





Article

Modified Firefly Optimization Algorithm-Based IDS for Nature-Inspired Cybersecurity

Shishir Kumar Shandilya ¹, Bong Jun Choi ^{2,*}, Ajit Kumar ² and Saket Upadhyay ¹¹ Vellore Institute of Technology, VIT Bhopal University, Bhopal 466114, India² School of Computer Science and Engineering, Soongsil University, Seoul 06978, Republic of Korea

* Correspondence: davidchoi@soongsil.ac.kr

Abstract: The new paradigm of nature-inspired cybersecurity can establish a robust defense by utilizing well-established nature-inspired computing algorithms to analyze networks and act quickly. The existing research focuses primarily on the efficient selection of features for quick and optimized detection rates using firefly and other nature-inspired optimization techniques. However, selecting the most appropriate features may be specific to the network, and a different set of features may work better than the selected one. Therefore, there is a need for a generalized pre-processing step based on the standard network monitoring parameters for the early detection of suspicious nodes before applying feature-based or any other type of monitoring. This paper proposes a modified version of the firefly optimization algorithm to effectively monitor the network by introducing a novel health function for the early detection of suspicious nodes. We implement event management schemes based on the proposed algorithm and optimize the observation priority list based on a genetic evolution algorithm for real-time events in the network. The obtained simulation results demonstrate the effectiveness of the proposed algorithm under various attack scenarios. In addition, the results indicate that the proposed method reduces approximately 60–80% of the number of suspicious nodes while increasing the turnaround time by only approximately 1–2%. The proposed method also focuses specifically on accurate network health monitoring to protect the network proactively.

Keywords: adaptive defense; nature-inspired cybersecurity; firefly algorithm, information security; early intrusion detection



Citation: Shandilya, S.K.; Choi, B.J.; Kumar, A.; Upadhyay, S. Modified Firefly Optimization Algorithm-Based IDS for Nature-Inspired Cybersecurity. *Processes* **2023**, *11*, 715. <https://doi.org/10.3390/pr11030715>

Academic Editor: Olympia Roeva

Received: 24 January 2023

Revised: 17 February 2023

Accepted: 22 February 2023

Published: 28 February 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

Nature-inspired cybersecurity (NICS) focuses on maintaining a network's trustworthiness and security by mimicking nature's processes, behaviors, and phenomena. As nature-inspired computing is fundamentally tolerant of incompleteness and vague data, NICS is expected to exhibit the same ability to produce robust cyber resilience. An intrusion detection system (IDS) ensures security by constantly monitoring unusual activities that may lead to an attack. IDS utilizes models of known attacks to analyze new and unknown scenarios [1]. Conventional IDSs are based on a feature selection approach to identify and classify malicious activities, which provides moderate security in the era of the rapid development of intrusion methods. There are two major types of IDSs: host-based intrusion detection systems (HIDSs) and network-based intrusion detection systems (NIDSs). NIDS is based on network traffic patterns that are used for the prediction of possible attacks. In addition, several researchers have adopted hybrid IDS, which combines signature- and activity-based methods to identify attacks while accurately avoiding false positives. This hybrid IDS approach is currently amalgamated with artificial intelligence (AI), machine learning (ML), and data mining techniques to improve accuracy and resilience. K-means and firefly optimization algorithms are the most common methods used in IDS [2] for feature selection, and NSL-KDD and KDD Cup'99 are the datasets used in these studies [3]. In feature selection methods, available feature datasets are cleaned and processed via various

classifiers, such as support vector machines (SVMs) and Bayesian network classifiers, to attain high accuracy at a low false-positive rate [4].

Existing works related to IDSs and fireflies are limited to feature selection [5], and they provide optimized features to machine learning and artificial intelligence-based systems. These approaches do not adapt to changing environments. They are relatively inefficient because they can be resource heavy and decrease the expected performance of the overall network by considering all nodes for scanning and detection. In contrast, the proposed method adopts the firefly algorithm for the early detection of suspicious nodes in a given network, which reduces the burden of further processing and analysis for IDS.

Figure 1 compares the use of the firefly algorithm in the existing approaches and the proposed approach, where the placements of both approaches in the network intrusion detection pipeline are highlighted.

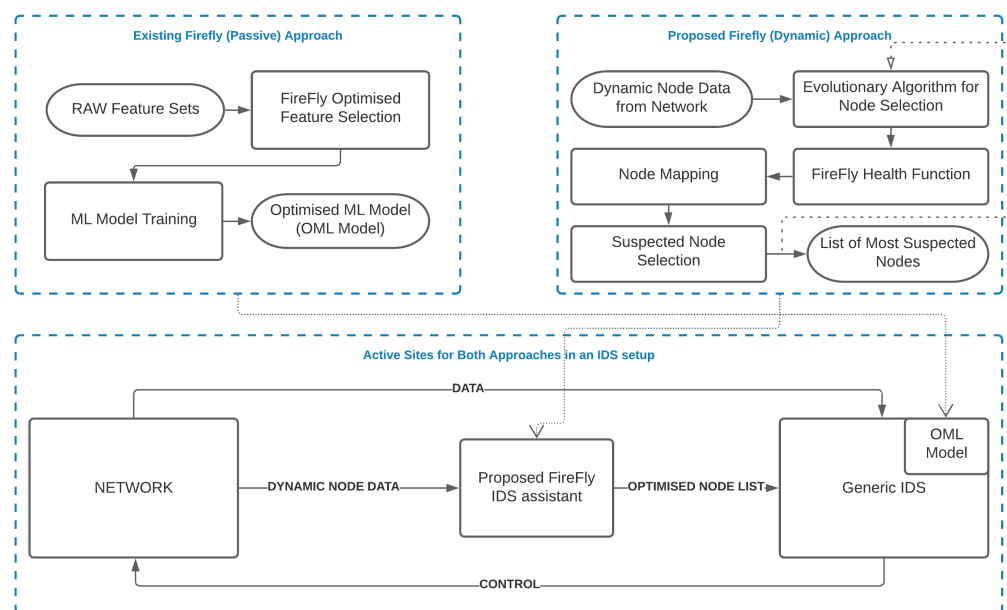


Figure 1. Comparison between the existing approach and the proposed approach.

Therefore, the proposed method is adaptive and efficient for optimizing and assisting network monitoring for IDS, even for large networks. The contribution of the proposed method is outlined below:

1. The proposed modified firefly algorithm supports IDS for the efficient detection and shortlisting of intruders for the early detection of suspicious nodes. It achieves 84% reduction in the number of nodes that requires observation to detect suspicious activity or an attack; 19 out of 50 nodes were flagged suspicious in normal IDS before attack detection, whereas only 3 out of 50 nodes were flagged in the proposed algorithm.
2. A novel health function is proposed to consider more realistic parameters for network monitoring. Whereas the earlier works only focused on feature selection and optimization of the preprocessing of network data, the proposed health function helps in the early identification of suspicious nodes. The proposed health function calculates three important network parameters (normalized ideal throughput, end-to-end delay, and packet delivery ratio) and attacks throughput with a negligible overhead; the introduction of the health function increases the average run time by only 2% from 5.30 to 5.403 s.

