

Article

# Adaptive Artificial Bee Colony Algorithm for Nature-Inspired Cyber Defense

Chirag Ganguli <sup>1,\*</sup>, Shishir Kumar Shandilya <sup>1,†</sup>, Maryna Nehrey <sup>2,†</sup> and Myroslav Havryliuk <sup>3,†</sup><sup>1</sup> Vellore Institute of Technology, VIT Bhopal University, Bhopal 466114, India<sup>2</sup> Department, National University of Life and Environmental Sciences of Ukraine, 03041 Kyiv, Ukraine<sup>3</sup> Department of Artificial Intelligence, Lviv Polytechnic National University, 79013 Lviv, Ukraine

\* Correspondence: chirag.ganguli2019@vitbhopal.ac.in

† These authors contributed equally to this work.

**Abstract:** With the significant growth of the cyber environment over recent years, defensive mechanisms against adversaries have become an important step in maintaining online safety. The adaptive defense mechanism is an evolving approach that, when combined with nature-inspired algorithms, allows users to effectively run a series of artificial intelligence-driven tests on their customized networks to detect normal and under attack behavior of the nodes or machines attached to the network. This includes a detailed analysis of the difference in the throughput, end-to-end delay, and packet delivery ratio of the nodes before and after an attack. In this paper, we compare the behavior and fitness of the nodes when nodes under a simulated attack are altered, aiding several nature-inspired cyber security-based adaptive defense mechanism approaches and achieving clear experimental results. The simulation results show the effectiveness of the fitness of the nodes and their differences through a specially crafted metric value defined using the network performance statistics and the actual throughput difference of the attacked node before and after the attack.

**Keywords:** nature-inspired cyber security (NICS); artificial bee colonization algorithm (ABC); adaptive defense; node-based fitness analysis



**Citation:** Ganguli, C.; Kumar Shandilya, S.; Nehrey, M.; Havryliuk M. Adaptive Artificial Bee Colony Algorithm for Nature-Inspired Cyber Defense. *Systems* **2023**, *11*, 27. <https://doi.org/10.3390/systems11010027>

Academic Editor: William T. Scherer

Received: 31 October 2022

Revised: 30 November 2022

Accepted: 20 December 2022

Published: 5 January 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The Internet has become an integral part of people's lives, and there is an enormous amount of incoming and outgoing traffic flow every day as people browse the World Wide Web for various needs. This has led to the growth of adversaries who tend to detect anomalies in their hosts and their networks and perform various operations to gain privileges that are equivalent to the real users. The adversaries referred to in this case are those who try to gain unauthorized access to the critical assets. Therefore, there is the utmost need for quick and advanced protection of these hosts and networks from adversaries. The first line of defense includes the detection of the anomaly in all the nodes of a network. Invasion of a network starts from a singular node which then spreads to infect the entire network. This can therefore be justified in the case of defensive action from the point an attack is detected to the mitigation stage. The proposed methodology in this paper revolves around the process of the efficient detection of malicious activity in a network and flagging the activity using nature-inspired algorithms.

Anomaly detection can be classified into supervised and unsupervised learning techniques, of which supervised techniques perform better than unsupervised techniques if the entire set of test cases contain all known types of attacks. However, this is unrealistic in the presence of ever-growing attacks. Non-linear methods, such as support vector machines, multilayer perceptrons, and heuristic-based detections, tend to outperform traditional learning techniques [1]. Modern applications require rigorous analysis of test data to filter out the outliers and anomalies present in order to maintain the system's reliability. These techniques significantly help in fraud detection, where analyzing and removing outliers

is a major factor. These anomalies in test data can be defined as observations that tend to differ notably from the known observations, such that it can be concluded that they were generated from a different process altogether. These anomalies (outliers) are very low in number as compared to the nominal test cases [2]. Accuracy is considered to be the key metric in the said case, as the detection and classification of system outliers is of the highest priority. The precision metric is not important in this scenario, as the false positives that may arise can be processed later on [3].

It is evident that there is large-scale usage of these aforementioned data-based detection algorithmic techniques in industrial processes, thereby making the detection of anomalies (outliers) for industrial process data very crucial [4]. The main disadvantage of supervised learning techniques is that they contain the break-in detection of unknown patterns often termed zero-day exploits, as they are not present in the learning phase of the algorithm. Threats that are present in information systems have become significantly intelligent and they can easily bypass basic security solutions (firewalls and anti-viruses). Anomaly-based intrusion detection systems allow this traffic and system call classification in normal and attack conditions and their efficiency greatly depends on the techniques that are used in these systems [5].

The artificial bee colonization algorithm is an optimization algorithm that is defined by the foraging behavior of honeybees. It is a member of the swarm intelligence algorithm category. This algorithm consists of two types of bees—employed and unemployed forager bees—performing their own optimization process to find an optimal food source keeping in consideration the distance of the discovered food source from its nest. As part of the optimization process, a bee can either abandon the source because of unmet criteria or continue foraging recursively unless an optimal source is found. The artificial bee colonization (ABC) algorithm is based solely on the foraging behavior of honeybees. This algorithm is only possible because of the high degree of organization of work that bee colonies exhibit, where thousands of bees work in a group to make critical decisions, create stability, etc. Several studies have shown that this aforementioned colony-level organization and decision-making rely greatly on individual bees performing their own tasks. For example, foraging bees perform a waggle dance, thereby recruiting other bees to the advantageous foraging locations in congruous densities [6]. It is worth noting that the artificial bee colony algorithm (ABC) performs better than other algorithms, such as population-based algorithms. ABC is a meta-heuristic algorithm, and it is, therefore, important to mention that collective animal behavior is part of animals' social interaction process, which is used for effective coordination and forms a major part of their working concept in larger teams. This communication stature is achieved using several processes, such as group motion, decision-making, aligned information passing, and synchronizing [7].

Nature-inspired cyber security is imposed on a security attack apparatus to define a new point of attack or to execute a defensive mechanism against an attack. The artificial bee colonization algorithm defines a metric-level optimization of the foraging behavior of the honeybees, where a swarm of honeybees can perform several operations by dividing itself into groups. To influence the anomaly detection defensive mechanism, artificial bee colony algorithms can be used to define the fitness of the nodes attached to a network and find the first or the nearest point (or points) to an intrusion that leads to an infection in the network. These nearest points can then be detached from the network, a backup of the nodes can be created, such that the attack does not spread, and the first point can be investigated for vulnerable services that might be operating on the node that can then be fixed to prevent similar intrusions in the future. Because the artificial bee colony algorithm is an optimization approach for solving this problem, it tends to perform the anomaly detection process in an optimized manner, thus preserving the efficiency of the network.

The primary research problem that is solved using the proposed algorithm concerns the early detection and prevention of attacks imposed by an adversary or group of adversaries on a specified system. Adaptive defense-based intrusion detection provides a mechanism for reducing the count of false positives in the detection phase of present-day firewall

systems. This paper provides a mechanism that performs detection of malicious packets while they pass through a network based on specified network properties and flags them based on the proposed fitness mechanism of the nodes before their signatures are evaluated by a firewall. The fitness criteria have been graphically displayed in Scenario 1 of the Results section and the fitness is further implemented to detect the probable set of malicious nodes that might be present in a network based on the packet hop count and network stability. This process is defined and presented in Scenario 2 of the Results section in this paper.

## 2. Related Work

AI-assisted Computer Network Operations testbed [8] provides an architecture for simulating the results of defense mechanisms on several network topologies. The scheme shows how nature-inspired cyber security can efficiently find solutions to provide adaptive security and exemption and flexibility in network operations. The nature-inspired cyber security (NICS) algorithm analyzed for fitness evaluation in this paper is based on Artificial Bees Colony algorithm.

In accordance with [9], the ABC algorithm can be improved to overcome the shortcomings of the actual algorithm. There can be several of the same for example slow convergence speed. Considering this problem of the slow convergence speed of the ABC algorithm and its falling in the local extreme, an improved ABC can be prepared using a uniform distribution point set that combines the number theory and crossover method. According to [10], attacks occurring in several distributed systems begin at the network layer. Defense against these attacks forms the core motive behind the self-adaptation of the defense mechanisms or adaptive defense.

