Received: February 11, 2024. Revised: April 25, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Novel Approach for Detecting Unauthorized Requests in Software-Defined Networks Using Hybrid Particle Swarm and Automated Grey Wolf Optimizer Algorithm

Aminata Dembele[1]* **Elijah Mwangi[2]** **Kennedy K. Ronoh[3]** **Edwin O. Ataro[4]**

[1]Departement of Electrical Engineering Pan African University Institute for Basic Sciences, Technology and Innovation (PAUSTI), Nairobi, Kenya
[2]Department of Electrical and Information Engineering, University of Nairobi, Kenya
[3]School of Computing and Engineering Sciences, Strathmore University, Nairobi, Kenya
[4]Department of Electrical and Information Engineering, Technical University, Nairobi, Kenya
* Corresponding author's Email: aminata.dembele@students.jkuat.ac.ke

Abstract: Software Defined Networking (SDN) is a technology that consolidates network management through a unified controller. However, it is vulnerable to attacks like distributed denial of service (DDoS) due to reliance on a single control plane. In order to address this, a new approach called Hybrid Particle Swarm Optimization (PSO) and Automated Modified Grey Wolf Optimizer Algorithm (AMGWOA) is proposed in this paper. We enhance the efficiency of detecting and preventing malicious requests in SDN frameworks by combining PSO and AMGWOA. Our PSOAMGWO method outperforms conventional grey wolf optimizer and particle swarm optimization techniques, achieving a remarkable 100% accuracy in detecting harmful requests within 0.5 seconds under the same sample size of traffic requests. This approach not only reduces detection time but also minimizes storage and computing resource utilization.

Keywords: SDN, Security, DDoS attacks, Metaheuristics algorithms, Particle swarm, Grey wolf optimizer.

1. Introduction

The extensive distribution of mobile devices, along with the introduction of advanced technologies like the Internet of Things and cloud computing, has led to an unparalleled rise in the number of networked devices on the Internet. Consequently, there has been a substantial rise in the expansion and intricacy of large-scale networks, which presents various challenges. The current network technologies and infrastructure lack the ability to efficiently handle large and complex networks in a flexible and easily controllable manner [1].

Software Defined working (SDN) facilitates "programmable networking", a pioneering method that decouples control decisions from routing hardware. This separation enables the delivery of flexible and dynamic services within wireless communication networks [2].

Within the domain concerning Software-Defined Networking (SDN), network intelligence is centralized within a software-based controller, identified as the control plane. This configuration empowers network devices, particularly OpenFlow switches, to operate as simple packet forwarding entities, known as the data plane. The OpenFlow protocol [3] allows for programming these devices using an open interface.

While the concept of separating the control plane and data plane in SDN technology offers notable benefits such as enhanced flexibility, cost-effectiveness, and efficient administration, it also introduces new vulnerabilities [4]. The SDN controller functions as the central intelligence of the network. If the controller is compromised, the entire network is exposed to risk. Multiple recent research

DOI: 10.22266/ijies2024.0831.12

Received: February 11, 2024. Revised: April 25, 2024.

studies have indicated that the SDN paradigm is susceptible to DDoS attacks perpetrated by malicious users [5] [6] [7]. These attacks are characterized by the controller manipulation by numerous puppet hosts to launch an assault on the target system. This leads to the exhaustion of the resources within the targeted system and presents a risk to its continuous functioning.

Throughout a DDoS assault on an SDN network, the switch produces a continuous flow of packet messages that the controller needs to handle. These events exert strain on the resources of the controller, leading to an expansion of switch routing tables, and provide a possible risk to the security of encrypted connections between controllers and switches. This situation has the potential to cause significant disruption to the entire SDN.

If a DDoS attack occurs in the SDN, the communication channel is quickly limited, causing a depletion of controller resources and a substantial deterioration in service quality. Current techniques for identifying DDoS attacks in SDN networks face challenges in extracting features effectively, leading to low detection accuracy and elevated false negative rates. Consequently, it becomes essential to employ advanced Optimization techniques, such as metaheuristic algorithms, to swiftly and accurately detect attacks.

Recently, two optimization methods, namely Grey Wolf Optimizer (GWO) and Particle Swarm Optimization (PSO), have been developed to tackle issues of network security and computing [8]. These algorithms all have the same goal in consideration: to identify the best solutions and to improve convergence performance. PSO [9] is an optimizer that utilizes a swarm of particles to explore a limited search space and identify the optimal solutions to a given problem. Particle swarm optimization is an optimization approach that is suited for dimensional optimization and has strong comprehensive search capabilities. Due to its ability to effectively explore global optimum solutions, rapid convergence, and ease of implementation, PSO is frequently integrated into hybrid methodologies [10].

GWO is a bio-inspired optimization method that emulates the social structure and hunting patterns of grey wolves to progressively enhance solutions to optimization issues [11]. The natural hunting and dominant behavior of grey wolves are modeled by this algorithm [11].The algorithm's underlying principles and has received positive feedback from the optimization community.

Our earlier study in [12], we proposed Automated Modified Grey Wolf Algorithm (AMGWOA) for the automatic detection of DDoS attacks in SDN networks. AMGWOA is an algorithm designed for the purpose of detection of DDoS attacks in SDN environments. The algorithm is tailored to improve the efficiency of DDoS detection mechanisms within an SDN framework by leveraging an automated and modified version of the Grey Wolf Optimizer. This study aims to enhance AMGWOA approach by proposing a novel approach that combines PSO with AMGWOA. Results show considerable improvement in the rapid convergence of attack detection by the new PSOAMGWO algorithm.

In order to improve detection accuracy and minimize system operating costs, it is necessary to react quickly within a limited timeframe. This study therefore presents a new approach that combines particle swarm optimization (PSO) and grey wolf optimization (GWO) to automatically prevent DDoS attacks in SDN networks in a significantly short time. This hybrid approach, PSOAMGWO, aims to detect and prevent the most harmful requests by the controller of the SDN. This strategy effectively reduces the danger of encountering a critical DDoS attack, minimizes latency, and ensures uninterrupted access to the controller for legitimate users.

This research presents a novel approach designed to significantly improve the efficacy of attack detection. The methodology integrates PSO with the Automated Modified Grey Wolf Optimization technique (AMGWOA), leveraging the progress achieved in AMGWOA as described in our prior research [12]. Particle Swarm Optimization is frequently incorporated into hybrid methodologies because of its efficacy in exploring the global optimum, swift convergence, and straightforwardness. The extensive utilization of the grey wolf optimization method in addressing optimization problems is supported by its rapid iteration process.

By combining the PSO-GWO algorithms, we leverage the benefits provided by both methods, leading to decreased electrical consumption, as well as overall execution time required for detecting attacks in SDN. This combination minimizes resource consumption and effectively leverages the strengths of each algorithm, therefore reducing their deficiencies [13]. The core concept is to enhance the exploitation capability of PSO while integrating the exploration potential of the GWO, harnessing the strengths of both methods in comparison to conventional metaheuristic algorithms [14]. To the best of our knowledge, the fusion of AMGWOA (Automated Modified Grey Wolf Optimization Algorithm) and PSO (Particle Swarm Optimization) has not been previously employed in an automated fashion for the identification of DDoS (Distributed

Denial of Service) attacks within Software-Defined Networking (SDN) contexts.

This amalgamation presents a novel approach that capitalizes on the unique strengths and adaptability of both algorithms, aiming to enhance the accuracy and efficiency of DDoS attack detection within the SDN framework. The integration of these optimization techniques introduces a fresh perspective, potentially unlocking improved performance and robustness in addressing the challenges associated with identifying and mitigating DDoS attacks in SDN environments.