The remainder of this paper is organized as follows. Section 2 provides a concise review of the applications of firefly algorithms in various domains. In Section 3, the details

of the proposed work are presented. Section 4 presents and discusses the experimental setup, environment, and results. Section 6 concludes the paper and discusses future work.

2. Related Work

Nature-inspired algorithms are extensively used in the network security field, specifically in IDS [2]. However, countermeasures for the same are also being researched in parallel [6]. Recently, Nijim et al. [7] surveyed the application of nature-inspired algorithms for cybersecurity. Authors have considered various cybersecurity domains, such as malware detection, IDS, and threat analysis. Although most of the IDS mechanisms perform well in feature extraction and analysis for malicious nodes, having an independent monitoring system that could assist the existing IDS serves two purposes: (a) in case of any incident where IDS itself is under attack, the independent mechanism can help the SOC team to trace the malicious nodes, and (b) the existing IDS will benefit by receiving the list of likely-to-be-malicious nodes to monitor more rigorously. Nature-inspired computing provides an excellent approach that works on the principle of attraction toward luminescence, which can be modeled in the cybersecurity domain as the key feature(s) for categorizing malicious nodes.

The firefly algorithm was introduced by [8] in 2008, and it adopts a meta-heuristic approach inspired by fireflies' unisex attraction and flashing behavior.

The firefly algorithm is one of the most reliable optimization techniques implemented on various problem domains and achieves good accuracy [9–13]. Although it has some inefficiency in terms of parameter dependence and computing complexity [14], it has been used in various applications because of its reliability and accuracy, such as power economic dispatch problems and spectrum access [15–17]. Apart from firefly, other nature-inspired algorithms are used for intrusion detection. For example, Zaid and Parul [18] used nature-inspired algorithms (particle swarm optimization (PSO) and whale optimization algorithm (WAO)) for building domain-independent IDS, i.e., the proposed system will be able to detect intrusion on both the network and computer. Authors have reduced computational costs using nature-inspired algorithms for building supervised and unsupervised models.

Many researchers also have used the firefly algorithm for feature selection in IDSs [3,5,19]. In 2018, Ram et al. introduced fuzzy firefly optimization for fast learning networks [20]. In 2019, Dhanarao et al. used a genetic algorithm with a hybrid firefly for efficient IDS for a mobile ad hoc network (MANET) [21]. This work was improved by Albadran et al. in 2020, who proposed a fast-learning network model based on IDS that provides a quick and efficient learning algorithm [22]. In addition, Pakdel et al. recently applied a firefly algorithm for wireless sensor networks (WSNs) [23]. Other recent studies on IDS focused primarily on the deployment and utilization of machine learning techniques to achieve high accuracy [24–27]. Kaur et al. [28] presented a hybrid form of K-mean and FA for anomaly detection in the network environment. Ghosh et al. [29] proposed a modified FA for feature selection to build an IDS system with a focus on detecting intrusion in a cloud environment. An optimal set of features can reduce storage space and training time with improved classification accuracy. Najeeb and Dhannoon [19] used FA as a binary feature selection algorithm. They solved a multi-objective problem (classification accuracy and the number of features) to reduce false alarms and enhance the performance of IDS. Together with the works mentioned above, Fister et al. [30] presented an extensive review of the firefly algorithm, which presents the variations and various applications of the firefly algorithm.

Table 1 summarizes the application of the firefly algorithm (FA) and modified FA for IDS, cryptography, and feature selection. Here, *passive firefly* means that the FA or its modification is not being actively used during anomaly detection or classification in IDS. Instead, FA is used in the early stages, i.e., building for ID system, as depicted in Figure 1. Several researchers utilize FA to support IDS decision making, although most focus on feature selection only. However, the FA can categorize suspicious nodes and hint IDS to pay special attention to those nodes while closely observing the health of individual nodes of the entire network. This idea led to the proposed work, where we scan the selected nodes

for their relative performance values according to their behavior (via health function) and prioritize informing the IDS about any suspicious nodes.

Table 1. Summary of application of firefly algorithm (FA) and Modified-FA.

Work	Contribution	Use of Firefly	Application Domain	Remarks
An IDS using modified-FA in cloud environment [29]	- Modified-FA - NSL-KDD dataset - Less storage space and training time	Feature Selection	ML-based IDS	Passive FA
A feature selection approach using binary FA for network IDS [18]	- Optimal number of features - Multi-objectives (classification accuracy and features) - Reduction of false alarms - Enhanced detection performance	Multi-objective feature selection	ML-based IDS	Passive FA
Hybridization of K-Means and FA for IDS [28]	- Hybridization of K-Means and FA - Higher detection performance than other combinations - NSL-KDD dataset.	Classification	ML and Nature-inspired IDS	Passive FA
An efficient IDS based on Fuzzy FA optimization and fast learning network [20]	- Fast learning system (FLN) - Fluffy firefly algorithm (FFA) - Interruption location - KDD99 dataset	To Obtain Optimal Weights and Threshold values.	Bio-inspired IDS	Passive FA
FA-based feature selection for network IDS [5]	- Experimented with feature selection (filter and wrapper) - Used FA as wrapper method	Feature Selection	ML-based IDS	Passive FA
A modified FA based on neighborhood search [14]	- Modified FA based on neighborhood search - Modified attraction strategy - Neighborhood search method for best neighborhood solutions - Dynamic parameter tuning	Firefly Optimization	Optimization	N/A
Anomaly detection using DSNS * and firefly harmonic clustering algorithm [10]	- Improve initialization in K-Means - Escape local optimal solutions - Efficient clustering the network traffic - Detect volume anomalies from MIB objects	Feature Optimization	Optimization, ML Model Training	Passive FA
PCA *-FA-based XGBoost classification model for intrusion detection in networks using GPU [31]	- Hybrid of PCA and FA - Dimensional reduction	Feature Optimization	Optimization, ML Model Training	Passive FA
Design of keystream generator utilizing FA [13]	- Use of FA for Local Key Generation - Keystream generation.	Random key Generation	Cryptography	N/A

* DSNS: Digital Signature of Network Segment. PCA: Principal component analysis.

3. Materials and Methods

3.1. Firefly-Inspired IDS Optimization

In this study, we modified the generic FA to facilitate network monitoring for an IDS. The prime objective of the modified FA is to build an effective and autonomous network monitoring functionality to support IDS. Troubleshooting and incident responses benefit from efficient network monitoring. Furthermore, the proposed algorithm notifies the IDS to observe suspicious nodes more carefully, as these nodes may require load balancing or intrusion analysis. In either case, it will benefit the overall performance of the firefly network while providing additional support for the IDS.

The nodes are assumed to be arranged in a 2-dimensional network with one unit distance apart. Each node has the following base properties: node name, health value, X coordinate, and Y coordinate. The proposed algorithm sets and corrects the properties of each participating node in the network. After each cycle, the position of the fireflies is determined by the health of the nodes. If multiple fireflies are near one or more nodes, it signifies that these nodes require attention. In addition, the transition of fireflies may reveal sudden and abrupt changes in the network.

