Computers & Security 124 (2023) 102957

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization

Abdullah Alzaqebah[a], Ibrahim Aljarah[b], Omar Al-Kadi[b,*]

[a] The World Islamic Sciences and Education University, Amman, Jordan
[b] King Abdullah II School for Information Technology, The University of Jordan, Amman, Jordan

ARTICLE INFO

Article history:
Received 3 July 2022
Revised 13 September 2022
Accepted 13 October 2022
Available online 18 October 2022

Keywords:
Intrusion detection system
Swarm intelligence
Extreme learning machine
Feature extraction
Crossover error rate

ABSTRACT

The surge in cyber-attacks has driven demand for robust Intrusion detection systems (IDSs) to protect underlying data and sustain availability of network services. Detecting and classifying multiple type of attacks requires robust machine learning approaches that can analyze network traffic and take appropriate measures. Traffic data usually consists of redundant, irrelevant, and noisy information, which could have a negative influence on the model performance. In this paper, we propose an improved bio-inspired meta-heuristic algorithm for efficient detection and classification of multi-stage attacks. The proposed model uses a one-versus-all sub-model based technique to deal with the multi-class classification problem. Each sub-model employs an enhanced Harris Hawk optimization with extreme learning machine (ELM) as the base classifier. This hierarchy produces the best subset of features per attack, along with optimized ELMs weights, which can improve the detection rate significantly. The proposed technique was tested against various meta-heuristic algorithms and multi-class classifiers using the UNSWNB-15 dataset. In seven different types of attacks, experimental results outperformed other existing methods in terms of decreasing the crossover-error rate and obtaining the best values for the G-mean measure.

1. Introduction

The demand for a robust Intrusion Detection System (IDS) is growing day by day with widespread internet applications and services. Online teaching, e-commerce, and video-conferencing, among many others, have expanded the utilization of web resources. In line with the increase of the computer network and internet spread, the cyber security breach and intrusion incidents were increased with the availability of the newly developed hacking tools, which were used to violate the CIA principles (confidentiality, integrity, and availability). Despite using different protection techniques such as firewalls, encryption, and anti-malware tools to prevent cyber threats, cyber-attacks are increased, and new attacks (known as zero-day attacks) become a significant problem for the security of the spread of computer networks (Moustafa et al., 2017a; Zhou et al., 2020).

As a result, creating and utilizing sophisticated IDSs is necessary to protect against recent attacks. An intrusion detection system (IDS) is a defensive wall that is in charge of spotting suspicious activity (intrusions) by keeping a close watch on network traffic and logs and taking appropriate countermeasures against threats (Krishnaveni et al., 2020). Recently, machine and deep learning techniques were adopted in building such an IDS (Al-Kadi, 2020; Alkadi et al., 2020b). Various IDSs have been proposed, which can be categorized based on different criteria, such as the source of data, detecting, and response mechanisms. Host-based and network-based are the categories of IDSs, distinguished based on the source of data, which are either the client devices or the network traffic according to these categories. The anomaly and signature-based techniques are distinguished categories according to the detection mechanism (Alkadi et al., 2020a). Signature-based IDSs detect threats by identifying the client activities' behavior pattern and comparing it with pre-saved intrusion patterns in the database. The database should be updated periodically in order to detect new threats with distinguished patterns. In contrast, an anomaly-based IDS uses a pre-trained normal activity profile, which is used to compare the traffic activity and recognize the abnormal one.

The feature selection (FS) process substantially impacts how well an IDS performs in addition to the machine learning classification techniques that comprise its foundation. Therefore, IDS effectiveness can be significantly increased by seamlessly choosing robust features and integrating them with the classification process. One of the approaches to implementing feature selection is the meta-heuristic algorithms. Bio-inspired meta-heuristic algorithms

* Corresponding author.
E-mail address: o.alkadi@ju.edu.jo (O. Al-Kadi).

https://doi.org/10.1016/j.cose.2022.102957
0167-4048/

imitate the common behavior of biological species followed by certain conditions, such as actions taken while searching 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 (Heidari et al., 2019). Due to their learning and adaptability, bio-inspired machine learning and deep learning techniques for IDSs can keep up with various threats and attacks. Furthermore, because IDSs handle a considerable quantity of data and recognize online intrusions in a dynamic and multi-dimensional domain, they can be utilized to build an efficient IDS.

This paper proposed an efficient multi-stage attack detection IDS with a bio-inspired feature selection strategy. The proposed model employs a one-versus-all approach for training, which results in N submodels, where N is the number of attacks in the dataset. Then, an enhanced Harris Hawks optimizer is used in each submodel, abbreviated as Improved Harris Hawks optimizer (IHHO). IHHO is executed in association with the Extreme Learning Machine (ELM) for feature selection and optimizing the weights. The suggested model identifies various forms of staged attacks in a dynamic and realistic manner, with the ability to recognize the most informative features at the attack level.

The contribution of this paper is summarized as follows:

- Improving the efficacy of the IDSs by minimizing the crossover-error rate, and strengthen the IDS to recognize different types of attacks.
- Developing a one-versus-all approach for converting the multi-class classification problem into N binary classification problems, each corresponding to a single type of attack.
- Improving the standard Harris Hawks optimizer with rapid convergence for best fitness with fewer iterations and an improved transfer function is proposed to widen the covered search space and better handle the local optima problem.
- Developing an integrated approach based on the IHHO and ELM for feature and parameter optimization. Thus, produce a useful feature set for each type of attack, and optimize the ELMs weights.
- Applying the proposed model to the different attacks within the UNSWNB-15 network intrusion dataset and ranking the features according to their importance.

The rest of the paper is organized as follows: Section 2 lists and discusses recent work IDSs, Section 3 explains in detail the proposed system and some key preliminaries of the utilized techniques. Then, our findings and the conducted results are discussed in Section 4. Finally, the paper is concluded in Section 5.

2. Related works

In the field of information and network security, IDSs are gaining prominence. As a result, an IDS performance was developed, enhanced, and improved using a variety of methodologies and procedures. Recently, machine learning and deep learning have revolutionized the trend in better attack detection and boosted IDS performance. Several works considered distinguishing the anomalies from the normal traffic as a binary classification problem without focusing on the underlying types of anomalies. Different Machine Learning (ML) techniques and algorithms were used and hybridized as the core of the IDS. These algorithms were used together with the optimization algorithms for feature selection and parameters optimization, which significantly affects the output results. The bio-inspired algorithms performed effectively in improving and optimizing the IDSs. Various algorithms, such as the GWO, PIO, and IWD, were used for feature selection, yet these algorithms faced two main problems: discretization and binarization. Table 1 summarize the previous state-of-the-art techniques and methods for IDS.

(Alazzam et al., 2020) used a wrapper feature selection algorithm for IDS based on a pigeon-inspired optimizer (PIO). A sigmoid transfer function was utilized to binarize the continuous pigeon optimization algorithm. Similarly, a cosine similarity technique was used for the discretization process. The results of the proposed PIO-based approach were compared with the traditional way of binarizing continuous swarm intelligent algorithms in terms of True Positive Rate, False Positive Rate, accuracy, and F-score for three different datasets: KDDCUPP99, NSL-KDD, and UNSW-NB15. The results showed that the PIO-based technique outperforms the other swarm intelligent algorithms according to the utilized measurements. Yet, the proposed IDS was used as a binary classification problem to detect the abnormal traffic without determining the type of abnormality.