The structure of our study is delineated as follows: Section 2 provides the contextual information for the research. Section 3 provides a succinct summary of previous research. Section 4 provides a comprehensive description of the design of our proposed algorithm, PSOAMGWO. Section 5 provides a detailed discussion of the experimental approach employed and the resulting conclusions. The final remarks of the paper are summarized in Section 6.

2. Background

This section presents the following: an overview of SDN, DDoS attacks targeting the SDN controller, an overview of the Grey Wolf Optimizer, and particle swarm optimization algorithms.

2.1 Software defined network

Within an SDN network, there is a separation between software and hardware. The control plane, which determines traffic routing, is transitioned to software, but the data plane, which physically forwards traffic, remains in hardware. This allows network managers to apply a standardized interface to program and oversee the whole network, minimizing the need to individually manage each device. A standard SDN architecture comprises three components:

1. Application layer: These applications convey resource requests or provide information about the network as a whole.
2. Control layer: These controllers utilize information from applications to make decisions on how to route data packets.
3. Infrastructure layer (Networking Devices): These devices receive instructions from the controller regarding the optimal path for data movement. The interaction between these layers is enabled through northbound and southbound application programming interfaces (APIs).

Fig. 1 depicts the architectural framework of this novel technology.

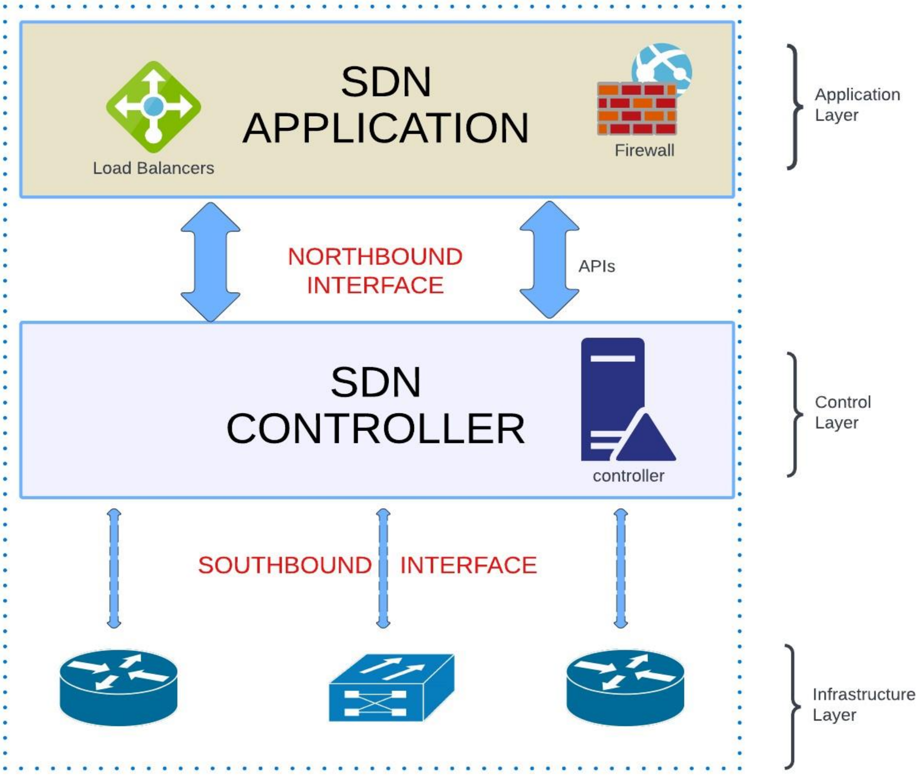

Figure. 1 SDN framework comprising three layers

2.2 DDoS attacks targeting the SDN controller

Controllers face a substantial vulnerability to DDoS threats because of the centralization of the computational capabilities of the SDN within the controller which may lend it to a single point of failure.

The main issue arises from a significant quantity of manipulated puppet individuals launching assaults on the designated system. This leads to a rapid depletion of the resources of the specific system, ultimately resulting in a decrease in the quality of service or a possible shutdown. Considering the significant importance of SDN, a DDoS attack can trigger a sudden increase in packets from switches that reach the controller. As a result, this puts pressure on the controller resources, jeopardizing the network stability because the SDN controller integrity is affected or if it is unable to meet the switch requests. Simultaneously, the flow table of the switch experiences a substantial surge, and an excess of messages may impede the safe connection between the controller and switches, potentially causing complete paralysis of the entire SDN.

The consequences of a DDoS attack on an SDN network include the possibility of the attack quickly blocking communication channels and rapidly depleting controller resources, leading to a significant degradation in the quality of service. The depicted attack in SDN, as presented in Fig. 2, has the objective of inundating all resources of the targeted host with the intention of disrupting its normal operation.

2.3 Grey wolf optimizer algorithm

The algorithm is derived from the hunting behaviors of grey wolves, involving tactics such as encircling, hunting, and attacking their prey [11].

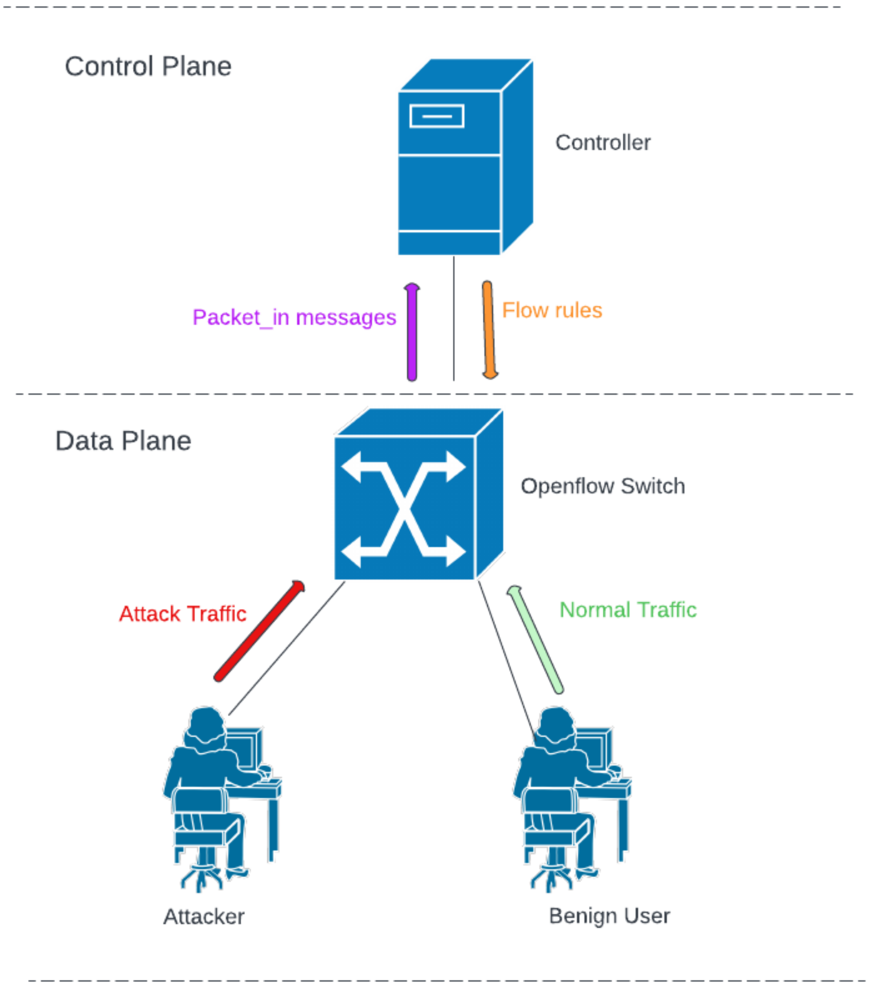

Figure. 2 Illustration of the scenario of DDoS within SDN

Grey wolves exhibit a clear social hierarchy within their packs, with the top-ranking wolves being referred to as alphas. In terms of hierarchy, Alpha is the first leader, Beta is the second, Delta ranked third, and the last position is Omega in the pool. Decision-making within the pack, such as when to rise, where to rest, and when to embark on a hunt, is overseen by the alpha wolves.