The proposed work is inspired by the recent research conducted by Karatas et al. [32], and Bhattacharya et al. [31] that improves the classification and prediction of attacks of IDS. In contrast, the proposed work focuses on observations based on available network

information. We aim at curtailing the number of nodes to be given attention in the intrusion detection process. The nodes in a network deviate significantly from their normal behavior under attack or certain conditions/loads. Therefore, it can be a good indicator for observing the abnormal behavior of nodes in detail.

IDS can observe the activities of suspicious nodes behaving abnormally and take appropriate actions in more detail. The challenge is to decide which node(s) is/are suspicious and which node(s) require(s) observation. Therefore, to identify suspicious nodes in the network that require attention, the proposed work uses the concept of attraction (the level of brightness) of the firefly algorithm.

3.2. Basic Firefly Algorithm

The firefly algorithm is a nature-inspired stochastic global optimization method. The core idea of this swarm-based meta-heuristic algorithm, as proposed by Yang et al. [8], is to design an objective function for the given problem based on the luminescence of each firefly, which directs the swarm of fireflies to move to brighter fireflies for optimal solutions.

The location of firefly i at each time iteration is calculated as

$$X_i^{t+1} = X_i^t + \beta^{-\gamma r_{ij}^2} (X_j^t - X_i^t) + \alpha_t \epsilon_t, \quad (1)$$

where β is the attractiveness, γ is the absorption coefficient, r is the distance between nodes x_i and x_j , α_t is the step size, and ϵ_t is the vector drawn from a Gaussian distribution. Here, β defines the brightness of the firefly, calculated as

$$\beta = \beta_0 e^{-\gamma r^2}. \quad (2)$$

The proposed work applies control to β_{min} . So, initially, each node is given a minimum brightness β_{min} , and its brightness is updated given an initial brightness β_{init} as

$$\beta_{new} = (\beta_{init} - \beta_{min}) e^{-\gamma r^2} + \beta_{min}. \quad (3)$$

Here, Equation (3) is modified from Equation (2) to ensure that the brightness of the nodes does not fall below the predefined minimum value β_{min} . The brightness is significantly affected by the absorption coefficient γ and the Euclidean distance between two fireflies, calculated as

$$r = d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}, \quad (4)$$

where p and q are two points in the Euclidean n -space, and q_i and p_i are Euclidean vectors, starting from the origin.

3.3. Health Function

Throughput is considered the most appropriate parameter for analyzing network performance. Still, considering additional network parameters, such as end-to-end delay and packet delivery ratio, can help build an even more effective IDS. Therefore, a novel health function is proposed to improve network monitoring by considering these additional parameters. Furthermore, the parameters can be readily customized to suit specific applications and types of networks. In addition, it can be fine-tuned by controlling each contributing parameter in the health function (HF).

The proposed HF uses a node (i) and a timestamp (t) as inputs and returns a constant float value corresponding to that node as an output, depending on the selected parameters. This returned value is then used to calculate the node's influence on the nearest firefly and update its brightness value.

HF depends on multiple parameters, such as the *number of nodes* in a network that can change the scan frequency, number, and distribution of fireflies, and the *type of deployment*

also plays a significant role, as it defines the network structure and how nodes are mapped in the grid. *Type of service provided via network* will define how much tolerance we can have for error and generation frequency of fireflies. *Type of network* or the type of data transmitted across the network along with the *general performance priority* of the network significantly influences the formulation and modification of the health function of the nodes.

In our system model, we used throughput (TP), end-to-end delay ($ETED$), and packet delivery ratio (PDR) as performance metrics. TP is the rate at which packets are sent through a network, as throughput is affected by network traffic (e.g., if the network is overloaded with traffic, packet losses will occur). Network throughput can be degraded if routers, switches, or other nodes are outdated or faulty. $ETED$ is the time a packet takes to reach the destination node from the source node and is inversely proportional to the network speed. PDR is the ratio of the total number of packets received at the destination node to the total number of packets sent from the source node, given by

$$PDR = \frac{\sum_{i=0}^P P_{dest_k}}{\sum_{j=0}^Q P_{src_j}}, \quad (5)$$

where P_{dest_k} is the k th packet received by the destination node, P_{src_j} is the j th packet sent by the source nodes, Q is the total number of packets sent, and P is the total number of packets received. Because it is a ratio ($1 > PDR > 0$), the **higher value of PDR gives better performance**.

Based on the parameters defined above, we define the health function of node i at time t as

$$hf(i, t) = \frac{(\frac{\sum_{x=0}^N NTP_x}{N} - ATP(i, t)) + ETED(i, t)}{PDR(i, t)}, \quad (6)$$

where NTP_x is the throughput of the normal operation of node x , N is the total number of nodes in the network, $ATP(i, t)$ is the current throughput of node i at time t (any scan apart from the ideal scan or normal scan is treated as a suspicious scan), $ETED(i, t)$ is the end-to-end delay of i at t , and $PDR(i, t)$ is the power delivery ratio of i at t . The defined health function will be large for poor performance and small for normal or good performance. The average normal throughput ($AvgNTP$) is calculated in the configuration phase using $AvgNTP = \sum_{x=0}^N NTP_x / N$. It is a major factor in determining the overall node health, as it is relative to the normal operation of the network. If the current throughput $ATP(i, t)$ is less than $AvgNTP$, then $hf(i, t)$ will produce a greater positive result as $ETED(i, t)$ increases. The value of the health function increases as $PDR(i, t)$ decreases. Conversely, if $ATP(i, t) > AvgNTP$, then $ETED(i, t)$ is lessened and $PDR(i, t)$ is close to 1 ($\frac{1}{1}$). Thus, the value of the health function is much smaller. Therefore, the health function will create high 'peaks' for unwanted behaviors (i.e., suspicious activities and performance drops) and 'valleys' for normal behaviors, making our problem a maximization problem. Hence, fireflies try to find the maximum values in the search space to locate suspicious nodes.

For the maximization problem, we can define the brightness of fireflies as

$$\beta \propto hf(i, t). \quad (7)$$

Based on the values of the health function, the fireflies attract suspicious nodes in the network to notify the IDS of appropriate measures. In particular, the fireflies' brightness is dominated by the proposed health function, making network monitoring more realistic by considering multiple network parameters. The proposed idea can also be applied to various other tasks, such as white-listing specific nodes or prioritizing incident responses.

3.4. Modified Firefly Algorithm

The proposed algorithm employs the benefits of evolutionary computation and swarm optimization to effectively reach out to multiple nodes while avoiding the issues of local maxima. Each firefly in the proposed method is independent and free to move in the

network, preventing local minimum conditions. It is very unlikely that each firefly will compute the network health similarly. Additionally, the main objective of the proposed method is to give early warning to the IDS system with confidence. Therefore, the nodes with more fireflies indicating bad health would be considered suspicious. Algorithm 1 outlines the basic strategy for the proposed firefly-based IDS. The algorithm applies the *NodeObjectList* of the nodes and firefly parameters, such as the number of fireflies (NF), step size (α), absorption coefficient (γ), and initial brightness (β_{init}). Table 2 summarizes the initialization parameters and their usage for the proposed firefly-inspired IDS scan. The proposed algorithm depends on functions, such as the health function (HF), which calculates the node's health defined by Equation (6). The *SetCoordinate* function takes the node object as an argument, arranges the nodes in a grid format, and returns the positional data in the *Node.Position* structure of the node object. The *UpdateBrightness* function takes a firefly object and updates its brightness using positional data and Equation (3). Next, the member function *step* of the firefly object moves the firefly according to the new health values obtained by HF when repeated for all objects in *FireflyObjectList*. This completes one iteration of the position update.

