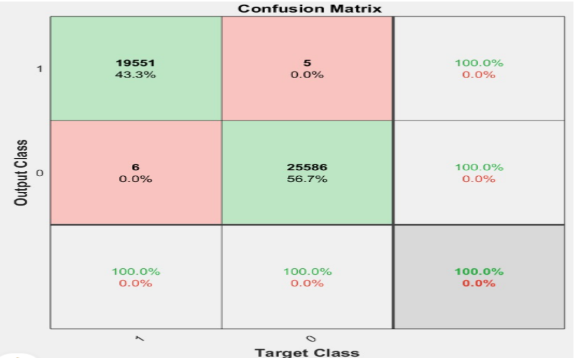
Output Class	1	0	
1	18755 41.5%	154 0.3%	99.2% 0.8%
0	820 1.8%	25419 56.3%	96.9% 3.1%
	95.8% 4.2%	99.4% 0.6%	97.8% 2.2%

Fig. 10. Evaluation results for the SSA+KNN algorithms



Output Class	1	0	
1	19580 43.4%	32 0.1%	99.8% 0.2%
0	16 0.0%	25520 56.5%	99.9% 0.1%
	99.9% 0.1%	99.9% 0.1%	99.9% 0.1%

Fig. 11. Evaluation results for the GWO+SVM algorithms



Output Class	1	0	
1	19551 43.3%	5 0.0%	100.0% 0.0%
0	6 0.0%	25586 56.7%	100.0% 0.0%
	100.0% 0.0%	100.0% 0.0%	100.0% 0.0%

Fig. 12. Evaluation results for the GWO+KNN algorithms

5. Conclusions

Detecting DDoS attacks is essential for safeguarding infrastructure upholding service availability and guaranteeing the security of online systems and services. To achieve this objective, in our work, we start by gathering information, which includes network activity and system records, for operations as well as instances of DDoS attacks. Then we identify characteristics of the data collected such as patterns in network traffic, packet details, IP addresses, types of protocols used and more. Next, we utilize algorithms for feature selection such as Salp Swarm Algorithm (SSA), Gray Wolf Algorithm (GWA), Particle Swarm Algorithm (PSO) to pinpoint the features that can distinguish between normal activities and DDoS attack patterns. After that, we divide the processed dataset into sections for training and testing purposes in order to develop and assess the machine learning models such as SVM (support vector machine), and KNN (K-nearest neighbor). Then we develop a classification model using machine learning techniques like decision trees, forests, support vector machines (SVM), logistic regression models or neural networks. Finally, we assess the effectiveness of models through metrics such as accuracy rates, precision levels, recall rates, and F1 scores. The results show that the proposed models achieve high results (99.9%).

Acknowledgment

This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia (Grant No. KFU241073).

References

Ahmad, R., Wazirali, R., Bsoul, Q., Abu-Ain, T., & Abu-Ain, W. (2021). Feature-selection and mutual-clustering approaches to improve DoS detection and maintain WSNs' lifetime. *Sensors*, *21*(14), 4821.

Alahmadi, A. A., Aljabri, M., Alhaidari, F., Alharthi, D. J., Rayani, G. E., Marghalani, L. A., ... & Bajandouh, S. A. (2023). DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions. *Electronics*, *12*(14), 3103.

Aljuhani, A. (2021). Machine learning approaches for combating distributed denial of service attacks in modern networking environments. *IEEE Access*, *9*, 42236-42264.

Alzahrani, R. J., & Alzahrani, A. (2021). Security analysis of DDoS attacks using machine learning algorithms in networks traffic. *Electronics*, *10*(23), 2919.

Anirudh, M., Thileeban, S. A., & Nallathambi, D. J. (2017, January). Use of honeypots for mitigating DoS attacks targeted on IoT networks. In *2017 International conference on computer, communication and signal processing (ICCCSP)* (pp. 1-4). IEEE.

Fauzi, M. A., Hanuranto, A. T., & Setianingsih, C. (2020, October). Intrusion detection system using genetic algorithm and K-NN algorithm on dos attack. In *2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS)* (pp. 1-6). IEEE.

Halim, Z., Yousaf, M. N., Waqas, M., Sulaiman, M., Abbas, G., Hussain, M., ... & Hanif, M. (2021). An effective genetic algorithm-based feature selection method for intrusion detection systems. *Computers & Security*, *110*, 102448.

Iftikhar, S., Al-Madani, D., Abdullah, S., Saeed, A., & Fatima, K. (2023). A supervised feature selection method for malicious intrusions detection in IoT based on genetic algorithm. *International Journal of Computer Science & Network Security*, *23*(3), 49-56.

Lee, S. H., Shiue, Y. L., Cheng, C. H., Li, Y. H., & Huang, Y. F. (2022). Detection and prevention of DDoS attacks on the IoT. *Applied Sciences*, *12*(23), 12407.

Liu, X., & Du, Y. (2023). Towards effective feature selection for iot botnet attack detection using a genetic algorithm. *Electronics*, *12*(5), 1260.

Marvi, M., Arfeen, A., & Uddin, R. (2021). A generalized machine learning-based model for the detection of DDoS attacks. *International Journal of Network Management*, *31*(6), e2152.

Mohmand, M. I., Hussain, H., Khan, A. A., Ullah, U., Zakarya, M., Ahmed, A., ... & Haleem, M. (2022). A machine learning-based classification and prediction technique for DDoS attacks. *IEEE Access*, *10*, 21443-21454.

Mohsin, A. H., Zaidan, A. A., Zaidan, B. B., Mohammed, K. I., Albahri, O. S., Albahri, A. S., & Alsalem, M. A. (2021). PSO–Blockchain-based image steganography: towards a new method to secure updating and sharing COVID-19 data in decentralised hospitals intelligence architecture. *Multimedia tools and applications*, *80*, 14137-14161.

Norouzi, M., Gürkaş-Aydın, Z., Turna, Ö. C., Yağci, M. Y., Aydin, M. A., & Souri, A. (2023). A Hybrid Genetic Algorithm-Based Random Forest Model for Intrusion Detection Approach in Internet of Medical Things. *Applied Sciences*, *13*(20), 11145.

Onah, J. O., Abdullahi, M., Hassan, I. H., & Al-Ghusham, A. (2021). Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment. *Machine Learning with applications*, *6*, 100156.

Ray, S., Mishra, K. N., & Dutta, S. (2022). Detection and prevention of DDoS attacks on M-healthcare sensitive data: a novel approach. *International Journal of Information Technology*, *14*(3), 1333-1341.

Seifousadati, A., Ghasemshirazi, S., & Fathian, M. (2021). A Machine Learning approach for DDoS detection on IoT devices. *arXiv preprint arXiv:2110.14911*.

Trab, S., Bajic, E., Zouinkhi, A., Abdelkrim, M. N., & Chekir, H. (2018). RFID IoT-enabled warehouse for safety management using product class-based storage and potential fields methods. *International Journal of Embedded Systems*, *10*(1), 71-88.

Ullah, S., Mahmood, Z., Ali, N., Ahmad, T., & Buriro, A. (2023). Machine learning-based dynamic attribute selection technique for ddos attack classification in iot networks. *Computers*, *12*(6), 115.

Zhao, J., Xu, M., Chen, Y., & Xu, G. (2023). A DNN architecture generation method for DDoS detection via genetic alogrithm. *Future Internet*, *15*(4), 122.