The Artificial Bees Colonization algorithm has improved the overall accuracy rate for both known and unknown attacks as compared to the traditional methods according to [10,11]. Ref. [12] provides a brief analysis of the various approaches that have the property to enhance the reliability of Intrusion Detection Systems in light of accuracy and execution time. Refs. [13,14] uses the properties of the Artificial Bees Colony algorithm to highlight improved efficiency on modern detection systems, where ABC is proved to have an accuracy of 97.5% for known attack scenarios and 93.2% for unknown attack scenarios. Both of these papers use the very popular KDD Cup 99 dataset to measure the accuracy of this Nature Inspired algorithm. Ref. [15] uses the Random Neural Network and Gradient Descent algorithms to showcase the property of the ABC Inspired Swarm Optimization technique that has provided an overall accuracy rate of 95.02%.

This paper solely focuses on the property of selecting the correct features based on the fitness function of the Artificial Bees Colony algorithm which can be used further to detect intrusions and maintain network infrastructures to act in a certain way in times of an attack. Counting the total required features that must be enough to detect an intrusion forms an important step of these algorithms. Ref. [16] presents a firefly-based approach for intrusion detection that contributes to confirming that 10 features are the actual count that is enough for providing improved accuracy. Ref. [17] uses the Grey Wolf Optimization algorithm and the meta-heuristic algorithmic approach to gain an effective accuracy rate of 81% with an improved F1 score of 78%.

To elaborate more on the process of including advanced artificial intelligence mechanisms in cyber defense. This includes the applications of artificial neural networks in cyber defense areas [18,19] offer a broader approach to the concept of active defense and the frameworks and principles involved in the process to differentiate between the types of active defense. In order to apply these defense mechanisms to an active security model, the process must be optimized using infinite-time horizon optimal control for strategic defense [20,21]. Refs. [22,23] provides a strategic analysis and computational model for adaptive defense against advanced persistent threats that demonstrates the attackers' types and defenders' motives in issuing deception techniques in shaping the information symmetry. A survey by [24] signifies that the nature of security mechanisms has several limitations as potential attackers cannot be prevented in advance. This issue is addressed

in this paper, as it focuses on the early detection of an attack. Applications of computational intelligence-based technologies were addressed in [25] to highlight their roles in information security concerns.

### 3. Materials and Methods

#### 3.1. Research Problem

In modern-day firewall systems, when there is a large flow of data packets among several machines present in a network, the primary task is to match the packet properties with some pre-defined set of signatures that are regularly updated. This may include the packet filtering firewalls that filter the packets based on the Source and Destination IP and Ports. This would also apply to protection against malware. There might be some logging concerns in such firewalls as the detection is purely based on pre-defined properties and this might lead to a large section of false positives and false negatives, which could increase the investigation time and lead to an attacker gain access to critical data before they are even found and eliminated from the network.

To have a better solution to this existing problem by using nature-inspired cyber security techniques and adaptive defense, a property can be derived out of the three core network characteristics—average throughput, average end-to-end delay, and packet delivery ratio, that can have the capacity to have constant monitoring of the nodes present in the network and calculate their health while using minimum system resources, and detect probable malicious packets or attacked nodes and pass them to the modern day firewalls for signature evaluation. This could lower the packet count passing through the firewall, thus enabling early detection and lower false positives and negatives, improving the efficiency of the network and lower wait times before an attack is flagged and eliminated.

#### 3.2. Artificial Bees Colonization Algorithm (ABC)

The adaptive defense mechanism discussed in this paper is solely focused on combining Artificial Bees Colony algorithm as a nature-based algorithm and Intrusion Detection System and System Outlier Detection as the security mechanism, thus forming a whole new adaptive defense concept of nature-inspired cyber security. The mechanism used here includes a NICS-based testbed where several network topologies are measured in their throughput, end-to-end delay, and packet delivery ratio metric values. The metric defined in this scope is a combination of the above three metrics to form a whole new concept of nodes' actual fitness and probabilistic fitness ratio that can be used to determine the part of the route that has been affected during an intrusion and to what extent they contribute to the entire process.

There exist two types of bees in a bee colony—employed and unemployed bees. The employed bees are responsible for exploiting a food source whereas the unemployed bees include onlookers (shadow the employed bees and search for the food source based on the greedy selection approach) and scout bees (start searching around the nest spontaneously to confirm if the nearer food sources do form the optimal solution).

This process can be emulated in the case of cyber security where employed bees are the algorithmic function that sits on every node present in the network monitoring their functionality (exploiting a food source) which includes constantly calculating their throughput, end-to-end delay and packet delivery ratio. The unemployed bees are the algorithmic function that performs two operations: (i) keeping a record of the calculations of the previous module and comparing them with actual values to determine the presence of any malicious activity or lower throughput, increased end-to-end delay, and lower packet delivery ratio (onlooker bees—shadowing the employed bees); and (ii) confirm any abnormality in the network and report them based on the calculated function by onlooker bees module (scout bees).

### 3.3. Proposed Method

The proposed methodology based on this paper is the concept of the ABC algorithm which is modified to showcase the effect of its actual fitness function on the nodes attached to the network before and after an attack has been initiated on them. Based on the ABC algorithm, there is a set of solutions or nodes in this case that needs to be considered for the solution set.

To generate the solution set the following Equation (1) is used:

$$v_i = x_i + \phi_i * (x_i - x_j) \quad (1)$$

where  $v$  forms a member of the solution set,  $\phi$  can be a random number within  $[-1,1]$ . In our algorithm, we are using Gauss randomization to define the value of  $\phi$  and select the nodes under attack. Furthermore, we take the value of ' $x$ ' as the combination of average throughput, end-to-end delay, and packet delivery ratio. This combination from Equation (1) generates the following Equations (2)–(4).

$$v_i = \phi * ((ATP + (ATP - node['TP']) + (ATP + (AED - node['AED']) + (ATP + (PDR - node['PDR']))) \quad (2)$$

$$v_j = \phi * ((AED + (ATP - node['TP']) + (AED + (AED - node['AED']) + (AED + (PDR - node['PDR']))) \quad (3)$$

$$v_k = \phi * ((PDR + (ATP - node['TP']) + (PDR + (AED - node['AED']) + (PDR + (PDR - node['PDR']))) \quad (4)$$

The Equations (2)–(4) are calculated for every node from 0 to 7 and the solution set is generated as the result of the fitness function. In this equation,  $\phi$  is the random metric, ATP is the average throughput, AED is the average end-to-end delay, and PDR is the packet delivery ratio.

$$Fitness = \frac{fit_i}{fit_j} \quad (5)$$

where ' $i$ ' is the fitness value of the nodes in normal conditions and ' $j$ ' is the fitness value of the nodes in the attack condition if the fitness of nodes in normal conditions is greater than the fitness of nodes in the attack condition else vice versa.

The set of results generated using Equations (2)–(4) are examined under both normal and attack scenarios along the metric value on set  $v$  for each node present in the clusters on the network. The average of the fitness values is captured using Equation (5). These fitness values when dropped on the network structure provide an overview of the state of each node under attack conditions, thereby providing a representation of the difference in the node properties in each of the mentioned conditions.

Equation (5), defined in the ABC algorithm, can be modified to determine the fitness value of each node present in a network architecture.  $fit_i$  can be modified to be the under-attack parametric value, henceforth calculated for every node which is divided by the sum of the under-attack and normal parametric values for every node to obtain an average fitness value for the selected node under attack. This equation can be segregated by calculating the fitness under attack and fitness under normal conditions separated and taking their sum, pointing to the solution of whichever is having a greater value. It is evident that for our under-attack selected nodes that have been calculated in the previous equation will be having a greater value than the normal nodes. This, experimentally, seems to have a better metric value, accurate to the highest decimal. The proposed fitness function

is determined based on the primary factors involving average network throughput, end-to-end delay, and packet delivery ratio, therefore, a change in the fitness function of a node is linked directly with an attack scenario or the node is infected. If other factors influence the core network parameters, they will further be detected by firewall systems. In the current scenario, the probable attack set is early determined and processed to enable efficient network operations.

The Algorithm 1 provides an overview of calculating the metric value of the nodes connected to the network. This provides an updated approach to the Artificial Bees Colony algorithm that helps in measuring the fitness of the nodes based on the metric value. This metric value is calculated using the core network properties, throughput, end-to-end delay, and packet delivery ratio, using Equations (2)–(4).