Algorithm 1: One cycle of firefly-inspired IDS scan

Input: Collection of *Node* objects as *NodeObjectList* and Firefly parameters NF, α, γ

Output: *FireflyObjectList*

```

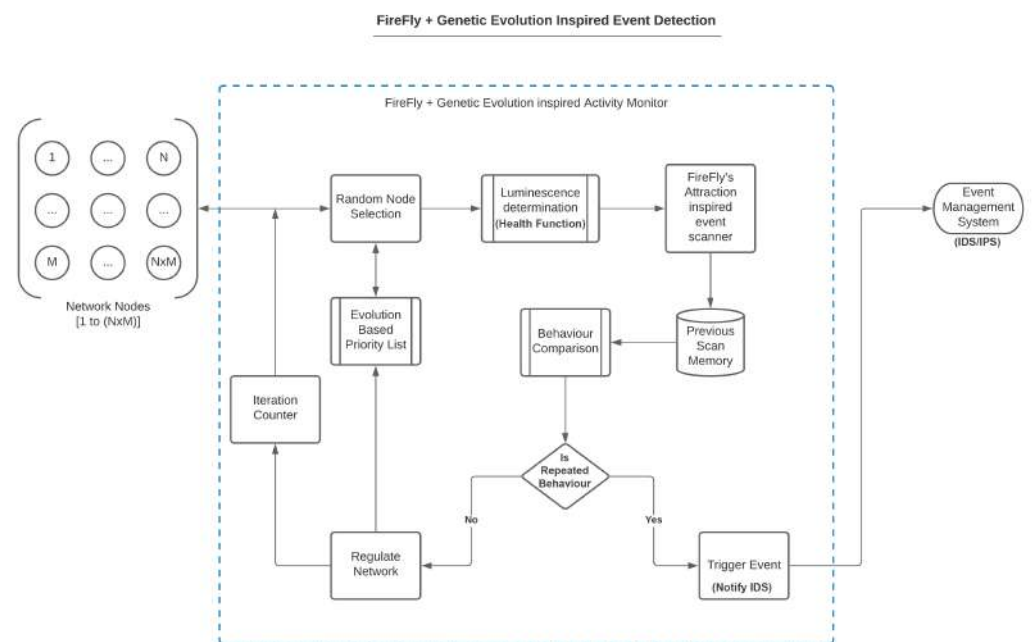
2 FFCluster  $\leftarrow$  NodeObjectList, NF
4 Initialize FireFlies(NF)
6 for Node in NodeObjectList do
8   Node.Position  $\leftarrow$  SetCoordinate(Node)
10  while currentGen < MaxGeneration do
12    for time  $\leftarrow$  0 to MAX_SIM_TIME do
14      for Node in NodeObjectList do
16        | Node.health  $\leftarrow$  HF(Node, time)
17      end
19      FireflyObjectList  $\leftarrow$  new generateFirefly(NF,  $\alpha$ ,  $\gamma$ )
21      for Fly in FireflyObjectList do
23        | Fly.brightness  $\leftarrow$  UpdateBrightness(Fly)
24      end
26      for Fly in FireflyObjectList do
28        | if  $\beta_j > \beta_i$  then
30          | | Fly.position  $\leftarrow$  Fly.step()
31        | end
32      end
33    end
35    currentGen  $\leftarrow$  currentGen + 1
36  end
37 end
39 return FireflyObjectList

```

Table 2. Summary of Initialization Parameters for Firefly-inspired IDS Scan.

Parameter	Description and Use
NodeObjectList	Node Object used by SetCoordinate() function.
NF	Number of fireflies
α	Step size: used in newgenerateFirefly() function.
γ	Absorption coefficient: used in newgenerateFirefly() function.
β_{init}	Initial brightness: used in UpdateBrightness() function.

We assumed the entire network as a grid of nodes to implement the proposed method based on the firefly algorithm for effective IDS operations. Then, we randomly set a list of nodes requiring special observation. The proposed algorithm determines luminescence according to a novel health function that directs the attention of the firefly. After each scan cycle, the positioning of fireflies determines the part of the network or specific nodes that should be considered. Interestingly, as will be demonstrated in the experimental results, in contrast to the general limitation of the firefly algorithm of local maxima, the modified firefly algorithm can address the limitations introduced by the local maxima in conventional methods by considering all nodes equally in each cycle. Figure 2 shows a general flowchart of the proposed method.

**Figure 2.** Modified firefly-inspired IDS flowchart.

The proposed method also determines and utilizes previous scan cycles for more realistic identification of suspicious nodes. However, the role of the proposed health function is most important, as it directs the fireflies and provides a comprehensive analysis of the health of specific nodes by analyzing the three most essential network parameters: throughput, end-to-end delay, and packet delivery ratio.

4. Result

4.1. Experimental Setup

For the experiment, we used an NS2-based NISC network architecture generator coded in Python3 to simulate an attack and normal scenarios in NS2 with configurable nodes and dynamic clusters. We implemented a low-rate TCP denial of service (DoS) attack on

cluster-based network architecture [33]. The cluster-based network architecture is used because it is easy to isolate specific nodes and observe their behavior and effect on the overall network.

To analyze the behavior of the proposed modified firefly algorithm under different networks and protocols, we designed and used a NICS-based testbed, which comprises scalable clusters of nodes of various network architectures and platforms (<https://github.com/Saket-Upadhyay/nics-testbed>, accessed on 27 January 2023). This testbed was explicitly designed for NICS-based experiments on adaptive defense. The size and network protocols of the testbed can be customized according to the requirements and domain of the application.

We used trace files generated by the NS2 simulator as inputs. Each trace file contains 60 s of simulation data of $5N + 8 + M$ nodes, where N is the number of nodes per cluster and M is the number of malicious nodes.

The parameters above simulate a “low-rate TCP DoS” attack on $R2$, which should significantly decrease the performance of $R3$ and $R2$ and affect other clusters. In our experiment, we used $N = 10$ with $M = 1$, which resulted in 59 nodes in our network with 5 clusters, as shown in Figure 3a, in which the malicious node is attached to the $R3$ node, as shown in Figure 3b. Every cluster participates in inter-cluster communication using the following protocols: HTTP, FTP, TELNET, SMTP, and TCP-CBR. Table 3 presents a summary of the properties of each network cluster used for the experiments.

