mathematics

Article

A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System

Abdullah Alzaqebah [1], Ibrahim Aljarah [1,*], Omar Al-Kadi [1] and Robertas Damaševičius [2,*]

[1] King Abdullah II School for Information Technology, The University of Jordan, Amman 11942, Jordan; abd9160100@ju.edu.jo (A.A.); o.alkadi@ju.edu.jo (O.A.-K.)
[2] Department of Applied Informatics, Vytautas Magnus University, 44404 Kaunas, Lithuania
* Correspondence: i.aljarah@ju.edu.jo (I.A.); robertas.damasevicius@vdu.lt (R.D.)

Abstract: Cyber-attacks and unauthorized application usage have increased due to the extensive use of Internet services and applications over computer networks, posing a threat to the service's availability and consumers' privacy. A network Intrusion Detection System (IDS) aims to detect aberrant traffic behavior that firewalls cannot detect. In IDSs, dimension reduction using the feature selection strategy has been shown to be more efficient. By reducing the data dimension and eliminating irrelevant and noisy data, several bio-inspired algorithms have been employed to improve the performance of an IDS. This paper discusses a modified bio-inspired algorithm, which is the Grey Wolf Optimization algorithm (GWO), that enhances the efficacy of the IDS in detecting both normal and anomalous traffic in the network. The main improvements cover the smart initialization phase that combines the filter and wrapper approaches to ensure that the informative features will be included in early iterations. In addition, we adopted a high-speed classification method, the Extreme Learning Machine (ELM), and used the modified GWO to tune the ELM's parameters. The proposed technique was tested against various meta-heuristic algorithms using the UNSWNB-15 dataset. Because the generic attack is the most common attack type in the dataset, the primary goal of this paper was to detect generic attacks in network traffic. The proposed model outperformed other methods in minimizing the crossover error rate and false positive rate to less than 30%. Furthermore, it obtained the best results with 81%, 78%, and 84% for the accuracy, F1-score, and G-mean measures, respectively.

Keywords: intrusion detection system; bio-inspired algorithms; extreme learning machine; feature selection; information gain

check for updates

Citation: Alzaqebah, A.; Aljarah, I.; Al-Kadi, O.; Damaševičius, R. A Modified Grey Wolf Optimization Algorithm for an Intrusion Detection System. *Mathematics* **2022**, *10*, 999. https://doi.org/10.3390/math10060999

Academic Editor: Ioannis G. Tsoulos

Received: 8 February 2022
Accepted: 16 March 2022
Published: 21 March 2022

1. Introduction

The use of Internet applications and services is growing day by day in different applications such as e-learning and e-commerce, which raises concerns about security and privacy [1]. Simultaneously with this use, breaching cyber-security using additional and newly developed phishing [2] and hacking [3] tools has increased with the aims to violate the Confidentiality, Integrity, and Availability (CIA) principles [4,5]. Malicious software (malware) is any code that can be used to steal data, bypass access controls, or harm or compromise a software system, a computer network, or an Internet of Things (IoT) system [6,7]. For that, different protection techniques such as firewalls, encryption, and anti-malware tools [8] have been used to prevent cyber threats [9], while digital forensics techniques have been used to investigate the attacks [10]. The rise of new cyber-attacks and zero-day-attacks make the defense against them a significant security problem in the wide-spread networks [11,12].

An Intrusion Detection System (IDS) is a software application or device that is considered as a defensive wall. The primary role of the IDS is monitoring the activities and behavior of the network traffic to detect abnormal and malicious activities and generating alerts and reports of this behavior. Instead of using a standard IDS, a robust and intelligent IDS is needed since several systems have the ability to hide suspicious network traffic [13,14].

The categorization of IDSs is based on various criteria, such as the source of the data and the detection and response mechanisms. Host-based and network-based mechanisms are the leading categories of IDSs based on the source of the data, while anomaly- and signature-based mechanisms are the main techniques based on the detection mechanism. In the signature-based mechanism, the IDS detects the abnormal activities by comparing them with well-known and saved patterns. Hence, the pattern database should be updated periodically to make the IDS recognize the newly developed attacks, whereas an anomaly-based IDS uses a pre-trained normal activity profile to detect different abnormal activities [15].

An IDS deals with a large amount of data in the network traffic; these data contain redundant, noisy, and irrelevant features, which affect the IDS's performance and consume more resources. For that, dimension reduction is needed to enhance the IDS's efficacy [16].

Besides the machine learning classification algorithms, which are the core of the IDS process, the feature selection process has a significant influence on the performance of an IDS. Robust feature selection with seamless integration with the classification process can remarkably improve the IDS's efficiency. One of the approaches to implement feature selection is the meta-heuristic algorithms. Bio-inspired meta-heuristic algorithms imitate the typical behavior of biological species in certain conditions, such as actions taken while searching for and chasing prey. These algorithms can be suitable for dynamic environments and with data having different dimensions. Moreover, these algorithms have shown superior performance in solving optimization problems. In addition, the bio-inspired meta-heuristic algorithms have been widely used to resolve real-world, engineering, and complex problems, as used and reported in [17,18]. Accordingly, they can be used to build an efficient IDS, as IDSs process large amounts of data and have to detect online intrusions in a dynamic and multi-dimensional domain, which is considered a real-world problem.

Feature Selection (FS) is a preprocessing step for obtaining the most relevant features in order to build a robust model. This step is a crucial step that has a direct effect on the IDS's efficacy [19,20]. FS can be divided into two main approaches: filter-based and wrapper-based approaches. The filter-based approach uses the correlation between the data and the corresponding class label without the consultation of the learning algorithm, while the wrapper-based approach evaluates the solution based on the learning algorithm during the searching and optimization processes. Instead of the less-expensive filter-based approach, the proven findings of the wrapper-based approach make it the commonly used approach [21,22]. Bio-inspired meta-heuristic algorithms are commonly used in the wrapper-based approach for the feature selection process in intrusion detection systems due to their outperforming accuracy [23]. Here, the works in the literature are commonly concerned with the binarization process by using different transfer functions. At the algorithm level, the use of different operators such as crossover and mutation is also employed to enhance the searching capability and avoid being trapped in local optima. The random initialization technique is usually used without concern about taking advantage of the filter-based approach and merging it with the rapper-based approach. The most feasible initial population has a direct effect on the convergence speed by achieving the best fitness in early iterations.

In this paper, we propose a modified bio-inspired algorithm, which is Grey Wolf Optimization (GWO), to enhance the IDS's efficacy. The UNW-NB15 dataset was the targeted dataset in this paper to evaluate the proposed model in detecting the generic attack. The modified GWO (MGW) is used in this paper to select the best informative features in the wrapper-based approach and optimize the ELM's weights and biases during the optimization process. The main modifications of the MGWO were: first, the intelligent initialization phase by considering the information from the filter-based approach, especially the Information Gain (IG), to initialize the population, which speeds up the algorithm's convergence and enhances the efficacy of the GWO algorithm; second, adopting the ELM as the base classifier to overcome the added complexity, simultaneously tuning the ELM's weights and biases using the MGWO. Overall, the contributions of this paper can be summarized as follows:

- Improving the quality of the initial population of the meta-heuristic GWO algorithm by including the most relevant features in the initialization phase as evaluated by the IG. Accordingly, a hybrid approach of filter-based and wrapper-based techniques was implemented. An initial guided population speeds up the algorithm's convergence by obtaining the best fitness solutions in early iterations;
- Speeding up the optimization process using the ELM as a base classifier. As mentioned, the ELM is considered a very fast Single-Layer Feed-forward Neural network (SLFN);
- Enhancing the efficacy of the IDS to distinguish and detect the generic attack in the UNSW-NB15 dataset with the most relevant features.