Similarly, (Acharya and Singh, 2018) proposed a novel mechanism that utilizes the Intelligence Water Drops (IWD) algorithm for enhancing the performance of the IDS. IWD is a nature-inspired algorithm that starts by representing the search space as a graph with a set of nodes N and edges E. The IWD initializes a set of paths over the graph, then uses these paths to form the feature subset. Each subset is evaluated using the SVM classifier, and the best-performed solution will be kept as the best solution so far till the termination condition is met. The KDDCUP99 dataset was utilized to assess the suggested algorithm using false alarms, detection rate, and accuracy criteria for evaluation purposes. The main drawback of this technique is the exhaustive complexity in terms of time and resources when the dimension of the dataset becomes larger since the graph will be long and have different branches with various lengths. The proposed model was developed to classify the input into normal and anomalous activity without distinguishing between different types of attacks.

Apart from the significance of the feature selection process, various IDSs were improved by focusing on enhancing the classification process regarding the bio-inspired meta-heuristic for parameter and configuration optimization. Ensemble classification was used by (Tama and Rhee, 2015), using the supervised particle swarm optimization algorithm (PSO). PSO was combined with a correlation-based feature selection and the ensemble of tree-based classifiers (C4.5, Random Forest, and CART) with a majority voting technique for classification. The proposed approach was evaluated using the NSL-KDD dataset in terms of accuracy and false positive rate in normal vs. abnormal form. Different types of attacks were not recognized.

On the other hand, the need to recognize the underlying types of attacks has risen and gotten attention in the cyber-security field. Therefore, researchers tried to consider the IDS as a multi-class classification problem. (Alzubi et al., 2020) modified the binary version of grey wolf optimizer (GWO) and applied it for the feature selection problem for multi-attacks classification. The modified GWO improves the population generation process by using the crossover operator and adding omega search agent ω to the three best grey leaders α, β, and δ to increase and diversify the search agents. As the features are selected, the SVM classifier is used to classify the instances into various attacks. The solutions' fitness is measured to preserve the best one. The results of the modified GWO-SVM approach were evaluated based on the NSL-KDD dataset compared to the results of the original version of GWO and PSO in terms of accuracy, detection rate, and false-positive rate with different splitting data scenarios.

Although deep learning (DL) as a branch of machine learning was proved to be efficient with the ability to be used for various applications, these algorithms required extra time and computational resources. With the spread and the memory of the computational resources, DL can be used to improve the IDS performance

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

Publication	Dataset	Layers	Classification Mode	Method
(Alzubi et al., 2020)	NSL-KDD	Single	Multi-class	ML
(Alazzam et al., 2020)	KDDCUP99 NLS-KDD UNSW-NB15	Single	Binary	ML
(Acharya and Singh, 2018)	KDDCUP99	Single	Binary	ML
(Tama and Rhee, 2015)	NSL-KDD	Single	Binary	ML
(Khalvati et al., 2018)	KDDCUP99	Single	Binary	ML
(Sharma et al., 2019)	UNSW-NB15 KDDCUP99	Multi	Multi-class	ML
(Basnet et al., 2019)	CSE-CIC-IDS2018	Single	Multi-class	DL
(Kasongo and Sun, 2019)	NSL-KDD	Single	Multi-class	DL
(Qaddoura et al., 2021)	IoTID20	Multi	Multi-class	DL

supported by the availability of the frameworks and packages that implement DL, such as Pytorch, Keras, and Tensorflow. Using these packages to implement an improved IDS system was presented by (Basnet et al., 2019). The CSE-CIC-IDS2018 dataset was used to evaluate the IDS, which was implemented using the DL packages on both CPU and GPU infrastructure. No feature selection was used with a binary classification problem. Accordingly, the implemented IDS forms a basic and a simple DL-based IDS. A further improvement in the implemented IDS is required to achieve the state-of-the-art performance for such IDS.

Dimensionality reduction and feature selection can also be used with DL algorithms. A filter approach for feature reduction was used with the DL model for IDS in a wireless network by (Kasongo and Sun, 2019). As for the feature selection, the information gain (IG) filter approach was used, and then a DL feed-forward neural network was developed for the classification purpose. The accuracy, precision, and recall measurements are used to evaluate the proposed IDS compared to various ML algorithms based on the NSL-Knowledge Discovery and Data mining (NSL-KDD) dataset. The results showed that the DL-based approach outperformed the ML algorithms. The drawback of the proposed method is selecting features is implemented on the overall dataset by considering all attacks. Hence, the feature x may be insignificant for the particular attack, but it will be selected for another attack.

The works mentioned earlier for binary and multi-class classification were built as a single layer to improve IDS' performance. The multi-layers approach was recently adopted and applied to improve the IDSs performance using ML and DL techniques. In contrast, various multi-class classification-based IDS were proposed to implement a multi-attack detecting IDS. A one-versus-all technique was proposed by (Sharma et al., 2019) using ELM for classification and an extra-trees algorithm for feature selection. The multi-class classification problem is formed as N binary classification problems to detect each attack separately in the UNSWNB-15 and NLS-KDD datasets. The output of the ELM classifiers is merged in the final step using a softmax layer to produce the final predictions.

Although, A Multi-layer DL strategy was also used to improve the performance of the DL-based IDS. Multi-layer deep learning-based IDS for internet-of-things (IoT) was proposed by Qaddoura et al. (2021). This approach aims to detect different types of attacks in the IoT environment. Accordingly, two layers with the oversampling technique were used; the first layer is responsible for recognizing the normal traffic. Accordingly, the second layer is used to recognize the type of attack using two stages; sequential and long-short term memory (LSTM). The IoTID20 dataset was used to evaluate the multi-layer DL approach.

Similar to the reviewed work, the proposed approach aims to solve all the problems using tightly integrated steps for multi-classification, feature selection, and parameter and configuration optimization. Accordingly, feature reduction and ELM weight optimization will be accomplished using a one-versus-all technique and an improved bio-inspired algorithm (IHHO). As a result, rather than using the entire dataset, the suggested approach can identify the most informative features at the attack level.

3. Bio-inspired intrusion detection system

3.1. Enhanced harris Hawk's optimizer

Harris Hawks Optimizer (HHO) was recently established as an optimization algorithm by (Heidari et al., 2019) to be used for solving global optimization problems. The HHO mimics the intelligent hunting behavior of the Harris Hawks birds in nature. The HHO in particular and optimization algorithms, in general, depend on the optimized solution's consecutive building, which depends on the best solutions built iteratively. In order to cope with the nature of the optimization process, some of the worst solutions will be considered in the initial solutions, and these solutions may form the best solution in the upcoming generations. Fig. 1 shows the major phases of the HHO (Heidari et al., 2019; Piri and Mohapatra, 2021; Too et al., 2019).