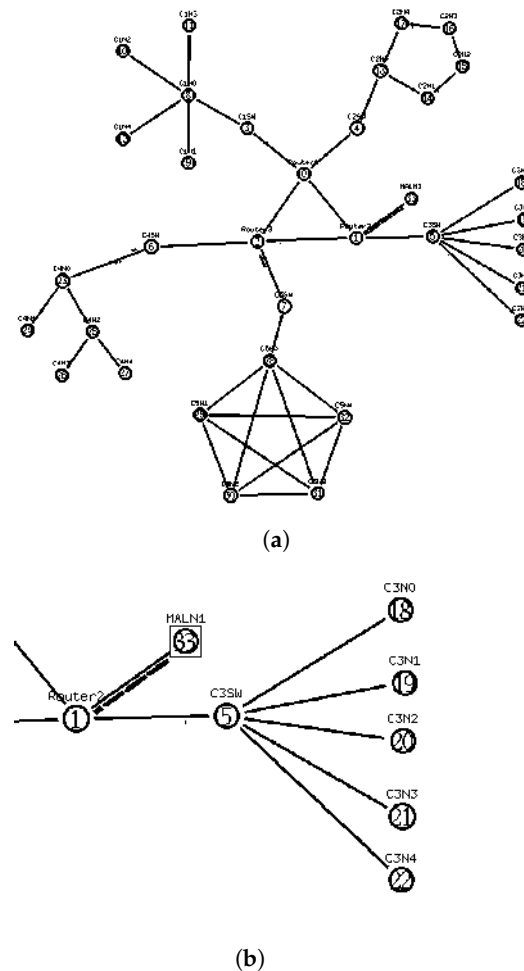


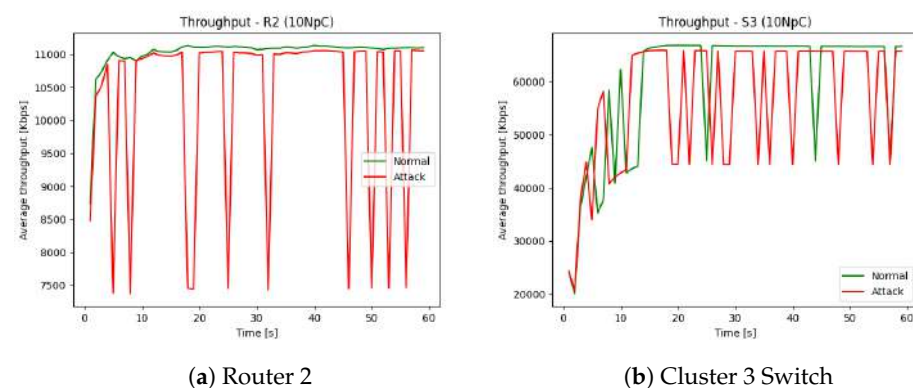
Figure 3. Simulation network configuration. (a) Network structure with 5 clusters and 5 nodes per cluster; (b) Malicious node attached to router R2.

Table 3. Network Cluster Properties.

Cluster	Protocol	Configuration	Connection
C1N(x^*)	Telnet	500 MB at interval of 0.01 s	C5N
C2N(x)	FTP	500 MB at interval of 0.01 s	C4N
C3N(x)	SMTP	20,000 bytes with <i>burst_time</i> = 50 ms and <i>idle_time</i> = 50 ms at 100 KBps	C3N
C4N(x)	HTTP	100 MB at 1 MBps	C3N
C5N(x)	TCP CBR	50 MB	C1N
MALN1	TCP CBR	100 MB	C5N via R3 and R2

We initialized the testbed and marked the throughput performance on the entire network before proceeding with the proposed algorithm. These results can be further analyzed to evaluate the performance of the proposed approach because of efficient network monitoring.

In Figure 4a, the red color plot represents the throughput of the nodes under attack, and the green plot represents a normal operation. The red plot has more frequent valleys than stable peaks; this shows a significant drop in the throughput of the node, which in turn contributes to the drop in performance. We can observe a significant drop in R2 under attack compared to a much more stable and higher throughput of normal operation from the given plot. Similarly, in Figure 4b, a significant performance drop under attack for S3 can be observed.

**Figure 4.** Throughput under normal and attack scenarios for Router 2 (R2) and Cluster Switch 3 (S3).

4.2. Detection of Suspicious Node(s)

The proposed method focuses on observations based on available network information with an aim to curtail the number of nodes to be given attention in the intrusion detection system. The nodes in a network deviate significantly from their normal behavior under attack or certain conditions/loads. Therefore, it can be a good indicator for observing the abnormal behavior of nodes in detail. IDS can observe the activities of suspicious nodes behaving abnormally and take appropriate actions in more detail. The challenge is to decide which node(s) is/are suspicious and which node(s) require(s) observation. Therefore, to identify suspicious nodes in the network that require attention, the proposed method uses the firefly algorithm's concept of attraction (the brightness level). The proposed method can detect suspicious nodes early, irrespective of their location. In addition, owing to a novel health function implemented at the core of the proposed algorithm, the modified firefly algorithm addresses the limitation of the local maxima. The proposed health function is intended to accurately estimate the condition of a node by analyzing the throughput, end-to-end delay, and packet delivery ratio for the entire network traffic.

4.2.1. Test Case #1 [Identification of Malicious Nodes]

To identify malicious nodes in a given network, the first step is to have a grid layout of all the nodes to apply the proposed algorithm. Once configured, the proposed algorithm

proceeds in various cycles while calculating the locations of fireflies in each cycle as per the health function. Figure 5a–d show the results of the modified firefly-inspired strategy at different stages of evolution of a firefly, and the final result at generation 22 is shown in Figure 5d.