The rest of the paper is organized as follows: Section 2 presents the recent work on IDSs. Section 3 shows in detail the proposed model. Our findings and the conducted results are discussed in Section 4. Finally, the paper is concluded in Section 5.

2. Related Works

Recently, IDSs have become mainstream and received more attention from researchers in the network security field [24]. The data in the network traffic can be huge in size and high-dimensional [25]. For that, dimension reduction is required in the IDS applications using different techniques and approaches to enhance the IDS's performance.

Deep learning methods can be used for network intrusion detection, but they require large amounts of data and huge computational resources for training [26]. For example, Wozniak et al. [27] developed a Recurrent Neural Network (RNN) model to find threats in Android-based IoT cyber-physical systems. Toldinas et al. [28] used multistage deep learning image recognition to present a unique technique for network intrusion detection, in which the network features were converted into four-channel (red, green, blue, and alpha) images, then they were used to train and evaluate the pre-trained deep learning network ResNet50. The proposed method was tested against two publicly accessible benchmark datasets, UNSW-NB15 and BOUN Ddos.

Bio-inspired machine learning techniques show superior performance in selecting the best informative subset of the data features in IDSs, which directly affects the IDSs' performance. Various algorithms have been used in the literature, such as the GWO, the Pigeon-Inspired Optimizer (PIO), Intelligent Water Drops (IWD), the Genetic Algorithm (GA), and the Particle Swarm Optimization algorithm (PSO) for feature selection, yet these algorithms have their advantages and drawbacks. The main drawbacks that the FS bio-inspired algorithms face are: balancing between exploration and exploitation and being trapped in local optima. This section shows some of the recent studies in FS for IDS applications. Table 1 summarizes the recent state-of-the-art techniques and methods for FS in IDSs.

In Alharbi et al. [29], a Local–Global Best Bat Algorithm for Neural Networks (LGBA-NN) was presented to choose both feature subsets and hyperparameters for efficient detection of botnet assaults, based on data from commercial IoT devices infected by two botnets: Gafgyt and Mirai. The proposed LGBA-NN method was evaluated on an N-BaIoT dataset that included comprehensive real-time traffic data from both benign and malicious target classes.

In Khare et al. [30], the feature dimensionality was decreased using the Spider Monkey Optimization (SMO) technique, and the resulting dataset was put into a deep neural network. To achieve homogeneity, the dataset was cleaned using the min–max normalizing approach and then sent via the 1-N encoding method. The approach was evaluated on the benchmark NSL-KDD and KDD Cup 99 datasets.

In order to achieve an optimum detection rate, Natesan et al. [31] suggested a Hadoop-based parallel binary bat algorithm approach for efficient feature selection and classification.

Hadoop's MapReduce programming style increased the computational complexity; the parallel binary bat method optimized the selection of significant characteristics; parallel naive Bayes provided cost-effective categorization. The performance was evaluated on the KDDCup99 test dataset.

Alazzam et al. [32] employed a feature selection algorithm for IDS based on PIO in the wrapper-based approach. They used the sigmoid Transfer Function (TF) to binarize the continuous version of PIO, and for the discretization process, they used a cosine similarity technique. The proposed PIO model was evaluated on the KDDCUPP99, NSL-KDD, and UNSW-NB15 datasets and compared with traditional binarization techniques in terms of the True Positive Rate (TPR), False Positive Rate (FPR), accuracy, and F-score. According to the results, PIO outperformed the other swarm intelligence algorithms. Yet, the proposed IDS did not ensure the local and global search, which made it possible to become trapped in local optima.

In the same way, Acharya and Sing [33] utilized the Intelligent Water Drops (IWD) algorithm to enhance the performance of the IDS. The nature-inspired IWD algorithm represents the overall search space as a graph, consisting of a set of nodes and edges. The IWD initializes a set of paths from the root node to the leaf nodes in the graph. These paths form the solution in terms of the feature subset. These solutions were evaluated using the SVM classifier in order to keep the best-performing solution. The authors evaluated the IWD model using the KDDCUP99 dataset in terms of false alarms, the detection rate, and accuracy. Because they dealt with a connected graph, there was an exhaustive complexity in terms of the time and resources when the dimension of the graph became larger.

Additionally, Alzubi et al. [34] proposed a modified binary version of GWO to enhance the IDS's efficacy by selecting the most relevant features. The main modification in the proposed version was to choose the next position based on the four positions (α), (β), (δ), and (ω) instead of the first three solutions. Thus, the wolf's impact rate decreased to 0.25 rather than 0.33. The proposed modification was evaluated on the NSL-KDD dataset using a Support Vector Machine (SVM) for multi-class classification. The results showed that the proposed version outperformed the other versions by 81.56% in terms of the average accuracy. Here, the authors used the random initialization technique without considering starting from a good enough solutions, which affected the convergence speed of the algorithm.

Furthermore, a multi-objective GWO to solve the FS problem in IDSs was introduced by [35]. The authors used both the accuracy and the reduction rate in the fitness evaluation function with manageable importance weights. Here, the authors employed the random subset generation technique based on the heuristic search for the population initialization phase. The proposed work was evaluated using the SVM and NSL-KDD dataset with a multi-class classification solution. The results showed a superior reduction rate in the number of selected features. The proposed model achieved the best classification accuracies in all types instead of the DoS attack. Similar to our work, we used multiple objectives, which were the crossover error rate and the reduction rate. By using a heuristic search, additional complexity is added to the subset generation process, which we avoided by using our smart initialization technique and forcing the algorithm to converge faster.

Since each bio-inspired algorithm has its drawbacks, the hybridization technique has been widely used by considering the strengthens of algorithms to overcome the weaknesses of other algorithms. Hosseini et al. [36] proposed a hybrid two-phase intrusion detection method. Feature selection was implemented as the first phase by employing the GA algorithm with the SVM classifier, and the crossover and mutation operators were used with the multi-parent strategy. The chosen features were then fed to the second phase, which employed the Artificial Neural Network (ANN) to detect intrusions. Furthermore, Hybrid Gravitational Search (HGS) was used along with a PSO to enhance the model efficacy. The proposed model showed very good results in terms of the detection rate and reduction rate on the NSL-KDD dataset compared with other algorithms combined with the ANN.

On the other hand, using the signature- and anomaly-based IDS together was proposed by [37]. They used the C5.0 decision tree classifier to create the signature-based IDS, while they used the one-class SVM to create the anomaly-based IDS. The main target of the proposed model was to detect the zero-day attacks. The model was evaluated using the NSL-KDD and the Australian Defense Force Academy (ADFA) datasets, which showed superior performance compared with the signature- and anomaly-based IDSs.

Researchers have used an ensemble approach by using majority voting between different algorithms and classifiers. Tama et al. [38] proposed an ensemble classification model for IDSs. The supervised PSO algorithm was combined with a correlation-based feature selection to select the best features subset. For the classification task, they used a majority voting approach with three different tree-based classifiers, which were: C4.5, random forest, and CART. The approach was evaluated using the NSL-KDD dataset in terms of accuracy and the false positive rate in normal vs. abnormal attacks.