The decisions made by the Alpha must be adhered to by the rest of the wolves. The Beta wolf fulfills a supportive function by aiding the Alpha in deciding on actions and assuming leadership responsibilities in case the Alpha becomes unable to do so or passes away. While the Beta complies with the Alpha decisions, it issues orders to wolves of lower rank. Wolves classified as Delta, such as sentinels, scouts, elders, and caregivers, hold intermediate positions. Omegas occupy the lowest tier and are to be consumed last, following the established hierarchy.

The wolves are social creatures that exhibit common behaviours, such as engaging in group hunting. Initially, they follow, chase, and approach their prey. Subsequently, the targeted prey is pursued, encircled, and subjected to harassment until it ceases movement. The final stage of the hunt involves the wolves launching an attack on their prey.

The algorithm known as GWO replicates two social traits observed in wolf packs: social dominance and cooperative hunting. Within the framework, each wolf represents distinct tactics aimed at attaining ideal results. Alpha (α) is the optimal or most favorable response, beta (β) is the subsequent or second-best option, and delta (δ) is the third-ranked choice. The top three choices are the leading options and actively sought in the hunt, while all other alternatives are regarded as omega (ω) solutions. Eq. (1) represents the analytical encircling behavior modeling of the wolves.

$$\vec{Y}(t+1) = \vec{Y}_p(t) + \vec{B} \cdot \vec{E} \tag{1}$$

Where: $\vec{Y}_p$ is the position of the prey, $\vec{Y}$ is the position of the grey wolf, $\vec{E}$ is as stated in Eq. (2), t is the iteration number, $\vec{B}$ and $\vec{D}$ are coefficient vectors as defined in Eqs. (3) and (4).

$$\vec{E} = \mid \vec{D} \cdot \vec{Y}_p(t) - \vec{Y}(t) \tag{2}$$

$$\vec{B} = 2a \cdot \overrightarrow{r_1} - \vec{a} \tag{3}$$

$$\vec{D} = 2.\overrightarrow{r_2} \tag{4}$$

Throughout each iteration, the parameter a drops in a linear manner through the interval 2 to 0. Both r_1 and r_2 are vectors of randomness that fall within the interval [0, 1]. Like grey wolves, alpha, beta, and delta have in-depth knowledge of where their prey is likely to be. After the best spots for the main search agents (alpha, beta, and delta) have been found, the placements of the other wolves are modified appropriately. According to the instructions given in Eq. (5), the placements of the wolves are updated as follows:

$$\vec{Y}(t+1) = \frac{\vec{Y}_1 + \vec{Y}_2 + \vec{Y}_3}{3} \tag{5}$$

Here $\overrightarrow{Y_1}$, $\overrightarrow{Y_2}$, and $\overrightarrow{Y_3}$ are defined in Eqs. (6)-(8).

$$\overrightarrow{Y_1} = \overrightarrow{Y_\alpha} - \overrightarrow{B_1} \cdot \left(\overrightarrow{E_\alpha}\right) \tag{6}$$

$$\overrightarrow{Y_2} = \overrightarrow{Y_\beta} - \overrightarrow{B_2} \cdot \left(\overrightarrow{E_\beta}\right) \tag{7}$$

$$\overrightarrow{Y_3} = \overrightarrow{Y_\delta} - \overrightarrow{B_3} \cdot \left(\overrightarrow{E_\delta}\right) \tag{8}$$

Here $\overrightarrow{Y_\alpha}$, $\overrightarrow{Y_\beta}$, and $\overrightarrow{Y_\delta}$ represent the positions of the top three solutions. Additionally, $\overrightarrow{B_1}$, $\overrightarrow{B_2}$, and $\overrightarrow{B_3}$ are defined in Eqs. (6)-(8) respectively while the vectors $\overrightarrow{E_\alpha}, \overrightarrow{E_\beta}, and\ \overrightarrow{E_\delta}$ are defined in Eqs. (9)-(11).

$$\overrightarrow{E_\alpha} = \left|\overrightarrow{D_1} \cdot \overrightarrow{Y_\alpha} - \vec{Y}\right| \tag{9}$$

$$\overrightarrow{E_\beta} = \left|\overrightarrow{D_2} \cdot \overrightarrow{Y_\beta} - \vec{Y}\right| \tag{10}$$

$$\overrightarrow{E_\delta} = \left|\overrightarrow{D_3} \cdot \overrightarrow{Y_\delta} - \vec{Y}\right| \tag{11}$$

Where $\overrightarrow{D_1}$, $\overrightarrow{D_2}$, and $\overrightarrow{D_3}$ are as defined by Eq. (4). The parameter a, which controls the balance of exploration and exploitation, is updated based on Eq. (12).

$$a = 2 - t\frac{2}{\mathrm{MaxIter}} \tag{12}$$

Where t the number of iterations and M is the maximum number of iterations. The pseudocode for the GWO algorithm is represented by Algorithm 1.

Algorithm 1: Grey Wolf Optimizer

```
Initialize the grey wolf population $Xi$ (i=1,2,…,n)
Initialize $a$, $A$ and C
Compute the fitness of each wolf
Set $\vec{X}_\alpha$ as the best wolf
Set $\vec{X}_\beta$ as the second best wolf.
Set $\vec{X}_\delta$ as the third best wolf.
while (t < $MaxIter$)
    for each wolf do
        Update the current wolf position using Eq. (5)
    end
    Update a, A and C
    Compute the fitness of all search agents
    Update $\vec{X}_\alpha$, $\vec{X}_\beta$and $\vec{X}_\delta$
    $t = t + 1$
end while
return $\vec{X}_\alpha$
```

2.4 Particle swarm optimization algorithm

The fundamental decision-making process of PSO was primarily influenced by the social behaviour observed in animals, including the collective movement of fish in schools and the coordinated flight of birds in flocks [15]. When birds are looking for meals, they exhibit either a behaviour of spreading out or traveling together before coming together at a certain spot where they can get sustenance. As birds navigate between different habitats in pursuit of nutrition, there is consistently a bird with a keen sense of smell, effectively aware of the specific position where food is available, and possessing accurate food resource information. Through continuous transmission of messages, particularly valuable ones throughout the search for food across different positions, the birds ultimately converge at the location at which aliments are available.