---

**Algorithm 1:** Fitness measurement of ABC algorithm

---

**Input:** Target Function  
**Output:** Modified ABC Algorithm for calculating Node Fitness (Solution)

- 1: Initialize Nodes: 8 (Start: 0, End: 7)
- 2:  $Cluster \leftarrow$  Average Throughput,
- 3: Average E2E Delay, Average Packet Delivery Ratio
- 4: **for** nodes  $\leftarrow$  0 **to** 7 **do**
- 5: normalSelectSet = checkNode(Normal\_Output(nodes),  
AverageThroughput(normal\_nodes), AverageED(normal\_nodes),  
AveragePDR(normal\_nodes), start, end)
- 6: attackSet = checkNode(Attack\_Output(nodes),  
AverageThroughput(attack\_nodes), AverageED(attack\_nodes),  
AveragePDR(attack\_nodes), start, end)
- 7:  $fitness \leftarrow$  checkFitness((fitness\_attack + fitness\_normal)/2)
- 8: fitnessSum += fitness
- end**
- 9: Dump the metric value of checkNode
- 10: fitnessAvg = fitnessSum / Nodes[8]
- 11: checkNode():
- 12:  $randomMetric \leftarrow$  gauss(-1, 1)
- 13: **for**  $i \leftarrow$  start **to** end **do**
- 14: Append Parametric Values
- 15:  $v1 = \phi * ((ATP + (ATP - node['TP']) + (ATP + (AED - node['AED']) + (ATP - node['PDR'])))$
- 16:  $v2 = \phi * ((AED + (ATP - node['TP']) + (AED + (AED - node['AED']) + (AED - node['PDR'])))$
- 17:  $v3 = \phi * ((PDR + (ATP - node['TP']) + (PDR + (AED - node['AED']) + (PDR - node['PDR'])))$
- end**
- 18: Find Minimum of  $v1$  value for each node [ $min_v$ ]
- 19: Find Maximum of  $v2$  and  $v3$  value for each node [ $max_v$ ]
- 20: Calculate  $param\_avg = ((min_v + max_v) / (node\_to\_check))$
- 21: checkFitness():
- 22: **for**  $i \leftarrow$  start **to** end **do**
- 23: fitness += (attackSet / ((fitness\_attack + fitness\_normal)/2))
- end**

---

### 3.4. Experimental Setup

The network architecture used in this paper is derived from [8]. The architecture diagram is displayed in Figure 1.

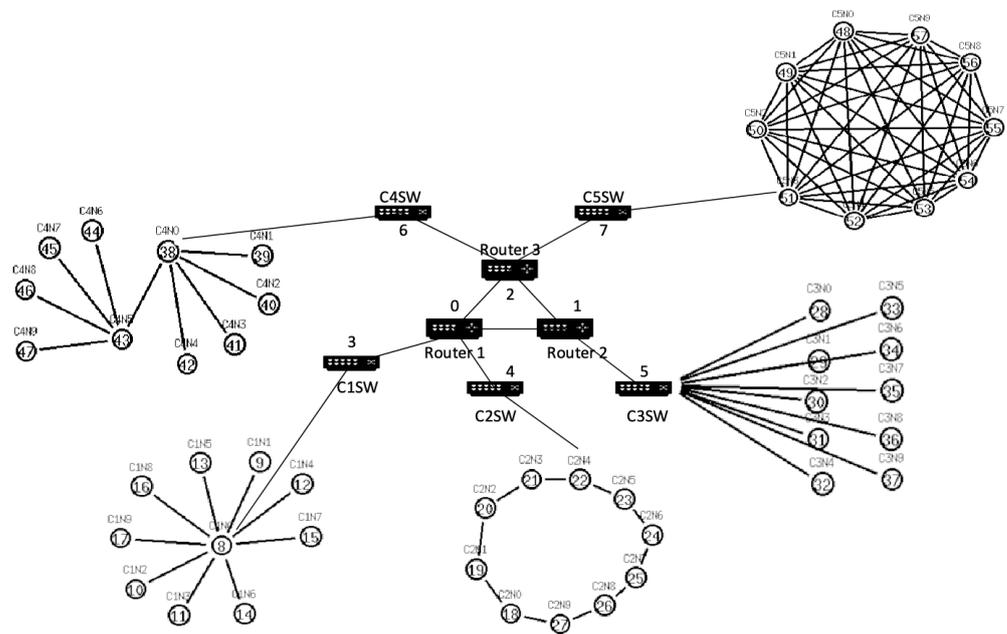


Figure 1. Network architecture [8].

In Figure 1, there exist 3 routers (R1, R2, R3) connected with one another. They are, in turn, connected to the 5 clusters each connected in different topologies, such as star, ring, tree, and mesh.

Each cluster in Table 1 is connected to 10 nodes. To define the attack consequences in the above architecture, there is a malicious node that is attached to each of the clusters one by one, and the results are calculated based on their default and modified metric values.

Table 1. Cluster Information.

Cluster Name	Network Device
R1—Node 0	Router (Router 1)
R2—Node 1	Router (Router 2)
R3—Node 2	Router (Router 3)
SW1—Node 3	Switch (Cluster 1)
SW2—Node 4	Switch (Cluster 2)
SW3—Node 5	Switch (Cluster 3)
SW4—Node 6	Switch (Cluster 4)
SW5—Node 7	Switch (Cluster 5)

#### 4. Results

The presented scenarios include the Distributed Denial of Service attack as a criterion for detecting malicious activity in the defined network. The proposed Nature Inspired algorithm is a modified version that is enhanced with network parameters, such as throughput, end-to-end delay, and packet delivery ratio, to determine the working of the nodes based on the flow of the packets. This concept is implemented to demonstrate the proposed health function in Scenario 1 to define the effective detection of the presence of malicious nodes based on the packet flow. To further supplement the proposed methodology, Scenario 2 is provided with a metric that implements the health function to calculate the nodes' fitness ratio and generate a probable set of malicious nodes based on the properties defined out of packet delivery rate. This research proposal can also be extended to other attack scenarios other than the Denial of Service attack by modifying the proposed equations. However, this paper uses DoS as it focuses on the availability clause of the nodes present in the network, such that the nodes remain active under attack scenarios while the affected nodes are evaluated and the attack is mitigated.

4.1. Scenario 1: Demonstration of the Fitness Function

Using the proposed methodology, this paper demonstrates results by attaching the malicious node to individual clusters randomly, to execute a low-rate Denial of Service (DoS) on the network. In this experiment, the network comprises five clusters and three routers. In a larger corporate network, there could be multiples of such clusters and nodes attached and this algorithm could help to measure the fitness of each node attached by implementing the Artificial Bees Colony algorithm. This fitness value can provide an overview of the health of each node before and after an attack. The malicious traffic flow is rate-limited to 5 MB, in this case to substantiate the fact of the low-rate Denial of Service concept.

In Figures 2, 4, 6, 8 and 10, we are not getting to the defensive part of the analysis, instead, we are focusing on how the metric value itself affects the nodes connected to the network. This metric is based on the Artificial Bees Colony algorithm and it continuously measures the fitness of each node as traffic flows through them. ‘Green dot’ signifies the working condition of the node and ‘Red dot’ means that the specific node is failing. When the metric falls below 0, it means the node has stopped working completely. The ‘blue line’ signifies the best fit possible based on the defined metric as displayed in Algorithm 1 which defines the best points between a node being in an active state and it failing under an attack condition.

- In Case 1: when the malicious node is connected to Cluster 1 Node 9, whose traffic is supposed to flow to Cluster 5 Node 9 shows a metric composition as displayed in Figure 2 with fitness as in Figure 3.

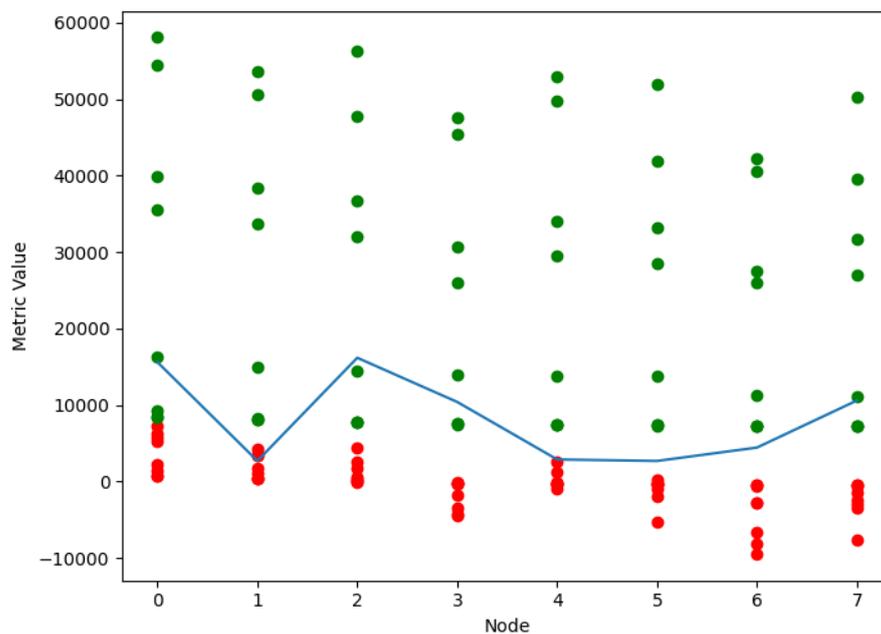


Figure 2. Case 1: metric composition.