The HHO's dynamic hunting habit allows it to operate in dynamic and realistic situations. Because network traffic fluctuates, the FS for IDS is seen as a dynamic environment. We adopt the HHO in this work for optimization and feature selection. Although HHO excels in terms of ease of implementation, speed of computation, and efficiency in traversing search space, it, like all metaheuristic algorithms, suffers from delayed convergence and falls into local optima in some circumstances (Hussien and Amin, 2022; Kardani et al., 2021). For that, in this work, we proposed enhancements to the basic HHO to strengthen the searching capabilities to ensure local and global optimization, avoid trapping into local optima during optimization, and speed up the HHO's convergence.

The information gain (IG) values in the suggested model will direct initialize the first population in the wrapper-based method. This methodology combines both procedures into a single step to take advantage of the wrapper-based method's high accuracy and the filter-based method's speed. In other words, using the filter-based method, the classic merging method chooses a certain number of the highest ranking features. The selected features will then be given to the wrapper-based, done in different processes, to select the best from the input features set.

IG is a popular measure of the effect of attributes or the attribute importance in a specific dataset; according to the information theory, when the IG value of an attribute is increased, that means more importance of that feature (Gao et al., 2014).

The information theory metric, which characterizes the purity of an arbitrary collection of samples, typically uses entropy. The Information Gain (IG) and Gain Ratio (GR) are built on this foundation. The entropy measure is used to determine the unpredictability of a system (Malathi and Manimekalai, 2021).

$$H(Y) = \sum_{y \in Y} p(y) \log_2 (p(y)) \quad (1)$$

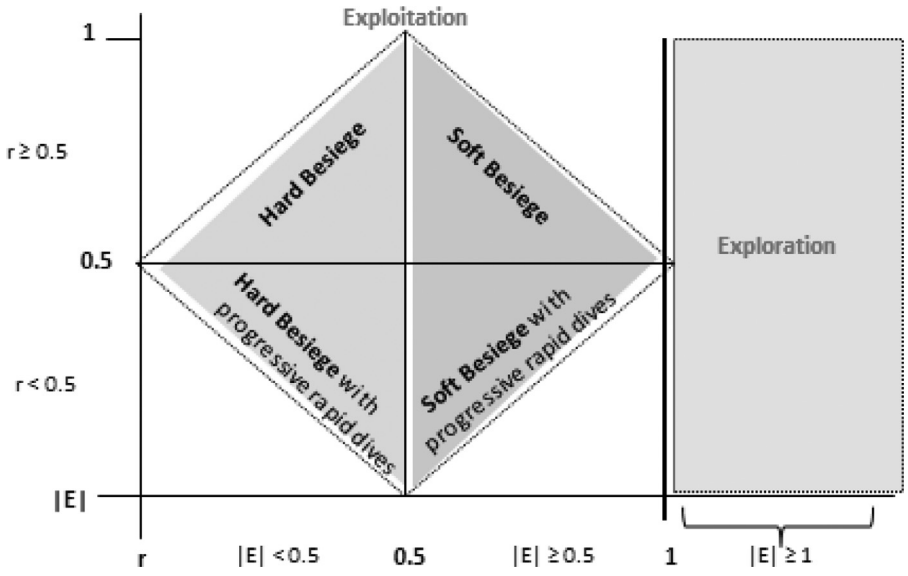

Fig. 1. Standard HHO phases.

The marginal probability density function of the random feature Y is p(y). There is an association between features X and Y if the detected Y values in the training data set S have divided consorting to the values of the following feature X. The entropy of Y concerning the split produced by X is less than the entropy of Y before dividing. Where the conditional probability of y is $p(y|x)$ for given x (Malathi and Manimekalai, 2021).

$$H(Y|X) = \sum_{x \in X} p(x) \sum_{y \in Y} p(y|x) \log_2 (p(y|x)) \tag{2}$$

Given that entropy is a condition of impurity in a training set S, it can be used to provide a metric that represents the degree by which the entropy of Y lowers by excogitating extra information about Y rendered by X. IG is the abbreviation for this measurement (Malathi and Manimekalai, 2021).

$$IG(X) = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{3}$$

After calculating the IG values, the min-max normalization technique is applied to make the lowest value is equal to zero, and the largest one is equal to 1 using Eq. 4.

$$NormalizedIG(i) = \frac{(IG(i) - MinIG)}{(MaxIG - MinIG)} \tag{4}$$

Where $IG(i)$ represents the information gain of the i^{th} feature, *MinIG* and *MaxIG* indicate the minimum and the maximum values of the IG in the IG vector, respectively.

In population-based algorithms, some probabilities are considered the worst solutions, which may form the best solutions in the upcoming generations. For that, the proposed model works by dividing the population into two parts:

1. The first part represents the injected population ratio (25%, 50%, 75%, and 100%) from the original population size, which will be initialized based on the IG value of each feature by generating a random number between 0 and 1. After that, check if the random number is less than the feature's IG, then select the feature; otherwise, the feature will never be selected. Here, the injected ratio is the ratio of the population size that will be initialized using the proposed initialization technique. Furthermore, when we used a 25% ratio, we mean that 25% of the initial population will be generated based on the IG values. In contrast, the rest of the population (75%) will be generated in a random way. As such, the overall population will be generated using the IG values when the injection ratio is equal to 100%. This operator is employed to give a high probability of the high IG value to be included in the first population, as shown in Eq. 5.

$$P(i) = \begin{cases} 1, & if\ rnd\ <\ Normalized\ IG(i) \\ 0, & if\ rnd\ \geq\ Normalized\ IG(i) \end{cases} \tag{5}$$

where P_i is the binary representation of the i^{th} 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, which will be initialized using the traditional random way by generating a random number between 0 and 1. Then check the values, in case the value is greater than 0.5, then select this feature; otherwise, deselect this feature, as shown in Eq. 6.

$$P(i) = \begin{cases} 1, & if\ rnd > 0.5 \\ 0, & if\ rnd \leq 0.5 \end{cases} \tag{6}$$

Where P_i is the binary representation of the i^{th} feature in the initial population and *rnd* indicates a random number bounded by 0 and 1.

The final step is to combine the two generated halves to create the initial population set, which the IHHO algorithm will then process to pick the optimal feature set.

The transfer functions (TFs) are commonly used to map the continuous search space into binary search spaceis build commonly using S-shaped and V-shaped TFs families (Alzaqebah et al., 2020; Ghosh et al., 2020; Mirjalili and Lewis, 2013).To enhance the searching capabilities, the proposed model uses the X-shaped function (Ghosh et al., 2020) as giving in Eq. 7 and Eq. 8 (Beheshti, 2021)) and illustrated in Fig. 3.

$$X_1(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

$$X_2(x) = \frac{1}{1 + e^{x}} \tag{8}$$

The algorithm's increased complexity is the main disadvantage of employing the X-shaped TF. Due to this complexity, it is necessary to assess each potential position in order to keep the best one. Despite this shortcoming, this approach has the benefit of covering