In the same manner, Almomani et al. [39] proposed a hybrid IDS model to distinguish between the generic attack and normal behavior. The authors employed a hybrid bio-inspired model for feature selection and SVM, C4.5 decision tree, and Random Forest (RF) classifiers for classification of the traffic into generic and normal. The authors hybridized the meta-heuristic algorithms PSO, Multi-Verse Optimization (MVO), GWO, Moth Flame Optimization (MFO), the Whale Optimization Algorithm (WOA), the Firefly Algorithm (FFA), and the Bat algorithm (BAT). The UNSW-NB15 dataset was used to evaluate the proposed model, which showed the superior performance of the C4.5 decision tree classifier against the others, while the MFO-WO and FFA-GWO achieved the best reduction rate and in the evaluation measures.

The author of [40] proposed a feature selection model for IDS by employing PSO, GWO, the FFA, and the GA. The author used Mutual Information (MI) between the used algorithms in order to generate more robust features set by using pre-defined rules. Each algorithm generated its own best feature set, and by applying the set operations between these sets, new sets would be generated. The proposed model was evaluated using the SVM and J48 classifiers on the UNSW-NB15 dataset, which achieved 90.119% and 90.484% in terms of accuracy for the SVM and J48, respectively. The R13 with 30 features was the best performing one. As before, the author used the same random initialization phase, which directly affected the convergence speed.

Despite GWO's success in solving many optimization issues, it has a flaw in that it becomes trapped in local optima, resulting in an unsatisfactory solution. To address this issue, a balance between the exploration and exploitation stages was adopted in [41]. The authors combined the GWO and the Harris Hawks Optimization (HHO) algorithms to address the feature selection problem by incorporating HHO's balancing strengths in the GWO algorithm. The final results demonstrated the significant impact of the original GWO's flaw.

Similarly, in order to address GWO's shortcomings, some researchers combined GWO with other methods. Al-Tashi et al. [42] hybridized GWO with PSO, GWO with the GA [43], and GWO with the Artificial Bee Colony (ABC) algorithm [44]. These solutions were deployed to overcome the drawbacks of GWO, and it was concluded that the limitation of the original GWO algorithm was being trapped in local optima. GWO was also used for the classification task of IDSs and was proven to obtain more accurate results compared to PSO, the FFA, and the GA [34,35,40].

In general, the hybrid models are more complex in terms of behavior and resource consumption. In addition, the hybridization techniques suffer an imbalance between exploration and exploitation. The ensemble approaches have their strengths, but keeping in consideration the added complexity of the model's building. Yet, each algorithm works separately, which means they still have their drawbacks. From the recent studies, we considered improving a single bio-inspired algorithm for IDSs, adopting techniques and approaches to overcome the main drawbacks that this algorithm face, which are: being trapped in local optima and balancing between the exploration and exploitation phases.

Table 1. Previous state-of-the-art techniques for intrusion detection systems.

Publication	Dataset	Algorithm	Classifier	Technique
[32]	KDDCUPP99, NSL-KDD, UNSW-NB15	PIO	DT	Single
[33]	KDDCUP99	IWD	SVM	Single
[34]	NSL-KDD	MBGWO	SVM	Single
[35]	NSL-KDD	Multi-objective GWO	SVM	Single
[36]	NSL-KDD	GA+SVM	ANN-HGS	Hybrid
[37]	NSL-KDD, ADFA	-	C5.0 + OC-SVM	Hybrid
[38]	NSL-KDD	PSO + correlation-based	C4.5 + RF + CART	Ensemble
[39]	UNSW-NB15	PSO, MVO, GWO, MFO, WOA, FFA, BAT	SVM + C4.5 + RF	Ensemble + hybrid
[40]	UNSW-NB15	PSO, GWO, FFA, and GA with MI	SVM + J48	Ensemble

3. Intrusion Detection System Based on the MGWO

Grey Wolf Optimization (GWO) mimics the intelligent leadership and hunting behavior of grey wolves in nature, where they hunt in a pack of 5–12 wolves. GWO was introduced by [45] to solve optimization problems by indicating four types of wolf leadership hierarchies, which are: alpha, beta, delta, and omega. Exploration, encircling, and attacking the prey are the main hunting phases of the GWO's behavior. Alpha (α) is the strongest member, who is the decision-maker in the group. Beta (β) acts as the advisor of alpha. Delta (δ) and omega (ω) are placed in the third and fourth positions in the wolf hierarchy, as shown in Figure 1. The first three wolves are responsible for optimization, while the fourth one is responsible for tracking the other wolves [46].

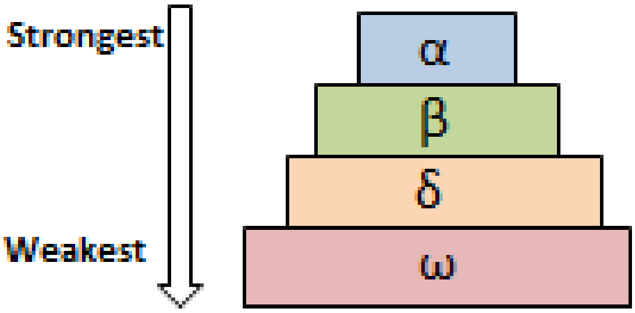

Figure 1. Hierarchy of grey wolves.

GWO is a population-based bio-inspired algorithm, which means building the initial random population and iteratively changing the positions of the search agents to form the best solutions. Dimension reduction by using feature selection techniques is an important phase in data mining and machine learning applications. One of the meta-heuristic applications is feature selection by selecting the most relevant and informative features and ignoring the noisy and redundant features. Therefore, feature selection is considered a hard and complex problem when the search space becomes extremely large. The binary version of GWO shows superior performance in solving the FS problem [47,48].

Here, we propose a modified version of the binary GWO, especially in the population initialization phase. We propose a smart initialization technique to reach the best solution in early iterations, which speeds up the algorithm's convergence. The main modification was achieved by initializing the population by a consultation of the filter-based information and using it in the wrapper-based approach instead of in a random way. The initial population is generated based on the IG value, which determines whether or not to select a feature. The calculation of the IG for a specific feature t is represented as the following equation [49]:

$$IG(t) = -\Sigma P(ci) \log P(ci) + P(t)\Sigma P(ci|t) \log P(ci|t) + P(t')\Sigma P(ci|t') \log P(ci|t') \qquad (1)$$

where c represents a set of features and i is the number of class labels, while the probability of the ith class is indicated by $P(ci)$. The values of $p(t)$ and $p(t')$ represent the classes' probabilities, and $P(ci|t)$ and $P(ci|t')$ are the conditional probabilities of the class with the feature t.

The population size of the MGWO is divided into two parts:

1. The first part represents the injected ratio of the population (25%, 50%, 75%, and 100%) from the proposed modified technique. A feature with a high IG value means it is significant for classifying the instance. Here, and by using the following equation, the proposed technique ensures that features with high IG values will be included in the initial population. The injected population is initialized based on the IG values, as follows:

$$P(i) = \begin{cases} 1, & \text{if } rnd < \text{Normalized IG}(i) \\ 0, & \text{if } rnd \geq \text{Normalized IG}(i) \end{cases} \qquad (2)$$

   where P_i is the binary representation of the ith feature in the initial population and rnd is a random number in the range [0, 1];
2. The second part represents the rest of the population ($1 -$ injection ratio), which is initialized randomly, as shown in the following equation.

$$P(i) = \begin{cases} 1, & \text{if } rnd > 0.5 \\ 0, & \text{if } rnd \leq 0.5 \end{cases} \qquad (3)$$

As mentioned before, and to overcome the added complexity, we adopted a fast classifier for evaluation, which was the ELM. The Extreme Learning Machine (ELM) was proposed by [50] as a new learning scheme for Feed-Forward Neural Networks (FFNNs) to overcome the drawbacks of the Single-Layer Feed-forward Neural network (SLFN). The ELM begins by assigning random weights and biases, then in just one step, computing the hidden layer's output. Then, by using the Moore–Penrose (MP) generalized inverse, the output weights were assigned. Thus, it has been proven that the ELM is an extremely fast process [50,51].

