Automated Modified Grey Wolf Optimizer for Identification of Unauthorized Requests in Software-defined Networks

Aminata Dembele[1], Elijah Mwangi[2], Abderrahim Bouchair[3], Kennedy K Ronoh[4], Edwin O Ataro[5]
Pan African University, Institute for Basic Sciences, Technology and Innovation (PAUSTI), Nairobi, Kenya[1]
University of Nairobi, Electrical and Information Engineering, Nairobi, Kenya[2]
University of Oran1 Ahmed Ben Bella, Oran, Algeria[3]
School of Computing and Informatics, Strathmore University, Nairobi, Kenya[4]
The Technical University of Kenya, Electrical and Information Engineering, Nairobi, Kenya[5]

***Abstract*—Software Defined Networking (SDN) is utilized to centralize network control within a controller, but its reliance on a single control plane can make it vulnerable to attacks such as DDoS. This highlights the importance of developing effective security mechanisms and using proactive measures such as detection and prevention strategies to mitigate the risk of attacks. Many DDoS attack detection technologies within SDN focus on detecting and mitigating the attack once it has occurred in the controller, which leads to more seconds of exposure, diminished precision, and high overhead. In this work, we have developed an Automated Modified Grey Wolf Optimizer Algorithm (AMGWOA) to design the detection of this malicious activity in an SDN environment to prevent the attack in the controller. Our methodology involves the development of the AMGWOA, which incorporates a mechanism to facilitate the blocking of malicious requests while reducing detection time and minimizing the use of storage and data resources for detection purposes. The results obtained show that our model performs well, with an ability to minimize a very large number of malicious requests in a minimum of time of less than 1 second compared to Grey Wolf Optimizer and particle swarm optimization algorithms evaluated using the same datasets.**

***Keywords*—Software-defined networks; security; DDoS attacks; metaheuristic algorithms; Grey Wolf Optimizer**

I. Introduction

Prior to very recently, communication service providers (CSPs) used proprietary physical equipment and devices to carry out network activities, making security a crucial component of wireless communication systems. This typical network design does not provide a scalable and manageable solution for such big and complicated networks, and as user needs rise, more hardware devices are needed to satisfy consumer expectations [1].

With 5G and edge computing, software-defined networking (SDN), which separates the control plane from the data plane, can be used to deliver more adaptable and dynamic services across the wireless communication network [2].

Although this SDN technology has several significant advantages, such as flexibility and economical, effective administration, it also introduces new risks [3]. The SDN controller serves as the brain of the system. The entire network will be at risk if the controller is compromised, or worse, destroyed. The SDN paradigm is vulnerable to DDoS attacks from malicious users, according to a number of recent research studies [4] [5] [6]. This attack is characterized by a large number of puppet hosts controlled by the controller launching an attack on the targeted system, rapidly depleting its resources and threatening its continued operation.

When a DDoS attack affects an SDN network, the switches generate a flood of incoming packet messages for the controller to process. This places a strain on the controller's assets. causes the switch routing table to grow and potentially compromises the integrity of the encrypted connection between the controller and the switches. This has the potential to bring the whole SDN network down.

If the DDoS happens in SDN, communication channels might be quickly blocked, and controller resources would be used up, drastically reducing service quality.

In this research, we provide a novel model based on the Grey Wolf optimization technique that acts in an automated way to prevent DDoS attacks in the SDN network.

Mirjalili [7] describes the Grey Wolf Optimizer (GWO), a novel population-based algorithm motivated by the hunting strategies of a wolf pack. While other evolutionary computation-based methods, such as particle swarm optimization (PSO), fast evolutionary programming (FEP), and the gravitational search algorithm (GSA), achieve comparable performance, GWO has the advantage of requiring fewer adjustment parameters [7].

The design of an improved algorithm based on an automated modified grey wolf (GWO) is the contribution of this paper. This modified GWO will identify the most malicious requests and facilitate blocking them by the SDN Controller, which will minimize the risk of dealing with a critical DDoS attack, minimize the latency, and maintain the continuous availability of the controller for legitimate users.

This work is organized as follows: The background of the research is provided in Section II. Section III gives a brief summary of previous work. Section IV presents the design of our proposed algorithm (AMGWOA) for DDoS detection in SDN. Section V contains the experimental methodology and findings. Section VI is the concluding section of the paper.

II. Background

A. Overview of the SDN Concept

The control and infrastructure layers of conventional networks function as a single unit. The control layer is responsible for determining the best route for data packets to take across the network, while the Infrastructure layer is responsible for carrying out those instructions. In software-defined networking (SDN), the Infrastructure land control layers are two distinct entities, with the control layer controlling several data planes. Through "softwarization," it can centrally monitor and regulate the network. The architecture of this new software technology is shown in Fig. 1.