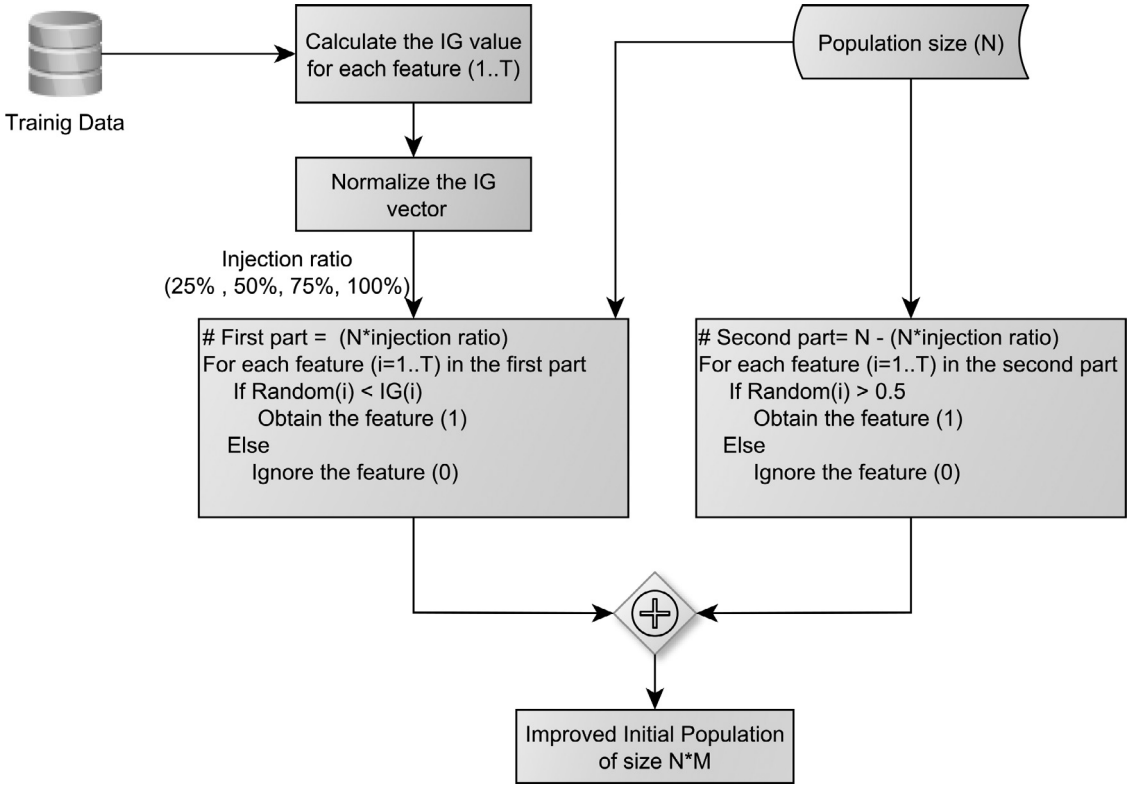

Fig. 2. Improved initialization phase of IHHO.

a large portion of the search space and guarantees greater dynamicity in handling the IDS environment. To address this issue, we focus on accelerating the evaluation process in our work.

Extreme Learning Machine (ELM) was introduced by Huang et al. (2004) as a new learning scheme for feed-forward neural network (FFNN) to overcome the weakness of the FFNN. The ELM works by assigning random values for the input weights and hidden biases, then computing the hidden layers output in one step. Then, the output weights are assigned using Moore Penrose (MP) generalized inverse. Thus, it was proven that ELM achieves the lowest training fault, the lowest norm of weights, and the best generalization performance with a high-speed training process (Feng et al., 2021; Huang et al., 2004).

In summary of this work, a novel binary version of the HHO algorithm is proposed, which is denoted as the improved Harris Hawk Optimization algorithm (IHHO). First, the population initialization phase was improved by using an integration of the FS mechanism in the filter-based approach with the mechanism of the wrapper-based approach, as illustrated in Fig. 2. This improvement was attained by measuring each feature's importance using a filter-approach technique, then using the resulting values to guide the subset features in the initial population. This improvement guarantees that the most suitable solutions will be included in early iterations, which immediately accelerates the algorithm's convergence.

Consecutively, an X-shaped transfer function is used to improve the searching capability of the standard HHO. The additional advantage is acquired using the X-shaped TF by balancing between the exploration and exploitation phases. Because these processes add complexity when calling the fitness function to evaluate the generated solutions, the balancing of exploration and exploitation is also subject to affordable resources. Because the ELM is regarded as a high-speed method, it is utilized as a basis classifier in the IHHO to overcome this drawback. (Huang et al., 2004). Algorithm 1 shows the proposed IHHO. Hence the IHHO differs from standard HHO by using the intelligent initialization technique presented in figure. 2 to adopt the most relative features from the beginning. In addition, the X-shaped TF was utilized during update positions processes to widen the search strategy. And in each fitness evaluation call, the ELM was also used as the base classifier.

It's worth noting that the IHHO's computing complexity is mainly determined by three processes: hawk initialization, fitness evaluation, and updating. The computing complexity of the initialization operation with N hawks is $O(N+D)$ since the IG should be computed for D features. Let us assume that the T and D represent the maximum number of iterations and the problem's dimension. The updating process has a computational complexity of $O(T \times N \times D)$, which is made up of searching for the optimal location and updating the location vector of all hawks. In addition, the evaluation of N hawks requires $O(N)$. The X-shaped TF will call the fitness function more than one time in each updating process. For that, the ELM was utilized to overcome this additional calling complexity. Finally, there are no changes in the other steps' complexity presented in the original HHO. Therefore, the computational complexity of IHHO is $O(N \times T \times D)$.

3.2. Hierarchical approach for attacks classification

An efficient IDS is proposed for multi-attack classification based on n models and an optimized IHHO optimization algorithm. In the proposed approach, the one-vs-all technique is used by breaking down the multi-class classification problem into N binary classification problems, The hierarchy is represented by splitting the overall multi-class problem into binary classification problems. While N is the number of class labels in the used dataset in order to minimize the load and the complexity since the multi-class classifier is more complex compared with the binary classifier (Sharma et al., 2019).

Each sub-model is trained to recognize and detect this attack rather than others, acting as a binary classifier and an optimizer for

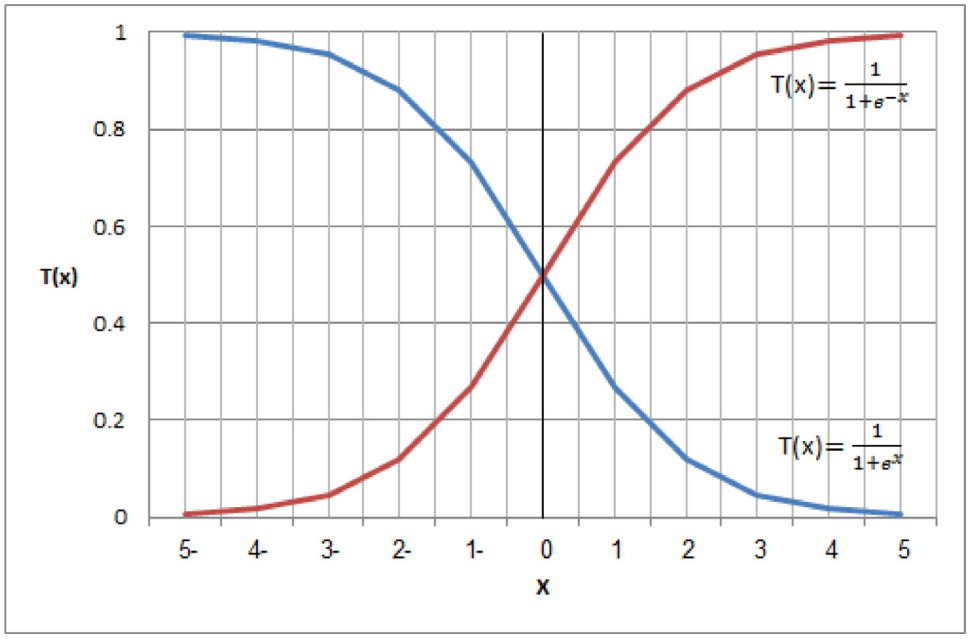

Fig. 3. Function of X-shaped Transfer.

a particular attack. As a result, each sub-model will generate the optimal selection of features for detecting this attack, along with the optimized ELM weights and biases.