According to the domain of the proposed approach (cyber-security), the Crossover Error Rate (CER), which is also called the Equal Error Rate (ERR), was considered as the fitness function in this paper. The main goal of the CER is to minimize the difference between the False Negative Rate (FNR) and the False Positive Rate (FPR). In other words, a lower CER means better performance [52,53]. The fitness value is calculated as the following formula:

$$\downarrow \text{Fitness} = \alpha \times (|\text{FPR} - \text{FNR}|) + \beta \times \frac{|R|}{|N|} \qquad (4)$$

where α and β are parameters between zero and one to represent the weight of each objective ($\beta = 1 - \alpha$). R indicates the number of chosen features. The overall number of features is represented as N. FNR and FPR are the False Negative Rate and False Positive Rate, respectively, based on the literature; α was set to 0.99, and β was equal to 0.01 [47,54].

In summary, a modified version of the GWO algorithm is proposed, which is denoted as the Modified Grey Wolf Optimization algorithm (MGWO). First, utilizing a filter-based technique to measure the significance of each feature and using the values informing the subset features in the initial population to make it smarter, this speeds up the algorithm's convergence. Second, the ELM was adopted as a base classifier since it is considered a very fast method to overcome the added complexity. In addition, the MGWO was used also to tune the ELM's weights and biases. Figure 2 summarizes the proposed model.

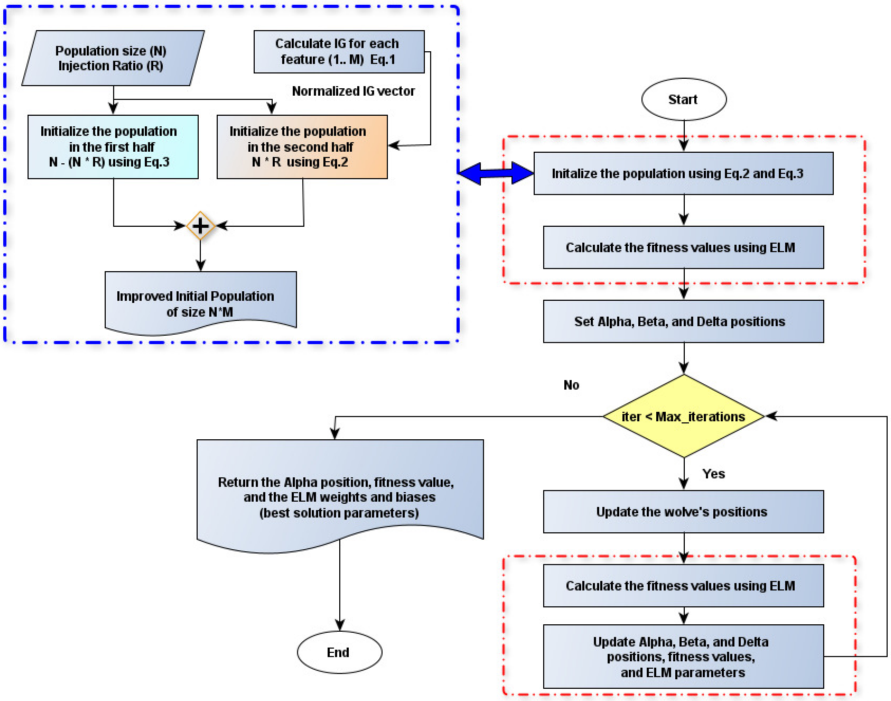
Population size (N)
Injection Ratio (R)
Calculate IG for each feature (1.. M) Eq.1
Normalized IG vector
Initialize the population in the first half N - (N * R) using Eq.3
Initialize the population in the second half N * R using Eq.2
Improved Initial Population of size N*M
Start
Initalize the population using Eq.2 and Eq.3
Calculate the fitness values using ELM
Set Alpha, Beta, and Delta positions
No
iter < Max_iterations
Yes
Return the Alpha position, fitness value, and the ELM weights and biases (best solution parameters)
Update the wolve's positions
Calculate the fitness values using ELM
Update Alpha, Beta, and Delta positions, fitness values, and ELM parameters
End

Figure 2. The proposed MGWO flowchart.

4. Experimental Results and Discussion

4.1. Dataset Description and Data Preparation

The UNSW-NB15 dataset was utilized using the IXIA perfect storm tool. Pcap files with a size of 100 GB (for each file) were generated using the tcpdump tool to simplify the packet's analysis process. Bro-IDS tools with 12 algorithms were utilized to generate 49 features with the class label. The dataset was divided into training and testing sets, where the training set contained 175,341 instances and the testing set included 82,332 instances. This dataset is commonly used to evaluate different techniques adopted in IDSs. The list of anomalies in the UNSW-NB15 dataset are DoS, ShellCode, worms, fuzzers, backdoors, exploits, analysis, generic, and reconnaissance. The generic attack is the one that this paper focused on because it is the most common type of attack. Hence, 40,000 instances represented a generic attack in the training set, while 18,871 instances in the testing set [55,56].

The data preparation step went through the following stages:

1. Feature removal: Some features in the original dataset should be removed since they do not have a relationship with the detection process. These features were: source IP address (srcip), source port number (sport), destination IP address (dstip), destination port number (dsport), record Start time (Stime), and record end time (Ltime) [32]. These features represent static data, such as the source IP and the port number, which can vary from site to site, and this variation is not determinant of whether the traffic has an attack or not. Additionally, the attacks can occur at any time instead of the start and end time. For that, these attributes cannot be considered as features for the traffic, which was eliminated by the work of [32,57];
2. Data encoding: This was implemented by converting the symbolic data into numerical representations, such as the state, protocol, and service type, having a string value that is critical to encode into numerical values to fit with the classifier;
3. For data normalization, the min–max approach was used to scale the data in the range of [0, 1]

On the other hand, to simplify the training process, a stratified sampling technique was used in this work by considering 10,000 samples from the normal type and the same ratio for the generic type.