This approach is based on the examination of animal conduct to compute global optimization functions or problems, wherein every individual inside the group is referred to as a particle. The PSO methodology entails modifying the geographical location of each member of the collective inside the global search region through the utilization of two mathematical formulas. The subsequent equations are:

$$v_i^{k+1} \& = v_i^k + c_1 r_1 \left(p_i^k - x_i^k\right) + c_2 r_2 \left(g_{\text{best}} - x_i^k\right) \quad (13)$$

$$x_i^{k+1} \& = x_i^k + v_i^{k+1} \quad (14)$$

Where:
v_i^{k+1}: Updated velocity of particle i in the *(k + 1)* iteration
v_i^k : Current velocity of particle *i* in the *k* iteration c_1, c_2 : Cognitive and social acceleration coefficients
r_1, r_2 : Random values between 0 and 1
p_i^k : Best-known position of particle *i* so far (personal best)
x_i^k : Current position of particle i in the k iteration
g_{best} : Best-known position in the entire swarm (global best)

These equations are applied to each particle in the swarm during each iteration of the PSO algorithm. Algorithm 2 presents the pseudocode that demonstrates the PSO algorithm.

Algorithm 2: Particle Swarm Optimization Algorithm

```
Initialize number of particles, $c_1$ , $c_2$ , ω, and Umin,Umax
Initialize particle with random power values that are within allowed range;
Evaluate particles;
while maximum iterations has not been reached do
    foreach particle do
        Calculate fitness value;
        if the fitness value is better than the best fitness value $P_i$ in history then
            Set current value as the new $P_i$
    end
    foreach particle do
        Choose the particle with the best fitness value of all the particles as the $g_{\text{best}}$ ;
        if the current $P_{\text{best}}$ is better than $P_{\text{best}}$ then
            Set current $P_i$ as new $g_{\text{best}}$ ;
    end
    foreach particle do
        Calculate particle velocity;
        Update particle position;
    end
end
While maximum iterations has not been reached;
Set $g_{\text{best}}$ at the final solution PSO
```

3. Existing studies on DDoS detection in SDN

Extensive debates have emerged on the security issues linked to SDN. The literature in [16] indicates that in recent years, numerous researchers have integrated various artificial intelligence algorithms to provide a hybrid strategy for identifying and mitigating SDN DDoS attacks. However, just a limited number of modern Hybrid techniques are addressed in this discussion.

Several hybrid machine learning systems, such as the one discussed in reference [17], have included random forest (RF) and support vector machine (SVM) classification algorithms for efficiently identifying regular and DDoS attacks on traffic across networks. Their approach underwent testing and evaluation using a realistic Software-Defined Networking (SDN) dataset, resulting in a high accuracy rate of 98.8% and a minimal number of false alarms. The disadvantage of that approach lies in its singular emphasis on mitigating severe attacks known as DDoS. Its efficacy stems on its capacity to precisely anticipate such attacks, mostly because of the abundance of heavily modified network traffic.

In [18], the utilization of naıve Bayes (NV), k-nearest neighbours algorithm (K-NN), RF, and SVM, along with decision tree (DT) algorithms, was explored. The assessment, conducted on the NSL-KDD dataset, revealed notable performance, with DT achieving a high accuracy of 99.97%, while SVM exhibited a considerably lower accuracy of 60.19%. It is essential to note that the model proposed in this study underwent testing and training using artificial datasets that do not accurately capture the distinctive features of SDN networks.

The study referenced in [19] utilized machine learning methods, including SVM, DT, K-NN, and ANN, to categorize SDN flow and differentiate between regular traffic and DDoS attacks. The results revealed that among the classification algorithms, DT exhibited the highest level of accuracy rate at 99.75%, while SVM had the lowest accuracy at 81.48%. It is important to highlight that the methodology is specifically tailored to address high-rate DDoS attacks, which are easily identifiable because of the significant volume of network traffic they generate.

In [20], a hybrid method was introduced, by integrating a convolutional neural network (CNN) and a transformer that consists of an encoding and a decoding algorithm, for attack detection. The results of this proposed approach, conducted on the CICDDoS2019 dataset, demonstrated superior performance compared to alternative methods. Despite its success, the approach exhibited elevated frequencies of incorrect positive results, and poor accuracy, highlighting the need for improvement in these areas.

In [21], a set of machine learning algorithms, including KNN, SVM, and RF, together with deep learning techniques such as MLP, CNN, GRU, and LSTM, were employed to achieve a 95% accuracy in detecting DDoS assaults at the application layer.

In [22], a novel hybrid design for the software-defined networking (SDN) controller was introduced combining a one class SVM and an autoencoder for identifying DDoS assaults. The model attained a mean accuracy of 99.35%. However, it imposes superfluous burden and overhead in addition has been trained on a synthetic dataset, which may not accurately reflect the SDN network’s reality.

In [23], the technique has been proposed for detecting and addressing DDoS and port-scanning threats on the SDN application layer using fuzzy logic, Shannon entropy, and LSTM algorithms. The technique was tested in two scenarios using the CICDDoS 2019 dataset. The method showed excellent performance in the first scenario but could potentially overload the SDN controller during DDoS attacks.

A detection method was proposed in [24], which employed a Deep Neural Network (DNN) and SVM to accurately identify anomaly-based DDoS threats. The model was tested using the KDD CUP dataset, yielding a rate of detection of 92.3%. The suggested technique underwent evaluation and training using an artificial dataset that failed to accurately reflect the characteristics of the network’s SDN system.

In conclusion, hybrid-based approaches are predominantly crafted to address high-rate DDoS attacks, leading to heightened precision. Nonetheless, certain methods demonstrate lower precision in identifying or alleviating DDoS attacks within the SDN environment. The objective is to create cost-effective, computationally straightforward solutions that avoid imposing excessive burdens on the network.

Many researchers [25] have applied optimization techniques to address network intrusion problems, and in the proposed method, these techniques will be leveraged to tackle DDoS attacks. Metaheuristics, employed in tasks like facial, gene selection, disease diagnosis, intrusion detection systems, and emotion recognition, have effectively tackled diverse optimization challenges. In comparison to exact search mechanisms, metaheuristics demonstrate outstanding performance. Contrary to complete search algorithms, they do not necessitate exploring the entire space of searches in order to locate the best answer, offering benefits regarding computing complexity and resource efficiency.

4. A hybrid particle swarm-automated modified grey wolf optimizer algorithm (PSOAMGWO) for detecting DDoS attacks in software defined networking (SDN)

The presented method employs a hybrid PSO and GWO for the identification and prevention of malicious requests originating from the SDN controller. Leveraging collective intelligence principles and a GWO model, it discerns anomalous traffic patterns and detects DDoS attacks. The algorithm scrutinizes the input of the network traffic to discern unique characteristics, flagging suspicious traffic and implementing proactive measures to safeguard the network. The adjustments are used to explore the application of PSO in GWO, resulting in the generation of variant strength, which provides an additional benefit for the mode. The Hybrid GWO-PSO algorithm utilizes a mathematical equation to determine the positions of the initial three agents in the space of search. The governance of the investigation and use of the GWO is overseen inside the designated area by the inertia of constant w.

This work demonstrates the utilization of a low-level co-evolutionary combined hybrid to merge the PSO and GWO algorithms. The hybrid is at a lower level as a result of combining functionalities from both categories. The phenomenon is considered co-evolutionary due to the absence of sequential utilization of both variations. Alternatively, they operate simultaneously. By incorporating the exploration capabilities of GWO with the exploitation abilities of PSO, we strengthen the overall performance of the algorithm by using strengths from both GWO and PSO.

