applied sciences

Article

WS-AWRE: Intrusion Detection Using Optimized Whale Sine Feature Selection and Artificial Neural Network (ANN) Weighted Random Forest Classifier

Omar Abdulkhaleq Aldabash * and Mehmet Fatih Akay

Department of Computer Engineering, Faculty of Engineering, Çukurova University, Adana 34000, Turkey; mfakay@cu.edu.tr
* Correspondence: omarabd1970@gmail.com

Abstract: An IDS (Intrusion Detection System) is essential for network security experts, as it allows one to identify and respond to abnormal traffic present in a network. An IDS can be utilized for evaluating the various types of malicious attacks. Hence, detecting intrusions has become a significant research area in the contemporary era, especially with the evolution of technologies. With the progress of ML (Machine Learning)-based algorithms, researchers have striven to perform optimal ID. However, most of these studies lag in accordance with their accuracy rate. Thus, to attain a high accuracy rate in ID, the present study proposes ML-based meta-heuristic algorithms, as these approaches possess innate merits of determining near-optimal solutions in limited time and are capable of dealing with multi-dimensional data. The study proposes OWSA (Optimal Whale Sine Algorithm) for selecting suitable and relevant features. With an exclusive optimization process using the SCA (Sine Cosine Algorithm), this study proposes to combine SCA with WOA (Whale Optimization Algorithm) for mitigating the demerits of both, with its hybridization thereby achieving OWSA. Following this, AWRF (Artificial Neural Network Weighted Random Forest) is proposed for classification. The main intention of this process is to propose a weight-updating process for discrete trees in the RF model. The proposed approach is motivated by avoiding overfitting and attaining stability and flexibility. This approach is assessed with regard to performance via a comparative analysis, so as to uncover the best performance of this proposed technique in ID.

Keywords: intrusion detection; machine learning; Sine Cosine Algorithm; Whale Optimization Algorithm; Artificial Neural Network; Random Forest

check for updates

Citation: Aldabash, O.A.; Akay, M.F. WS-AWRE: Intrusion Detection Using Optimized Whale Sine Feature Selection and Artificial Neural Network (ANN) Weighted Random Forest Classifier. *Appl. Sci.* **2024**, *14*, 2172. https://doi.org/10.3390/app14052172

Academic Editors: Chilukuri K. Mohan and Mustafa M. Matalgah

Received: 25 October 2023
Revised: 21 February 2024
Accepted: 27 February 2024
Published: 5 March 2024

1. Introduction

In recent years, the advancement of technologies, namely, modern Industrial Control Systems, Cloud Computing, IoT (Internet of Things) and other web applications, has been rapidly increasing. These systems often handle large volumes of information and work in symbiosis in complex and vast communication networks. This results in intrusions by various ill-intentioned entities and hackers, who find new ways to break computer systems [1]. So, it will be important to improve the security of these network systems. Generally, an IDS monitors network traffic so as to detect suspicious activities, and creates an alert when these activities are found. It represents a software application that scans a system or network for harmful events or policy-breaching. Violations or malicious ventures are typically reported to a proprietor, or centrally gathered with the use of SIEM (Security Information and Event Management). An SIEM system collects the results from multiple sources and utilizes alarm filtering methodologies for differentiating malicious events from FAR (False Alarm Rates). Though IDS can identify potential malicious activities, it is also liable to FARs. Thus, organizations need to fine-tune their IDS products once they have already been installed. This suggests that establishing the proper settings of an IDS for recognizing threats in network traffic is of the utmost importance, and these setting can be categorized based on the detection technique employed [2]. Moreover, in the area of cyber-security, ML methodologies have been productively implemented in developing effective approaches. These ML-based techniques show extensive potential to identify various types of intrusions, perform malware classification, protect privacy, evade cyber-attacks etc. Hence, ML has become a valuable tool for defenders. Contemporary threats have evolved to a sophisticated and complex level with the rapid evolution of adversarial methodologies. Most of the present security technologies will encounter various threats. Consequently, self-learning methodologies should be used, as they are capable of handling these issues. Accordingly, ML tools have emerged as crucial across the whole security sector. ML is an area of AI associated with data mining, data science and computational statistics. It is predominantly concerned with training machines to learn efficiently from input data. Hence, ML is considered a data-driven method, the initial phase of which is understanding the raw data to construct smart security frameworks and generate forecasts [3].

Correspondingly, several studies have endorsed the use of various models for enhancing the performance of IDSs with the use of diverse methods, such as wrapper, filter, optimization, and bio-inspired algorithms, etc. Specifically, bio-inspired algorithms are utilized for improving the performance of network IDSs via their ability to find the most efficient solutions in less time. Each of the bio-inspired algorithms possesses its own merits and demerits. Using hybrid methods, each of the algorithms employed could adopt the strengths of the other, as well as resolving its demerits [4]. Several recent studies have also found that hybridization enhances the performance of bio-inspired methods. In accordance with this, Ref. [5] used an efficient method to detect and classify attacks based on a UNSW-NB15 dataset. For improving accuracy, feature selection has been developed with the use of the OSS (Optimized Sine Swarm) approach, which chooses significant features. Lastly, ID has been performed with OSS-RF (OSS-Random Forest). The performances of classifiers deployed have been assessed by use of various evaluation metrics. Comparisons have been made with various conventional models for confirming the efficacy of this approach. The endorsed system achieved a better performance, with 98.15% accuracy. Following this, the research is [6] aimed to enhance the classification rate and minimize the redundancy arising in feature selection during intrusion detection. To achieve this, an improved BWO-BOA (Black Widow Optimization–Butterfly Optimization Algorithm) was utilized, which employs a dynamic adaptive search strategy for improving the actual BOA and solving the issues of low precision, easy entrapment in the local optima, and slow convergence. Satisfactory results have been attained.

Additionally, Ref. [7] applied a hybrid model to perform feature selection and thus accomplish ID. The execution model utilized PSO (Particle Swarm Optimization) and GWO (Grey Wolf Optimization). Further, two models were used, namely, PSO-GWO-ANN (PSO-GWO-Artificial Neural Network) and PSO-GWO-NB (PSO-GWO-Naïve Bayes). Frequently repeating features from the suggested model were assessed. PSO-GWO was run for a specific iteration count. An individual feature selection framework was run independently, and the chosen feature set was saved. The features of PSO-GWO were tested in the subsequent phase. The UNSW-NB15 dataset was utilized for evaluation. Further, experimentations were executed with two classifiers, ANN and NB. The outcomes reveal that PSO-GWO shows acceptable performance in selecting the features of ID. Moreover, the combination of the features of PSO-GWO afforded better results with fewer features. Initially, common features of attack and normal behavior were eliminated. The RF algorithm was utilized to assess the significance of predictors for ID. Subsequently, RF was employed to work with minimal selected predictors for classifying attack behavior and normal users. Lastly, classifications were considered for intruder detection. Experimentations were undertaken, and it was confirmed that the classifier's performance could be enhanced with regard to accuracy with the use of RF [8]. In spite of several attempts having been made to use existing models, there is scope for a further improvement in accordance with accuracy. Moreover, only a few studies have considered three datasets when performing ID. Hence, to alleviate such pitfalls, the current study proposes appropriate ML-based methods for effective IDS detection.

The main objectives of this study are:

- To pre-process the data via the identification of missing values, and the scaling and filling of the missing values so as to accomplish effective prediction;
- To select suitable and relevant features using the proposed OWSA (Optimal Whale Sine Algorithm) for enhancing classifier performance;
- To predict the intrusions by deploying the proposed AWRF (ANN Weighted Random Forest) to attain a better prediction rate;
- To assess the performance rate of the proposed work with regard to evaluation metrics for proving the effectiveness of this system.

Paper Organization

The paper is structured as follows: Section 2 contains the review of conventional systems, with problem identification. This is followed by Section 3, with the proposed flow, algorithm, and relevant steps. The results procured via the execution of the proposed system are presented in Section 4. Lastly, the overall research is concluded in Section 5 with the outlining of possible future works.

2. Review of Existing Work

Existing studies have struggled to use different ML-based methodologies for intrusion detection. These methods are discussed in this section, with the identification of their problems.