The utilized dataset, which contains ten class labels (nine attacks and one for normal), will be split into ten datasets. Each one includes instances of the i^{th} attack and the same ratio from other cases and is set to be a non-attack label. This process trains each sub-model to distinguish the assigned type of attack. In the training phase, each sub-model utilizes the proposed IHHO to select the best features set that describe the assigned attack and optimize the ELM weights and biases for the used ELM in the sub-model. Furthermore, each sub-model produces a trained model with the best features set and the optimized ELM's weights and biases for the significant type of attack. Finally, the testing set will be passed through each sub-model and extracting the correct predictions of the attack that trained for. The $N-1$ prediction sets (all attacks instead of the normal one) will be grouped to form the final predictions of the model, while all other unclassified attacks will be considered normal traffic. Fig. 4 shows the overall environment of the proposed work.

It is essential to mention that, in the ensemble approach, each sub-model predicts the class label for a specific instance. Then, using techniques such as voting or averaging, the model produces the final prediction. But for multi-class classification, each sub-model is responsible for detecting a specific type of attack. If only one sub-model recognizes the attack, there is no problem, but if more than one sub-model recognizes the tested instance as an attack, which one will be obtained as the predicted attack?. Therefore, we kept the training accuracy for each sub-model in the proposed model during the training phase. If more than one sub-model indicates the attack, the sub-model with the highest training accuracy will be considered the most reliable recognizer and set the prediction as the attack for which this sub-model is responsible.

4. Experimental results and discussion

4.1. Dataset

4.1.1. Dataset description

UNSW-NB15 dataset is an intrusion detection dataset developed by IXIA perfect storm, and it targets more realistic network traffic and novel types of modern attacks (Shushlevska et al., 2022). Tcpdump tool used to generate pcap files with a size of almost 100GB with a hybrid of real normal activities and synthetic contemporary attack behaviors. It employed Bro-IDS tools with twelve algorithms to generate 49 attributes with nine different attacks and 1 class for normal traffic. It was widely used for proving and testing algorithms and techniques for solving the intrusion detection system since the multi-class problem is more challenging than the binary-classes problem (Sharma et al., 2019). The attacks are DOS, ShellCode, Worms, Fuzzers, Backdoors, Exploits, Analysis, Generic, and Reconnaissance. The generated attributes are grouped into six main categories (Alazzam et al., 2020). For more details about the dataset and features description, see the researches from the University of New South Wales - Sydney (Moustafa et al., 2017a; Moustafa and Slay, 2015; 2016; Moustafa et al., 2017b; Sarhan et al., 2020).

On the other hand, the proposed model includes a sub-model for identifying normal traffic and dealing with other forms of attacks. This sub-unidentified model's traffic will be deemed anomalous traffic. The unrecognized traffic in the normal sub-model will be treated as a separate category for future work. Which will be used in the categorization process to identify it as atypical behavior without defining the type of anomaly.

The dataset is divided into training and testing sets, where the training set contains 175,341 instances and the testing set includes 82,332 instances. Fig. 5 shows the data distribution (training and testing) among all class labels in the dataset.

4.1.2. Data preprocessing

In this early stage of data preparation, the data is pre-processed and cleaned form to feed into the algorithm in order to avoid overfitting and process the outliers. The preparation stage implements the following processes in general: missing value removal, duplicate data removal, data transformation, and data normalization. As the utilized dataset does not include any missing or duplicate values, only data encoding is implemented by converting the symbolic data into numerical representations. The class labels will be encoded into numbers from 1 to 10 as follows: {'analysis': 1, 'backdoor': 2, 'dos': 3, 'exploits': 4, 'fuzzers': 5, 'generic': 6, 'normal': 7, 'reconnaissance': 8, 'shellcode': 9, 'worms': 10}.

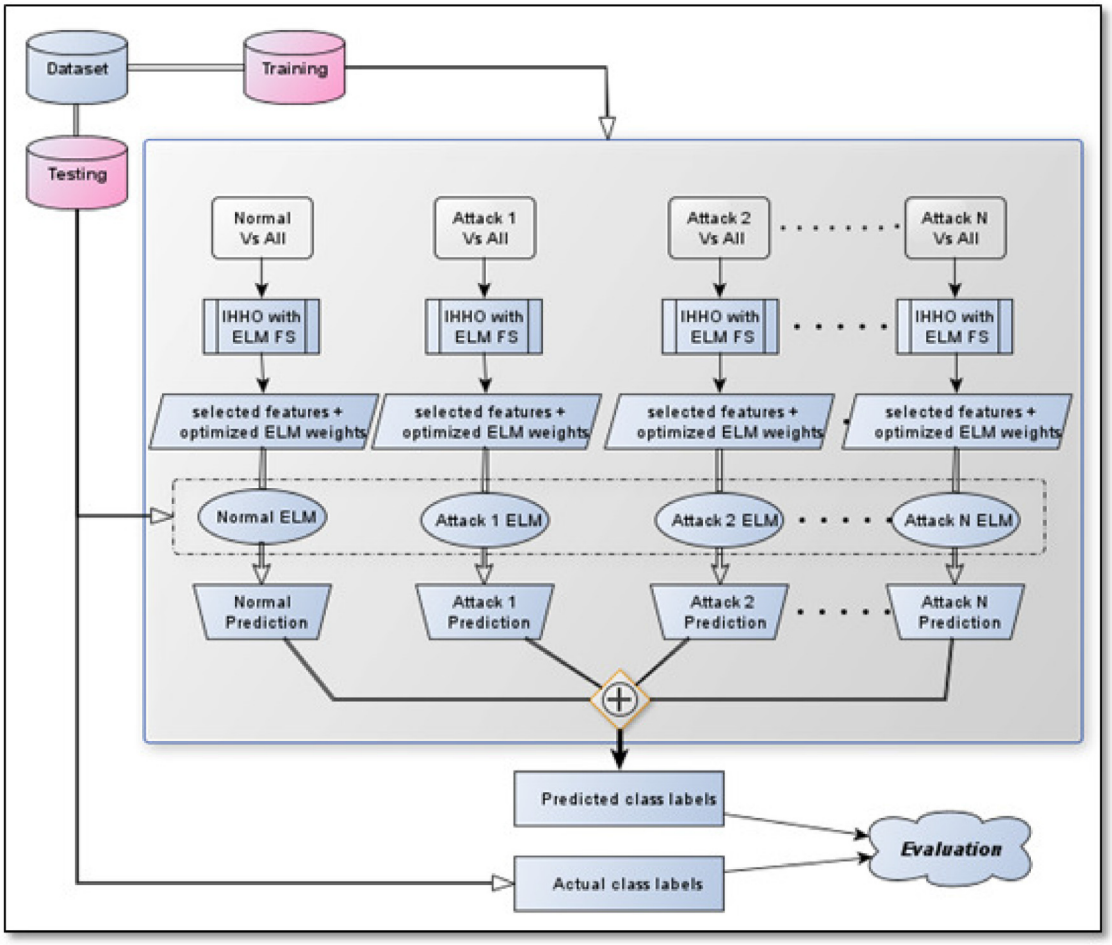

Fig. 4. The Proposed Hierarchical IDS.