4.2. Evaluation Metrics

To evaluate the efficacy of the proposed model, the confusion-matrix-based measures were used. The confusion matrix compares prediction results to real results in the dataset, where actual classes are represented as rows and predictions are represented as columns. Here, the True Positive (TP) is when the actual positive cases are predicted as positive cases. Simultaneously, the True Negative (TN) measure is when the negative cases are predicted as negative. In contrast, False Positive (FP) and False Negative (FN) occurs when the predicted instances differ from the actual classes. From these measures, the Accuracy, F-measure, FPR, CER, and G-Mean measures were calculated using the following equations:

- Classification accuracy: This is the total accuracy of the IDS in classifying attacks and is calculated as:
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}}; \tag{5}$$
- False Positive Rate (FPR): The proportion of normal traffic that is identified as an attack was measured, which is calculated as:
$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}; \tag{6}$$
- False Negative Rate (FNR): This is the proportion of anomalies that is identified as normal. The FNR is calculated as:
$$\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{FN}}; \tag{7}$$
- Crossover Error Rate (CER): This is the difference between the FNR and the FPR, which is calculated as:
$$\text{CER} = |\text{FPR} - \text{FNR}|; \tag{8}$$
- Precision (P): This is the percentage of total TP instances divided by the total number of TP and FP instances:
$$\text{P} = \frac{\text{TP}}{\text{TP} + \text{FP}}; \tag{9}$$
- Recall (R): This is the the percentage of total instances that are correctly classified, TPs, divided by the total true positive and False Negative (FN) instances:
$$\text{R} = \frac{\text{TP}}{\text{TP} + \text{FN}}; \tag{10}$$
- F1-score (F-measure): The FM is the mean of the precision and recall, which is calculated as:
$$\text{F1-Score} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}; \tag{11}$$
- G-Mean: Sensitivity and specificity can be combined into a single score that balances both. The G-Mean is calculated as follows:
$$\text{G-Mean} = \sqrt{\text{Recall} * \text{Precision}}. \tag{12}$$

4.3. Experimental and Parameter Settings

The MATLAB R2019a tool was used to implement the proposed approach on an Intel Core I7 machine, 2.6 GHz with 16 GB ram. The advantages of using MATLAB are the simplicity and the availability of the supported toolboxes, such as the parallel toolbox, which speeds up the computation. Moreover, MATLAB processes complex data and is used for complex simulations and engineering problems, while the Python programming language is used with Pandas and Sklearn libraries for data preparation and preprocessing, which already have functions and procedures to preprocess and transform data such as the preprocessing library [32]. The proposed and compared approaches were implemented using the same platform, the same programming language, and the same parameters (fitness function, population size, and number of iterations) to have a fair comparison.

Different parameter settings were tested and analyzed; these parameters included the number of hidden neurons in the ELM network with multiple values: 20, 40, 60, 80, and 100 neurons. The most common activation functions, sigmoid and ReLU, were also tested, as presented by the following equations, respectively:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{13}$$

$$f(x) = Max(0, x) \tag{14}$$

Accordingly, four injection ratios of 25%, 50%, 75%, and 100% for the initialization phase were tested, and the average was used. The overall analysis is shown in Figure 3. It shows that the model performed best with 20 hidden neurons and with the sigmoid activation function. On the other hand, The population size was set to 10 according to the sensitivity analysis of different values: 5, 10, 25, 75, and 100 search agents, as shown in Figure 4. It is clearly shown that the model performed better in terms of accuracy and the CER with 10 search agents, while the number of iterations was set to 100 based on [58,59].

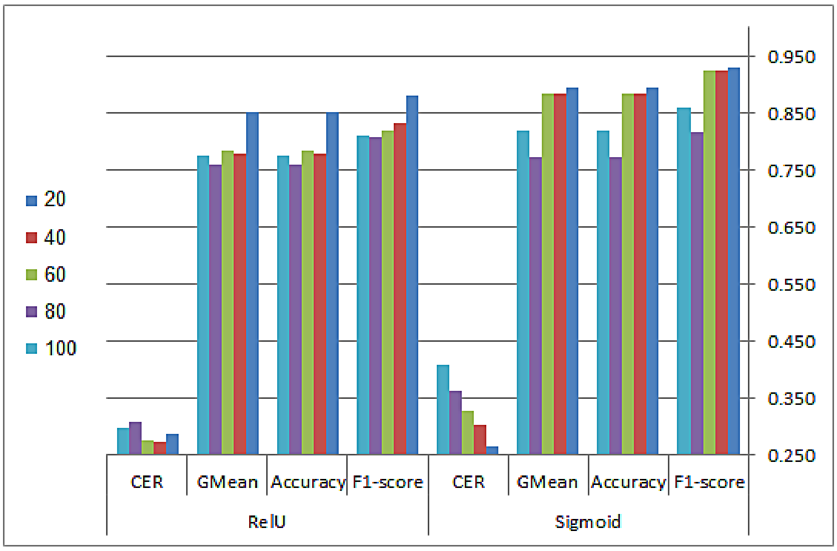

Figure 3. Sensitivity results over the ReLU and sigmoid activation functions using different numbers of hidden neurons.

Finally, the best-performing version based on the injection ratio was chosen. Table 2 shows this analysis, and as clearly shown, the MGWO-25% outperformed other versions in terms of the accuracy, F-score, FPR, CER, and G-mean measures. Therefore, we considered it further and refer to it as the MGWO for the rest of the experiments. Table 3 shows the parameter settings that were used in this paper. Table 4 shows the compared algorithms' parameters.

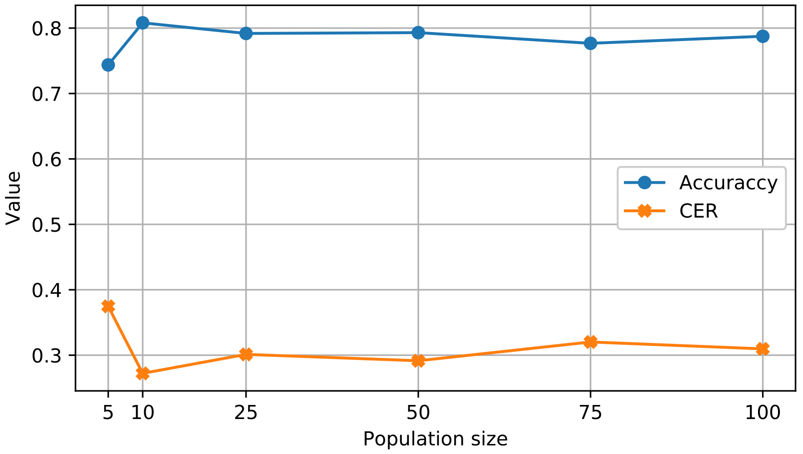

Figure 4. Sensitivity analysis of the population size.

Table 2. Analysis of different injection ratios compared with the standard GWO.

Algorithms	F1_Score	Accuracy	FPR	CER	G-Mean
GWO	0.7656	0.7894	0.3121	0.3007	0.8215
MGWO-25%	0.7808	0.8093	0.2808	0.2669	0.8403
MGWO-50%	0.7637	0.7868	0.3154	0.3025	0.8184
MGWO-75%	0.7700	0.7932	0.3062	0.2944	0.8241
MGWO-100%	0.7572	0.7791	0.3283	0.3180	0.8116

Table 3. List of the parameters used in the experiments.

No.	Parameter	Value
1.	ELM type	Basic
2.	Activation function	Sigmoid
3.	Number of hidden neurons	20
4.	Population size	10
5.	Max number of iterations	100
6.	Injection ratio	25%

Table 4. The parameter settings of the compared algorithms.

Algorithm	Parameter	Value
GA	Crossover percentage	0.8
	Mutation percentage	0.3
	Mutation rate	0.02
	Selection scheme	Random
	Tournament size	3
	Beta	8
PSO	Inertia weight	2
	Max inertia weight	0.9
	Min inertia weight	0.4
	c1, c2	2
GWO	Convergence constant α	[2 0]
HHO	Upper bound	1
	Lower bound	0
	Transfer function	S2

4.4. Classification

In this subsection, the proposed model is evaluated and compared with a well-known meta-heuristic algorithms. Table 5 shows the comparison results of the proposed MGWO, standard GWO, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grasshopper Optimization Algorithm (GOA), and Harris Hawks Optimization (HHO) in terms of classification accuracy, F1-score, G-mean, FPR, and CER.