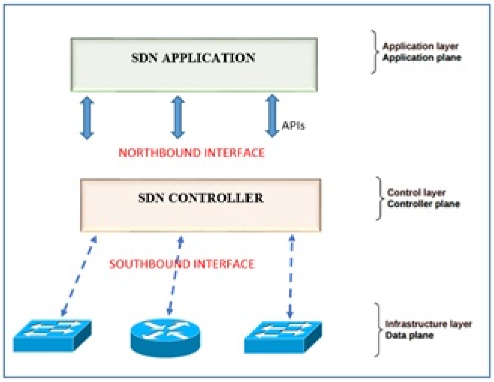


Fig. 1. A three-layer software-defined networking (SDN) architecture.

1) Application layer: It contains network applications like firewalls, load balancing, monitoring, and routing. The management plane is in charge of establishing regulations and guidelines.

2) Control layer: It is responsible for setting up the forwarding devices. The program that interacts with the hardware and software parts of the network in an SDN is called a controller. It is a focal point of the network since it coordinates information transfer between northbound and southbound APIs and connects the data plane and application plane.

3) Infrastructure layer: In the data plane, a physical network architecture is defined. Switches and routers are examples of forwarding devices, and they can communicate with one another using either wired or wireless means. In the data plane, the header, match, and actions fields constitute the main part of forwarding tables. The ternary content addressable memory (TCAM) contains the flow entries into tables for the data plane. Another name for it is a forwarding plane (FP).

4) Northern connection: This connection interface is called the northbound interface. Communication between the management plan and the control plan. It gives the southbound interface low-level instructions. It's also known by its alternate name, the Management to Control Plane Interface (MCPI). As of yet, there are no agreed-upon protocols for the northbound interface [8, 9].

5) Southern connection: The southern interface is known as the southbound interface. it provides a means of communication between the control and infrastructure layers, made possible by a protocol called OpenFlow. The control plane and data plane are separated by the OpenFlow standard protocol for SDNs [10].

B. DDoS Attacks in Software Defined Networks

when the processing power of the network is centralized, one central point of vulnerability is created. In simpler terms, the network will fail if the state of the SDN controller is compromised or cannot fulfill the requests of the switches. The DDoS attack in SDN illustrated in Fig. 2 aims to overwhelm the target host resources in order to disrupt the benign host.

The following possible attack scenarios could be carried out by attackers:

1) Attack on the application plane: The attack takes place via the applications that are present in the application plane. The rogue application uses up all the resources, hurting honest users.

2) Attack on the controller: The controller would be able to handle all of the packet requests sent by the attacker, which would result in a malfunction. As a result, all requests from respectable users suffer.

3) The transmission path of information transfer between the control plane and the data plane: An attacker could attempt to assault the communication channel connecting the control and data planes by sending many Packets as requests.

4) Attack using a table overflow: Innocent users are harmed when an attacker utilizes phony IP sources to fill the switch flow table to its maximum capacity. As a result, a decent traffic sender will be denied access to the services.

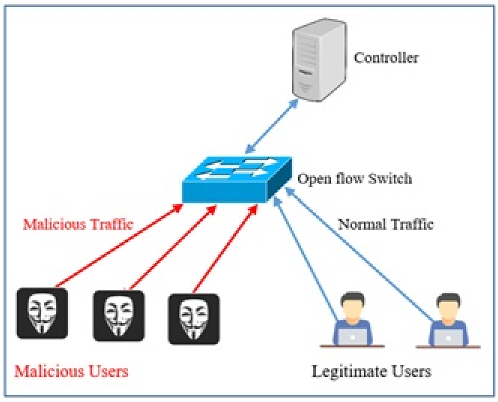


Fig. 2. DDOS attack scenario in SDN an example.

C. Overview of the Grey Wolf Optimizer

Grey wolves tend to move and hunt in packs of 10 to 17 individuals, and this social behavior inspired GWO, a population-based metaheuristic algorithm [11],[12],[13],[14],[15]. Grey wolves have a distinct social hierarchy. Wolves at the top of the social hierarchy of a pack are called alphas. In the hierarchy, Beta, Delta, and Omega are ranked second, third, and fourth, respectively. When to get up, where to sleep, and when to go hunting are all decisions made by the Alpha.

The rest of the wolves must follow the decision of the Alpha. Beta wolf provides assistance to the Alpha wolf when making decisions and will take over if the Alpha dies or is incapacitated.

The Beta defers to the Alpha's decision but gives orders to lower-ranked wolves. Sentinels, scouts, elders, and caregivers are all examples of Delta wolves. The Omegas are at the bottom of the list and should be consumed last. Omega wolves take orders from all wolves. In the same pack, the delta wolf, in turn, dominates omega but follows the instructions of alpha and beta.

Grey wolves are sociable animals with many shared traits, including group hunting. Grey wolves will follow, pursue, and approach their prey first. The target will then be pursued, encircled, and harassed until it stops moving.

Wolves will then attack their prey in the final phase of the hunt.

The GWO algorithm mimics two social behaviors typical of wolves: social hierarchy and collective hunting. Each of the individual wolves represents various strategies for achieving optimal performance. Alpha (α) is the best possible response, while beta (β) is the second, and delta (δ) is the third choice. These three competitors are leading and being followed by the hunt. Every other option is assumed to be the omega (ω) solution. The wolves encircling behavior is modeled analytically using Equation (1).