The PSOAMGWO algorithm updates the positions of the initial three agents in the space of search using the mathematical Eqs. (15)-(17) specified in algorithm 3. Rather than using traditional mathematical calculations, the controls the balance between exploring and exploiting the search area of the Grey Wolf by employing an inertia constant. The revised set of equations that apply is as follows:

$$\overrightarrow{E_\alpha} = \left|\overrightarrow{D_1} \cdot \overrightarrow{Y_\alpha} - w \times \vec{Y}\right| \tag{15}$$

$$\overrightarrow{E_\beta} = \left|\overrightarrow{D_2} \cdot \overrightarrow{Y_\beta} - w \times \vec{Y}\right| \tag{16}$$

$$\overrightarrow{E_\delta} = \left|\overrightarrow{D_3} \cdot \overrightarrow{Y_\delta} - w \times \vec{Y}\right| \tag{17}$$

The proposed approach for integrating PSO and GWO variations involves the formulation of new equations for velocity and updates as in Eqs. (18)-(19):

$$v_i^{k+1}\& = w \times \left(v_i^k + c_1 r_1\left(p_i^k - x_i^k\right) + c_2 r_2\left(x_2 - x_i^k\right) + c_3 r_3\left(x_3 - x_i^k\right)\right) \tag{18}$$

$$x_i^{k+1}\& = x_i^k + v_i^{k+1} \tag{19}$$

4.1 Design of PSOAMGWO

To develop the proposed policy as part of our proposed hybrid optimization process, we define an objective function, also known as a fitness function. This objective function, denoted as Z in Eq. (20) below, is designed to capture the essence of the optimization task. Specifically, it minimizes the sum of $\mathrm{Re}\, q_i^m$ for all *n* components, where *Req_i* represents a certain characteristic of the system and m is a tunable parameter.

$$min\, Z = \sum_{i=1}^{n} \mathrm{Re}\, q_i^m \tag{20}$$

Subject to*:*

$$\mathrm{Req}\ q^\gamma = \tau, \tau \in [20,50]$$
$$\sigma^t = \mu, \mu \in [0.01,1]$$

The objective function Z is defined as the sum of $\mathrm{Re}\, q_i^m$ for all n components. The term $\mathrm{Re}\, q_i^m$ represents a characteristic of the system for the i-th component raised to the power m. Therefore:

- n is the total number of components in the system that are considered for optimization. The summation runs from i = 1 to n, indicating that we have n components, each with its own characteristic represented by $\mathrm{Re}\, q_i^m$.
- m is a tunable parameter used to control the exponent in the expression $\mathrm{Re}\, q_i^m$. The optimization process involves finding the optimal value for m that minimizes the objective function Z. It is a parameter that can be adjusted to influence how the characteristics of the components (Req_i) contribute to the overall objective.

The variable Req^m represents the specific malicious requests that are being targeted for minimization. Z is the objective function that is associated with two primary constraints:

- Several sets of requests Req^γ are directed towards a common resource (service/application).
- A time frame σ^t has been established with a specific duration measured in seconds, during which requests are received.

4.2 The fundamental components of PSOAMGWO

In order, to solve the given objective function, it is imperative to combine the Hybrid Particle Swarm Optimization Algorithm with the Grey Wolf Optimization Algorithm, while also incorporating a Resource-Constrained management method. The latter classifies the received queries by utilizing the threshold [λ +] as the upper limit, wherein each query is categorized according to the three most optimal solutions of GWO. To achieve the objective function, the locations of the three highest-ranking search agents, referred to as alpha (α), beta (β), and delta (δ), are adjusted inside the search area using the equations outlined in Eq. (18) of the hybrid PSOAMGWO. Therefore, this threshold [λ+] is established by computing the fitness function, which may be used to classify into three separate categories:

- The Alpha class refers to the initial optimal solution linked to the subsequent Req^m that will be eliminated if the condition $\omega \leq \lambda^-$ is met.
- The Beta class signifies the next most optimal answer. If the requirement$\omega \in [\lambda^-, \lambda^+]$ is met, the Req^m will be eliminated.
- The Delta category represents the third most favorable choice. If the rule $\omega \geq \lambda^+$ is achieved, the request will be denied, similar to the previous categories.

The equation $\omega = \frac{Bw}{C}$ defines the variable ω as the ratio of the observed bandwidth Bw to the capacity C of the cable in terms of bits per second. The upper and lower bounds, denoted by λ^+ and λ^- respectively, are initialized based on the constraint σ^t. The time range is separated into three intervals: $[20 - \lambda^-]$, $[\lambda^- - \lambda^+]$ and $[\lambda^+ - 50]$.

4.3 Implementation of PSOAMGWO

Algorithm 3 outlines the procedure for implementing the Hybrid Particle Swarm Automated Modified Grey Wolf Optimizer Algorithm (PSOAMGWO). The PSOAMGWO algorithm aims to minimize the fitness function, which represents the sum of the requests, to identify the most optimal solution.

Upon the receipt of requests, each will be carefully examined to confirm its authenticity and eliminate any possible malicious purpose before being sent out. In other words, if it does not meet predetermined standards of range and duration, then it is going to be promptly obstructed and withheld from reaching the controller.

The novel PSOAMGWO is summarized in the algorithm 3 provided below:

Algorithm 3: Proposed Hybrid PSO Automated Modified Grey Wolf Optimizer (PSOAMGWO)

```
Initialize the GWO population (solution): Yi (Y = 20)
Initialize a, $\vec{B}$, $\vec{D}$, $\lambda^+$, $\lambda^-$, w; // w = 0.5 + rand() / 2 and t = 0
Calculate the fitness of each solution using Eq. (20)
$\vec{Y_i}$      (e.g., i=1...20);
$\vec{Y_\alpha}$ the first malicious request;
$\vec{Y_\beta}$ the second malicious request ;
$\vec{Y_\delta}$ the third malicious request;
While (t < max number of iterations) do
    For each agent do
        If (number of requests = $\tau$ and time window = $\mu$ ) then Update the velocity and the position of the current agent using Eqs. (15)-(17)
        If the sum of $Req^\gamma$ in the Alpha class greater than the sum of $Req^\gamma$ in the other two classes then
        Block the next request
        Else
        Forward the request
    end
        Update a, $\vec{B}$, $\vec{D}$, and w
        Calculate the fitness value of each candidate solution (malicious requests)
        Update $\vec{Y_\alpha}$, $\vec{Y_\beta}$ and $\vec{Y_\delta}$
        t= t+1
End while
Return $\vec{Y_\alpha}$
```

5. Experimental configuration and discussion of results

This section describes the experimental environment we used and presents the results obtained from evaluating the proposed approach.

5.1 Experimental configuration

The simulation was conducted using a Matlab R2020a platform. The research experiments were performed on a laptop, namely the HP Pavilion X360, operating on the operating system Windows 10. The device has 8GB of DDR4 RAM, a 10th-generation Core i7 CPU, and a 512 GB SSD.

For this particular implementation, we have established 35 as the population size. The iteration limit has been set to 500. The values of c_1 and c_2 are both 0.5, while c_3 is set to 0.5. The formula for determining the value of w is 0.5 + rand()/2. The parameters that are set are utilized to assess the efficacy of hybrid and other metaheuristics.