2.1. Intrusion Detection with NSL-KDD Dataset

Existing studies have utilized several methods, such as ML algorithms [9,10], swarm-intelligence and ANN (Artificial Neural Network), for detecting intrusions. Accordingly, Ref. [11] suggested a feature selection method using a GA (Genetic Algorithm), wherein the GA finds the ideal features from an NSL-KDD dataset. Furthermore, hybrid classifiers have been used, encompassing LR (Logistic Regression) and DT (Decision Tree) to achieve the ideal accuracy and detection rate. The study has also employed and compared the performances of varied meta-heuristic methods in optimizing the chosen ideal features. The empirical outcomes reveal that the GWO (Grey Wolf Optimization) method showed a better accuracy rate of 99.44%. Moreover, the integration of several feature-selection-based approaches and the ML methods has been pursued for efficient intrusion detection. Initially, the most relevant features were retrieved with the use of the hybrid meta-heuristic algorithm, and later, supervised ML methods were employed for detecting various attacks with a satisfactory accuracy rate. Accordingly, the NSL-KDD dataset was used with the suggested AdaBoost model. A cost-sensitive classifier has been employed to enhance the accuracy of the minority class, affording an 81.1% overall accuracy [12]. Further, Ref. [13] considered a method relying on the HOA (Horse Herd Optimization Algorithm) to detect network intrusions. To perform the classification, KNN (K-Nearest Neighbour) was employed. A suitable performance was secured. To improve the performance further, Ref. [14] employed a hybrid meta-heuristic approach with an OWKELM (Optimal Wavelet Kernel Extreme Learning Machine)-based classifier. The suggested system included a hybrid MFO (Moth Flame Optimization) with HC (Hill Climbing)-based process for feature selection. Extensive simulations have been undertaken on the NSL-KDD dataset, and the better performance of the suggested model was revealed. Besides this, intrusion detection has been performed in several applications using the NSL-KDD dataset [15]. For instance, Ref. [16] set out a model to detect and classify IoT network intrusions for use in agriculture. NSL-KDD was employed as the input dataset. Feature extraction was performed with PCA (Principal Component Analysis). For the classification of the dataset, ML methods such as SVM (Support Vector Machine), RF (Random Forest) and LR (Linear Regression) have been employed. The performances of the suggested ML classifiers have been assessed. The accuracy rate was found to be 98% when using SVM, 78% when employing LR and 85% when using RF. Contrarily, Ref. [17] utilized an optimized strategy based on GB-EGWO (Genetic-Based–Enhanced Grey Wolf Optimization) to detect intrusions. The simulation outcomes confirm the better performance of the suggested system. Likewise, Ref. [18] employed a bio-inspired method, namely, GWO, to improve its effectiveness in detecting anomalous and normal traffic. Additionally, the ELM (Extreme Learning Machine) has been considered for use as a classifier, and modified GWO has been used for tuning the parameters of ELM. The accuracy rate has been found to be 81%. Following this, Ref. [19] considered EBGWO (Enhanced Binary Grey Wolf Optimization) to perform feature selection by balancing the parameters so as to detect anomalies. Evaluations of this method have been performed on the NSL-KDD dataset, containing varied attack classes, in comparison to other benchmark methods such as Binary PSO, Binary Bat Algorithm, and four variants of GWO, in performing feature selection. The empirical outcomes show that EBGWO achieved a better performance than supplementary classifiers, with a classification rate of 87.46%. To improve the performance rate further, Ref. [20] suggested using GSO (Glowworm Swarm Optimization) with PCA (Principal Component Analysis). The detection rate was 94.08%.

2.2. Intrusion Detection with CICIDS Dataset

The study in [21] tried to enhance the classification rate and avoid FPs (False Positives). To accomplish this, ideal features were selected. An intrusion detection model has been suggested based on ML methods such as DTs (Decision Trees), SVM and RF (Random Forest). After the model's training, an ensemble method-based voting classifier was integrated, which attained a 96.25% accuracy rate. Further, the suggested system also incorporated explainable AI, which contributed to attaining better outcomes. Moreover, Ref. [22] assessed the performance of the CICIDS-2017 dataset when employing various ML classifiers, such as CNN (Convolutional Neural Network), RF (Random Forest) and NB (Naïve Bayes). Among these other classifiers, RF showed satisfactory performance. Unlike CICIDS-2017, some works have also used the CICIDS-2018 dataset [23]. Accordingly, the study in [24] employed an approach that improves time effectiveness and enables better intrusion detection. The LightGBM (Light Gradient Boosting Machine) has been utilized for constructing a model that accords with the considered dataset. The suggested system achieved 97.73% accuracy. As an enhancement, Ref. [25] integrated the NTLBO (New Teaching Learning Based Optimization Algorithm), ELM (Extreme Learning Machine), LR (Logistic Regression) and SVM to perform the selection of subsets of features. Experiments have been undertaken on identifying intrusion from ML datasets, such as CICIDS-2017, wherein significant improvements have been made, reaching 97% accuracy.

Similarly, Ref. [26] used an enhanced intrusion detection system to perform binary classification. This work included various optimizers, such as Rao Optimization, SVM, LR and ELM, as well as hybrid Rao–SVM, with supervised ML methods for the selection of feature subset. The selection of fewer features without a loss of accuracy in selecting feature subsets has been regarded as the primary aim of optimization. The developed model showed a 97% accuracy rate when applied to the considered CICIDS dataset. Similarly, Ref. [27] used the CICIDS dataset. Duplicate information was eliminated. Subsequently, with the use of specific software, the dataset was reformatted into a compatible form with the WEKA tool. Subsequently, eight ML algorithms (AdaBoost, MLP (Multi-Layer Perceptron), NB, KNN (K-Nearest Neighbor), J48, QDA (Quadratic Discriminant Analysis), ID3 (Iterative Dichotomizer 3) and RF) were utilized. After assessing the efficacy of the suggested system, the j48 method was chosen. Furthermore, in accordance with the outcomes of the j48 classifier, expert rules were created that could respond to intrusions in the network.

Furthermore, IDSs have been developed with embedded expert rules. These expert rules were executed in the snort-rule language. The snort operates via the network IDS method, and rules files were accessed as described in the snort configuration. This system achieved a better accuracy rate of 98%. Likewise, Ref. [28] employed a bottleneck layer approach to the CICIDS-2107 dataset so as to confirm its better performance. This performance was achieved with the incorporation of DL algorithms, such DNN (Deep Neural Network) and ANN, and has been compared with those of conventional DNN, SVM and ANN. Experimentation revealed the 92.35% accuracy of DNN, while the accuracy rate was found to be 90.98% when using ANN. In addition, Ref. [29] utilized SVM. Two other methodologies have also been benchmarked—NB and DT. The results reveal that SVM reached 76.47% accuracy when applied to the CICIDS dataset, whereas DT reached 63.71% accuracy, while NB reached 41.58% accuracy.

2.3. Intrusion Detection with UNSW-BOT Dataset

The study in [30] presented a BOT-IoT dataset that incorporates network traffic and IoT-related traffic with several attack kinds. The dataset was developed in real time, and it has been categorized with label features that represented the flow of attack, the category of attack, and sub-categories for the probable intentions of multi-class classification. Further, additional features have been generated to enhance the predictive ability of the classifiers that were trained on the suggested model. With statistical evaluation, a subset of the actual dataset was generated encompassing ten ideal features. Lastly, four metrics have been utilized in assessing the dataset—precision, fall-out, recall and accuracy. A high accuracy rate and high recall rate were found for the SVM model. The recommended models could be further optimized so as to attain better outcomes. Accordingly, DL-based models have been considered. Correspondingly, the study in [31] designed a system to detect intrusions in accordance with DL for revealing IoT-DDoS (Distributed Denial of Service) Botnet attacks. The considered dataset was developed and designed within a real-time environment. Traffic data that were incorporated included attack and normal traffic data. The highly extendible DNN has been recommended for use in robust attack detection in IoT networks. Analysis has revealed that DNN works better than conventional algorithms, with a better precision and accuracy rate. Following this, Ref. [32] developed a training method for tuning the parameters of the utilized deep model. To achieve this, the NSBPSO (Neighborhood Search-based PSO) has been utilized to improve the exploration or exploitation of PSO. The merits of NSBPSO have been utilized for the optimal training of deep models in order to attain better predictions. To assess the performance, the BOT dataset and UNSW-NB-15 dataset have been used, and better outcomes have been procured.

2.4. Problem Identification

Recently, ML- and meta-heuristic-based approaches have found extensively use in resolving the issues related to intrusion detection. Various issues have been identified through the review of conventional works, and they are discussed in this section, as follows:

- Existing models have concentrated on determining the impacts of the network dataset. They have suggested using the ML model to improve prediction performance. They have also suggested the development of schemes related to feature engineering to obtain a higher accuracy [9];
- Though Ref. [10] sought to perform intrusion detection based on ML, the system they suggested did not employ any feature selection methodologies;
- Conventional models have used the NSL-KDD dataset. Accordingly, Ref. [12] employed a cost-sensitive classifier, and achieved 81.1% accuracy. Similarly, Ref. [18] used Modified GWO, with an accuracy rate of 81%. Moreover, Ref. [19] employed EBGWO and achieved a classification accuracy of 87.46%. In addition, Ref. [20] utilized GSO-PCA and reached 94.08% accuracy. In addition, other research works have also considered the CICIDS-2017 dataset. In accordance with this, Ref. [21] used an ensemble method that achieved a 96.25% accuracy rate. Further, Ref. [25] used TLBO-GA, and achieved a 97% accuracy rate. Furthermore, Ref. [26] employed Rao–SVM and achieved 97% accuracy. Though better performances have been achieved, there is scope for further improvement;

- The article [13] suggested that future works could consider employing various semi-supervised and meta-heuristic approaches in intrusion detection so as to accomplish a high accuracy rate and low computational complexity when detecting extensive attacks in CNs (computer networks).