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

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

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

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

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

where a is reduced linearly from 2 to 0 over iterations and r_1 and r_2 are random vectors in $[0,1]$. The alpha, beta, and delta are said to have superior knowledge of the likely whereabouts of prey in order to mimic the hunting behavior of grey wolves. Once the best search agents' locations have been determined (alpha, beta, and delta), the positions of the other wolves will be updated accordingly. The wolves' positions are updated in accordance with equation (5).

$$\overrightarrow{Y}(t+1)=(\overrightarrow{Y_1}+\overrightarrow{Y_2}+\overrightarrow{Y_3})/3 \tag{5}$$

Where $\overrightarrow{Y_1}$, $\overrightarrow{Y_2}$)and $\overrightarrow{Y_3}$ are defined in equations (6), (7) and (8).

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

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

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

where $\overrightarrow{Y_\alpha}$, $\overrightarrow{Y_\beta}$ and $\overrightarrow{Y_\delta}$ are the positions of the first best three solutions, $\overrightarrow{B_1}$, $\overrightarrow{B_2}$ and $\overrightarrow{B_3}$ are defined in equations (6), (7) and (8) and $\overrightarrow{E_\alpha}$, $\overrightarrow{E_\beta}$, and $\overrightarrow{E_\delta}$ are defined in equations (9), (10) and (11).

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

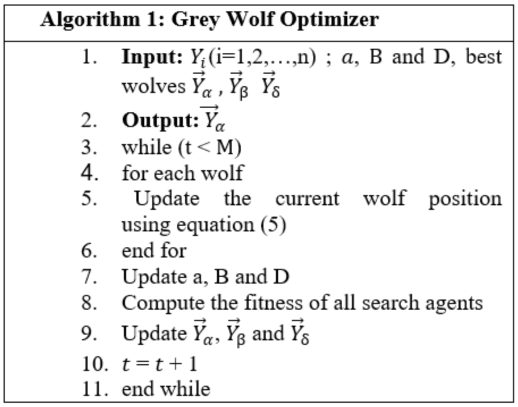
$$\overrightarrow{E_\beta}=|\overrightarrow{D_2}\cdot\overrightarrow{Y_\beta}-\overrightarrow{Y}| \tag{10}$$

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

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

$$A=2-\frac{2t}{M} \tag{12}$$

where t is 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

```
Input: $Y_i$(i=1,2,…,n) ; $a$, B and D, best wolves $\vec{Y}_\alpha$ , $\vec{Y}_\beta$  $\vec{Y}_\delta$
Output: $\overrightarrow{Y_\alpha}$
while (t < M)
    for each wolf
        Update the current wolf position using equation (5)
    end for
    Update a, B and D
    Compute the fitness of all search agents
    Update $\vec{Y}_\alpha$, $\vec{Y}_\beta$ and $\vec{Y}_\delta$
    $t = t + 1$
end while
```

III. Related Works on DDOS Detection in SDN

There has been a lot of discussion about the security risks that SDN faces. DDoS attacks are the most frequent and well-known SDN attacks. Numerous DDoS detection algorithms have been proposed thus far, but only a selected few are presented here.

A. The Detection Methods Based on Information Entropy

In [16] DDoS is detected by evaluating the unpredictability of incoming packets and using two elements: window size and threshold. The entropy of packets is calculated and if it goes below a threshold, the attack is detected. This strategy merely identifies the DDos attack, but does not eliminate it.

Authors in [17] proposed a fusion entropy method. In this method, the benefits of log energy entropy and information entropy are combined to achieve complementarity. Attackers can take advantage of fusion entropy's ease of detection and the transparency of its entropy value variations. Since it is challenging to discern between normal network traffic and low-rate DDoS attacks when they occur, this method makes it more challenging to detect low-rate DDoS assaults.

Low-rate and high-rate DDoS attacks against the controller are both detectable using an entropy-based DDoS attack detection method, which the authors of [18] and [19] assess in terms of detection rate (DR) and false-positive rate (FPR), as well as whether the attacks originate from a single host, multiple hosts, or both. Eight different scenarios were tested, each representing a different level of traffic rate during a distributed denial of service (DDoS) attack on the controller. Experimental results show that the average DR for identifying high-rate DDoS attack traffic is improved by 6.25% points, 20.6% points, 6.74 % points, and 8.81% points using the entropy-based method.

These information entropy detection-based techniques fail to optimize most of the limited resources of the controller.

B. Detection Based on Machine Learning

DDoS attacks can be detected with SDN security. Machine learning-based detection approaches are used more frequently than those based on information entropy [20].

Relevant feature selection methods for DDoS detection using ML are discussed in [21]. The final feature selection is based on the classification accuracy of the machine learning methods and the efficiency of the SDN controller. Comparative research on feature selection and machine learning classifiers for SDN attack detection has also been conducted. Using a subset of features determined by the Recursive Feature Elimination (RFE) approach, the Random Forest (RF) classifier is able to train a model with an accuracy of 99.97 %, as shown by their experimental findings.