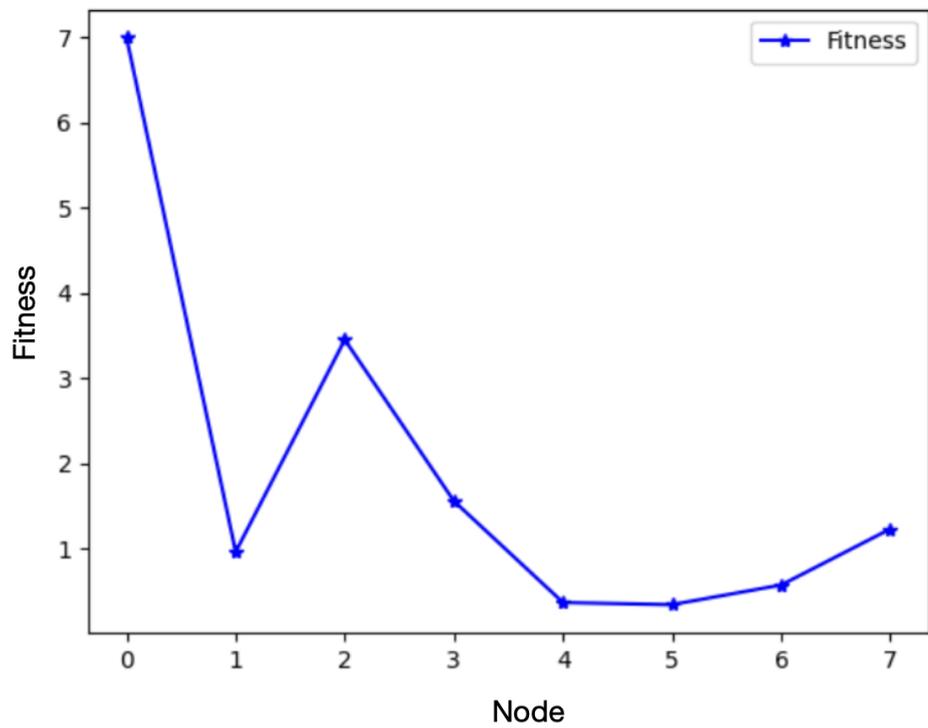


Figure 3. Case 1: fitness.

- In Case 2: when a malicious node is connected to Cluster 2 Node 9, whose traffic is supposed to flow to Cluster 4 Node 9 shows a metric composition as displayed in Figure 4 with fitness as in Figure 5.

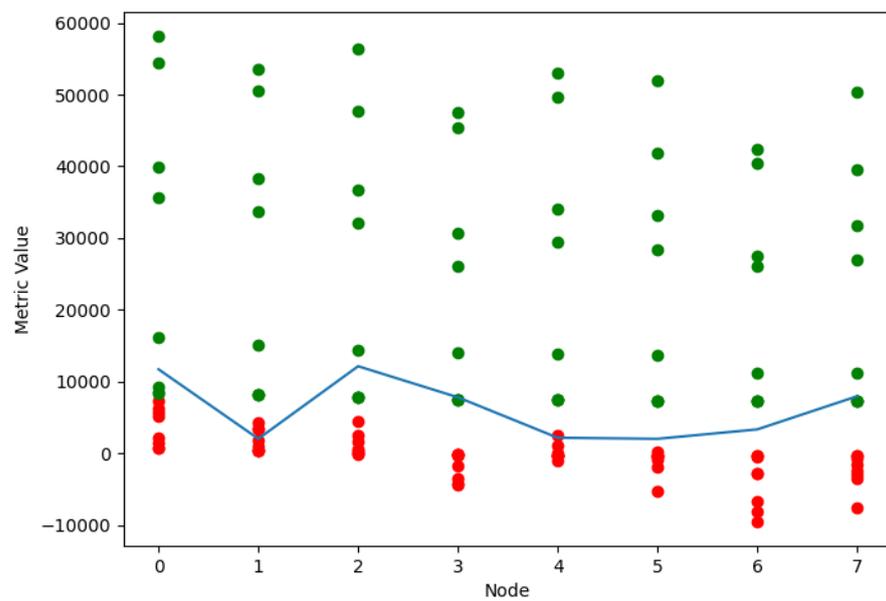


Figure 4. Case 2: metric composition.

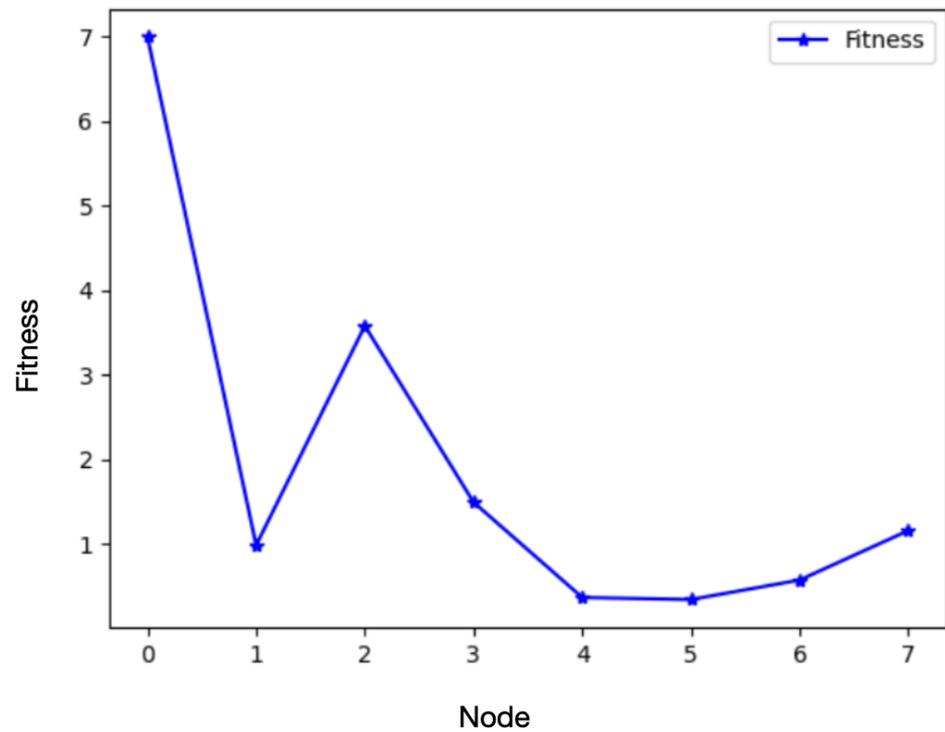


Figure 5. Case 2: fitness.

- In Case 3: when the malicious node is connected to Cluster 3 Node 4, whose traffic is supposed to flow to Cluster 1 Node 9 shows a metric composition as displayed in Figure 6 with fitness as in Figure 7.

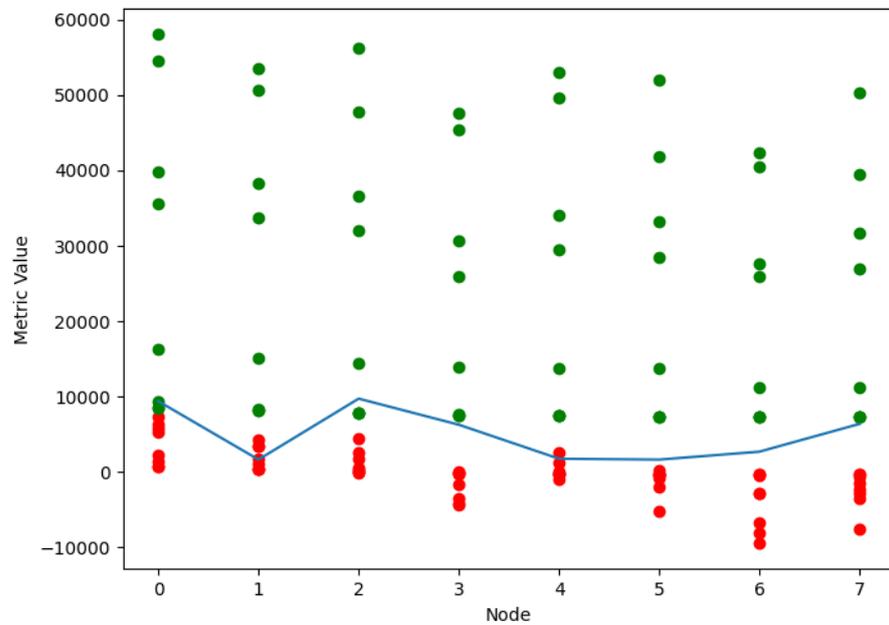


Figure 6. Case 3: metric composition.