3. Proposed Methodology

This study proposes an IDS model developed using ML-based approaches. Though existing studies have attempted this, the systems developed have been deficient with regard to accuracy. To resolve the prevailing issues, the current research considers three datasets (UNSW-Bot, NSL-KDD and CICIDS-2107) and performs a series of steps, as shown in Figure 1. Initially, pre-processing is performed, whereby inconsistent and missing data are eliminated so as to avoid error. This process also helps enhance the dataset's quality and accuracy by making the data more consistent and reliable. After pre-processing, features are selected using OWSA. Feature selection is performed by considering the diversity and number of features of user behavior and network traffic, and the selection of a subset of features is performed to improve the accuracy of model classification. An approach to selecting server traffic feature subsets based on OWSA has been used to determine the features that are the most important and related to the class label. This process contributes to minimizing computational costs and enhancing the performance rate. After feature selection, the chosen data are used in the training and testing phases, with a ratio of 80:20. The use of 80% of the data for training involves presenting a labeled dataset to the model, and allowing it to learn patterns and relationships between input features and corresponding labels. The proposed model adjusts its parameters through optimization techniques, aiming to make accurate predictions. Finally, regarding the 20% of data used as a testing set, we evaluate the model trained on a separate dataset in order to calculate the performance of the model. In the final phase, classification is accomplished with AWRF. The model is evaluated with regard to performance so as to confirm its efficacy.

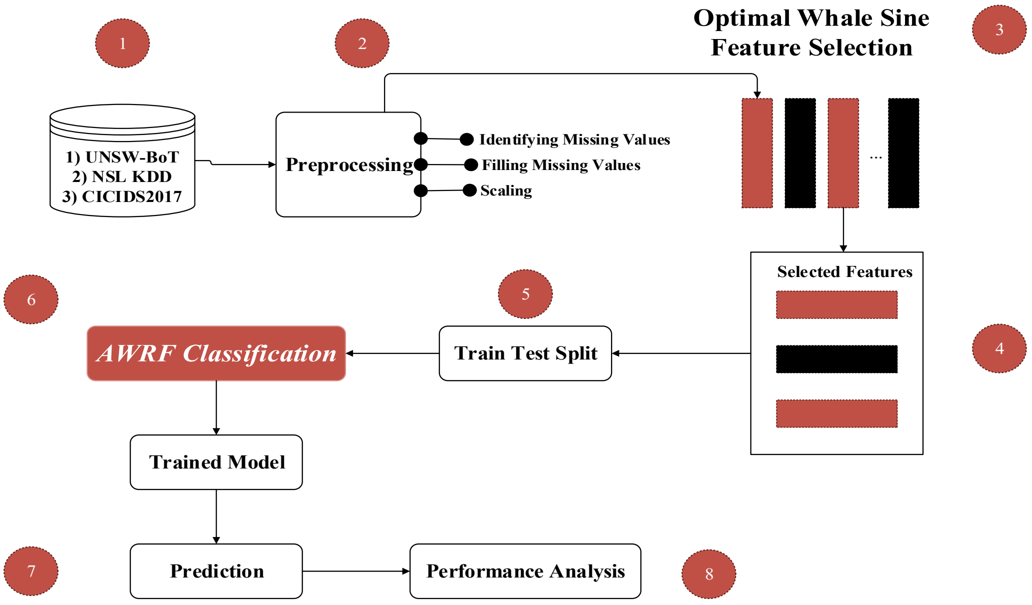


Figure 1. Overview of the proposed methodology.

3.1. OWSA (Optimal Whale Sine Algorithm) for Feature Selection

Initially, the WOA (Whale Optimization Algorithm) was inspired by the bubble-net hunting mannerism of humpback whales. The population-based WOA possesses the capacity for avoiding local optima, thereby attaining optimal solutions. Such merits could identify the WOA as a suitable approach to solving various unconstrained or constrained optimization issues related to practical employment without reformatting the structure of the algorithm. Humpback whales produce huge spirals of bubbles as they approach closer to their prey top contain them. Then, the prey is hunted. In the hunting phase, the humpback whales pursue two predation methodologies, so as to minimize the ensuing steps. Subsequently, the whales will enact the spiral upraise-position technique. Throughout the hunting phases, both of these methodologies are utilized concurrently. Accordingly, two bubble-related techniques are employed—upward spirals and double loops. In the first stage, the humpback whales dive twelve meters down; following this, bubbles are produced that surround the prey and float upwards in a spiral form. The subsequent step includes three phases—lob-tail, coral loop and capture loop.

Phase 1: Encircling the prey—The humpback whales lactate the prey and then surround them. With the ideal design position within the search area already known, the WOA selects a candidate outcome that targets prey, or else it moves close to the optimal. Then, better search agents are described. These search agents attempt to direct the search towards the ideal agent.

Phase 2: Exploitation—Bubble-net attack. This process encompasses two main methodologies, namely, shrinking encircling and spiral updating;

Phase 3: Exploration—Aside from bubble net processing, humpback whales search for their prey in a random manner.

For pursuing the two approaches, the WOA uses random choice possibility ($x\ \left(x^2[0,\ 1]\right)$). Moreover, when ($x$ < 0.5), the humpback whale makes use of the spiral upraising position methodology. On the contrary, when (x > 0.5), the humpback whale uses the methodology of shrinking encircling. During shrinking encircling, whales search for their prey with consideration of each other’s locations. To reflect the uncertainty in this algorithm, $\overrightarrow{A}$ is presented as the co-efficient vector. The search agent is set as $\left|\overrightarrow{p}\right| < 1$ to update the area of the search agents. As this process involves several search strategies, its effect is limited in identifying the global ideal outcome with the highest merit in comparison to conventional optimization. A mathematical model of the search distances of position vectors is given in Equation (1),

$$\left|\overrightarrow{r}\right| = \overrightarrow{q}.\overrightarrow{z}^{*}(t) - \overrightarrow{z}(t) \tag{1}$$

In Equation (1), t denotes the existing iteration count, $(\overrightarrow{z})$ and (z^{*}) represent the position vectors of the ideal outcome, $\overrightarrow{r}$ represents the search distance and $\overrightarrow{q}$ denotes the coefficient vector. Equation (2) updates the whale’s movement (location) around the victim, which can be described in a mathematical form as follows:

$$\overrightarrow{z}(t+1) = \overrightarrow{z}^{*}(t) - \overrightarrow{A}.\overrightarrow{r} \tag{2}$$

The random search agent is given by Equation (2),

$$\overrightarrow{z}(t+1) = \overrightarrow{z}_{rand} - \overrightarrow{p}\left(\overrightarrow{r}\right) \tag{3}$$

In Equation (3), $\overrightarrow{z}_{rand}$ represents the random position vector of the current population, while $\overrightarrow{p}$ indicates the coefficient vector. Humpback whales also use the spiral position updating technique to hunt. The spiral positioning of the whale is given in Equation (3),

$$\overrightarrow{z}(t+1) = \overrightarrow{r}'\left(e^{bl}\right)(cos(2\pi l)) + \overrightarrow{z^{*}}(t) \tag{4}$$

In Equation (4), $\overrightarrow{r}' = \left|\overrightarrow{z^{*}}(t) - \overrightarrow{z}(t)\right|$ reveals the distance of the ith whale from the prey (satisfactory outcome gained here), and b represents the constant defining the logarithmic spiral format. $\overrightarrow{z}(t+1)$ is the coefficient vector. $\overrightarrow{r}'\left(e^{bl}\right)(cos(2\pi l)) + \overrightarrow{z^{*}}(t)$ denotes the coefficient scalar. l is a random digit (in (−1, 1)). The stepwise procedure for WOA is given in Pseudocode 1.