In [22] authors proposes a deep learning (DL) based ensemble solution to address the problem of DDoS attack detection in SDN. In order to enhance SDN traffic classification, four hybrid models have been provided that combine three ensemble methodologies with three distinct DL architectures (convolutional neural network, long short-term memory, and gated recurrent unit). The CICIDS2017 flow dataset served as the basis for the experiments. The findings demonstrated high detection accuracy (99.77%). This method has a higher controller resource requirement because it uses four distinct hybrid models.

The authors in [23] have studied several machine learning models for DDoS detection in SDN. The question of how to improve the accuracy of DDoS attack detection has been studied using a well-known DDoS dataset called CICDDoS2019. In addition, the DDoS dataset has been preprocessed using two main approaches to obtain the most relevant features. Four machine learning models have been selected for the DDoS dataset. According to the results obtained from real experiments, the Random Forest machine learning model offered the best detection accuracy with (99.9974%), with an enhancement over the recently developed DDoS detection systems.

Authors in [24] attend to detect DDoS attacks by classifying the normal and malicious traffic. The study solves the data shift issues by using the introduced Decision Tree Detection (DTD) model encompassing of Greedy Feature Selection (GFS) algorithm and Decision Tree Algorithm (DTA). Initially, the gureKddcup dataset is loaded to perform preprocessing. After this, feature selection is performed to select only the relevant features, removing the irrelevant data. The results of the investigation revealed that the proposed system achieved an accuracy of 98.42% in the test data. this technique is based on the decision tree, which often involves higher time to train the model, which is costly in terms of detection time.

The authors of [25] use machine learning algorithms in conjunction with Neighborhood Component Analysis (NCA) to categorize SDN traffic as either benign or malicious. The project leveraged a publicly available “DDoS attack SDN dataset,” which had a total of 23 features. Through feature selection, the NCA algorithm reveals the most important features, allowing for accurate categorization. The acquired dataset was then categorized using the k-Nearest Neighbor (kNN), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM) methods, after the preparation and feature selection phases. The experimental findings demonstrate that DT achieves a perfect 100% classification rate, which is far higher than any of the competing methods. This method could not be continued with other types of high-volume datasets because DT is sometimes unstable, meaning that a small change in the data can lead to a large change in the optimal decision structure.

Authors in [26], did a study that was focused on DDoS attack detection using machine learning-based methods. The primary goal of the study was to reduce misclassification error in DDoS detection and this was made possible by using Mutual information and Random Forest Feature Importance. From the features selected, Random forest, Gradient Boosting, Weighted voting ensemble, and KNN were applied and they had better accuracy when using the features selected. Random Forest, performed better in DDoS attack detection and only misclassified 1.

Although the aforementioned studies are grounded in machine learning, the most majority rely on inefficient, time-consuming, and costly fixed detection approaches that require immediate control.

C. Detection Based on Optimization Algorithms

In order to create an innovative solution, the modern new approach to DDoS detection in SDN relies on optimization algorithms techniques. The study in [27] designed an efficient and low-power SDSN topology by using the Degree Constrained Topology Generation (DCTG) algorithm and a novel formulation of the optimization target. The primary purpose of the method is to design a topology for a Software-Defined Satellite Network (SDSN) that minimizes power consumption. In addition to considering all possible link states, the proposed method strives to reduce the aggregate power usage of the network.

The authors of [28] proposed a satellite network topology optimization technique that incorporated NIDS (Network Intrusion Detection System) using federated learning distributed NIDS in STN. This program Could evaluate and filter harmful traffic as well as fairly distribute resources across each domain. Additionally, it can reduce malicious packet tracking challenges brought by frequent network changes. Malicious traffic could be identified with greater accuracy than typical NIDS, yet with less CPU usage.

Defending DDoS attacks with a metaheuristic strategy, the authors of [29] presented a whale optimization algorithm-based clustering for DDoS detection (WOA-DD). The WOA is a metaheuristic algorithm that takes inspiration from nature. With this historical data, WOA-DD hopes to distinguish between typical traffic and malicious DDoS attacks. After the clusters have been created, any incoming requests will be distributed among them at random. WOA-DD prevents DDoS attacks, but the technology has a major drawback: the clustering process significantly slows down decision-making.

In conclusion, the SDN ecosystem is abundant with DDoS attack detection solutions that aim to identify and counteract the issue of data memory. This requires the creation of a memory-efficient, computationally straightforward, and network overhead solution.

The Optimization techniques have been utilized by many researchers to solve the network Intrusion problem. These techniques will be employed in solving the DDoS attack in the proposed method. Several optimization issues have recently been solved successfully with the help of metaheuristics (e.g., Facial emotion recognition, disease diagnosis, gene selection, and intrusion detection systems) [30]. In contrast to exact search mechanisms, metaheuristics deliver exceptional performance, because unlike full search algorithms, they don't need to traverse the entire search space to find the optimal solution, which is an advantage in terms of computational complexity and memory.

IV. Automated Modified Grey Wolf Optimizer Algorithm (AMGWOA) for DDOS Detection in SDN

A discussion of the proposed technique is presented in this section. It is based on the Grey Wolf algorithm and applied in blocking malicious requests from the controller.

The algorithm is based on the concept of pack intelligence and uses a grey wolf optimization model to identify the best combination of attack detection techniques.