The modified GWO (MGWO) outperformed the other algorithms in all measures. This improvement in the results was due to using the smart initialization technique, which ensured the best fitness values in the early iterations, which made the MGWO perform better compared to the other algorithms.

As clearly shown in Table 5, the MGWO minimized the difference between the FNR and FPR (CER) to less than 0.27. In other words, it minimized both attacks to be classified as normal and vice versa, which led to a more robust IDS. By achieving the lowest CER, the MGWO also achieved the minimum FPR by classifying the normal instances as attacks. On the other hand, the MGWO reached the highest G-mean value, which indicated the proper sensitivity and specificity values with the highest classification accuracy as well.

Table 5. Comparison of the MGWO, GWO, HHO, GA, PSO, and GOA in terms of average classification accuracy, F1-score, G-mean, FPR, and CER over 30 runs.

Algorithms	F1_Score	Accuracy	FPR	CER	G-Mean
GA	0.7511	0.7827	0.3173	0.3164	0.8151
PSO	0.7397	0.7659	0.3431	0.3316	0.7997
GOA	0.7461	0.7710	0.3389	0.3264	0.8061
HHO	0.7627	0.7862	0.3182	0.3090	0.8191
GWO	0.7656	0.7894	0.3121	0.3007	0.8215
MGWO	0.7808	0.8093	0.2808	0.2669	0.8403

Moreover, the MGWO showed a faster convergence speed, as shown in Figure 5. It was clearly shown that the MGWO performed better compared to the standard GWO. The smart initialization technique gave the algorithm the power to obtain the best fitness values in the early iterations. Accordingly, the MGWO improved the performance of the GWO algorithm and sped up the algorithm's convergence, as illustrated in Figure 5.

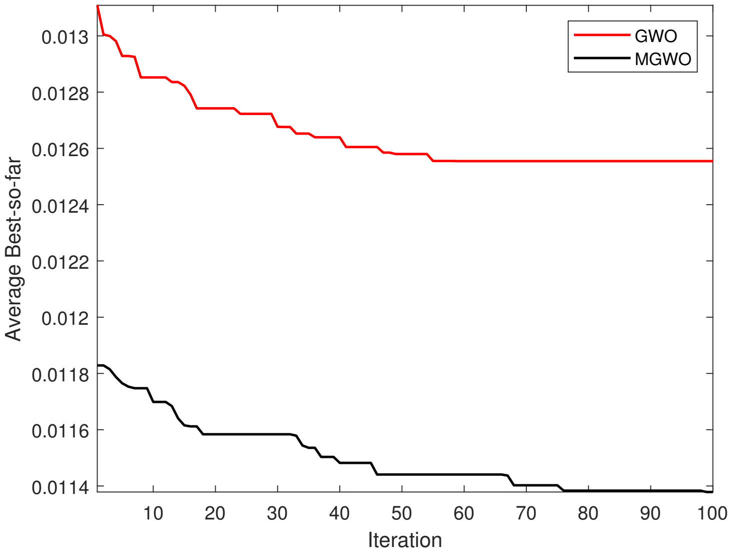

Figure 5. Convergence curve of the UNSW-NB15 dataset for GWO and the MGWO.

On the other hand, and in terms of the reduction rate and best fitness values, Figure 6 shows the best obtained fitness values and the selected number of features. GOA performed better by obtaining a 0.0063 fitness value and an 81% reduction rate in terms of the dimension reduction rate. The MGWO achieved a better reduction rate of 61% compared with a 51% reduction rate for the standard GWO. Hence, as the detection performance is the measure considered most in IDS applications, the MGWO had the best performance on the detection measures.

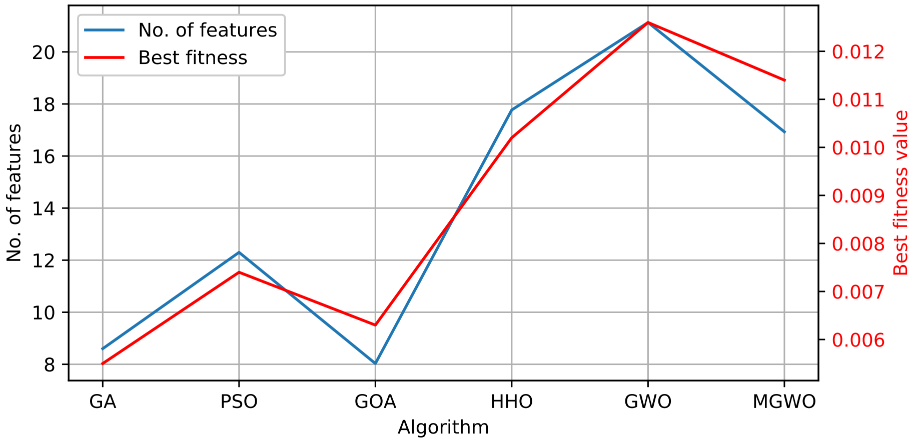

Figure 6. Comparison between the tested algorithms in terms of the best fitness values and the number of selected features.

In the literature, different research papers used the same dataset, but with some variants. The work of [40] proposed a hybrid model with a mutual information approach. The author used the UNSW-NB15 dataset to solve a multi-class classification problem (all attacks). The author's model achieved 90.119% and 90.484% in terms of accuracy for the SVM and J48 respectively, using 30 features. The authors of [60] proposed a hybrid model by overcoming the GWO drawbacks by using the Crow Search Algorithm (CSA) for feature selection and the Deep Sparse Autoencoder (DSAE) for multi-class classification on the same dataset. The proposed model showed interesting results with almost 99% overall accuracy. The authors did not provide any details about the number of selected features. Our proposed model was specified for a certain type of attack (generic), and our model achieved 80.1% accuracy using almost 17 features. However, it showed a faster convergence compared to the standard GWO.

As the summary of this section, the MGWO showed an exciting result for IDS applications, while the modified initialization phase achieved the targeted objectives by enhancing the IDS's efficacy in terms of the detection rate and minimizing the CER. Additionally, it showed how the convergence was sped up by reaching the best fitness values in the early iterations compared with the standard GWO.

Finally, since several domains and applications can be enhanced using ML techniques, it is difficult to find an algorithm that applies to all optimization problems with reference to the No-Free-Lunch theorem (NFL) [61]. Meta-heuristic algorithms have been successfully implemented in various domains and obtained superior results with different applications.

5. Conclusions

To improve the efficacy of IDSs, we proposed an improved bio-inspired meta-heuristic algorithm in this paper. To select the best feature set and eliminate irrelevant and noisy features, the modified GWO was used and refined. By employing an intelligent initialization strategy, the modified GWO (MGWO) accelerated the algorithm's convergence speed. The IG values from the filter-based approach were used to initialize the population in the wrapper-based approach by this intelligent strategy. The ELM was utilized as a base classifier to overcome the algorithm's increased complexity. As a result, in addition to the FS procedure, the MGWO was employed to tune the ELM's weights and biases.

The UNSW-NB15 dataset was used to evaluate the proposed model. With a 25% population injection ratio, the results showed that the MGWO performed better in terms of accuracy, G-mean, F1-score, FPR, and CER, achieving the highest G-mean value of 84%, while minimizing the CER and FPR to 27% and 28%, respectively.

For future directions, the transfer function is an essential part of binarizing the continuous search space. The new transfer function can be adopted in the algorithm to ensure both local and global search, such as the X-shaped transfer function. For IDS applications, the algorithm will be generalized to deal with multi-classification problems to detect different types of attacks with good performance.