Pseudocode 1: WOA (Whale Optimization Algorithm)

```
Modify the whale populace $z_i$ $(i = 1, 2, \ldots., n)$
Analyze the value of fitness function
Randomly select the search agent $z^*$
Initialize $t = 1$ and $t <$ maximum iterations
  for every search agent
  Modify $a,\ p,\ q,\ l,$ and $x//a$ represents linearly reduced within the range of 2 to 0 over the
    course of iteration, $p$ and $q$ represents coefficient vectors, $l$ reresents absolute value
    and $x$ represents the constant for explaining the shape of the logarithmic spiral.
    if 1 $(p < 0.5)$
      if 2 $(|a| < 1)$
Update the position of current solution by Equation (2)
        else if 2 $(|a| \geq 1)$
Update the position of current solution by Equation (3)
        end if 2
      else if 1 $(p \geq 0.5)$
        Random solution is generated
      Update the position of the current solution by Equation (4)
    end if 1
  end for
if some search agent drives away from the search space or else amend it
Analyze the value of fitness function
Modify $z^*$ if it is better
$t = t + 1$
end while
return $z^*$
```

Based on the pseudocode, the whale population is altered and the fitness values are assessed. Thus, the search agent is modified and the ideal search agent is finally attained. On the contrary, the SCA (Sine Cosine Algorithm) is considered; the trigonometric operations of SC are the basis for this algorithm. Typically, SCA shows a better acceleration and convergence rate. It also shows a reliable implementation time. Several initial random solutions are generated by the SCA. It also helps in transferring the ideal solution that employs a mathematical framework in accordance with the functionalities of SC. Varied random and adaptive variables are integrated within this algorithm to maintain the search spaces in the optimization processes. The population search approach and local search approach are the major techniques in SCA. This algorithm shows certain innate advantages, such as simple execution and flexibility, due to which it could be utilized for solving several optimization issues. These features have enabled SCA to resolve various optimization issues. Taking into account the n-dimension optimization issue,

$$\underset{f}{min}(a_1,\ a_2 \ldots a_n) \tag{5}$$

$$l_i \leq a_i \leq u_i;\ i = 1,\ 2 \ldots n \tag{6}$$

In Equation (5), a_i indicates the ith-decision variable, l_i denotes the lower bound, and u_i represents the upper bound. Moreover, n indicates the problem dimension. In Equation (5), SCA utilizes the oscillatory functions of both functions of SC, which alters the capacities of individuals to observe the global ideal solution. The precise process is given below.

An assumption is made such that, in the SCA, the population size is termed N, and the ith individual's position in the tth generation is represented as $\left(a_i^t = a_{i1}^t + a_{i2}^t, \ldots, a_{in}^t\right)$, wherein $i = 1, 2, \ldots N$. In addition, the fitness value is computed for individuals, and the position of the ideal individual is recorded as a_*^t $\left(a_*^t = arg - min\left(f_{a_i^t}\right)\right.$. The *j*th dimension for the *i*th individual in the populace is updated in Equation (7),

$$a_{ij}^t(t+1) = \begin{cases} a_{ij}^t + d_1(sin(r_2)).\left|d_3\left(a_{*j}^t - a_{ij}^t\right|;\ d_4 < 0.5 \\ a_{ij}^t + d_1(cos(r_2)).\left|d_3\left(a_{*j}^t - a_{ij}^t\right|;\ d_4 \geq 0.5 \end{cases} \tag{7}$$

In Equations (6) and (7), $d_3 \in (0,\ 2\pi)$ and $d_4 \in (0,\ 1)$ indicate the random count of the uniform distribution, wherein d_1 indicates the control parameter. The computation process is given in Equation (8),

$$d_1 = x\left(1 - \frac{t}{t_{max}}\right) \tag{8}$$

In Equation (8), x is the constant, t represents current iterations and t_{max} is the maximum iteration count. Further, let $b_*^{t+1} = arg - min\left(f_{a_i}{}^{t+1}\right)$; this equation employs acquisitive searching for attaining $f_{a_i}{}^{t+1}$, which is represented via Equation (8),

$$\begin{array}{c} a_*^{t+1} = b_*^{t+1} \qquad f\left(b_*^{t+1}\right) < f\left(a_*^t\right) \\ \mathrm{a}_*^{\mathrm{t}} \text{ otherwise} \end{array} \tag{9}$$

Furthermore, let $t = t + 1$, and the process (in Equations (6)–(8)) repeats until the termination condition is attained. The overall procedure of SCA is given in Pseudocode 2.

Pseudocode 2: SCA (Sine Cosine Algorithm)

```
Randomly initialize search agents (solutions)
Do
  Estimate each search agents by objective function
  Update the solution attained so far ( $p = a.$ ) by using Equation (8)
  Update the control parameter $d_1$
  Update the position of search agents by using Equation (9)
  While ($t$ < $maximum\ number\ of\ iterations$)
Return the best solution attained so far as the global optimum
```

The initial phase involves the initialization of search agents. With the use of the objective function, search agent evaluation is undertaken. With the use of Equation (8), the ideal solution is updated and the control parameter d_1 is updated. With the use of Equation (6), the positions of search agents are updated. This process continues until the maximum iterations are reached and the ideal solution is attained. Periodically, SCA faces issues such as getting stuck in a local area in the search space. This impacts the computational exertion that is required in searching for the ideal solution within the search space. The above issues could be resolved via enhancements to standard SCA that in turn enhance the SCA performance.

On the contrary, WOA also possess disadvantages such as easy localization and low convergence. Hence, the present study endeavors to resolve these issues via several enhancements so as to procure better accuracy. With the exclusive optimization standard of SCA, this study intends to combine WOA and SCA so as to limit the demerits of both via hybridization, thereby attaining OWSA. As hybridization involves an enhancement of optimization methodologies, the present research proposes a OWSA wherein operators from certain methods are combined with other operators from supplementary methods so as to generate efficient and reliable results. In the current study, the OWSA is proposed for the optimal selection of features. It possesses the greatest ability to enhance the exploration phase. The overall process of optimal feature selection using OWSA is shown in Figure 2.

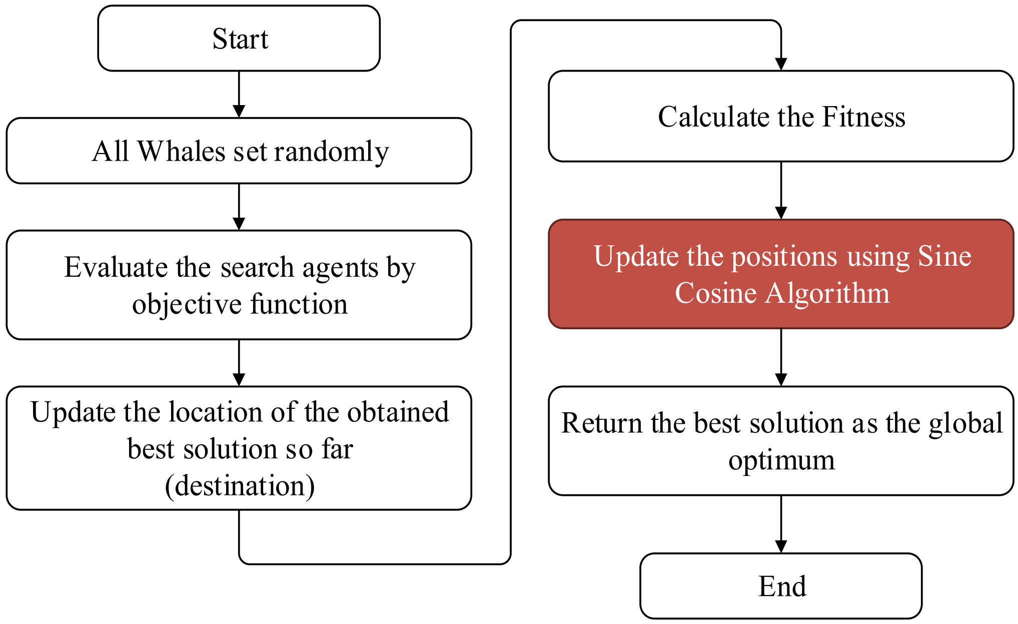

Start
All Whales set randomly
Evaluate the search agents by objective function
Update the location of the obtained best solution so far (destination)
Calculate the Fitness
Update the positions using Sine Cosine Algorithm
Return the best solution as the global optimum
End

Figure 2. Optimal selection of features with OWSA.

As depicted in Figure 2, all the whales are randomly set. Following this, the search agents are evaluated with the use of the objective function. Subsequently, the destination location is updated. Then, the fitness computation is performed. A position update is performed with SCA. Finally, the global optimal solution is attained. The overall sequence is presented in Pseudocode 3.

Pseudocode 3: OWSA (Optimal Whale Sine Algorithm)

```
Modify the whale populace $z_i$ ($i$ = 1,2,. . .. , $n$)
Analyze the value of fitness function
Randomly select the search agent $z^*$
Calculate objective function for each ai
                              **Initialize** $t = 1$ and $t <$ maximum iterations
    **for** every search agent
      *Modify* $a,\ p,\ q,\ l,$ *and* $x$
      **if 1** $(p < 0.5)$
         **if 2** $(|a| < 1)$
Update the position of current solution using Equation (2)
            **else if** 2 $(|a| \geq 1)$
Update the position of current solution using Equation (3)
         **end if 2**
         **else if 1** $(p \geq 0.5)$
         Update the position of the current solution using Equation (4)
      **end if 1**
    **end for**
if some search agent drives away from the search space or else amend it
Analyze the value of fitness function
Calculate Objective function for each ai
Modify $z^*$ if it is better
$t = t + 1$
**end while**
**return** $z^*$
```

3.2. AWRF (ANN Weighted Random Forest) for Classification

Generally, ANN refers to the biologically inspired sub-field of AI modeled after the brain. Typically, ANN is a computational network relying on biological NNs (Neural

Networks), which mirror human brain's structure. Like the human brain, this NN possess neurons interconnected with one another, and ANNs possess neurons that are associated with one another in several network layers. Further, RF (Random Forest) is a well-renowned ML method that relies on a supervised learning method. The RF's function relies on the idea of EL (Ensemble Learning), whereby multiple classifiers are integrated to solve complex issues, thereby enhancing the model's performance. The main intention of the current study is to propose a weight updating process that is applicable to an individual tree in the RF model, and the comprehensive evaluation of ideal parameter tuning. The proposed approach is recommended by its stability, flexibility and avoidance of over-fitting. The overall process is shown in Figure 3.