5.2 Results discussion

5.2.1 Comparison of PSOAMGWO with AMGWOA, Particle Swarm(PSO) and conventional GWO

In order to evaluate the effectiveness of our technique, we performed a comparative analysis between AMGWOA [12], PSO [9], and the standard

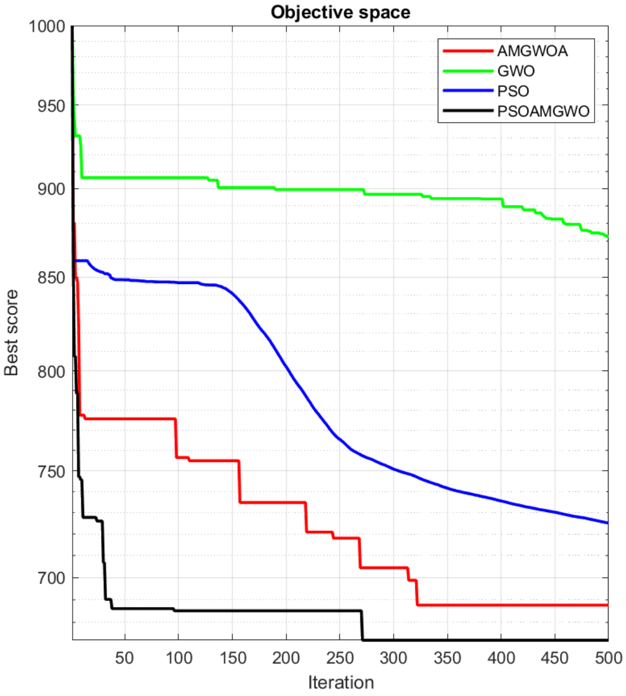

Figure. 3 Comparing the Convergence Curves for AMGWOA, GWO, PSO, and PSOAMGWO

Table 1. Analysis of the comparison values for the four algorithms: AMGWOA, GWO, PSO, AND PSOAMGWO

Algorithm	Best Solution	Running time(seconds)	Time Difference
AMGWOA	696	0.008	12.5%
GWO	873	0.238	96.7%
PSO	814	0.079	89.9%
PSOAMGWO	663	0.009	

GWO [8]. To ensure an equitable assessment, all three methods were subjected to identical parameter configurations, including the population size and number of iterations as specified in the Experimental Configuration section. The algorithms' performances are evaluated based on metrics such as running times and the best solution of the objective function values.

Table 1 presents an analysis of the comparison values for the four algorithms: AMGWOA, GWO, PSO, and PSOAMGWO.

Table 1 demonstrates that our suggested PSOAMGWO outperforms traditional GWO, AMGWOA, and PSO in terms of the value of the objective function represented by Eq. (20), achieving the lowest score. This phenomenon is a direct result of the synergistic integration of the robustness of both PSO and GWO algorithms. Regarding the duration of execution, AMGWOA demonstrates a substantial reduction in execution time, with a decrease of 96.7% compared to the GWO algorithm and 89.9% compared to PSO.

The PSOAMGWO algorithm exhibits superior performance compared to GWO and PSO in terms of both running time and the quality of optimal "Minimum" values it generates, surpassing even AMGWOA. AMGWOA outperforms PSOAMGWO in terms of time by a margin of 12.5%, while the difference is not highly significant.

The PSOAMGWO algorithm significantly enhances the accuracy of the PSO and GWO algorithms relative to both the quality of the results and the computational efforts required.

Fig. 3 displays the average value of a test function graphed against the iteration count for the AMGWOA, the standard GWO, PSOAMGWO, and PSO algorithms. The plot illustrates that the PSOAMGWO algorithm exhibits significantly faster convergence compared to GWO, PSO, and AMGWOA. The enhanced efficacy of the integrated PSO and (GWO) Optimization algorithms can be explained by their capacity to exploit and explore.

5.2.2 Comparative analysis of DDoS detection graphs utilizing our proposed methodology versus AMGWOA

The two figures below Figs. 4 and 5 depict the outcomes of a comparison analysis that evaluated the effectiveness of the conventional detection approach, AMGWOA, and our novel approach PSOAMGWO, in identifying fraudulent requests. The x-axis represents the quantity of requests, while the y-axis represents the anticipated arrival time of these requests.

The results depicted in both Figs. 4 and 5 demonstrate the classification of cumulative requests in the Controller before optimization, after applying AMGWOA, and following PSOAMGWO Optimization. Analysis of these figures reveals that the DDoS detection capability of the proposed PSOAMGWO method surpasses existing methods, leading to a significant reduction in the total amount of requests within the time window μ ∈ [0.01, 1]. Fig. 4 displays a request count of 600, whereas Fig. 5 exhibits an increase in requests to 1200. These statistics demonstrate when the amount of requests gets higher, our PSOAMGWO approach becomes more efficient under the same setting.

Before classification, it is important to acknowledge that the volume of requests arriving is significant and beyond the allotted time for processing. However, after implementing AMGWOA optimization, it is demonstrated that regardless of the number of requests, their total is minimized. Furthermore, the application of

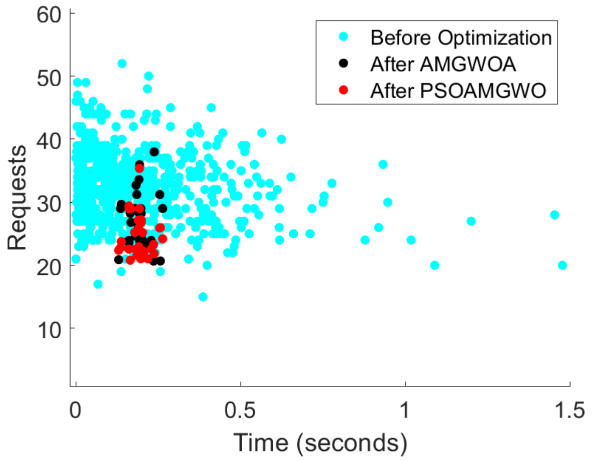

Figure. 4 A graph depicting the number of 600 requests over time

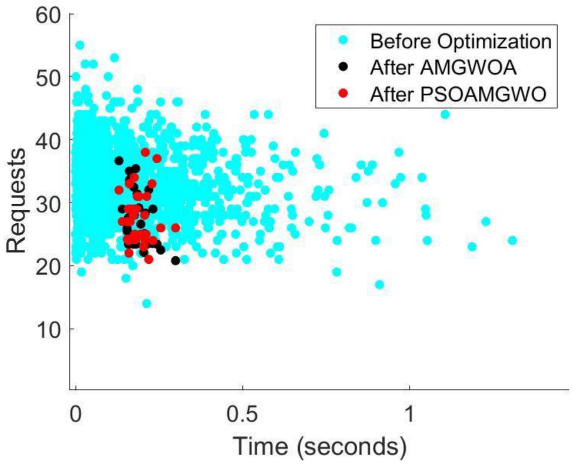

Figure. 5 A graph depicting the number of 1200 requests over time

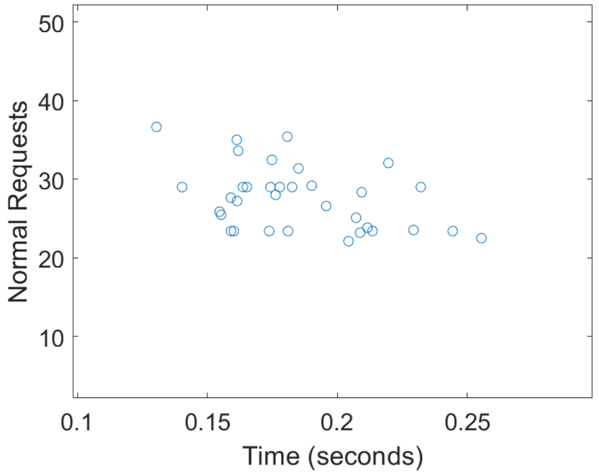

Figure. 6 Normal requests versus time after PSOAMGWO

PSOAMGWO leads to a further decrease in the volume of requests when compared with AMGWOA.