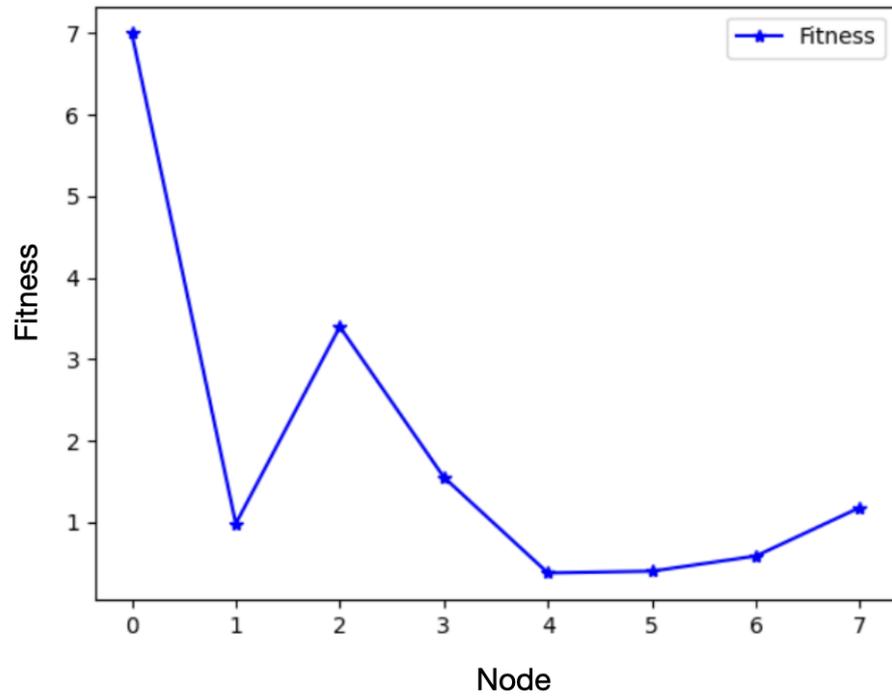


Figure 7. Case 3: fitness.

- In Case 4: when the malicious node is connected to Cluster 4 Node 6, whose traffic is supposed to flow to Cluster 3 Node 9 shows a metric composition as displayed in Figure 8 with fitness as in Figure 9.

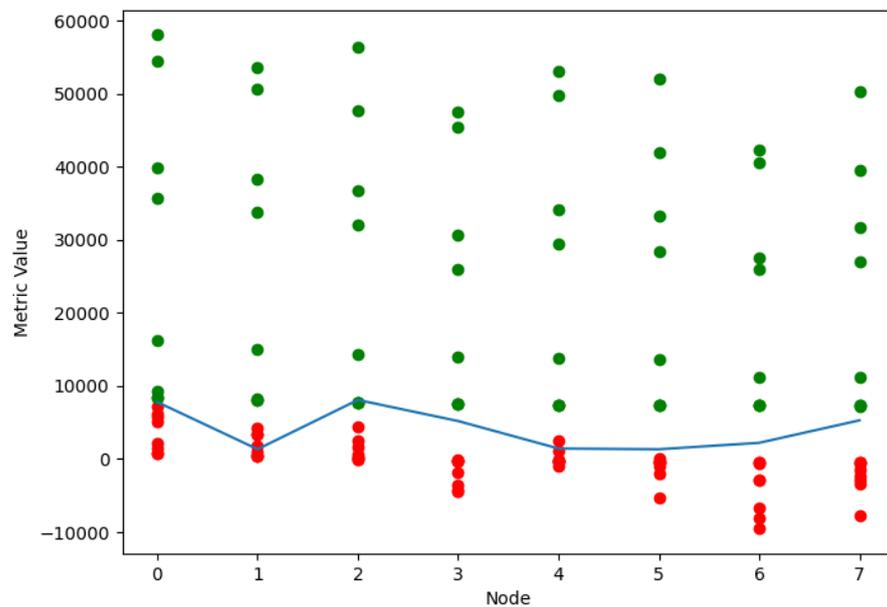


Figure 8. Case 4: metric composition.

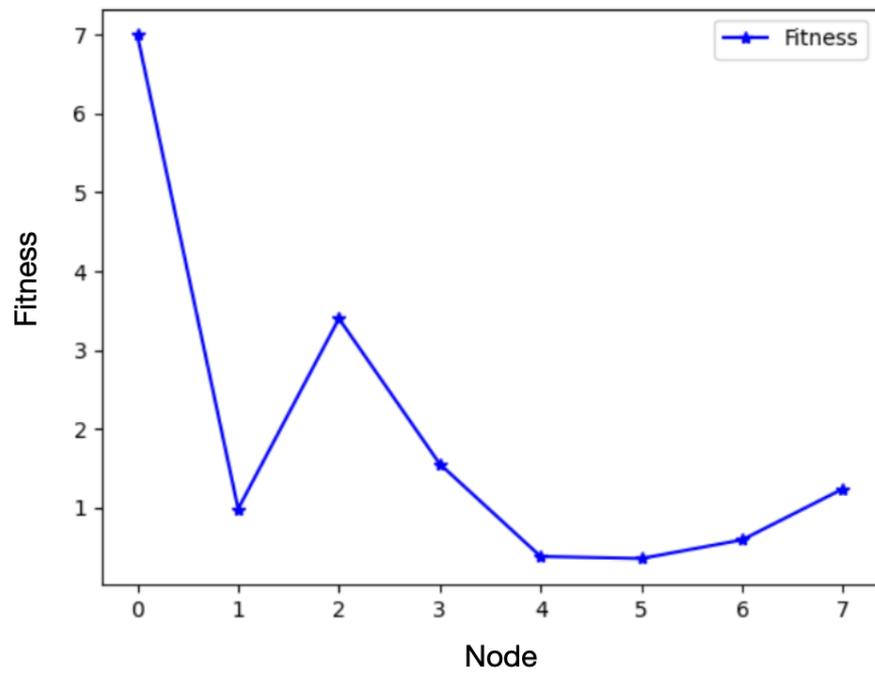


Figure 9. Case 4: fitness.

- In Case 5: when the malicious node is connected to Cluster 5 Node 9, whose traffic is supposed to flow to Cluster 1 shows a metric composition as displayed in Figure 10 with fitness as in Figure 11.

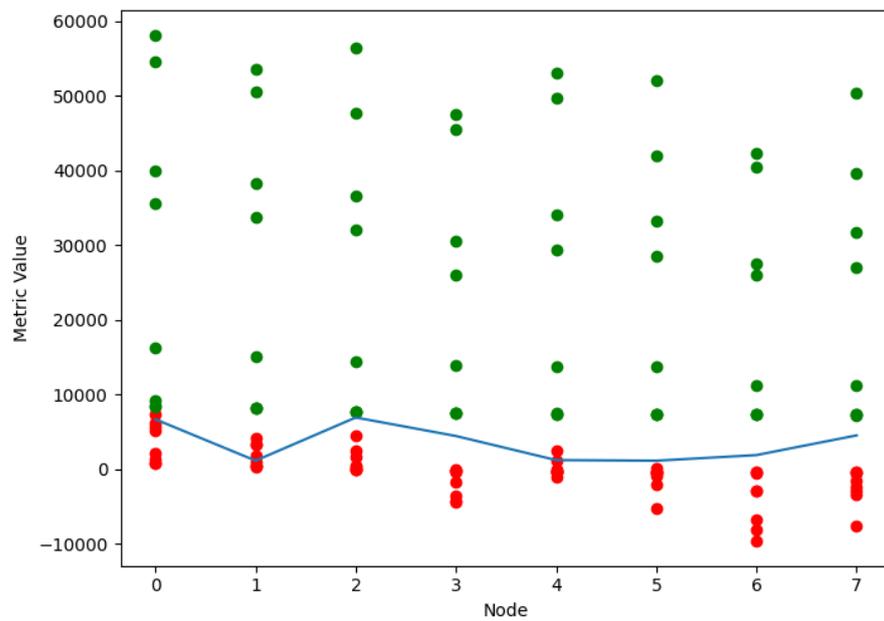


Figure 10. Case 5: metric composition.