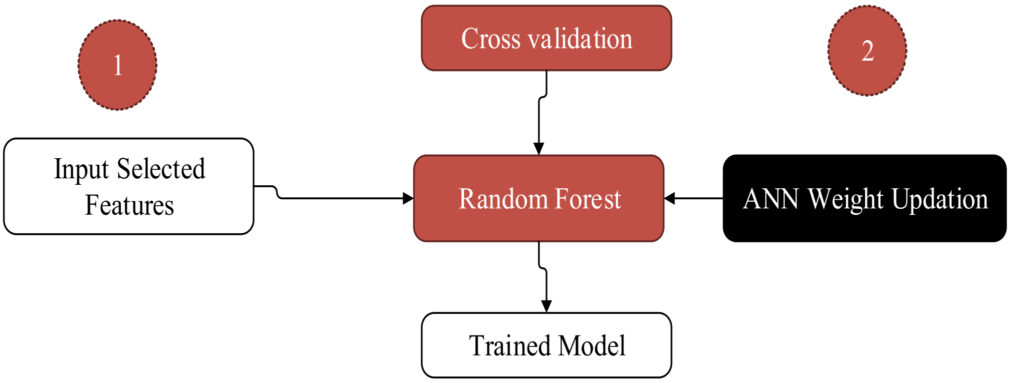


Figure 3. Overall process of AWRF.

As shown in Figure 3, the selected features are fed into RF. Based on weight updating, the RF generates optimal outcomes. RF encompasses numerous DTs for several subsets of the dataset, and by averaging, it enhances the prediction rate of a specific dataset. The RF model determines the predictions of all the DTs (Decision Trees), and the overall outcomes are predicted in accordance with the prediction that receives the most votes. Besides this, weight in the ANN indicates a parameter that possesses converted input data in the network's hidden layers. The resulting layer repeatedly tunes the input within the hidden layers to produce a desirable count in the specific range, as the less weighted value does not alter the input. However, on the contrary, high-weighted values are applied, causing significant alterations in the results. Thus, the chosen features are fed into RF. Concurrently, the ANN weight updates are fed into RF. Several input data subsets are employed for training the ML models. DT acts as the core element of the RF model. From the actual data, a set of DTs related to the bootstrap samples is generated. Usually, the bootstrapping method assists the RF with the collection of sufficient DT counts, which enhance the classification rate via the overlap-thinking concept. With the voting approach, optimal trees are chosen by bagging. The chosen features are subjected to cross-validation. These features are also fed into RF. Lastly, the ANN weight updating process assist the RF model in performing effective classification. The overall process is presented in Pseudocode 4.

To receive effective outcomes with the contribution of the considered classifier, all the input is weighted. The solution relies on majority voting. Each of the quadrant solutions is compared with the overall solution. When there is a match, the fixed value (β_+) is added to the weight. Contrarily, when it does not match, (β_-) is taken from the classifier's weight, which minimizes the negative impact on further processes. The algorithm is thus updated with persistent weights, and hence the system can be subjected to internal and external alterations. This ensures the classifier's reliability. It is significant for pre-defining the initial weight value φ_{0i}, wherein $i = 1,\ 2\ldots N$, and N denotes the classifier count. Values have a huge influence on the evolution of the system, as the system's reliability is in accordance with the weights of ANN.

Pseudocode 4: AWRF (ANN Weighted Random Forest)

```
Inputs:
Votes: array of votes from each source
$X$: array of ANN weights for each source
Outputs:
decision: category voted by the algorithm
$X \leftarrow \phi$; $\leftarrow\rightarrow$ Initialize ANN weights
for every new measure do
  ANN weighted majority:
  for all $k$ $\epsilon$ sources do
    Contributions $(k) \leftarrow X(k) * votes\ (k)$
  **end for**
  decision ← max_index(contributions);
  Update ANN weights:
  **for all** k ϵ sources do
    **if** decision = votes(s) then
      $X(k) = X(k) + \beta_{+}$
    **else**
      $X(k) = X(k) + \beta_{-}$
    **end if**
  **end for**
**end for**
```

4. Results and Discussion

The significant results attained via the execution of this proposed work are discussed in this section. Further, the dataset description, metrics and performance analysis, and comparative analysis are outlined.

4.1. Dataset Description

The present research has considered the three datasets such CICIDS-2017, NSL-KDD and UNSW-Bot.

UNSW-Bot: This dataset encompasses various categories of IoT, namely, normal traffic, IoT traffic, and several types of botnet attacks. The IoT-Bot training dataset encompasses 3,037,933 IoT traffic archives, with 1018 normal traffic archives and 3,036,915 attack traffic archives. On the contrary, the test dataset encompasses 3,668,552 IoT traffic archives, consisting of 477 normal traffic archives and 3,668,552 attack traffic archives. Hence, these IoT traffic archives are generated using IoT composed of smart home-based devices, like smart refrigerators, smart lights, remote garage doors, etc. Overall, this dataset comprises three main groups, namely, time-based, byte-based and packet-based. This dataset can be consulted at https://research.unsw.edu.au/projects/bot-iot-dataset (accessed on 1 February 2022).

NSL-KDD: The NSL (Network Security Laboratory) dataset is the advanced form of the DARPA-98 dataset. It is composed of specific records given by the complete KDD dataset. Currently, NSL-KDD is used for the analysis of the efficacy of several classification approaches used in finding abnormalities in the structures of network-traffic. Four kinds of attacks can be found in NSL-KDD, including R2L (Remote to Local), U2R (User to Root), Probe attacks and DoS (Denial of Service). This dataset can be consulted at https://www.unb.ca/cic/datasets/nsl.html (accessed on 15 February 2022).

CICIDS-2017: The Canadian Institute of Cyber-security has developed a conventional dataset termed CICIDS-2017, comprising recent threats and features. This dataset encompasses non-threatening and recent commonly occurring attacks, using actual real-time data. It also includes the results of network traffic enquiries made with a flow meter consisting of destination ports, attacks, protocols and time-stamps. The dataset covers 5 days of data collection, with 2, 25 and 745 sets containing almost 80 features. In this dataset, attacks are divided into seven categories, known as infiltration attack, DoS attack, brute-force attack, web attack, Botnet attack, DDoS attack and heart bleed attack. This dataset can be consulted at https://www.unb.ca/cic/datasets/ids-2017.html (accessed on 1 April 2022).

4.2. Performance Metrics

The main metrics considered for the analysis of the proposed work are discussed in this section.

(A) Precision

Precision is defined as the co-variance of the method, which is attained by suitably comparing cases to the overall cases. It is computed using Equation (10); TrP represents True Positive and FlP indicates False Positive.

$$Precision = \frac{TrP}{TrP + FlP} \tag{10}$$

(B) Recall

Recall is known as a production metric that finds the overall number of accurate positive groups composed of all the optimistic groups, and it is computed using Equation (11); TrP represents True Positive and FlN denotes False Negative.

$$Recall = \frac{TrP}{TrP + FlN} \tag{11}$$

(C) F1-score

This is also termed the F-measure. It is the weighted harmonic value of precision and recall. It is given in Equation (12),

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{12}$$

(D) Accuracy

Accuracy can be understood as the degree of systematic organization, which can be exposed by correctly classifying the groups into the overall dataset. It is computed with the use of Equation (13), where TrN indicates True Negative, TrP represents True Positive, FlN denotes False Negative and FlP indicates False Positive.

$$Accuracy = \frac{TrN + TrP}{TrP + TrN + FlN + FlP} \tag{13}$$

(E) Specificity

This is understood as the quality or state of remaining unique and specific to groups or individuals. It is computed by Equation (14), where TrN indicates True Negative and FlN denotes False Negative.

$$Specificity = \frac{TrN}{FlN + TrN} \tag{14}$$

4.3. Performance Analysis

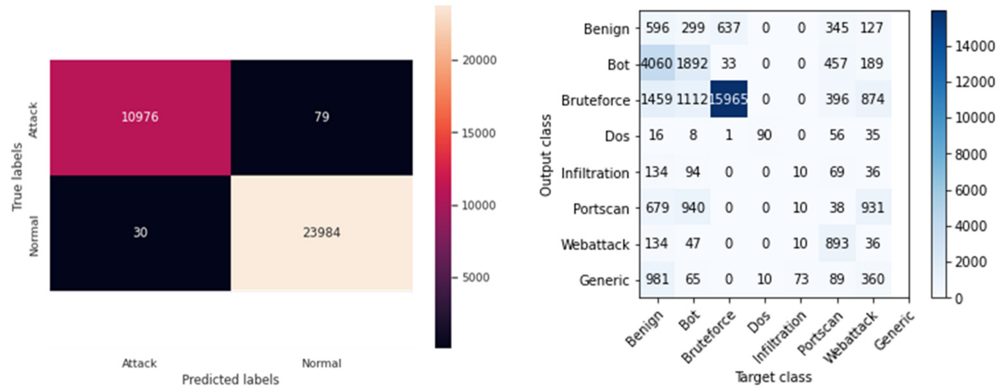
The performance of the proposed system has been evaluated in accordance with F1-Score, Precision, Recall and Accuracy in relation to the three considered datasets. The outcomes are discussed in this section.