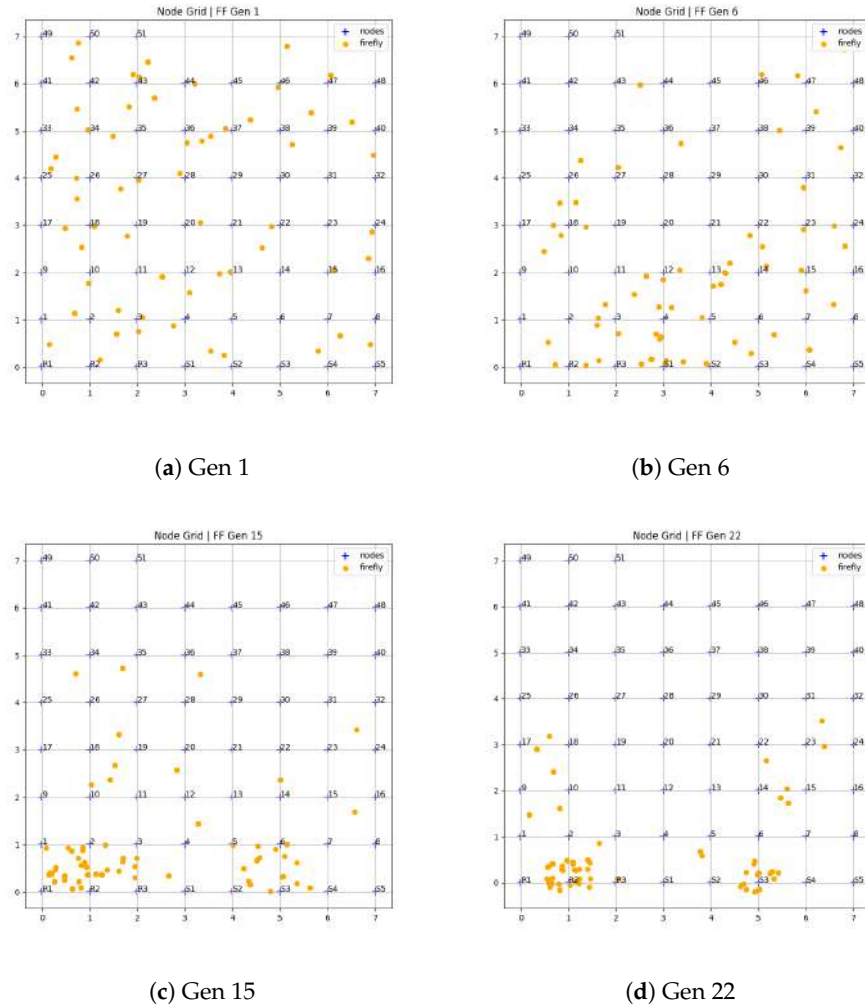


Figure 5. FireFly detectors for Test Case #1.

Figure 5a shows the initial setup of the fireflies. The fireflies in this figure are scattered randomly as per the initial priority list. The fireflies are now expected to converge toward the suspicious nodes of the network to notify the IDS to concentrate on those nodes for possible intrusion. Figure 5b shows the positioning of the fireflies at iteration 6. No prominent positioning is reflected owing to the lack of previous knowledge. However, each iteration is stored, and respective knowledge is utilized to position the next generation of fireflies. Figure 5c shows the noticeable positioning of fireflies near the suspicious nodes. Most importantly, the fireflies do not exhibit local maxima, and they are still well scattered in the network. Finally, Figure 5d illustrates the final inclination of fireflies. Most fireflies are concentrated near the suspicious nodes, which are the nodes connected to a malicious node or are directly affected by the attack. One of the drawbacks of the traditional firefly optimization algorithm is that it gets stuck in the local maxima. However, the proposed method can identify multiple suspicious nodes and is not limited to one worst-performing node in the whole network (global maxima). As shown in Figure 5d, R2 and S3 are highlighted, whereas the nodes R1 and R3 are not. This is because in the local range of 2×2 area under $[(-1, 2), (3, 2), (3, -2), (-1, -2)]$, R2 is the local maxima and covers $\{R1, R2, R3, S1, 1, 2, 3, 4, 9, 10, 11, 12\}$ nodes. Conversely, S3 is a local maxima of the area under $[(3, 2), (7, 2), (7, -2), (3, -2)]$ and covers $\{S1, S2, S3, S4, S5, 4, 5, 6, 7, 8, 12, 13, 14, 15, 16\}$.

As shown in Figure 3a,b, R3 is directly connected to the malicious node, and S3 is under suspicion because it is connected to R3. In addition, its performance drops significantly, as shown in Figure 4.

4.2.2. Test Case #2 [Isolating Cluster 4]

We implemented the proposed algorithm on the network testbed’s isolated section (Cluster 4). The objective of this experiment is twofold: first, to observe the behavior of the proposed algorithm in a homogeneous network setup, and second, to determine the convergence rate of the algorithm. All the nodes of Cluster 4 ($C4Nx$, where $x \in \{1, \dots, n\}$ and n are the max nodes in Cluster 4 (C4)), send HTTP packets of size $1000b$ at a rate of 1.0 Mbps to the nodes of Cluster 3, $C3Ny$ (where $y \in \{1, \dots, m\}$ and m are the maximum nodes in Cluster 3 (C3)). Cluster 4 uses two pivot nodes, $C4N0$ and $C4N(\lfloor (n/2) \rfloor)$. All the nodes from $C4N1$ to $C4N(\lfloor (n/2) - 1 \rfloor)$ are connected to $C4N0$, and all the nodes from $C4N(\lfloor (n/2) + 1 \rfloor)$ to $C4N[n]$ are connected to $C4N(\lfloor (n/2) \rfloor)$, as in Figure 3a, where n is the maximum number of nodes in C4.

Figure 6a–d show the snapshots of intermediate results at different phases of the proposed algorithm. Figure 6a shows the initial random position of the fireflies. After 5 generations, the fireflies started concentrating on certain areas of the node grid, as observed in Figure 6b.

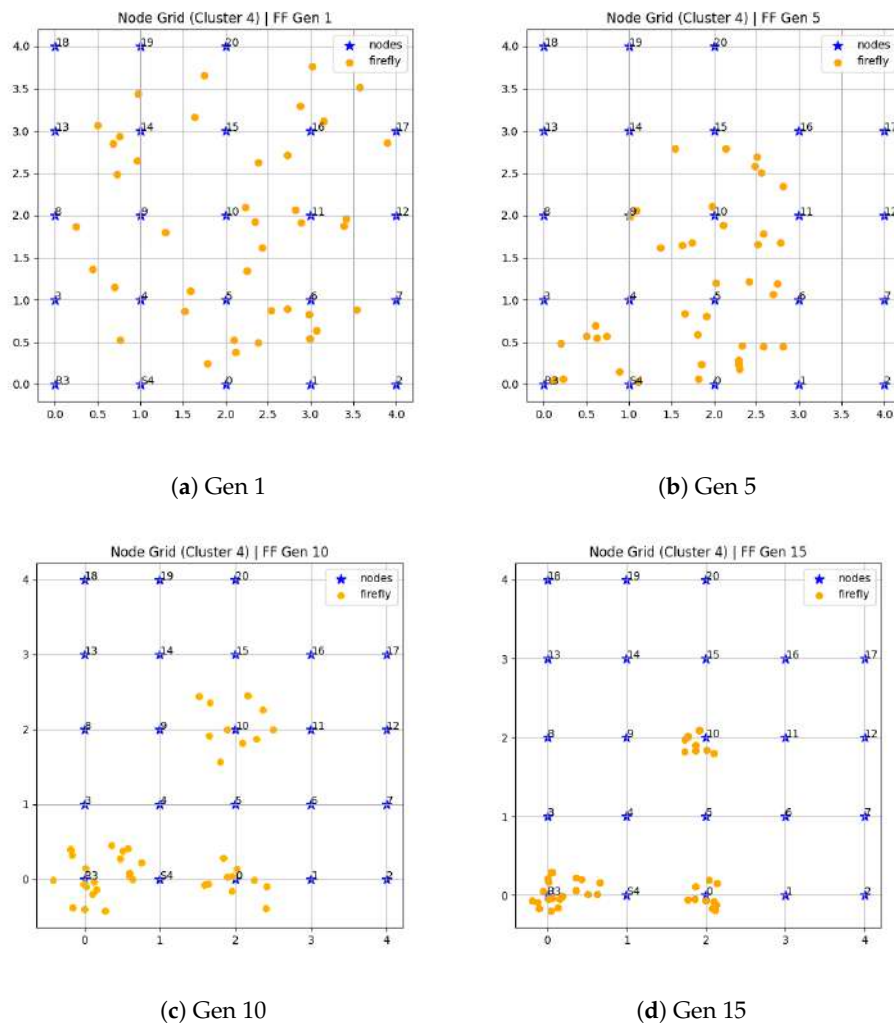


Figure 6. FireFly detectors for Test Case #2.