Six features were eliminated from the dataset from an expert in the domain since they were repeated; these features have no relationships with the detection or classification process. These features are 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) (Alazzam et al., 2020). 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 times. For that, these attributes cannot be considered as features for the traffic, which was eliminated by the work of Alazzam et al. (2020); Sharma et al. (2019);

For data normalization, the min-max approach was used to scale the data in the range of [0,1] as given in Eq. 9

$$X_{Normalized} = \frac{X - X_{Min}}{X_{Max} - X_{Min}}. \tag{9}$$

For the class imbalance problem, sampling is implemented. Given that the distribution of the training data among class labels, as shown in Fig. 5, is imbalanced, which can directly affect the performance of the classifier. Accordingly, for each data that belonged to a specific class label, a sub-set is selected for the training process. Thus, let n be the size of instances that belong to the class, and $n/(N-1)$ is selected only. Yet, given that the worm attack contains only 139 tiny instances, the oversampling technique is used to increase the number of instances in this attack by 800%.

4.2. Experimental and parameter settings

4.2.1. Programming language and tools

Matlab R2019a tool is used for implementing 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 supported toolboxes, such as parallel toolbox, which speeds up the computation. Moreover, Matlab processes complex data and is used for complex simulations and engineering problems. At the same time, the Python programming language is used with pandas and Sklearn libraries for data preparation and preprocessing, which already has functions and procedures to preprocess and transform data, such as preprocessing library (Alazzam et al., 2020). The proposed and compared approaches are implemented using the same platform and programming language to get fair comparisons.

4.2.2. Sensitivity analysis and parameter settings

The parameters used in the experiments were carefully established based on the sensitivity analysis. The analysis set up the number of hidden neurons in the ELM network based on testing multiple values; these are 20, 40,60, 80, and 100 neurons (Alzaqebah et al., 2022) under two main activation functions, Sigmoid and Relu. Eqs. 10 and 11 show the formulas that represent these functions respectively (Sharma et al., 2019).

$$f1(x) = \frac{1}{1 + e^{-x}} \tag{10}$$

$$f2(x) = Max(0, x) \tag{11}$$

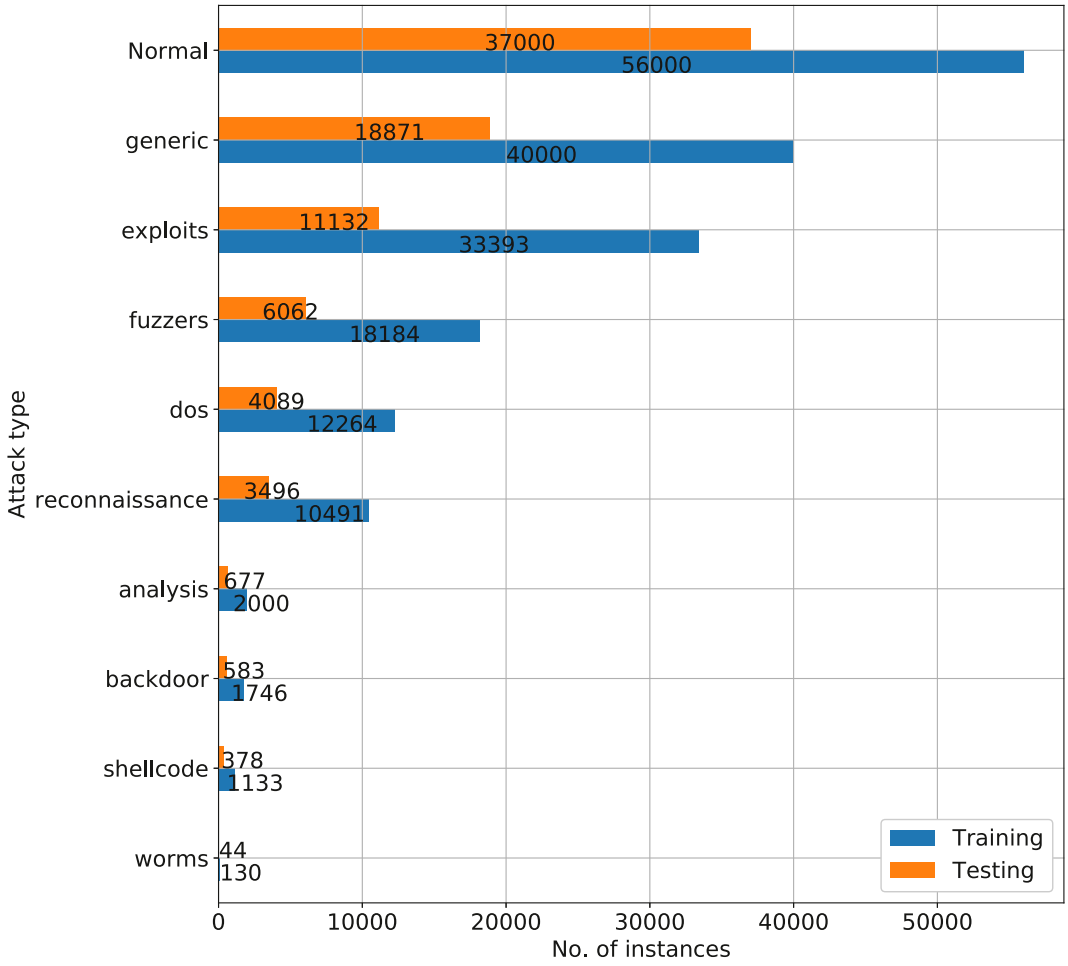

Fig. 5. UNSW-NB15 Data distribution.

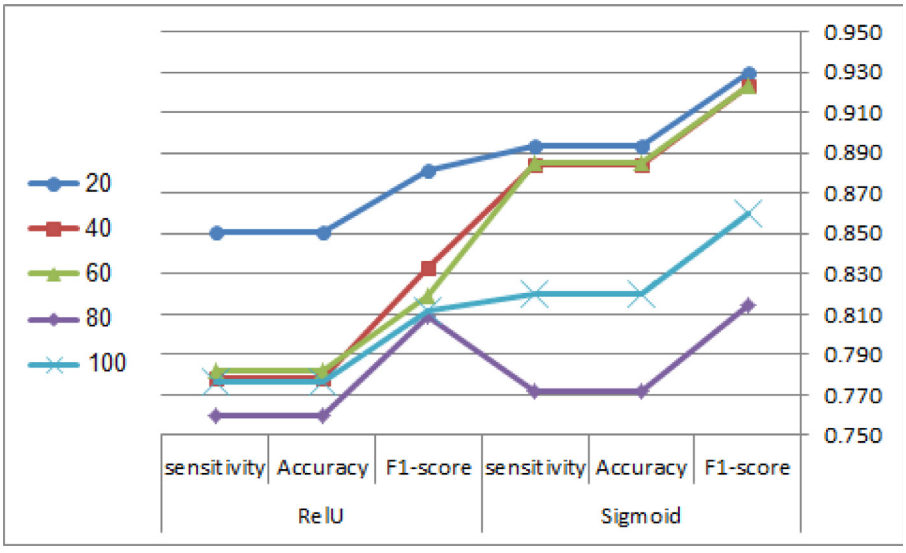

Fig. 6. Sensitivity results over RelU and Sigmoid activation functions using different number of hidden neurons.