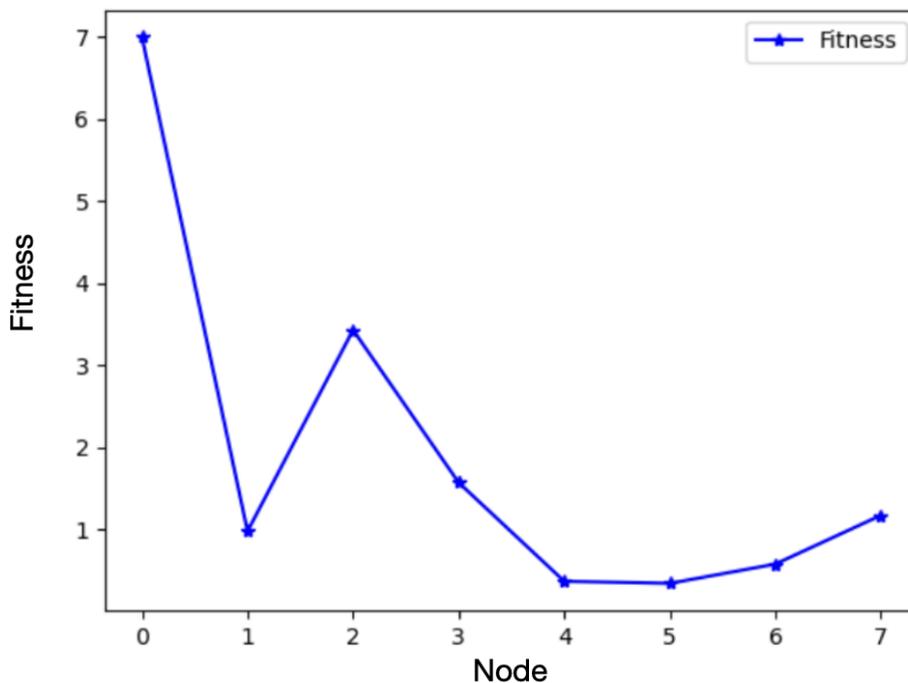


Figure 11. Case 5: fitness.

The fitness graph generated in the above case has a very low difference in their values because of the low-rate DoS traffic flow from the attached malicious node. The graph generated in the above cases is the zoomed visual of the parametric graph with Node on the X axis and metric value on the Y axis. To substantiate the above result, the values of the average fitness of the nodes in the above case are attached in Table 2. Upon a closer look at the metric composition graphs, it can be stated that the descent of these graphs gradually falls based on the topology of the clusters that are attached to the network which indicates the spread of an attack based on various network topologies.

Table 2. Fitness Chart.

No.	Malicious Node	Avg Fitness
1	Cluster 1-Node 9	1.9378411722984015
2	Cluster 2-Node 9	1.9398358282325272
3	Cluster 3-Node 4	1.9394194911657021
4	Cluster 4-Node 6	1.9420874115930948
5	Cluster 5-Node 9	1.9384687410596444

We can see that the average fitness of the nodes shows a significant accuracy to the third decimal. Node-wise values can also be generated individually and have been used in this paper to formulate the fitness graph, as shown in the figures above.

4.2. Scenario 2: Metric for Intrusion Detection and Fitness Evaluation

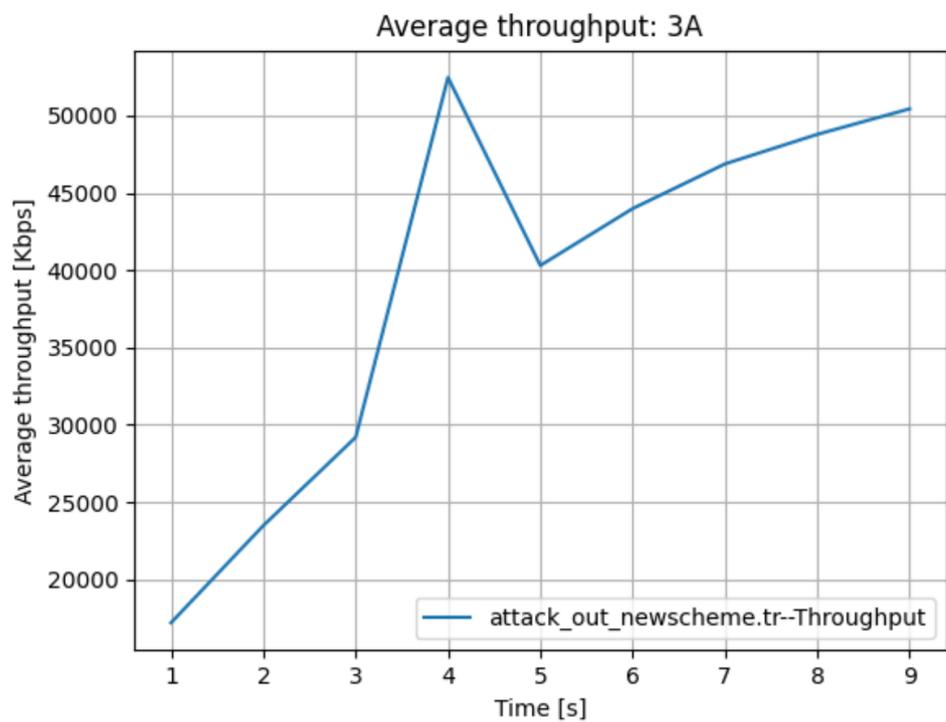
To support the algorithmic progress of the proposed algorithm, this paper also uses a second scenario. In this scenario, the number of nodes connected to each cluster is increased to 15 in place of the 10, as described in Scenario 1. The five new nodes introduced in each cluster are connected to the previous nodes to maintain the original topology of the network.

The results and observations on the new network were further extended from displaying metric composition to applying them on a cluster in order to measure the possibility of intrusions is added in Table 3:

**Table 3.** Affected nodes chart.

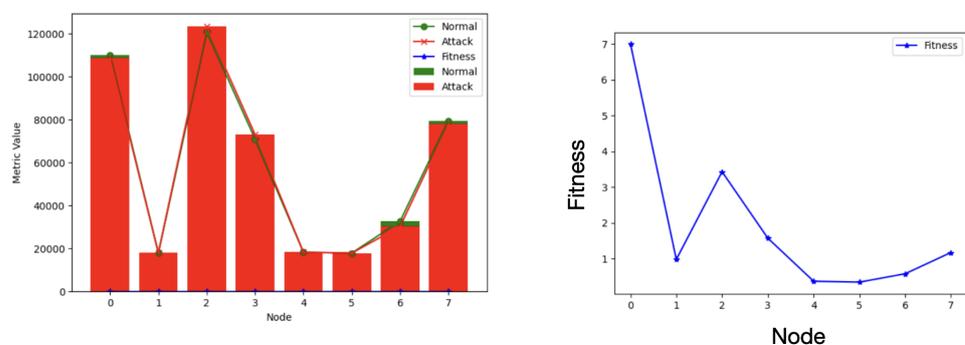
No.	Malicious Node at	Affected	Type
1	Cluster 1: Node 14	Nodes 2, 3	R3, SW1
2	Cluster 2: Node 12	Node 2	R3
3	Cluster 5: Node 10	Node 3	SW1

- In Case 1: when the malicious node is connected to Cluster 1 Node 14 whose traffic flows to Cluster 1 Node 0, so the traffic flows through R3 and Switch 1 to reach Cluster 5 Node 14
  - The average throughput under attack generated in this case is as shown in Figure 12.



**Figure 12.** Case 1: average throughput.

- The parametric value and fitness graphs can be seen below in Figure 13.



**Figure 13.** Case 1: intrusion chart and fitness.

- In Case 2: when the malicious node is attached to Cluster 2 Node 12 which is connected to Cluster 2 Node 0 but since no traffic is allotted to flow through Cluster 2 Node 12, on the connection of the malicious node, there is no significant denial of service attack

that is applicable on the network but on traffic flow through Router 3, it is affected by the attack.

- The average throughput under attack generated in this case is as shown in Figure 14.