The Grey Wolf algorithm detects suspicious traffic patterns and DDoS attacks by employing a set of heuristics. The algorithm analyzes traffic input to look for specific characteristics. These traits are used to detect malicious behavior. If any predefined characteristics are found in the data, the system flags the traffic as suspicious and takes proactive measures to block it to protect the network.

To formulate the proposed policy, we define the objective function (fitness function) is defined as follows:

A. Design of AMGWOA

$$\min Z = \sum_{i=1}^{n} Req_i^m \tag{13}$$

subject to:

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

Where Req^m represents the targeted malicious requests to be minimized. The objective function Z is linked to two main constraints:

- A set of requests Req^γ targeting the same resource (service/application).
- An instantiated time window σ^t (in seconds), in which requests are received.

B. Core Components of AMGWOA

To solve the proposed objective function, combine the GWO with a Resource-Constrained (RC) management. This latter tends to classify the received requests based on a threshold (λ) where each request is classified in regards to the three best solutions of GWO. Accordingly, this threshold is defined through the computation of the fitness function, which can be used to identify three classes:

- The Alpha class: it represents the first best solution where in our context is associated to the next Req^m that will be dropped in case of the condition $\omega \leq \lambda^-$ is attained.
- The Beta class: it represents the second best solution. The Req^m will be dropped in case the condition $\omega \in [\lambda^-, \lambda^+]$is satisfied.
- The Delta class: it represents the third best solution. Similar to the previous classes but with the following condition $\omega \geq \lambda^+$, the request will be blocked.

We note that the condition ω is calculated as the following $\omega = Bw/C$ where Bw is the measured bandwidth in the network (bits per second) and the capacity C represents the number of bits that a cable can transfer. The two thresholds λ^+ and λ^- (i.e., upper bound and lower bound values) are initialized based on the constraint σ^t where the defined time range is partitioned into three periods: $[20 - \lambda^-]$, $[\lambda^- - \lambda^+]$ and $[\lambda^+ - 50]$. The overall algorithm as shown on algotithm2 of our modified GWO is described on the following:

C. Implementation of AMGWOA

As shown on algorithm 2 the implementation of Automated modified Grey Wolf is described on the following:

The proposed AMGWOA algorithm will act by minimizing the fitness function, which is the total of the requests, to find the best possible solution.

When the sum of new requests arrives, each request will be examined to ensure that it is not an attack before being sent. That is to say that if it does not respect the pre-established conditions (range and time), it will be automatically blocked and not forwarded to the controller.

V. Experimental Setup and Results Discussions

In this section, we discuss our experimental setup and report our findings from testing the proposed methodology.

A. Experimental Setup

Our Simulation was done using Matlab R2020a. Due to the availability of diverse Matlab is selected because it has diverse mathematical functions.

Experiments are carried out on a personal computer PC HP Pavilion X360, Windows 10 OS with 8GB DDR4, Core i7, 10th generation CPU, and 512 GB SSD.

B. Results Discussion

In this implementation, we set the population size to 35 and the maximum number of iterations to 500 in the proposed Automated Modified Grey Wolf algorithm (AMGWOA).

Algorithm 2: Proposed Automated Modified Grey Wolf Optimizer (AMGWOA)

```
Initialize the GWO population (solution): $Y_i$ ($Y = 20$) ;   /* number of requests considered before blocking the next request. */
Initialize $a, \overrightarrow{B}, \overrightarrow{D}, \lambda^+$ and $\lambda^-$. $t = 0$;
Calculate the fitness of each solution $\overrightarrow{Y_i}$ (e.g., $i = 1 \cdots 20$);
$\overrightarrow{Y_\alpha}$: the first malicious request;
$\overrightarrow{Y_\beta}$: the second malicious request;
$\overrightarrow{Y_\delta}$: the third malicious request;
while ($t < M$) do
    foreach agent do
        if (number of requests = $\tau$ & time window = $\mu$) then
            Update the position of the current agent using equation (1);
            if (the sum of $Req^\gamma$ in the Alpha class is greater than the sum of $Req^\gamma$ in the other two classes) then
                Block the next request;
            else
                Forward the request;
            end
        end
    Calculate the fitness value of each candidate solution (malicious requests);
    Update $\overrightarrow{Y_\alpha}, \overrightarrow{Y_\beta}, \overrightarrow{Y_\delta}$;
    $t \leftarrow t + 1$;
end
Return $\overrightarrow{Y_\alpha}$;
```

1) AMGWOA compared with standard GWO and Particle Swarm(PSO): To verify the performance of our approach, AMGWOA is compared with standard GWO [7] and PSO [31] algorithms. For a fair comparison among the three algorithms, they were tested using the same settings of the parameters, specifically, a population size of 35 and a maximum number of iterations of 500 for all test functions. The performance of the algorithms is compared using the following metrics: Objective function values (Best solution) and running times.

Comparison parameters for the three algorithms AMGWOA, GWO, and PSO are shown in Table I below.