As for the parameters of the proposed improvements in the initialization phase, four injection ratios, 25%, 50%, 75%, and 100%, were tested, and the average was used. The outcomes of the conducted sensitivity are shown in Fig. 6, which are measured using the F1-Score, accuracy, and sensitivity, respectively. The higher the value, the better the output is for these measures. As a result, it is evident that 20 hidden neurons and the sigmoid activation function were the optimal selections for the model.

It is to be mentioned that all the experiments were obtained as the average of conducting 30 runs. The same environment and programming language were used to discuss and analyze fair results for all experiments.

Input: The population size N, the maximum number of iterations T, and the normalized IG vector, current iteration $t = 1$.
Output: The best location of the rabbit.
Initialize the population X_i(i= 1, 2, …, N) using the proposed improved initialization phase.
while $(t \leq Max_iteration)$ **do**
  Calculate the fitness values of hawks using ELM.
  Set X_{rabbit} as the location of the rabbit (best location).
  for *(each hawk (X_i))* **do**
    Update the initial energy E_0 and jump strength J randomly.
    if $(|E| \geq 1)$ **then** ▷ *Exploration phase*
      Update the location vector to explore and detect the prey in the search space.
      Calculate the probability using X-shaped TFs.
      Update the position of hawks based on best solution's position.
    end
    else ▷ *Exploitation phase*
      if $(r \geq 0.5$ *and* $|E| \geq 0.5)$ **then** ▷ *Soft besiege*
        vector to encircle the prey softly.
        Calculate the probability using X-shaped TFs.
        Update the position of hawks based on best solution's position.
      end
      else if $(r \geq 0.5$ *and* $|E| < 0.5)$ **then** ▷ *Hard besiege*
        Update the location vector to encircle the prey hardly.
        Calculate the probability using X-shaped TFs.
        Update the position of hawks based on best solution's position.
      else if $(r < 0.5$ *and* $|E| \geq 0.5)$ **then** ▷
        Soft besiege with progressive rapid dives
        Update the location vector using leafy flights technique.
        Calculate the probability using X-shaped TFs.
        Update the position of hawks based on best solution's position.
      else if $(r < 0.5$ *and* $|E| < 0.5)$ **then** ▷
        Hard besiege with progressive rapid dives
        Update the location vector to minimize the average distance to the prey.
        Calculate the probability using X-shaped TFs.
        Update the position of hawks based on best solution's position.
    end
  Update the best position of X_{rabbit}
  t=t+1
  end
Return X_{rabbit} $(bestsolution)$

Algorithm 1: Pseudo-code of the proposed IHHO algorithm.

The population size is set to 10 according to (Alzaqebah et al., 2022; Faris et al., 2018; Hammouri et al., 2020; Mafarja et al., 2019; Mafarja and Mirjalili, 2018), while the number of iterations is set to 100 based on (Mafarja et al., 2019; Mafarja and Mirjalili, 2018). The proposed approach achieves faster convergence, which means achieving the best fitness value in early iterations. Table 3 shows the parameter settings that will be used in this work after analyzing the sensitivity of each one, as shown in the afterward section. Table 4 shows the compared algorithms' parameters.

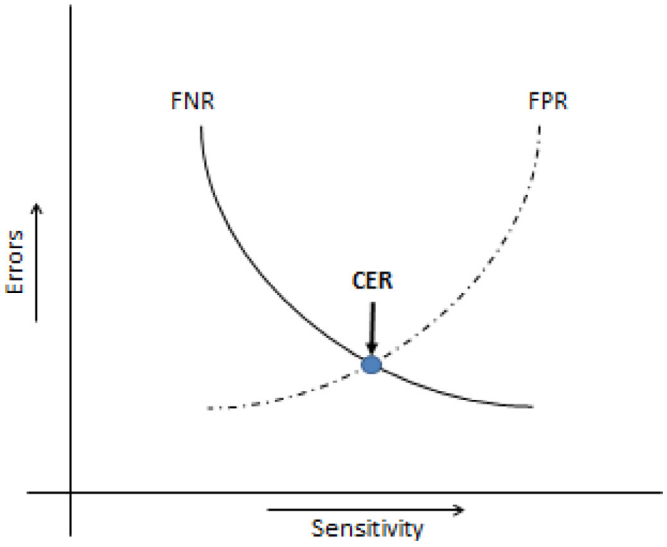

Fig. 7. CER for IDS system.

4.3. Fitness evaluation

The fitness function that is used within the optimization algorithm model is the objective that these algorithms aim to optimize. Several objective functions can be considered within the optimization algorithm based on the nature of the problem. Based on the NFL theorem (Wolpert and Macready, 1997) these objectives may vary. Two evaluation functions are used and tested in the proposed approach during experiments. First, classification accuracy is an important measure to be considered in the fitness function. On the other hand, the crossover error rate (CER), also known as an equal error rate (ERR), is considered because the domain of the proposed approach is connected with a security application. The CER aims to minimize the difference between the false-negative rate (FNR) and the false positive rate (FPR), which are also known as the false acceptance rate (FAR) and false rejecting rate (FRR) respectively. The lower the CER, the better the performance is. Fig. 7 shows the general concept of the CER for the IDS system (Awasthi et al., 2020; Liu et al., 2009).

Moreover, since the feature selection processes aim to decrease the datasets dimensionality by selecting the minimum number of features. Specifically, the wrapper-based algorithms work by selecting a subset from the original features set, then iteratively evaluating the performance using the selected subset of features and keeping the best performing features set. Thus, a small number of features is better in the FS. For that reason, the feature reduction rate will be considered too in both fitness functions. Accuracy and reduction rates are included as objectives of the fitness function that represented in Eq. 12; hence the aim is to minimize the number of features and increase the accuracy of classification. To avoid this contradiction, the accuracy is converted to a minimization problem by taking the error rate instead of the accuracy (1-accuracy) (Mafarja et al., 2017). While Eq. 13 represents the fitness function that is concerned with minimizing the CER concerning the reduction rate.

$$\downarrow Fitness = \alpha \times (ERRrate) + \beta \times \frac{|R|}{|N|} \tag{12}$$

$$\downarrow Fitness = \alpha \times (|FAR - FRR|) + \beta \times \frac{|R|}{|N|} \tag{13}$$

Table 2
Description of classes in the UNSW-NB15 dataset.