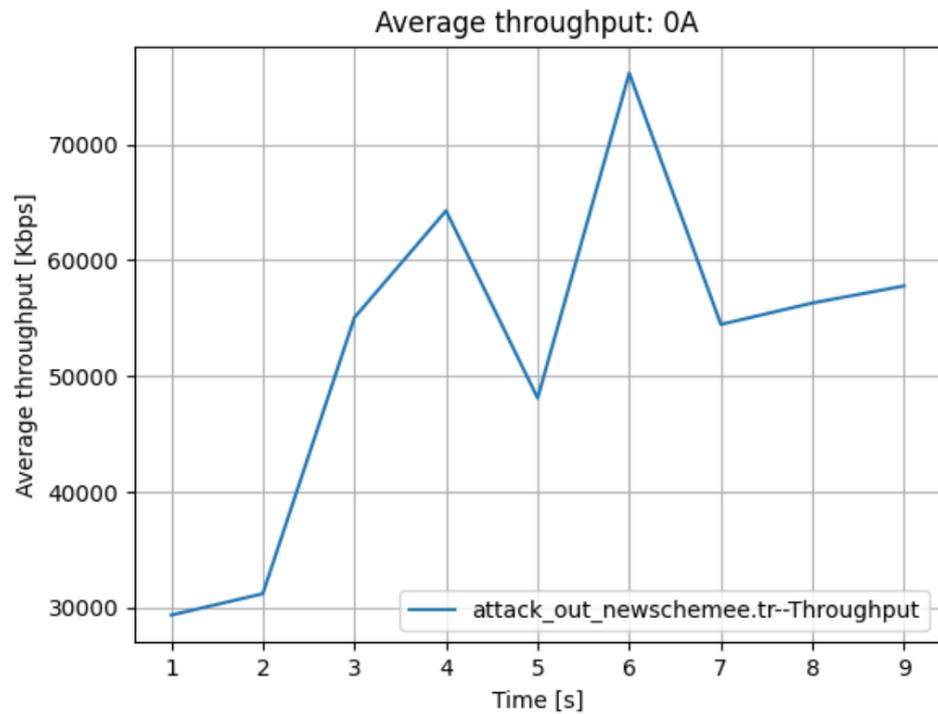


Figure 14. Case 2: average throughput.

- The parametric value and fitness graphs can be seen below in Figure 15.

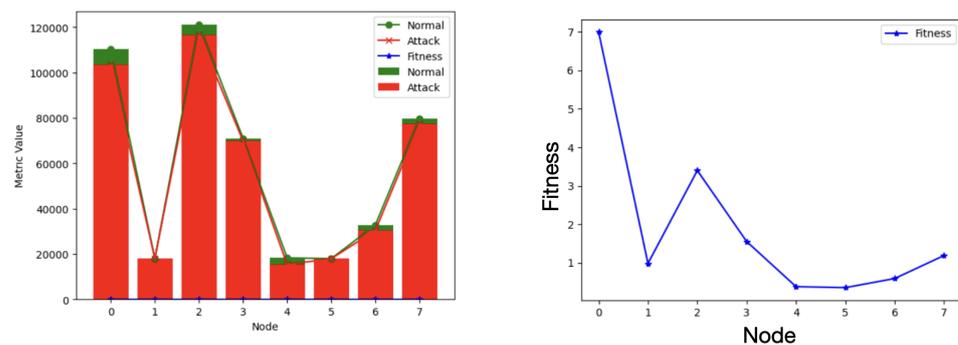


Figure 15. Case 2: intrusion chart and fitness.

- In Case 3: when the malicious node is connected to Cluster 5 Node 10, which is directly attached to Cluster 5 Node 11 in a mesh network, therefore on successful traffic flow, Switch 1 is affected the flow as all the traffic from Cluster 5 moves through Switch 1 to Cluster 1 so, in accordance to the proposed algorithm Switch 1 seems to be affected the most.
  - The average throughput under attack generated in this case is as shown in Figure 16.

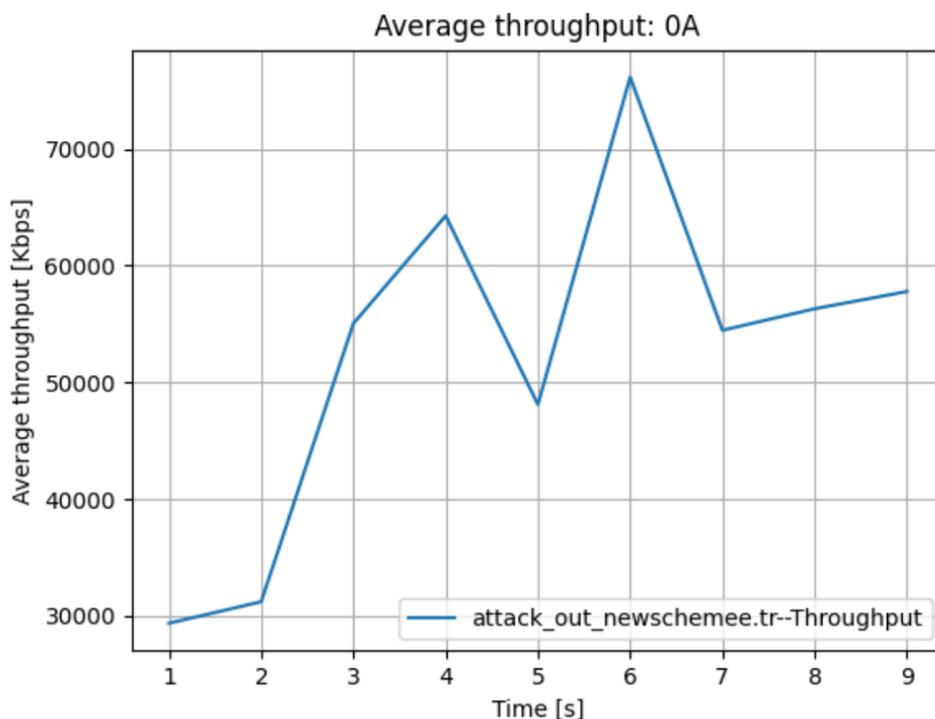


Figure 16. Case 3: average throughput

- The parametric value and fitness graphs can be seen below in Figure 17.

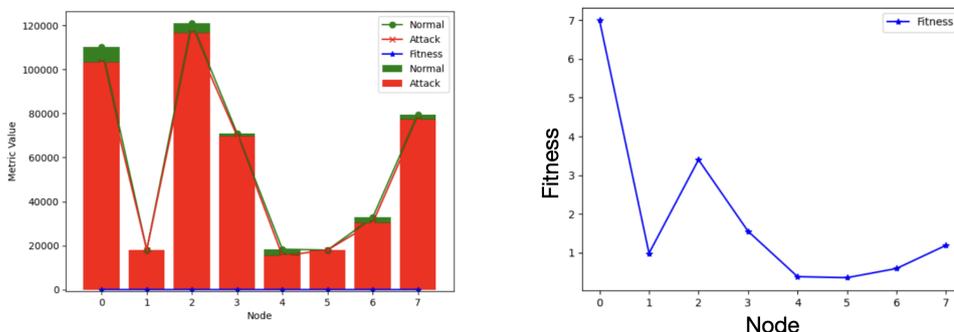


Figure 17. Case 3: intrusion chart and fitness.

The affected nodes, as tabulated above, show a slight deviation due to the introduction of the nodes and the change of the traffic flow to and from the newly introduced nodes, therefore supporting the proposed algorithm to be operational in any defined network topology and structure.

With respect to the proposed algorithm inspired by the concept of nature-inspired cyber security, it is visible that a partially modified nature-based ABC algorithm could help in optimizing a complex problem of searching affected nodes when a network is under attack. This statement is verified using five random test cases implemented on a simulated network collected from an adaptive defense testbed [8].

### 5. Discussion

This paper is presented to show the reaction of the nodes present in a network of a certain number of clusters constituting different network topologies and running different applications on the activities performed by malicious traffic flowing through one of the nodes that might be attached physically to the network or wirelessly through the network

modem. The base paper [8] provided a well-defined testbed to define the performance of the network throughput under normal conditions and attack conditions.

This paper has extended the working methodology of [8] by not only displaying the throughput difference in the clusters present in the network before and after the attack but also showing the attack surface which includes the nodes nearest to the point of attack or a path of attack traversal based on the topology of the network and the traversal speed of the malicious traffic from the infected node. Furthermore, this paper demonstrates a graphical representation of the distance of the nodes present in the network from the attack surface based on the traffic hop count as defined by the metric used as an adaptive defense technique which is a combination of nature-inspired Artificial Bees Colonization algorithm and Intrusion Detection System algorithm.

Therefore, this can help in narrowing the attack surface and isolate the infected nodes present in a network containing a huge number of nodes, thus greatly reducing the incident response time when under attack.

This work can be further extended to a nature-inspired network intrusion prevention system. Since the algorithm defined in this paper is based on an optimization algorithm it detects a search approach as is defined by the Artificial Bees Colony algorithm but to prevent the traffic from flowing to the nodes of the network, there must be an individual algorithm running in the network which is going to perform a sanity check on the traffic for each node before it leaves it such that as soon as the traffic is identified as being malicious, it would block the traffic then and there and not allow the traffic to flow through the network.