It can be seen from Table I that compared to standard GWO and PSO our proposed AMGWOA produces the best score (lowest) objective function value represented by equation (13) This is in fact due to the tolerable iterations of the algorithm. In terms of running time Compared with the GWO algorithm, AMGWOA reduces execution time by 96.7% and 89.9% with PSO. Compared to GWO The PSO does not take much running time, but it does not converge, giving the best optimal "Minimum" values compared to AMGWOA.

Fig. 3 displays the average value of a test function as a function of the number of iterations of the standard GWO,PSO and AMGWOA algorithms. The graph demonstrates that compared with GWO and PSO, AMGWOA converges much faster.This is because of the two parameters $[\lambda^-, \lambda^+]$ that are introduced into GWO initialization, which improve its exploration ability and enhance global convergence.

TABLE I. AMGWOA Compared with Standard GWO and PSO

Algorithm	Best solution	Running time(seconds)	% Time Improvement
AMGWOA	696	0.008	
GWO	873	0.238	96.7 %
PSO	814	0.079	89.9 %

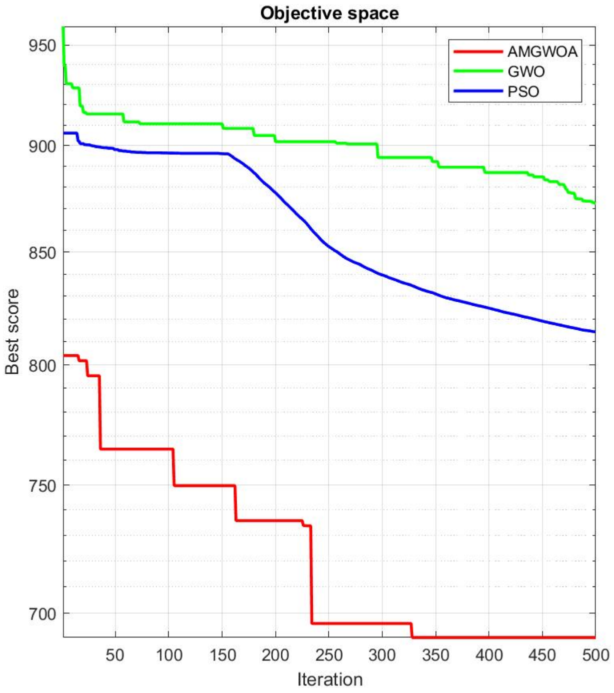


Fig. 3. Comparison of convergence curve for AMGWOA and PSO.

2) Comparison of DDoS detection graphs with and without our proposed method: Fig. 4 and Fig. 5 show the results of a comparison between the use of our AMGWOA model and a standard detection system for fraudulent requests. The number of requests is plotted along the horizontal axis, and the time at which those requests are expected to arrive is plotted along the vertical axis.

The results in Fig. 4 and Fig. 5 represent the classification of the sum of the requests in the Controller before and after AMGWOA Optimization. From Fig. 4 and Fig. 5, it can be observed that the detection of DDoS in the proposed method is superior to the current methods; as a result, the number of requests in the controller within the time window μ is reduced significantly. In Fig. 4 the number of requests is 600 whereas in Fig. 5 the requests are increased to 1200. The figures show that when the number of requests increases, the performance of our method AMGWOA also increases under the same circumstances. It can be noted that before classification the number of incoming requests is huge and exceeds the time that is allocated to them but after optimization, it is shown that whatever the number of requests their sums are minimized, and those that respect the conditions as defined in the constraints $\tau \in [20, 50]$ and, $\mu \in [0.01, 1]$ are maintained as normal requests. The results in Fig. 6 show the normal requests We can see that they do not overflow unlike malicious queries and respect the defined range $[20, 50]$ and time $\mu \in [0, 01, 1]$. The requests after this range and time are not visible because they are blocked and dropped by the algorithm automatically.

Our solution AMGWOA metaheuristic algorithm has been shown to be more effective than Standard GWO optimizer and other competing optimization methods such as PSO in preventing unauthorized requests.

In general, the exploration and exploitation capacity of a population-based metaheuristic algorithm determines its performance [32], [33]. If we further increase the initialization parameters threshold λ^- for our suggested approach, we are increasing the possibility that zombie hosts will pass as regular hosts by significantly altering their query pattern. However, the system's efficiency will drop if we put any otherwise healthy hosts into the zombie host group by lowering the value of the threshold λ^+. Therefore, it is essential to select an accurate threshold value for AMGWOA.

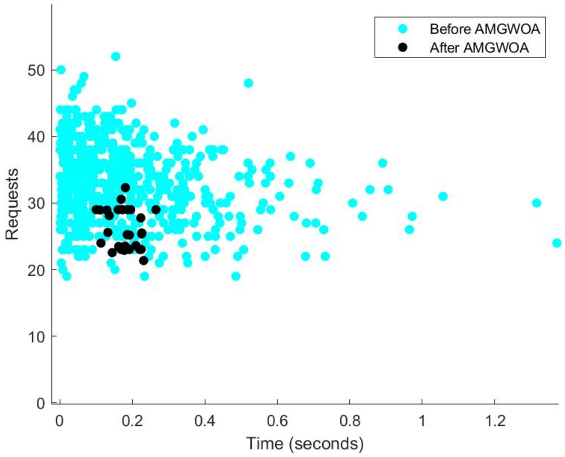


Fig. 4. 600 Requests versus time.