Regular requests are defined as those that meet the required parameters in the constraints, where $\tau \in [20, 50]$ and $\mu \in [0.01, 1]$

The data depicted in Fig. 6 illustrate the typical requests following the application of PSOAMGWO. It is evident that these requests do not exceed the defined range of [20, 50] and adhere to the specified time interval of $\mu \in [0.01, 1]$, unlike the malicious inquiries. It is evident that the queries are reduced to less than 0.4 seconds after optimizing PSOAMGWO. Requests occurring beyond this range and time frame become invisible, as the algorithm automatically blocks and discards them.

In summary, the performance of the proposed PSOAMGWO method, which integrates the strengths of GWO and PSO, is superior to the Standard GWO, PSO, and AMGWOA optimizer in its ability to prevent illegitimate requests and demonstrates encouraging outcomes in properly balancing the exploration and exploitation of optimization threats.

5.2.3 Algorithms accuracy analysis

In the context of accuracy classification, it is imperative to assess the efficacy of the algorithms based on their performance across different sample sizes. Table 2 presents a comprehensive comparison of the accuracy achieved by the Automated Modified Grey Wolf Optimizer Algorithm (AMGWOA), Grey Wolf Optimizer (GWO), Particle Swarm Optimization (PSO), and our proposed Hybrid Particle Swarm Optimization and Automated Modified Grey Wolf Optimizer Algorithm (PSOAMGWO) for sample sizes of 600 and 1200 requests.

Among the algorithms, AMGWOA exhibits a commendable accuracy of 93.3% for a sample size of 600 requests, showcasing its robust performance in identifying unauthorized requests. As the sample size increases to 1200 requests, AMGWOA continues to outperform the GWO and PSO algorithms, achieving an accuracy of 96.66%.

Table 2. Algorithms Accuracy Analysis

Algorithms	**Accuracy Sample size=600**	**Accuracy Sample size=1200**
AMGWOA	93.3%	96.66%
GWO	73.33%	83.33%
PSO	86.67%	90.3%
PSOAMGWO	96.6%	100%

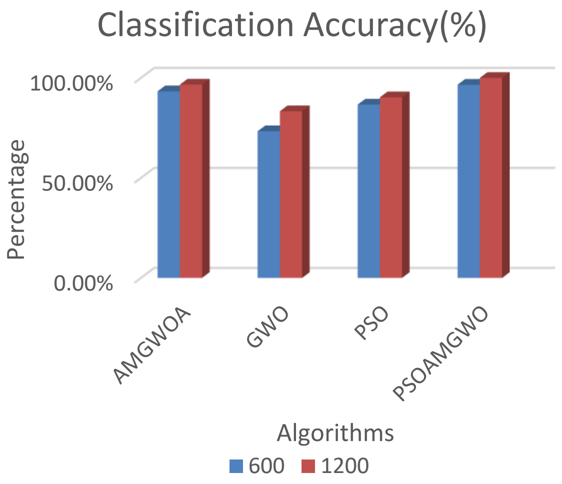

Figure. 7 Algorithms Accuracy Classification

Comparatively, the standard Grey Wolf Optimizer demonstrates lower accuracy levels, achieving 73.33% and 83.33% for sample sizes of 600 and 1200 requests, respectively. Particle Swarm Optimization shows improved accuracy, reaching 86.67% and 90.3% for the corresponding sample sizes.

Remarkably, our proposed PSOAMGWO algorithm emerges as the most effective in detecting unauthorized requests. With an accuracy of 96.6% for a sample size of 600 requests and a perfect accuracy of 100% for 1200 requests, PSOAMGWO surpasses all other algorithms. This underscores the synergistic benefits of combining Particle Swarm Optimization and Automated Modified Grey Wolf Optimization, resulting in a highly efficient algorithm for rapid and accurate detection of malicious activities within Software-Defined Networks.

The superior performance of PSOAMGWO indicates its potential for practical implementation in enhancing security measures, providing reduced detection time, and optimizing resource utilization in SDN environments. The information is presented in Fig. 7.

5.2.4 Comparative analysis

Table 3 compares the results produced with our approach versus existing methodologies. Our proposed method was compared to previous works to evaluate its effectiveness. The proposed PSOAMGWO obtained a better accuracy of 100% when compared to existing methods in the literature survey such as [17-24]. Based on the comparison of approaches detecting the same type of attacks (DDoS) in SDN, our suggested IDS outperforms all current IDSs. Fig. 8 below shows a comparison of performance measurements.

Table 3. Comparative analysis between our new approach and existing method

Authors	***Methodologies***	***Dataset***	***Accuracy (%)***
[17]	RF, and SVM	Created SDN traffic Dataset into CSV file	98.8
[18]	NV , KNN, RF, SVM, DT	NSLKDD dataset	99.97
[19]	SVM, DT, K-NN, and ANN	Created SDN traffic Dataset into CSV file	99.75
[20]	CNN	CICDDoS2019	97.94
[21]	KNN, SVM, and RF, MLP, CNN, GRU, and LSTM	CICIDS20127 dataset	95
[22]	SAE-1SVM	CICIDS20127 dataset	99.35
[23]	fuzzy logic, Shannon entropy, and LSTM	CICDDoS 2019 dataset	99.98
[24]	DNN and SVM	KDD CUP dataset	92.3%
Proposed Method	PSOAMGWO	-	100

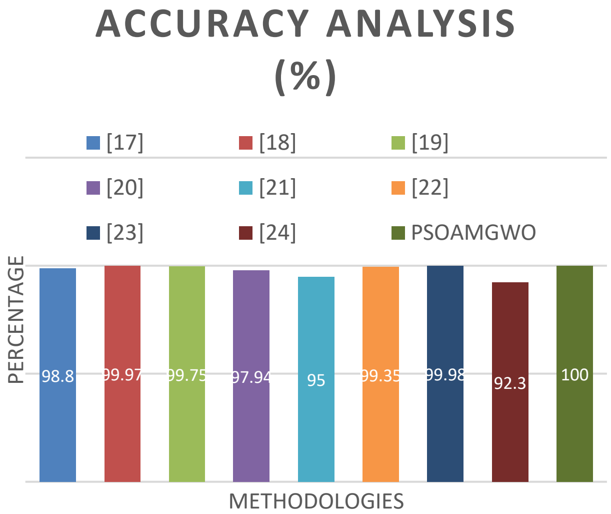

Figure. 8 Comparison with state-of-the-art IDSs

6. Conclusion

This paper presents the integration of the GWO (Grey Wolf Optimizer) with PSO (Particle Swarm Optimization) algorithms to effectively counteract Distributed Denial of Service (DDoS) attacks on Software-Defined Networking (SDN) systems. The majority of the strategies previously mentioned in Section 3 focus on the identification and mitigation of assaults that have already taken place. These methods necessitate a significant quantity of data storage, which might pose difficulties for devices with restricted memory capacity. Additionally, they expose the controller to significant dangers, as certain attacks can have instant consequences on the system before being detected. Our method effectively reduced the number of flows without requiring excessive storage or processing capacity to distinguish between fraudulent and legitimate requests.

The experiments we conducted illustrate the comparatively low time and space requirements of our technique. The minimization of the objective function graph for illegitimate requests in Fig. 3 demonstrates the results of our experiments, which support the efficacy of our strategy. This optimization is carried out while respecting the criteria specified for normal requests, which must be assigned in the allocated timing of µ in [0.01, 1] and lie within the range of the range [20, 50]. For future studies, we recommend employing a variety of hybrid metaheuristic algorithms to implement our architecture, followed by a thorough evaluation and comparison of the obtained results.