Author Contributions: Conceptualization, I.A. and A.A.; methodology, A.A., O.A.-K., I.A. and R.D.; validation, A.A. and O.A.-K.; data curation, A.A.; writing—original draft preparation, A.A., O.A.-K. and I.A.; writing—review and editing, A.A., O.A.-K., I.A. and R.D.; supervision O.A.-K.; project administration, I.A., A.A. and R.D. 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 data are available in a publicly accessible repository at https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed on 10 December 2021). The data presented in this study are available at https://ieeexplore.ieee.org/abstract/document/7348942 and https://www.researchgate.net/profile/Nour-Moustafa/publication/287330529_UNSW-NB15_a_comprehensive_data_set_for_network_intrusion_detection_systems_UNSW-NB15_network_data_set/links/567bf71708ae051f9ae029b6/UNSW-NB15-a-comprehensive-data-set-for-network-intrusion-detection-systems-UNSW-NB15-network-data-set.pdf (accessed on 10 December 2021) [55].

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

References

1. Dixit, P.; Kohli, R.; Acevedo-Duque, A.; Gonzalez-Diaz, R.R.; Jhaveri, R.H. Comparing and Analyzing Applications of Intelligent Techniques in Cyberattack Detection. *Secur. Commun. Netw.* **2021**, *2021*, 5561816. [CrossRef]
2. Azeez, N.A.; Salaudeen, B.B.; Misra, S.; Damaševičius, R.; Maskeliūnas, R. Identifying phishing attacks in communication networks using URL consistency features. *Int. J. Electron. Secur. Digit. Forensics* **2020**, *12*, 200–213. [CrossRef]
3. Rotimi, O.J.; Misra, S.; Agrawal, A.; Azubuike, E.; Maskeliunas, R.; Damasevicius, R. Curbing Criminal Acts on Mobile Phone Network. In *Cyber Security and Digital Forensics*; Springer:Berlin/Heidelberg, Germany, 2022; pp. 99–111.
4. Damaševičius, R.; Toldinas, J.; Venčkauskas, A.; Grigaliūnas, Š.; Morkevičius, N.; Jukavičius, V. Visual analytics for cyber security domain: State-of-the-art and challenges. In *International Conference on Information and Software Technologies*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 256–270.
5. Damasevicius, R.; Toldinas, J.; Venckauskas, A.; Grigaliunas, S.; Morkevicius, N. Technical Threat Intelligence Analytics: What and How to Visualize for Analytic Process. In Proceedings of the 2020 24th International Conference Electronics, Palanga, Lithuania, 15–17 June 2020; pp. 1–4.
6. Odusami, M.; Abayomi-Alli, O.; Misra, S.; Shobayo, O.; Damasevicius, R.; Maskeliunas, R. Android malware detection: A survey. In Proceedings of the International Conference on Applied Informatics, Bogotá, Colombia, 1–3 November 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 255–266.
7. Subairu, S.O.; Alhassan, J.; Misra, S.; Abayomi-Alli, O.; Ahuja, R.; Damasevicius, R.; Maskeliunas, R. An experimental approach to unravel effects of malware on system network interface. In *Advances in Data Sciences, Security and Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 225–235.
8. Rudd, E.M.; Rozsa, A.; Günther, M.; Boult, T.E. A Survey of Stealth Malware Attacks, Mitigation Measures, and Steps Toward Autonomous Open World Solutions. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 1145–1172. [CrossRef]
9. Cascavilla, G.; Tamburri, D.A.; Van Den Heuvel, W. Cybercrime threat intelligence: A systematic multi-vocal literature review. *Comput. Secur.* **2021**, *105*, 102258. [CrossRef]
10. Grigaliunas, S.; Toldinas, J.; Venckauskas, A.; Morkevicius, N.; Damaševičius, R. Digital evidence object model for situation awareness and decision making in digital forensics investigation. *IEEE Intell. Syst.* **2020**, *36*, 39–48. [CrossRef]
11. Moustafa, N.; Creech, G.; Slay, J. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In *Data Analytics and Decision Support for Cybersecurity*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 127–156.
12. Zhou, Y.; Cheng, G.; Jiang, S.; Dai, M. Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Comput. Netw.* **2020**, *174*, 107247. [CrossRef]
13. Scarfone, K.; Mell, P. Guide to intrusion detection and prevention systems (idps). *NIST Spec. Publ.* **2007**, *800*, 94.