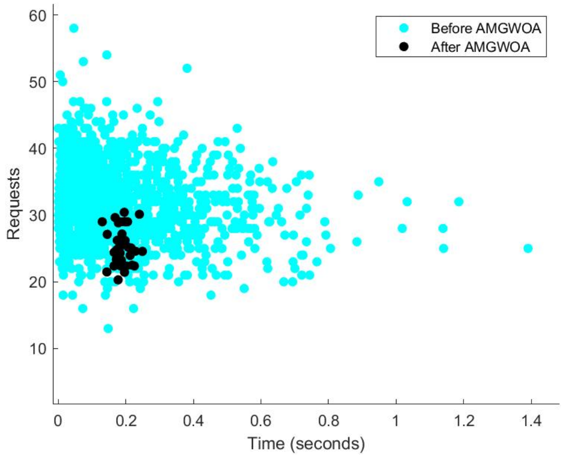


Fig. 5. 1200 Requests versus time.

VI. Conclusion

In this paper, we have developed an automated efficient Grey Wolf optimizer algorithm to prevent the DDoS attack on SDN. The results of our simulations demonstrate the relatively minimal time and space complexity of our approach. Most of the previously proposed methods for detecting and mitigating attacks once happen. These methods necessitate a huge quantity of data storage, which might be problematic for devices with limited memory and exposes the controller to high risks because some of the attacks once they enter the system have immediate effects before their detection. Our solution limited the number of flows without using up a lot of storage or processing space to separate the fraudulent requests from the legal ones.

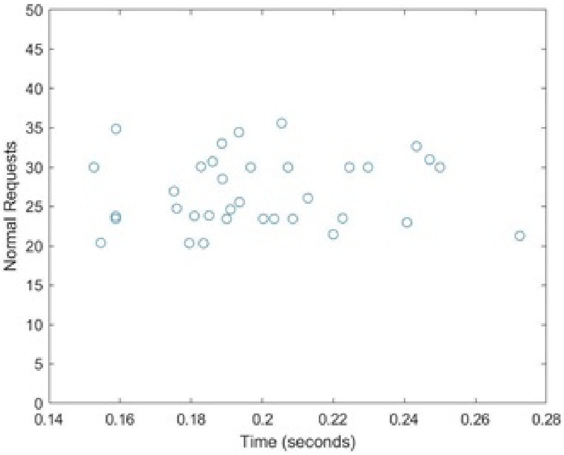


Fig. 6. 1200 Requests versus time.

The experimental findings demonstrate the efficacy of our method because the graph of the fitness function for malicious requests has been minimized while keeping the normal requests that respect the range [20, 50] and the time $\mu \in [0.01, 1]$ allocated to them.

In future work, we will implement our architecture using hybrid metaheuristic algorithms for more accuracy and compare the results obtained.

Acknowledgment

The authors would like to thank the Electrical Engineering (Telecommunications) option of the Pan African Institute of Basic Sciences, Technology and Innovation (PAUSTI) in Kenya for all their technical and financial support in carrying out this research project.

References

[1] F. Bannour, S. Souihi, and A. Mellouk. "Distributed SDN control: Survey, taxonomy, and challenges," *Communications Surveys and Tutorials*, vol. 20, no.1, pp. 333–354, 2018.

[2] N. Bizanis and F. Kuipers, and A. Mellouk, "SDN and virtualization solutions for the internet of things: A survey," *IEEE Access*, vol. 4, pp. 5591–5506, 2018.

[3] S. Faizullah and S. AlMutairi, "SVulnerabilities in SDN due to separation of data and control planes," *International Journal of Computer Applications*, vol. 31, pp. 21–24, 2018.

[4] S. Hameed and H. Khan, "SDN based collaborative scheme for mitigation of ddos attacks," *Future Internet*, vol. 10, no. 3, pp. 1–18, 2018.

[5] Y. Wang, T. Hu, G. Tang, J. Xie, and J. Lu, "SGS: Safe-guard scheme for protecting control plane against ddos attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34 699–34 710, 2019

[6] Y.Qiao and F. Yu, ”Distributed denial of service attacks in software-defined networking with cloud computing”. *Comm. Mag*. vol. 53, no. 4, pp. 52–59 (April 2015).

[7] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” *Adv. Eng. Softw*, vol. 69, pp. 46–61, Mar. 2014

[8] B. Almadani, A. Beg and A. Mahmoud, ”DSF: A Distributed SDN Control Plane Framework for the East/West Interface,” in *IEEE Access*, vol. 9, pp. 26735-26754, 2021.

[9] I. Ahmad, S. Namal, M. Ylianttila and Andrei Gurtov, ” Security in Software Defined Networks: A Survey”, In: IEEE *Communications Surveys and Tutorials*; Vol. 17, No. 4. pp. 2317-2346, 2015.

[10] W.Braun and M. Menth, ”Software-Defined Networking Using OpenFlow: Protocols, Applications and Architectural Design Choices”. *Future Internet*, 6, pp. 302-336, 2014.