The proposed algorithm can be used as a prelim stage of the network intrusion prevention system, to optimally choose and select the malicious node and this can then be followed by a node-based monitoring algorithm to stop traffic from the malicious node. This could also be devised by a nature-inspired algorithm, which would make the algorithms go hand-in-hand and perform significantly better as compared to native network intrusion detection systems.

## 6. Conclusions

This paper solely focuses on the concept of the nature-inspired cyber security algorithm in the security operations of network intrusion detection systems, which are used to detect the attack surface or the infected nodes that are present in the network. The concept proposed in this paper is different from the native intrusion detection systems as it is centered around the Artificial Bees Colony algorithm, which is an optimization algorithm showing the foraging behavior of the bees, which includes the employed bees to be associated with a specific food source, the onlooker bees to keep an eye on the employed bees to choose an optimal food source and the scout bees to search for a food source randomly.

The proposed method is able to accurately estimate the fitness of each node by observing and analyzing the throughput, end-to-end delay, and packet delivery ratio with respect to the entire network traffic. The proposed method provides an efficient method to early identify malicious nodes and facilitates quick response by intrusion detection or prevention system.

**Author Contributions:** Conceptualization, C.G., S.K.S., M.N. and M.H.; Methodology, C.G. and M.H.; Validation, C.G., S.K.S., M.N. and M.H.; Investigation, C.G. and M.H.; Data Curation, C.G., S.K.S., M.N. and M.H.; writing—original draft preparation, S.K.S. and C.G.; writing—review and editing, C.G., S.K.S.; Visualization, C.G., S.K.S., M.N. and M.H.; Supervision, S.K.S. and M.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The datasets generated during the current study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

### Abbreviations

The following abbreviations are used in this manuscript:

NICS	Nature-Inspired Cyber Security
ABC	Artificial Bee Colonization Algorithm
ATP	Average Throughput
TP	Throughput
AED	Average End-to-End Delay
PDR	Packet Delivery Ratio
DOS	Denial of Service

### References

- Omar, S.; Ngadi, M.; Jebur, H.; Benqdara, S. Machine Learning Techniques for Anomaly Detection An Overview. *Int. J. Comput. Appl.* **2013**, *79*, 2. [CrossRef]
- Domingues, R.; Filippone, M.; Zouaoui, J. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognit.* **2017**, *74*, 406–421. [CrossRef]
- Hodge, V.; Austin, J. An Evaluation of Classification and Outlier Detection Algorithms. *arXiv* **2018**, arXiv:1805.00811.
- Wang, B.; Mao, Z. Outlier Detection Based on Gaussian Process with Application to Industrial Processes. *Appl. Soft Comput.* **2019**, *76*, 505–516. [CrossRef]
- Meira, J.; Andrade, R.; Praça, I.; Carneiro, J.; Bolón-Canedo, V.; Alonso-Betanzos, A.; Marreiros, G. Performance evaluation of unsupervised techniques in cyber-attack anomaly detection. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 4477–4489. [CrossRef]
- Montovan, K.; Karst, N.; Jones, L.; Seeley, T. Local behavioral rules sustain the cell allocation pattern in the combs of honey bee colonies (*Apis mellifera*). *J. Theor. Biol.* **2013**, *336*, 75–86. [CrossRef] [PubMed]
- Ramsey, M.T.; Bencsik, M.; Newton, M.; Reyes, M.; Pioz, M.; Crauser, D.; Simon-Delso, N.; Le Conte, Y. The prediction of swarming in honeybee colonies using vibrational spectra. *Sci. Rep.* **2020**, *10*, 9798. [CrossRef] [PubMed]
- Shandilya, S.K.; Upadhyay, S.; Kumar, A.; Nagar, A. AI-assisted Computer Network Operations testbed for Nature-Inspired Cyber Security based adaptive defense simulation and analysis. *Future Gener. Comput. Syst.* **2021**, *127*, 297–308. [CrossRef]
- Zhang, P.; Liu, Y. Application of An Improved Artificial Bee Colony Algorithm. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *634*, 012056. [CrossRef]
- Atighetchi, M.; Pal, P.; Webber, F.; Jones, C. Adaptive Use of Network-Centric Mechanisms in Cyber-Defense. In Proceedings of the Sixth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing, Hokkaido, Japan, 14–16 May 2003; pp. 183–192. [CrossRef]
- Soliman, O.S.; Rassem, A. A Network Intrusions Detection System based on a Quantum Bio Inspired Algorithm. *arXiv* **2014**, arXiv:1405.1404.
- Bangui, H.; Buhnova, B. Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms. *Comput. Electr. Eng.* **2022**, *100*, 107901. [CrossRef]
- Aldwairi, M.; Khamayseh, Y.; Al-Masri, M. Application of artificial bee colony for intrusion detection systems. *Secur. Commun. Netw.* **2012**, *8*, 2730–2740. [CrossRef]
- Celik, M.; Kurban, R.; Kurban, T. Artificial Bee Colony Algorithm for Anomaly Based Intrusion Detection. 2021. Available online: [https://www.researchgate.net/profile/Rifat-Kurban/publication/356854870\\_Artificial\\_Bee\\_Colony\\_Algorithm\\_for\\_Anomaly\\_Based\\_Intrusion\\_Detection/links/61b0a7371a5f480388c19525/Artificial-Bee-Colony-Algorithm-for-Anomaly-Based-Intrusion-Detection.pdf](https://www.researchgate.net/profile/Rifat-Kurban/publication/356854870_Artificial_Bee_Colony_Algorithm_for_Anomaly_Based_Intrusion_Detection/links/61b0a7371a5f480388c19525/Artificial-Bee-Colony-Algorithm-for-Anomaly-Based-Intrusion-Detection.pdf) (accessed on 19 December 2022).
- Qureshi, A.; Larijani, H.; Mtetwa, N.; Javed, A.; Ahmad, J. RNN-ABC: A New Swarm Optimization Based Technique for Anomaly Detection. *Computers* **2019**, *8*, 59. [CrossRef]
- Selvakumar, B.; Muneeswaran, K. Firefly algorithm based Feature Selection for Network Intrusion Detection. *Comput. Secur.* **2018**, *81*, 148–155. [CrossRef]
- Alzaqebah, A.; Aljarah, I.; Al-Kadi, O.; Damasevicius, R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. *Mathematics* **2022**, *10*, 999. [CrossRef]
- Tyugu, E. Artificial intelligence in cyber defense. In Proceedings of the 2011 3rd International Conference on Cyber Conflict, Tallinn, Estonia, 31 May–3 June 2011; pp. 1–11.
- Denning, D.E. Framework and principles for active cyber defense. *Comput. Secur.* **2014**, *40*, 108–113. [CrossRef]
- Lu, W.; Xu, S.; Yi, X. Optimizing Active Cyber Defense. In Proceedings of the Decision and Game Theory for Security, Fort Worth, TX, USA, 11–12 November 2013; Das, S.K., Nita-Rotaru, C., Kantarcioglu, M., Eds.; Springer International Publishing: Cham, Switzerland, 2013; pp. 206–225.
- Tirenin, W.; Faatz, D. A concept for strategic cyber defense. In Proceedings of the MILCOM 1999, IEEE Military Communications Conference Proceedings (Cat. No.99CH36341), Atlantic City, NJ, USA, 31 October–3 November 1999; Volume 1, pp. 458–463. [CrossRef]

22. Huang, L.; Zhu, Q. Analysis and Computation of Adaptive Defense Strategies Against Advanced Persistent Threats for Cyber-Physical Systems. In Proceedings of the Decision and Game Theory for Security, Seattle, WA, USA, 29–31 October 2018; Bushnell, L., Poovendran, R., Başar, T., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 205–226.
23. Tambe, M. *Workshop on Adaptive Defense in the Cyber-Security Domain*; Technical Report; University of Southern California: Los Angeles, CA, USA, 2016.
24. Cho, J.H.; Sharma, D.P.; Alavizadeh, H.; Yoon, S.; Ben-Asher, N.; Moore, T.J.; Kim, D.S.; Lim, H.; Nelson, F.F. Toward Proactive, Adaptive Defense: A Survey on Moving Target Defense. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 709–745. [[CrossRef](#)]
25. Dasgupta, D. Computational intelligence in cyber security. In Proceedings of the 2006 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, Alexandria, VA, USA, 16–17 October 2006; pp. 2–3. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.