From Figure 6c,d, we can observe fireflies gathering around some nodes and then highlight R3, 0, and 10. This concludes the list of nodes M , where $M \leftarrow \{R3, C4N0, C4N10\}$,

successfully bringing suspicious nodes to our attention. Cluster 4 adopts two pivot nodes, C4N0 and C4N($\lfloor (n/2) \rfloor$) because $n = 20$ and $\lfloor (20/2) \rfloor = 10$, C4N0 is the first and C4N10 is the second pivot node. Because most of the traffic in Cluster 4 moves via these nodes and then via R3, they are most affected by the attack on R2; hence, their performance drops significantly. This experiment also confirmed the stability of the proposed algorithm for various network architectures and communication protocols.

4.2.3. Importance of Health Function

Based on the experimental results, the health function mentioned in Equation (6) can determine the actual performance of each node contributing to the overall network performance while also notifying the IDS to give more attention to the suspicious nodes. These suspicious nodes can be under attack or overloaded nodes that require load balancing. The health function performs this check while appropriately considering the throughput and other significant factors, such as packet delivery ratio and end-to-end delay.

The health function creates peaks at low performance and valleys with optimal or improved performance, as illustrated in Figure 7. The following figures also indicate a clear correlation between network performance and the output of the health function, which is the most critical parameter for any network monitoring system. We can also use this feature, which has an inverse relation to health function concerning the network performance, during root cause analysis (RCA) to infer knowledge about the role of all suspicious nodes involved in a network attack.

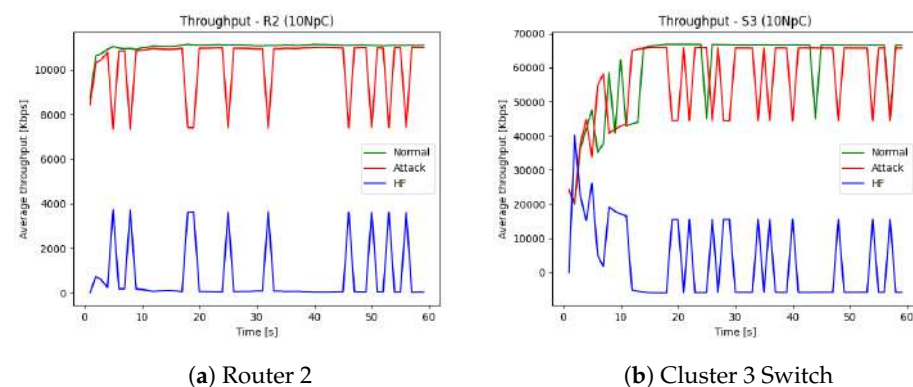


Figure 7. Throughput under normal and attack conditions and its $hf(i, t)$ transformation.

5. Discussion

- We proposed improving the host-based intrusion detection system (HIDS) using a nature-inspired algorithm.
- Our modified firefly algorithm will use input from host behaviors and identify the suspicious host in the network.
- A HIDS can use the proposed solution as a triggering step to improve the detection and computation performance of the detection system.
- The experimental result shows that the proposed solution achieves a notably low computation footprint on the host, and it can be compensated with the detection gain.

6. Conclusions

Nature-inspired cybersecurity requires a swift network analysis to respond appropriately to changes in nodes/networks. IDS needs to be strengthened via optimized network health monitoring as the first line of defense. To scan and shortlist the high-priority nodes required for achieving adaptive defense, which is the main objective of nature-inspired cybersecurity, we proposed a modified version of the firefly algorithm for the efficient network monitoring of IDS while considering multiple network parameters. The results of the modified firefly algorithm were investigated on multiple test cases and determined to be promising in all simulated attack scenarios. The proposed method can detect sus-

picious nodes early, irrespective of their location. In addition, owing to a novel health function implemented at the core of the proposed algorithm, the modified firefly algorithm addresses the limitation of the local maxima. The proposed health function is intended to accurately estimate the condition of a node by analyzing the throughput, end-to-end delay, and packet delivery ratio for the total network traffic. The immediate extension of the proposed approach could involve conducting network emulation with virtual machines to determine the possible ways of optimizing the grid layout of the nodes. Furthermore, it is important to know the grid layout's effect on the proposed method's performance and its correlation.

Author Contributions: Conceptualization, S.K.S. and B.J.C.; methodology, A.K. and S.U.; software, S.U.; validation, S.K.S., A.K., B.J.C. and S.U.; writing—original draft preparation, S.K.S. and S.U.; writing—review and editing, A.K. and B.J.C.; visualization, S.U.; supervision, S.K.S. and A.K.; funding acquisition, B.J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT Korea under the National Research Foundation (NRF) Korea (NRF-2022R1A2C4001270), the Innovative Human Resource Development for Local Intellectualization support program (IITP-2023-RS-2022-00156360) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and the Korea Institute for Advancement of Technology (KIAT) grant funded by the Korean government (MOTIE) (P0017123, The Competency Development Program for Industry Specialist).

Data Availability Statement: The proposed work generates data through simulation, and the code required for the simulation is available through the public code repository. <https://github.com/Saket-Upadhyay/nics-testbed>, accessed on 27 January 2023.

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

Abbreviations

The following abbreviations are used in this manuscript:

NICS	Nature-Inspired Cybersecurity
IDS	Intrusion Detection System
HIDS	Host-based Intrusion Detection System
NIDS	Network-based Intrusion Detection System
AI	Artificial Intelligence
ML	Machine Learning
SVM	Support Vector Machine
MANET	Mobile ad hoc Network
WSN	Wireless Sensor Network
FA	Firefly Algorithm
FFA	Fluffy Firefly Algorithm
FLN	Fast Learning System
PCA	Principal Component Analysis
TP	Throughput
ETED	End-to-End Delay
PDR	Packet Delivery Ratio
HF	Health Function
RCA	Root Cause Analysis
HTTP	Hyper Text Transfer Protocol
DoS	Denial of Service

References

1. Kumar, R.; K, D.; dumka, a.; Loganathan, J. RFA Reinforced Firefly Algorithm to Identify Optimal Feature Subsets for Network IDS. *Int. J. Grid High Perform. Comput.* **2020**, *12*, 5. [[CrossRef](#)]
2. Thakkar, A.; Lohiya, R. Role of swarm and evolutionary algorithms for intrusion detection system: A survey. *Swarm Evol. Comput.* **2020**, *53*, 100631. [[CrossRef](#)]