4.3.1. Analysis with UNSW-Bot Dataset

Initially, the proposed system was assessed with regard to four standard metrics for theUNSW-Bot dataset, and the corresponding outcomes are given in Table 1, with the relevant pictorial depiction shown in Figure 4.

Table 1. Analysis with the UNSW-Bot dataset.

Accuracy	Recall	Precision	F1-Score
99.783	99.748	99.875	99.705



(**a**) (**b**)

Figure 4. Confusion matrix for (**a**) binary classes and (**b**) different classes in UNSW-Bot.

From Table 1, we see that the proposed system achieved a 99.783 Accuracy rate, a 99.705 F1-Score, a 99.875 Precision rate and a 99.748 Recall rate. Following this, the accuracy values for varied weights in the UNSW-Bot dataset have been assessed, and the analytical outcomes derived are given in Table 2.

Table 2. Accuracy for varied weights in the UNSW-Bot dataset.

Weight	Accuracy
1	68.476
2	68.476
3	99.703
4	99.706
5	99.686
6	68.476
7	99.723
8	68.476
9	99.695
10	99.783

Table 2 shows that the accuracy value differs based on the weight. Accordingly, when weight = 1, the accuracy value is 68.476. This rate eventually increased with the increase in weights, and it reached 99.695 when weight = 9 and 99.783 when weight = 10. Subsequently, the outcomes of k-fold cross-validation for the UNSW-Bot dataset are shown in Table 3.

Table 3. Results of k-fold cross-validation for UNSW-Bot.

Folds	Accuracy
Fold 1	0.9971
Fold 2	0.9973
Fold 3	0.9964
Fold 4	0.997
Fold 5	0.9969

Table 3 shows that the accuracy value differs for each fold. Accordingly, the accuracy value was found to be 0.9971 for fold 1, 0.9973 for fold 2, 0.9964 for fold 3, 0.997 for fold 4 and 0.9969 for fold 5. Hence, the accuracy value differs based on the folds. Further, a confusion matrix for the UNSW-Bot dataset has been constructed so as to expose the correct and misclassification rates of the proposed system. The corresponding outcomes are shown in Figure 4.

A confusion matrix is calculated by comparing the predicted classifications of a model with the actual classifications in a dataset. It is often used in the context of classification problems. The matrix has four components: TP, denoting instances correctly predicted as positive; FP, as instances incorrectly predicted as positive; TN, signifying instances correctly predicted as negative; FN, as instances incorrectly predicted as negative. "Attack classes" refer to distinct categories of cyber threats that likely fit into specific malicious patterns. Evaluating the model's performance involves analyzing numerical values and statistics, including the distribution of attack classes in the dataset, developing a confusion matrix detailing true positives and false positives, and outlining class-specific metrics like precision and recall. These values offer insights into the model's ability to accurately detect various types of cyber-attacks.

In Figure 4, the number of correct classifications of attack classes is shown to be 10,976, while, the correct classification of normal classes numbered 23,984. Concurrently, 30 normal classes have been misinterpreted as attack classes. On the contrary, 79 attack classes have been misinterpreted as normal. As the correct classification rate seems to be higher than the misclassification rate, the proposed system has been confirmed to be effective.

4.3.2. Analysis with NSL-KDD Dataset

The proposed work has been evaluated in accordance with four standard metrics in relation to the NSL-KDD dataset and the equivalent outcomes are presented in Table 4.

Table 4. Analysis with NSL-KDD dataset.

Accuracy	Recall	Precision	F1-Score
99.92	99.38	99.97	99.54

Table 4 shows that the proposed system achieved a 99.92 Accuracy rate, a 99.54 F1-Score, a 99.97 Precision rate and a 99.38 Recall rate. Following this, the accuracy values for different weights in the NSL-KDD dataset have been evaluated, and the acquired analytical results are outlined in Table 5.

From Table 5, we see that the accuracy value changed in accordance with the weight. Correspondingly, when weight = 1, the accuracy value was 99.92. This rate gradually increased and decreased with the increase in weights, and it reached 42.5 when weight = 9 and 99.92 when weight = 10. The results of k-fold cross-validation for the NSL-KDD dataset are explored in Table 6.

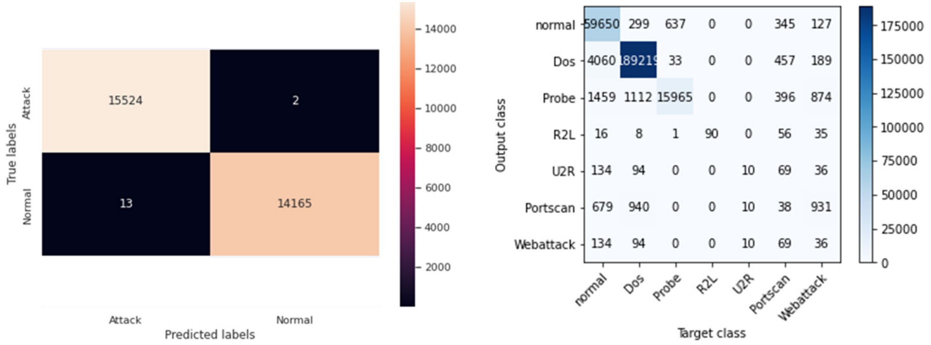
Table 5. Accuracy with varied weights in the NSL-KDD dataset.

Weight	Accuracy
1	99.92
2	99.92
3	42.5
4	99.92
5	42.5
6	99.92
7	99.92
8	42.5
9	42.5
10	99.92

Table 6. Results of k-fold cross-validation for NSL-KDD.

Folds	Accuracy
Fold 1	0.999
Fold 2	0.997
Fold 3	0.985
Fold 4	0.983
Fold 5	0.981

From Table 6, it can be seen that the accuracy value differed for each of the folds. Accordingly, the accuracy value was 0.9375 for fold 1, 0.956 for fold 2, 0.9375 for fold 3, 0.94 for fold 4 and 0.9375 for fold 5. Hence, the accuracy value varied based on the folds. Furthermore, a confusion matrix for the NSL-KDD dataset has been constructed to expose the correct and misclassification rates of the proposed work. The equivalent outcomes are projected in Figure 5.



(**a**) (**b**)

Figure 5. Confusion matrix for (**a**) binary classes and (**b**) different classes in NSL-KDD.

Figure 5 shows the number of correct classifications of attack classes to be 15,524, while the number of correct classifications of normal classes is 14,165. Concurrently, 13 normal classes have been misinterpreted as attack classes. Contrarily, two attack classes have been misinterpreted as normal. In this case, as the correct classification rate appears to be higher than the misclassification rates, the proposed system has been exposed to be effective.

4.3.3. Analysis with CICIDS2017 Dataset

The proposed system has been validated with four standard metrics in relation to the CICIDS2017 dataset, and the corresponding outcomes are presented in Table 7.

Table 7. Analysis with the CICIDS2017 dataset.

Accuracy	Recall	Recall	Precision
0.98	0.97	0.97	0.97

Table 7 shows that the proposed system reached a 0.98 Accuracy rate, a 0.97 F1-Score, a 0.97 Precision rate and 0.97 Recall rate. Then, the accuracy values for diverse weights in the CICIDS2017 dataset were assessed and the analytical outcomes obtained are shown in Table 8.

Table 8. Accuracy for varied weights in the CICIDS2017 dataset.