[11] N. Singh and S. B. Singh, “Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance,” *J. Appl. Math*, vol. 2017(1), pp. 1– 15, 2017.

[12] M. Panda and B. Das, “Grey Wolf Optimizer and Its Applications: A Survey,” in *Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems* , vol. 556, V. Nath and J. K. Mandal, Eds. Singapore: Springer Singapore, pp. 179–194, 2019.

[13] K. Ronoh, G. Kamucha, and T. Omwansa, “Comparison of Hybrid Firefly Algorithms for Power Allocation in a TV White Space Network,” *Int. J. Comput. Appl*, vol. 178, no. 38, pp. 37–43, Aug. 2019.

[14] K.Ronoh, G.Kamucha, W.Okelo-Odongo, O. Thomas, and T. Omwansa, “Firefly algorithm based power control in wireless TV white space network,” in *AFRICON*, 2017 IEEE, pp. 155–160, 2017.

[15] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary grey wolf optimization approaches for feature selection,” *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.

[16] N. I. Mowla, I. Doh and K. Chae, ”Multi-defense Mechanism against DDoS in SDN Based CDNi,” *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Birmingham, UK*, pp. 447-451,2014.

[17] C.Fan; N.M. Kaliyamurthy; S. Chen, H.Jiang; Y.Zhou; Campbell, C. ”Detection of DDoS Attacks in Software Defined Networking Using Entropy”. *Appl. Sci.* , vol 12, pp.370, 2022.

[18] A.M. Adnan; M. Anbar; A.J. Hintaw; I.H. Hasbullah; A.A.Bahashwan;S. Al-Sarawi. ”Renyi Joint Entropy-Based Dynamic Threshold Approach to Detect DDoS Attacks against SDN Controller with Various Traffic Rates” *Applied Sciences* 12, no. 12: 6127, 2022

[19] A.M. Adnan; M. Anbar; A.J. Hintaw; I. H. Hasbullah, A.A. Bahashwan; T.A. Al-Amiedy; D.R. Ibrahim.”Effectiveness of an Entropy-Based Approach for Detecting Low- and High-Rate DDoS Attacks against the SDN Controller: Experimental Analysis” *Applied Sciences 13*, no. 2: 775, 2023.

[20] A.A. Bahashwan; M. Anbar ; S. Manickam ; T.A. Al-Amiedy ; M.A. Aladaileh; I.H. A. Hasbullah, Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-Defined Networking. *Sensors*, 23, 4441, 2023

[21] M. W. Nadeem; H. G. Goh; V. Ponnusamy ; Y. Aun, ”Ddos detection in sdn using machine learning techniques,” *Computers, Materials & Continua*, vol. 71, no.1, pp. 771–789, 2022

[22] F. Alanazi; K. Jambi; F. Eassa; M. Khemakhem; A. Basuhail et al., ”Ensemble deep learning models for mitigating ddos attack in a software-defined network,” *Intelligent Automation & Soft Computing*, vol. 33, no.2, pp. 923–938, 2022.

[23] E.S. Alghoson, O.Abbass, “ Detecting Distributed Denial of Service Attacks using Machine Learning Models ” *International Journal of Advanced Computer Science and Applications(IJACSA)*, Vol. 12, No. 12, 2021

[24] Jeba Praba. J and R. Sridaran, “An SDN-based Decision Tree Detection (DTD) Model for Detecting DDoS Attacks in Cloud Environment” *International Journal of Advanced Computer Science and Applications(IJACSA)* , 13(7), 2022.

[25] 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.

[26] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, “Machine-learning-based ddos attack detection using mutual information and random forest feature importance method,” *Symmetry*, vol. 14, no. 6, p. 1095, 2022.

[27] 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.

[28] K. Li, H. Zhou, Z. Tu, W. Wang, and H. Zhang, “Distributed network intrusion detection system in satellite-terrestrial integrated networks using federated learning,” *IEEE Access*, vol. 8, pp. 214852–214865, 2020.

[29] M.Shakil, F.Y. Mohammed A, R.Arul, AK.Bashir, JK.Choi. A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering. *Trans Emerging Tel Tech*, 2019;e3622, 2019

[30] [38] Al-Tashi, Q., Rais, H.M, Abdulkadir, S.J, Mirjalili, S.M, & Alhussian, H.S, ”A Review of Grey Wolf Optimizer-Based Feature Selection Methods for Classification. *Algorithms for Intelligent Systems*, 2019.

[31] A.G,Gad, Particle Swarm Optimization Algorithm and Its Applications: *A Systematic Review. Arch Computat Methods Eng 29*, p.p 2531–2561,2023.

[32] P. J. Gaidhane and M. J. Nigam, “A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems,”*J. Comput. Sci* , vol. 27(1), pp. 284–302, 2018.

[33] N. Singh and S. B. Singh, “Hybrid Algorithm of Particle Swarm Optimization and Grey Wolf Optimizer for Improving Convergence Performance,” *J. Appl. Math*, vol. 2017(1), pp. 1– 15, 2017.