3. Pervez, M.S.; Farid, D. Feature selection and intrusion classification in NSL-KDD cup 99 dataset employing SVMs. In Proceedings of the SKIMA 2014—8th International Conference on Software, Knowledge, Information Management and Applications, Dhaka, Bangladesh, 15–17 December 2015. [[CrossRef](#)]
4. Çavuşoğlu, U. A new hybrid approach for intrusion detection using machine learning methods. *Appl. Intell.* **2019**, *49*, 2735–2761. [[CrossRef](#)]
5. Selvakumar, B.; Muneeswaran, K. Firefly algorithm based feature selection for network intrusion detection. *Comput. Secur.* **2019**, *81*, 148–155.
6. Chen, J.; Wu, D.; Zhao, Y.; Sharma, N.; Blumenstein, M.; Yu, S. Fooling intrusion detection systems using adversarially autoencoder. *Digit. Commun. Netw.* **2020**, *7*, 453–460. [[CrossRef](#)]
7. Nijim, M.; Goyal, A.; Mishra, A.; Hicks, D. A Review of Nature-Inspired Artificial Intelligence and Machine Learning Methods for Cybersecurity Applications. In *Advances in Nature-Inspired Cyber Security and Resilience*; Springer: Cham, Switzerland, 2022; pp. 109–118.
8. Yang, X.S. *Nature-Inspired Metaheuristic Algorithms*; Luniver Press: Cambridge, UK, 2008; Volume 12, ISBN 978-1-905986-10-1.
9. Ahmed, A.A.; Maheswari, D. Churn prediction on huge telecom data using hybrid firefly based classification. *Egypt. Inform. J.* **2017**, *18*, 215–220. [[CrossRef](#)]
10. Adaniya, M.H.; Carvalho, L.F.; Zarpelão, B.B.; Sampaio, L.D.; Abrão, T.; Jeszensky, P.J.E.; Proença, M.L., Jr. Firefly Algorithm in Telecommunications. In *Bio-Inspired Computation in Telecommunications*; Elsevier: Amsterdam, The Netherlands, 2015; pp. 43–72.
11. Adaniya, M.H.; Lima, M.F.; Rodrigues, J.J.; Abrao, T.; Proença, M.L. Anomaly detection using dns and firefly harmonic clustering algorithm. In Proceedings of the 2012 IEEE International Conference on Communications (ICC), Ottawa, ON, Canada, 10–15 June 2012; pp. 1183–1187.
12. Tuba, E.; Tuba, M.; Beko, M. Two stage wireless sensor node localization using firefly algorithm. In *Smart Trends in Systems, Security and Sustainability*; Springer: Singapore, 2018; pp. 113–120.
13. Mahdi, M.S.; Hassan, N.F. Design of keystream Generator utilizing Firefly Algorithm. *J. Al-Qadisiyah Comput. Sci. Math.* **2018**, *10*, 91.
14. Yu, G. A modified firefly algorithm based on neighborhood search. *Concurr. Comput. Pract. Exp.* **2020**, *33*, e6066. [[CrossRef](#)]
15. Liaquat, S.; Saleem, O.; Azeem, K. Comparison of Firefly and Hybrid Firefly-APSO Algorithm for Power Economic Dispatch Problem. In Proceedings of the IEEE 2020 International Conference on Technology and Policy in Energy and Electric Power (ICT-PEP), Bandung, Indonesia, 23–24 September 2020; pp. 94–99. [[CrossRef](#)]
16. Lakshmana Rao, K.; Sireesha, R.; Shanti, C. On the convergence and optimality of the firefly algorithm for opportunistic spectrum access. *Int. J. Adv. Intell. Paradig.* **2021**, *18*, 119. [[CrossRef](#)]
17. Koliass, C.; Kambourakis, G.; Stavrou, A.; Gritzalis, S. Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 184–208. [[CrossRef](#)]
18. Zaid, M.; Agarwal, P. Intelligent Intrusion Detection System Optimized using Nature-Inspired Algorithms. In Proceedings of the IEEE 2022 1st International Conference on Informatics (ICI), Noida, India, 14–16 April 2022; pp. 80–85.
19. Najeeb, R.F.; Dhannoon, B.N. A feature selection approach using binary firefly algorithm for network intrusion detection system. *ARN J. Eng. Appl. Sci.* **2018**, *13*, 2347–2352.
20. Ram, B.; Rao, B. An Efficient Ids Based on Fuzzy Firefly Optimization and Fast Learning Network. *Int. J. Eng. Technol.* **2018**, *7*, 557–561. [[CrossRef](#)]
21. Dhanarao, S.; Kumar, M. Efficient IDs for MANET Using Hybrid Firefly with a Genetic Algorithm. In Proceedings of the 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 11–12 July 2019.
22. Albadran, M. A new Firefly-Fast Learning Network model based Intrusion-Detection System. *Int. J. Innov. Technol. Explor. Eng.* **2020**, *8*, 146–152. [[CrossRef](#)]
23. Hossein, P.; Reza, F. A firefly algorithm for power management in wireless sensor networks (WSNs). *J. Supercomput.* **2021**, *77*, 9411–9432. [[CrossRef](#)]
24. Junlong, X.; Westerlund, M.; Sovilj, D.; Pulkkis, G. Using Extreme Learning Machine for Intrusion Detection in a Big Data Environment. In Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, Scottsdale, AZ, USA, 7 November 2014; Volume 2014. [[CrossRef](#)]
25. Deshmukh, D.; Ghorpade, T.; Padiya, P. Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset. In Proceedings of the 2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015, Mumbai, India, 15–17 January 2015. [[CrossRef](#)]
26. Al-Yaseen, W.; Othman, Z.; Ahmad Nazri, M.Z. Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Syst. Appl.* **2017**, *67*, 296–303. [[CrossRef](#)]
27. Singh, R. An Intrusion Detection System using Network Traffic Profiling and Online Sequential Extreme Learning Machine. *Expert Syst. Appl.* **2015**, *42*, 8609–8624. [[CrossRef](#)]
28. Kaur, A.; Pal, S.K.; Singh, A.P. Hybridization of K-Means and Firefly Algorithm for intrusion detection system. *Int. J. Syst. Assur. Eng. Manag.* **2018**, *9*, 901–910. [[CrossRef](#)]
29. Ghosh, P.; Sarkar, D.; Sharma, J.; Phadikar, S. An Intrusion Detection System Using Modified-Firefly Algorithm in Cloud Environment. *Int. J. Digit. Crime Forensics (IJDCF)* **2021**, *13*, 77–93. [[CrossRef](#)]

30. Fister, I.; Fister, I., Jr.; Yang, X.S.; Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.* **2013**, *13*, 34–46. [[CrossRef](#)]
31. Bhattacharya, S.; Somayaji, S.; Reddy, P.; Kaluri, R.; Singh, S.; Gadekallu, T.; Alazab, M.; Tariq, U. A Novel PCA-Firefly based XGBoost classification model for Intrusion Detection in Networks using GPU. *Electronics* **2020**, *9*, 219. [[CrossRef](#)]
32. Karatas, G.; Demir, O.; Sahingoz, O. Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset. *IEEE Access* **2020**, *8*, 32150–32162. [[CrossRef](#)]
33. Shandilya, S.K.; Upadhyay, S.; Kumar, A.; Nagar, A.K. AI-assisted Computer Network Operations testbed for Nature-Inspired Cyber Security based adaptive defense simulation and analysis. *Future Gener. Comput. Syst.* **2022**, *127*, 297–308. [[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.