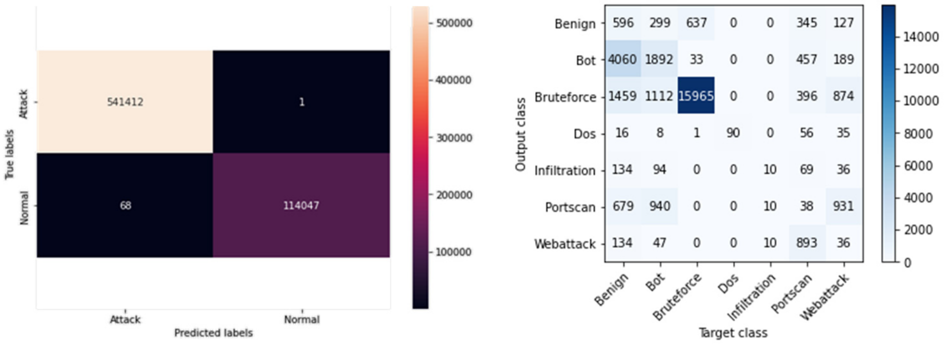
Weight	Accuracy
1	83.5
2	97
3	83.5
4	83.5
5	83.5
6	83.5
7	97
8	97
9	83.5
10	97

From Table 8, it is evident that the accuracy value varied based on weights. Similarly, when weight = 1, the accuracy value was 83.5. This rate progressively increased and decreased with the increase in weights, and it reached 83.5 when weight = 9 and 97 when weight = 10. Following this, the outcomes of k-fold cross-validation for the CICIDS2017 dataset are shown in Table 9.

Table 9. Results of k-fold cross-validation for CICIDS2017.

Folds	Accuracy
Fold 1	0.9375
Fold 2	0.9563
Fold 3	0.9688
Fold 4	0.975
Fold 5	0.98

From Table 9, it can be seen that the accuracy value changed for each of the folds. In accordance with this, the accuracy value was 0.9375 for fold 1, 0.9563 for fold 2, 0.9688 for fold 3, 0.975 for fold 4 and 0.98 for fold 5. Hence, the accuracy value differed based on the folds. Besides this, a confusion matrix for the CICIDS2017 dataset has been built in order to explore the correct and misclassification rates of the proposed system. The results are shown in Figure 6.



(**a**) (**b**)

Figure 6. Confusion matrix for (**a**) binary classes and (**b**) different classes in CICIDS2017.

Figure 6 shows that the number of correct classifications of attack classes was 541,412, while the number of correct classifications of normal classes was 114,047. Concurrently, 68 normal classes were misinterpreted as attack classes. Contrarily, one attack class was misinterpreted as normal. In this case, as the correct classification rate seemed to be higher than the misclassification rates, the proposed system has been inferred to be efficient. The performance analysis showed that the proposed system reached a 99.92 accuracy rate with the NSL-KDD dataset, 0.98 accuracy with the CICIDS2017 dataset and 99.783 accuracy with the UNSW-Bot dataset. In this case, the proposed work has shown high accuracy in relation to the NSL-KDD dataset.

4.4. Comparative Analysis

The proposed system has been compared with conventional methods in relation to the three considered datasets, and the obtained outcomes are presented in this section. The existing models, namely, Global Model, Local Model, Local Outlier Factor, Isolation Forest, GAN-AE (Generative Adversarial Network–Autoencoder), One Class SVM (One Class Support Vector Machine) and K-means, have been compared with the proposed system by considering the UNSW-Bot dataset, and the attained outcomes are presented in Table 10.

Table 10. Comparative analysis with UNSW-Bot [33].

Model	**Precision**	**Accuracy**	**F1-Score**	**Recall**
Global Model	0.9933	0.9711	0.9831	0.9733
K-means	0.9564	0.8801	0.9198	0.8676
One Class SVM	0.9427	0.5301	0.667	0.5274
Local Model	0.9654	0.9297	0.9426	0.921
GAN-AE	0.9874	0.9512	0.9603	0.9364
Isolation Forest	0.9008	0.7998	0.8397	0.8726
Local Outlier Factor	0.8093	0.6063	0.6502	0.4911
Proposed	0.9987	0.9978	0.9971	0.9975

From Table 10, we can see that existing algorithms such as K-means achieved 0.9711 accuracy rate and GAN-AE achieved a 0.9512 accuracy rate, while the Local Model showed a 0.9297 accuracy rate, whereas the proposed model achieved a high accuracy rate of 0.9978. Similarly, the precision, F1-Score and recall rate of the proposed work are better than those of existing methods. Following this, conventional models, namely, C5.0, RF (Random Forest), SVM (Support Vector Machine) and NB (Naïve Bayes), have been compared with the proposed work by considering the CICIDS2017 dataset, and the outcomes are shown in Table 11.

Table 11. Comparative analysis with CICIDS2017 [34].

Model	Accuracy	Recall	Precision
C5.0	0.86457	0.85925	0.99706
RF	0.86803	0.8629	0.9963
NB	0.79996	0.90069	0.86031
SVM	0.79887	0.92445	0.84364
Proposed Model	0.98	0.97	0.97

Table 11 shows that existing methodologies such as SVM reached a 0.79887 accuracy rate, while NB reached a 0.79996 accuracy rate, RF reached a 0.86803 accuracy rate and C5.0 reached a 0.86457 accuracy rate. However, the proposed model achieved a 0.98 accuracy rate. Moreover, the proposed model showed a better performance as regards precision and recall. Finally, a comparison has been undertaken with existing methodologies in relation to the NSL-KDD dataset. The conventional methods that have been considered include CFS + ANN (Correlation Feature Selection + Artificial Neural Network), SVM-RBF (Support Vector Machine–Radial Basis Function), SAE-SVM-RBF (Stacked Auto Encoder–SVM-RBF), CNN-Bi-LSTM (Convolutional Neural Network–Bi-directional Long Short-Term Memory), ensemble model, CART (Classification and Regression Tree), RF, C4.5, NB and MLP (Multi-Layer Perceptron). The respective results are shown in Table 12.

Table 12. Comparative Analysis with the NSL-KDD dataset [35].

Learning Techniques	Specificity	Accuracy
SVM	71.41	69.52
Bi-LSTM	79.64	76.37
CNN	80.75	95.01
MLP	79.57	77.41
C4.5	83.44	81
NB	83.21	81.47
RF	82.35	80.67
CART	82.71	80.3
Ensemble	89.41	87.28
CNN-BiLSTM	80.83	80.05
SAE-SVM-RBF	98.35	95.27
SVM-RBF	91.84	92.55
CFS + ANN	99.31	97.49
Proposed System	99.97	99.92

Table 12 shows that the existing methodologies, such as CFS + ANN, achieved 99.31 specificity, while CNN-Bi-LSTM reached 80.83% specificity, and other algorithms such as SAE-SVM reached 98.35 and SVM-RBF reached 91.84. However, the proposed model reached a high specificity rate of 99.97. Similarly, conventional models such as CNN reach a 95.01 accuracy rate, SAE-SVM-RB showed a 95.27 accuracy rate, SVM-RBF showed 92.55 and CFS + ANN showed 97.49. The proposed system reached a 99.92 accuracy rate. Hence, the comparative analysis shows that the proposed method reached an accuracy rate of 99.92 with the NSL-KDD dataset, a 0.98 accuracy rate with the CICIDS2017 dataset and a 99.783 accuracy rate with the UNSW-Bot dataset.

Further, by assigning different weights to accuracy scores, the overall weighted accuracy was influenced. If there are instances with higher weights, the model prioritizes them more during evaluation. Consequently, the proposed model's performance appears more favorable when heavily weighted instances are accurately predicted, leading to higher weighted accuracy. This weighting scheme emphasizes specific cases, potentially reflecting the model's effectiveness in handling critical scenarios, thus resulting in varying accuracy scores for different weights. Thus, the weights are chosen based on the importance of different instances, and the factors include cost sensitivity, class imbalance and domain knowledge. Thus, it is evident from the comparison that the proposed model outperforms other conventional methods. With the unique optimization capacity of SCA, the present research intends to integrate SCA and WOA so as to cancel out the shortcomings of both via hybridization, thereby procuring OWSA. Further, the proposed weight-updating process is made suitable for application to an individual tree in the RF model. These innate advantages of the proposed model allow it to yield ideal outcomes in ID.

5. Conclusions

This research proposed an IDS model based on appropriate ML-based algorithms, and to develop this, the study considered three datasets (UNSW-Bot, NSL-KDD and CICIDS2017). OWSA was proposed for selecting relevant features and AWRF was considered for the classification of intrusions. Performance was validated in accordance with standard metrics, in which the results suggest a better performance. Further, accuracy was assessed for various weights. In this case, although the accuracy rate varied with weights, better outcomes were attained when the weight = 10. Further, k-fold cross-validation was undertaken on the proposed system with three datasets. For the UNSW-Bot dataset, the accuracy was better when k = 2 and 4, with 0.9973 and 0.997 accuracy. Similarly, for the NSL-KDD dataset, the accuracy value was optimal when k = 1, with 0.999 accuracy, and for CICIDS2017, the accuracy rate was optimal when k = 5 with 0.98 accuracy. Furthermore, a confusion matrix was developed for identifying the correct and misclassification rate. The confusion matrices that were attained revealed that the proposed system reached a high rate of correct classification and a low rate of misclassification. This confirmed the better performance of the proposed system. However, for confirming the improved efficacy of the proposed system compared to conventional systems, a comparative analysis was undertaken. The outcomes show improved accuracy scores of 99.92 with the NSL-KDD dataset, 0.98 with the CICIDS2017 dataset and 99.783 with the UNSW-Bot dataset. The better performance rate of the proposed system makes it applicable for real-time execution. This study could also assist network security experts in determining intrusions, which would eventually help in the development of security measures for protecting a system. As a further extension, DL-based methodologies could be utilized for ID. In addition, various other datasets could be considered.