14. Odusami, M.; Misra, S.; Adetiba, E.; Abayomi-Alli, O.; Damasevicius, R.; Ahuja, R. An improved model for alleviating layer seven distributed denial of service intrusion on webserver. *J. Phys. Conf. Ser.* **2019**, *1235*, 012020. [CrossRef]
15. Alkadi, O.; Moustafa, N.; Turnbull, B. A review of intrusion detection and blockchain applications in the cloud: Approaches, challenges and solutions. *IEEE Access* **2020**, *8*, 104893–104917. [CrossRef]
16. Zaman, S.; Karray, F. Features selection for intrusion detection systems based on support vector machines. In Proceedings of the 2009 6th IEEE Consumer Communications and Networking Conference, Las Vegas, NV, USA, 10–13 January 2009; pp. 1–8.
17. Mnasri, S.; Bossche, A.V.D.; Nasri, N.; Val, T. The 3D redeployment of nodes in Wireless Sensor Networks with real testbed prototyping. In Proceedings of the International Conference on Ad-Hoc Networks and Wireless, Messina, Italy, 20–22 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 18–24.
18. Mnasri, S.; Nasri, N.; van den Bossche, A.; Thierry, V. 3D indoor redeployment in IoT collection networks: A real prototyping using a hybrid PI-NSGA-III-VF. In Proceedings of the 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC), Limassol, Cyprus, 25–29 June 2018; pp. 780–785.
19. Liu, H.; Motoda, H. *Feature Selection for Knowledge Discovery and Data Mining*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 454.
20. Tang, X.; Dai, Y.; Xiang, Y. Feature selection based on feature interactions with application to text categorization. *Expert Syst. Appl.* **2019**, *120*, 207–216. [CrossRef]
21. Glover, F.W.; Kochenberger, G.A. *Handbook of Metaheuristics*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 57.
22. Talbi, E.G. *Metaheuristics: From Design to Implementation*; John Wiley & Sons: Hoboken, NJ, USA, 2009; Volume 74.
23. Tubishat, M.; Ja'afar, S.; Alswaitti, M.; Mirjalili, S.; Idris, N.; Ismail, M.A.; Omar, M.S. Dynamic salp swarm algorithm for feature selection. *Expert Syst. Appl.* **2021**, *164*, 113873. [CrossRef]
24. Azeez, N.A.; Ayemobola, T.J.; Misra, S.; Maskeliūnas, R.; Damaševičius, R. Network intrusion detection with a hashing based apriori algorithm using Hadoop MapReduce. *Computers* **2019**, *8*, 86. [CrossRef]
25. Damasevicius, R.; Venckauskas, A.; Grigaliunas, S.; Toldinas, J.; Morkevicius, N.; Aleliunas, T.; Smuikys, P. LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. *Electronics* **2020**, *9*, 800. [CrossRef]
26. Li, G.; Sharma, P.; Pan, L.; Rajasegarar, S.; Karmakar, C.; Patterson, N. Deep learning algorithms for cyber security applications: A survey. *J. Comput. Secur.* **2021**, *29*, 447–471. [CrossRef]
27. Wozniak, M.; Silka, J.; Wieczorek, M.; Alrashoud, M. Recurrent Neural Network Model for IoT and Networking Malware Threat Detection. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5583–5594. [CrossRef]
28. Toldinas, J.; Venčkauskas, A.; Damaševičius, R.; Grigaliūnas, Š.; Morkevičius, N.; Baranauskas, E. A novel approach for network intrusion detection using multistage deep learning image recognition. *Electronics* **2021**, *10*, 1854. [CrossRef]
29. Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet attack detection using local global best bat algorithm for industrial Internet of things. *Electronics* **2021**, *10*, 1341. [CrossRef]
30. Khare, N.; Devan, P.; Chowdhary, C.L.; Bhattacharya, S.; Singh, G.; Singh, S.; Yoon, B. SMO-DNN: Spider monkey optimization and deep neural network hybrid classifier model for intrusion detection. *Electronics* **2020**, *9*, 692. [CrossRef]
31. Natesan, P.; Rajalaxmi, R.R.; Gowrison, G.; Balasubramanie, P. Hadoop Based Parallel Binary Bat Algorithm for Network Intrusion Detection. *Int. J. Parallel Program.* **2017**, *45*, 1194–1213. [CrossRef]
32. Alazzam, H.; Sharieh, A.; Sabri, K.E. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert Syst. Appl.* **2020**, *148*, 113249. [CrossRef]
33. Acharya, N.; Singh, S. An IWD-based feature selection method for intrusion detection system. *Soft Comput.* **2018**, *22*, 4407–4416. [CrossRef]
34. Alzubi, Q.M.; Anbar, M.; Alqattan, Z.N.; Al-Betar, M.A.; Abdullah, R. Intrusion detection system based on a modified binary grey wolf optimisation. *Neural Comput. Appl.* **2020**, *32*, 6125–6137. [CrossRef]
35. Alamiedy, T.A.; Anbar, M.; Alqattan, Z.N.; Alzubi, Q.M. Anomaly-based intrusion detection system using multi-objective grey wolf optimisation algorithm. *J. Ambient. Intell. Humaniz. Comput.* **2020**, *11*, 3735–3756. [CrossRef]
36. Hosseini, S.; Zade, B.M.H. New hybrid method for attack detection using combination of evolutionary algorithms, SVM, and ANN. *Comput. Netw.* **2020**, *173*, 107168. [CrossRef]
37. Khraisat, A.; Gondal, I.; Vamplew, P.; Kamruzzaman, J.; Alazab, A. Hybrid intrusion detection system based on the stacking ensemble of c5 decision tree classifier and one class support vector machine. *Electronics* **2020**, *9*, 173. [CrossRef]
38. Tama, B.A.; Rhee, K.H. A combination of PSO-based feature selection and tree-based classifiers ensemble for intrusion detection systems. In *Advances in Computer Science and Ubiquitous Computing*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 489–495.
39. Almomani, O. A Hybrid Model Using Bio-Inspired Metaheuristic Algorithms for Network Intrusion Detection System. *CMC-Comput. Mater. Contin.* **2021**, *68*, 409–429. [CrossRef]
40. Almomani, O. A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry* **2020**, *12*, 1046. [CrossRef]
41. Al-Wajih, R.; Abdulkadir, S.J.; Aziz, N.; Al-Tashi, Q.; Talpur, N. Hybrid binary grey wolf with Harris hawks optimizer for feature selection. *IEEE Access* **2021**, *9*, 31662–31677. [CrossRef]
42. Al-Tashi, Q.; Kadir, S.J.A.; Rais, H.M.; Mirjalili, S.; Alhussian, H. Binary optimization using hybrid grey wolf optimization for feature selection. *IEEE Access* **2019**, *7*, 39496–39508. [CrossRef]

43. Tawhid, M.A.; Ali, A.F. A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. *Memet. Comput.* **2017**, *9*, 347–359. [CrossRef]
44. Gaidhane, P.J.; Nigam, M.J. A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. *J. Comput. Sci.* **2018**, *27*, 284–302. [CrossRef]
45. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
46. Al-Tashi, Q.; Rais, H.M.; Abdulkadir, S.J.; Mirjalili, S.; Alhussian, H. A review of grey wolf optimizer-based feature selection methods for classification. *Evol. Mach. Learn. Tech.* **2020**, 273–286.
47. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, *172*, 371–381. [CrossRef]
48. Faris, H.; Aljarah, I.; Al-Betar, M.A.; Mirjalili, S. Grey wolf optimizer: A review of recent variants and applications. *Neural Comput. Appl.* **2018**, *30*, 413–435. [CrossRef]
49. Gao, Z.; Xu, Y.; Meng, F.; Qi, F.; Lin, Z. Improved information gain-based feature selection for text categorization. In Proceedings of the 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), Aalborg, Denmark, 11–14 May 2014; pp. 1–5.
50. Huang, G.B.; Zhu, Q.Y.; Siew, C.K. Extreme learning machine: A new learning scheme of feedforward neural networks. In Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541), Budapest, Hungary, 25–29 July 2004; Volume 2, pp. 985–990.
51. Feng, Z.k.; Niu, W.j.; Tang, Z.y.; Xu, Y.; Zhang, H.r. Evolutionary artificial intelligence model via cooperation search algorithm and extreme learning machine for multiple scales nonstationary hydrological time series prediction. *J. Hydrol.* **2021**, *595*, 126062. [CrossRef]
52. Liu, J.; Yu, F.R.; Lung, C.H.; Tang, H. Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 806–815. [CrossRef]
53. Urvashi; Awasthi, L.K.; Sikka, G. Behavior-Based Approach for Fog Data Analytics: An Approach toward Security and Privacy. In *Fog Data Analytics for IoT Applications*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 341–354.
54. Faris, H.; Mafarja, M.M.; Heidari, A.A.; Aljarah, I.; Ala'M, A.Z.; Mirjalili, S.; Fujita, H. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. *Knowl.-Based Syst.* **2018**, *154*, 43–67. [CrossRef]
55. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive dataset for network intrusion detection systems (UNSW-NB15 network dataset). In Proceedings of the 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
56. Moustafa, N.; Slay, J. The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 data set. *Inf. Secur. J. Glob. Perspect.* **2016**, *25*, 18–31. [CrossRef]
57. Sharma, J.; Giri, C.; Granmo, O.C.; Goodwin, M. Multi-layer intrusion detection system with ExtraTrees feature selection, extreme learning machine ensemble, and softmax aggregation. *EURASIP J. Inf. Secur.* **2019**, *2019*, 1–16. [CrossRef]
58. Mafarja, M.; Mirjalili, S. Whale optimization approaches for wrapper feature selection. *Appl. Soft Comput.* **2018**, *62*, 441–453. [CrossRef]
59. Mafarja, M.; Aljarah, I.; Faris, H.; Hammouri, A.I.; Ala'M, A.Z.; Mirjalili, S. Binary grasshopper optimisation algorithm approaches for feature selection problems. *Expert Syst. Appl.* **2019**, *117*, 267–286. [CrossRef]
60. Keserwani, P.K.; Govil, M.C.; Pilli, S.E. An Optimal Intrusion Detection System using GWO-CSA-DSAE Model. *Cyber-Phys. Syst.* **2021**, *7*, 197–220. [CrossRef]
61. Wolpert, D.H.; Macready, W.G. No Free Lunch Theorems for Optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]