Notation List:

Parameter	Description
$\vec{a}$	Represents a vector initialized to 2
$\vec{B}$	Coefficient vector at the iteration *t*
Bw	Bandwidth
C	Represents the capacity of the system
$\vec{D}$	Coefficient vector at the iteration *t*
m	Represents a tunable parameter used to control the exponent in the expression Re q_i^m
n	Represents the number of Components
$\vec{r_1}$, $\vec{r_2}$	Random values lies in the range [0,1]
Req^{γ}	Sets of requests
Req_i	Represents the i-th request
Re q_i^m	Represents the characteristic of the system for the i-th component raised to the power *m*
Req^m	Represents the specific malicious requests that are being targeted for minimization
$\vec{Y_P}$	Position of the prey
$\vec{Y_\alpha}$	The first malicious request
$\vec{Y_\beta}$	The second malicious request
$\vec{Y_\delta}$	The third malicious request
μ	The time windows
σ^t	The time frame during which requests are received
w	Inertia
λ^+	Upper bound
λ^-	Lower bound

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

“Conceptualization, E.M, K.K.R and E.O.A; investigation, A.D.; resources, A.D., E.M, K.K.R and E.O.A; data curation, A.D., E.M, K.K.R, and E.O.A; writing---original draft preparation, A.D.; writing---review and editing, A.D, E.M, K.K.R and E.O.A.; visualization, A.D., E.M, K.K.R, and E.O.A; supervision, E.M, K.K.R, and E.O.A; project administration, A.D.; funding acquisition, A.D. All authors have read and agreed to the published version of the manuscript.”

Acknowledgments

I acknowledge the invaluable contribution from Mawazo Institute and its partners in supporting my research, which has been instrumental in enabling me to achieve this.

References

[1] F. Bannour, S. Souihi, and A. Mellouk, “Distributed SDN control: Survey, taxonomy, and challenges”, *IEEE Commun. Surv. Tutor.*, Vol. 20, No. 1, pp. 333-354, 2017.

[2] N. Bizanis, and F. A. Kuipers, “SDN and virtualization solutions for the Internet of Things: A survey”, *IEEE Access*, Vol. 4, pp. 5591-5606, 2016.

[3] A. M. Bahaa-Eldin, E. E.-E. ElDessouky, and H. Dağ, “Protecting openflow switches against denial of service attacks”, In: *Proc. of 2017 12th International Conference on Computer Engineering and Systems (ICCES)*, pp. 479-484, 2017.

[4] S. Faizullah and S. AlMutairi, “Vulnerabilities in sdn due to the separation of data and control planes”, *Int. J. Comput. Appl.*, Vol. 31, pp. 21-24, 2018.

[5] Z. Tu, H. Zhou, K. Li, M. Li, and A. Tian, “An energy-efficient topology design and DDoS attacks mitigation for green software-defined satellite network”, *IEEE Access*, Vol. 8, pp. 211434-211450, 2020.

[6] B. Wang, Y. Sun, and X. Xu, “A scalable and energy-efficient anomaly detection scheme in wireless SDN-based mMTC networks for IoT”,

IEEE Internet Things J., Vol. 8, No. 3, pp. 1388-1405, 2020.

[7] S. G. Rawat, M. S. Obaidat, S. Pundir, M. Wazid, A. K. Das, D. P. Singh, and K. F. Hsiao, “A survey of ddos attacks detection schemes in SDN environment”, In: *Proc. of 2023 International Conference on Computer, Information and Telecommunication Systems (CITS)*, IEEE, pp. 01-06, 2023.

[8] F. Gul, I. Mir, L. Abualigah, P. Sumari, and A. Forestiero, “A consolidated review of path planning and optimization techniques: Technical perspectives and future directions”, *Electronics*, Vol. 10, No. 18, p. 2250, 2021.

[9] J. Kennedy and R. Eberhart, “Particle swarm optimization”, In: *Proc. of ICNN’95-international conference on neural networks*, ieee, pp. 1942-1948, 1995.

[10] S. Budilaksono, AA. Riyadi, L. Azhari, DD. Saputra, MA. Suwarno, IGA. Suwartane, and A. Fauzi, “Comparison of data mining algorithm: PSO-KNN, PSO-RF, and PSO-DT to measure attack detection accuracy levels on intrusion detection system”, In: *Proc. of Journal of Physics: Conference Series*, IOP Publishing, p. 012019, 2020.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer”, *Adv. Eng. Softw.*, Vol. 69, pp. 46-61, 2014.

[12] A. Dembele, E. Mwangi, A. Bouchair, K. K. Ronoh, and E. O. Ataro, “Automated Modified Grey Wolf Optimizer for Identification of Unauthorized Requests in Software-defined Networks”, *Int. J. Adv. Comput. Sci. Appl.*, Vol. 14, No. 7, 2023.

[13] D. T. Pham and T. T. B. Huynh, “An effective combination of genetic algorithms and the variable neighborhood search for solving travelling salesman problem”, In: *Proc. of 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 142-149, 2015.

[14] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, “Hybrid metaheuristics in combinatorial optimization: A survey”, *Appl. Soft Comput.*, Vol. 11, No. 6, pp. 4135-4151, 2011.

[15] M.-P. Song and G.-C. Gu, “Research on particle swarm optimization: a review”, In: *Proc. of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, Vol. 4, pp. 2236-2241, 2004.

[16] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh, and I. H. Hasbullah, “A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking”, *Sensors*, Vol. 23, No. 9, p. 4441, 2023.

[17] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, “Automated DDOS attack detection in software defined networking”, *J. Netw. Comput. Appl.*, Vol. 187, p. 103108, 2021.

[18] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, and S. Al-Sarawi, “Renyi joint entropy-based dynamic threshold approach to detect DDoS attacks against SDN controller with various traffic rates”, *Appl. Sci.*, Vol. 12, No. 12, p. 6127, 2022.

[19] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini, and M. Tornatore, “Machine-learning-enabled ddos attacks detection in p4 programmable networks”, *J. Netw. Syst. Manag.*, Vol. 30, pp. 1-27, 2022.

[20] M. W. Nadeem, H. G. Goh, V. Ponnusamy, and Y. Aun, “DDoS Detection in SDN using Machine Learning Techniques”, *Comput. Mater. Contin.*, Vol. 71, No. 1, 2022.

[21] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, and R. Kocaoğlu, “Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking”, *Electronics*, Vol. 10, No. 11, p. 1227, 2021.

[22] H. Wang, and W. Li, “DDosTC: A transformer-based network attack detection hybrid mechanism in SDN”, *Sensors*, Vol. 21, No. 15, p. 5047, 2021.

[23] N. M. Yungaicela-Naula, C. Vargas-Rosales, and J. A. Perez-Diaz, “SDN-based architecture for transport and application layer DDoS attack detection by using machine and deep learning”, *IEEE Access*, Vol. 9, pp. 108495-108512, 2021.

[24] L. Mhamdi, D. McLernon, F. El-Moussa, S. A. R. Zaidi, M. Ghogho, and T. Tang, “A deep learning approach combining autoencoder with one-class SVM for DDoS attack detection in SDNs”, In: *Proc. of 2020 IEEE Eighth International Conference on Communications and Networking (ComNet)*, IEEE, pp. 1-6, 2020.

[25] M. P. Novaes, L. F. Carvalho, J. Lloret, and M. L. Proença, “Long short-term memory and fuzzy logic for anomaly detection and mitigation in software-defined network environment”, *IEEE Access*, Vol. 8, pp. 83765-83781, 2020.