Author Contributions: Conceptualization, O.A.A. and M.F.A.; methodology, O.A.A.; Formal analysis, O.A.A.; Investigation, O.A.A. and M.F.A.; Project administration, M.F.A.; Supervision, M.F.A.; Writing—original draft, O.A.A.; Writing—review and editing, O.A.A. and M.F.A. 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: Data are contained within the article.

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

References

1. Kasongo, S.M. A deep learning technique for intrusion detection system using a Recurrent Neural Networks based framework. *Comput. Commun.* **2023**, *199*, 113–125. [CrossRef]
2. Das, S.K.; Samal, S.; Ranjan, P.; Das, S.P. An Efficient Data Mining Technique for an Intrusion Detection System in Network. In *Constraint Decision-Making Systems in Engineering*; IGI Global: Hershey, PA, USA, 2023; pp. 1–17.

3. Asif, M.; Abbas, S.; Khan, M.; Fatima, A.; Khan, M.A.; Lee, S.-W. MapReduce based intelligent model for intrusion detection using machine learning technique. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *34*, 9723–9731. [CrossRef]
4. Almomani, O. A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. *Comput. Mater. Contin* **2021**, *68*, 409–429. [CrossRef]
5. Brittorameshkumar, J.V.S. OSS-RF: Intrusion detection using optimized sine swarm based random forest classifier on unsw-nb15 dataset. *Int. J. Tech. Phys. Probl. Eng. (IJTPE)* **2022**, *14*, 275–283.
6. Xu, H.; Lu, Y.; Guo, Q. Application of Improved Butterfly Optimization Algorithm Combined with Black Widow Optimization in Feature Selection of Network Intrusion Detection. *Electronics* **2022**, *11*, 3531. [CrossRef]
7. Mohammad, A.H. Intrusion Detection Using a New Hybrid Feature Selection Model. *Intell. Autom. Soft Comput.* **2021**, *30*, 65–80. [CrossRef]
8. Ambikavathi, C.; Srivatsa, S.K. *Predictor Selection and Attack Classification Using Random Forest for Intrusion Detection*; NISCAIR-CSIR: New Delhi, India, 2020.
9. Magán-Carrión, R.; Urda, D.; Díaz-Cano, I.; Dorronsoro, B. Towards a reliable comparison and evaluation of network intrusion detection systems based on machine learning approaches. *Appl. Sci.* **2020**, *10*, 1775. [CrossRef]
10. Verma, A.; Ranga, V. Machine learning based intrusion detection systems for IoT applications. *Wirel. Pers. Commun.* **2020**, *111*, 2287–2310. [CrossRef]
11. Kunhare, N.; Tiwari, R.; Dhar, J. Intrusion detection system using hybrid classifiers with meta-heuristic algorithms for the optimization and feature selection by genetic algorithm. *Comput. Electr. Eng.* **2022**, *103*, 108383. [CrossRef]
12. Kaur, R.; Gupta, N. Network intrusion detection using meta-heuristic feature selection and cost-sensitive learning. *Int. J. Internet Technol. Secur. Trans.* **2023**, *13*, 105–138. [CrossRef]
13. Ghanbarzadeh, R.; Hosseinalipour, A.; Ghaffari, A. A novel network intrusion detection method based on metaheuristic optimisation algorithms. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 7575–7592. [CrossRef]
14. Kumar, B.V. Hybrid metaheuristic optimization based feature subset selection with classification model for intrusion detection in big data environment. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 2297–2308.
15. Naseri, T.S.; Gharehchopogh, F.S. A feature selection based on the farmland fertility algorithm for improved intrusion detection systems. *J. Netw. Syst. Manag.* **2022**, *30*, 40. [CrossRef]
16. Raghuvanshi, A.; Singh, U.K.; Sajja, G.S.; Pallathadka, H.; Asenso, E.; Kamal, M.; Singh, A.; Phasinam, K. Intrusion detection using machine learning for risk mitigation in IoT-enabled smart irrigation in smart farming. *J. Food Qual.* **2022**, *2022*, 3955514. [CrossRef]
17. Yerriswamy, T.; Murtugudde, G. An efficient algorithm for anomaly intrusion detection in a network. *Glob. Transit. Proc.* **2021**, *2*, 255–260.
18. Alzaqebah, A.; Aljarah, I.; Al-Kadi, O.; Damaševičius, R. A modified grey wolf optimization algorithm for an intrusion detection system. *Mathematics* **2022**, *10*, 999. [CrossRef]
19. Almazini, H.; Ku-Mahamud, K. Grey wolf optimization parameter control for feature selection in anomaly detection. *Int. J. Intell. Eng. Syst.* **2021**, *14*, 474–483. [CrossRef]
20. Khilar, R.; Mariyappan, K.; Christo, M.S.; Amutharaj, J.; Anitha, T.; Thavasimuthu, R. A Hybrid Network Anomaly Detection system using Glowworm Swarm Optimization with Principal Component Analysis. *Res. Sq.* **2021**, preprints. [CrossRef]
21. Patil, S.; Varadarajan, V.; Mazhar, S.M.; Sahibzada, A.; Ahmed, N.; Sinha, O.; Kumar, S.; Shaw, K.; Kotecha, K. Explainable Artificial Intelligence for Intrusion Detection System. *Electronics* **2022**, *11*, 3079. [CrossRef]
22. Priyanka, V.; Gireesh Kumar, T. Performance Assessment of IDS Based on CICIDS-2017 Dataset. In *Information and Communication Technology for Competitive Strategies (ICTCS 2020) ICT: Applications and Social Interfaces*; Springer: Singapore, 2022; pp. 611–621.
23. Jumabek, A.; Yang, S.; Noh, Y. CatBoost-Based Network Intrusion Detection on Imbalanced CIC-IDS-2018 Dataset. *J. Korean Soc. Commun. Stud.* **2021**, *46*, 2191–2197. [CrossRef]
24. Seth, S.; Singh, G.; Kaur Chahal, K. A novel time efficient learning-based approach for smart intrusion detection system. *J. Big Data* **2021**, *8*, 111. [CrossRef]
25. Al-Janabi, M.; Ismail, M.A. Improved intrusion detection algorithm based on TLBO and GA algorithms. *Int. Arab J. Inf. Technol.* **2021**, *18*, 170–179.
26. Alhayali, R.A.I.; Aljanabi, M.; Ali, A.H.; Mohammed, M.A.; Sutikno, T. Optimized machine learning algorithm for intrusion detection. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *24*, 590–599. [CrossRef]
27. Ивкин, А.Н.; Бурлаков, М.Е. Realization of expert intrusion detection system based on the results of datasets and machine learning algorithm analysis. *Casp.J. Manag. High Technol.* **2020**, *2*, 100–107.
28. Kavitha, S.; Manikandan, J. Design of a Bottleneck Layered DNN Algorithm for Intrusion Detection System. *Methods* **2022**, *5*, 6. [CrossRef]
29. Jama, A.M.; Khalifa, O.O.; Subramaniam, N.K. Novel Approach for IP-PBX Denial of Service Intrusion Detection Using Support Vector Machine Algorithm. *Int. J. Commun. Netw. Inf. Secur.* **2021**, *13*, 249–257. [CrossRef]
30. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Gener. Comput. Syst.* **2019**, *100*, 779–796. [CrossRef]
31. Shareena, J.; Ramdas, A.; AP, H. Intrusion detection system for iot botnet attacks using deep learning. *SN Comput. Sci.* **2021**, *2*, 205.

32. Baniasadi, S.; Rostami, O.; Martín, D.; Kaveh, M. A novel deep supervised learning-based approach for intrusion detection in IoT systems. *Sensors* **2022**, *22*, 4459. [CrossRef]
33. Zixu, T.; Liyanage, K.S.K.; Gurusamy, M. Generative adversarial network and auto encoder based anomaly detection in distributed IoT networks. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–7.
34. Abdulrahman, A.A.; Ibrahem, M.K. Evaluation of DDoS Attacks Detection in a CICIDS2017 Dataset Based on Classification Algorithms. *Iraqi J. Inf. Commun. Technol. (IJICT)* **2018**, *1*, 49–55.
35. Sumaiya Thaseen, I.; Saira Banu, J.; Lavanya, K.; Rukunuddin Ghalib, M.; Abhishek, K. An integrated intrusion detection system using correlation-based attribute selection and artificial neural network. *Trans. Emerg. Telecommun. Technol.* **2021**, *32*, e4014. [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.