Attack	# of instances	Description
Normal	1,550,712	Normal, non-malicious flows
Exploits	24,736	Are commands that influence the behavior of a host by exploiting a known vulnerability.
Fuzzers	19,463	An attack in which the attacker sends massive volumes of random data to cause a system to crash while also attempting to find security vulnerabilities.
Reconnaissance	12,291	A probe is a mechanism for acquiring information about a network host.
Generic	5570	A strategy that targets cryptography and causes each block-cipher to collision.
DoS	5051	Denial of Service (DoS) is an attempt to overburden the resources of a computer system in order to prohibit access to or availability of its data.
Analysis	1995	A group that uses ports, emails, and scripts to target online applications with a variety of threats.
Backdoor	1782	A method for getting beyond security measures by responding to specially designed client apps.
Shellcode	1365	A type of malware that infiltrates a code in order to take control of a victim's host.
Worms	153	Self-replicating attacks that spread to other computers.

Table 3
List of the used parameters 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

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
GOA	C_{max}	1
	C_{min}	0.00004
	Upper bound	1
	Lower bound	0
GWO	Convergence constant α	[2 0]
HHO	Upper bound	1
	Lower bound	0
	Transfer function	S2

Table 5
The confusion matrix.

		Predicted	
		Anomaly (+)	Normal (-)
Actual	Anomaly (+)	TP	FN
	Normal (-)	FP	TN

Where α and β are parameters between 0 and 1 to represent the weight of each objective (β = 1-α), *ERRrate* indicates the classification error rate, *R* indicates the number of chosen features and, the overall number of features is represented as *N*. *FAR* and *FRR* are the False Accepting and Rejecting Rate respectively, based on the literature; α is set to 0.99 and β equal to 0.01 (Emary et al., 2016; Faris et al., 2018).

4.4. Evaluation metrics

To evaluate the efficacy of the proposed model, the confusion matrix-based measures are used, which are; true-positive (TP), true-negative (TN), false-positive (FP), and false-negative (FN), as shown in Table 5. From these measures, the Accuracy, F-measure, FPR, CER, and G-Mean measures will be calculated using Eq. 14 through Eq. 21. Additionally, the IDS applications should be concerned with other measures instead of accuracy. The sensitivity, specificity, and cross-over error rate are crucial measures to evaluate the performance of the IDS.

- Classification Accuracy: the percentage of correctly classified classes in relation to the total number of classifications. and is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{14}$$

- False Positive Rate (FPR): The proportion of normal that is identified as an attack is measured. Which is calculated as:

$$FPR = \frac{FP}{FP + TN} \tag{15}$$

- False Negative Rate (FNR): The proportion of anomaly that is identified as normal. The FNR is calculated as:

$$FNR = \frac{FN}{TP + FN} \tag{16}$$

- Cross-over Error Rate (CER): Is the difference between false negative rate (FNR) and the false positive rate (FPR). Since the main aim of this work is to reduce the wrongly predicted instances for both types of errors FNR and FPR, and to minimize the intersection point between these two rates which calculated as:

$$CER = |FPR - FNR| \tag{17}$$

- Precision (P): the percentage of the total number of the indeed predicted attack instances divided by the total number of predicted attack instances:

$$Precision = \frac{TP}{TP + FP} \tag{18}$$

- Recall (R): also called sensitivity, which is the percentage of total attacks instances that were correctly classified, true positives (TP), divided by the total number of the actual attacks instances:

$$Recall = \frac{TP}{TP + FN} \tag{19}$$

- F1-Score (F-Measure): The FM is the mean of the precision and recall. Which is calculated as:

$$F1 - Score = \frac{2 * Recall * Precision}{Recall + Precision} \tag{20}$$

Table 6
Results in terms of average of the proposed IHHO compared with other algorithms over 30 runs based on fitness function in Eq. 12.

Algorithms	Sensitivity	F1_score	Accuracy	NumOf Features	Reduction Rate
GA	0.9968	0.8035	0.7281	15.75	62.50%
PSO	0.9990	0.7984	0.7193	15.05	64.17%
GOA	0.9979	0.7996	0.7181	**12.95**	**69.17%**
ALO	0.9994	0.7974	0.7173	16.55	60.60%
SSA	**0.9996**	0.7997	0.7228	23.95	42.98%
HHO	0.9995	0.7922	0.7098	22.85	45.60%
IHHO25%	0.9984	0.8073	0.7339	15.95	62.02%
IHHO50%	0.9988	0.8083	0.7373	17.60	58.10%
IHHO75%	0.9981	**0.8118**	**0.7421**	17.40	58.57%
IHHO100%	0.9938	0.7883	0.7022	16.30	61.19%

Table 7
Results in terms of average of the proposed IHHO compared with other algorithms over 30 runs based on fitness function in Eq. 13.

Algorithms	Sensitivity	F1_score	Accuracy	CER	NumOf Features	Reduction Rate
GA	0.9061	0.8069	0.7619	0.3209	20.05	52.26%
PSO	0.8969	0.8055	0.7622	0.2997	20.30	51.67%
GOA	0.8780	0.8001	0.7583	0.2665	**18.20**	**56.67%**
ALO	0.9003	0.8050	0.7603	0.3114	20.85	50.36%
SSA	**0.9226**	**0.8095**	0.7612	0.3590	24.95	40.60%
HHO	0.8954	0.8029	0.7583	0.3050	25.95	38.21%
IHHO25%	0.8939	0.8043	0.7609	0.2959	22.60	46.19%
IHHO50%	0.9147	0.8055	0.7572	0.3505	24.60	41.43%
IHHO75%	0.8701	0.7995	0.7602	**0.2447**	23.05	45.12%
IHHO100%	0.8887	0.8047	**0.7629**	0.2801	24.10	42.62%

- G-Mean: Sensitivity and Specificity can be combined into a single score that balances both concerns.in order to evaluate the classification algorithm to deal with an imbalanced data. Geometric mean (G-Mean) is calculated as follow:

$$G - Mean = \sqrt{Recall * Precision} \tag{21}$$

4.5. Binary classification results

In this section, preliminary results will be conducted in the binary framework by converting all the attacks labels into one abnormal class label and leaving the normal. Accordingly, the dataset will be formed of two class labels; normal and abnormal labels. For an accurate evaluation, the two fitness functions described in Section 4.3 were used separately in all tested algorithms. Moreover, the four versions of the proposed IHHO, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Grasshopper Optimization Algorithm (GOA), Ant Lion Optimization (ALO), Salp Swarm Algorithm (SSA), and the original HHO will be evaluated and compared.

Table 6 shows the results obtained using the fitness function that is represented by Eq. 12. These values were obtained over 30 runs, and the average was calculated for each measure. Based on the accuracy and the reduction rate, the IHHO with a 75% injection ratio achieved the best accuracy and F1-score with 74% and 81%, respectively. GOA performs the best in terms of reduction rate.

Table 7 shows the conducted the average results over 30 runs by using the crossover error rate as described in Eq. 13 for optimization. Although, as clearly shown, the best CER value (minimum) was obtained in the IHHO with a 75% injection ratio of 24%, the minimum CER did not mean the best accuracy since both CER and the reduction rate were employed in the fitness function.

The best-performed version of the IHHO will be obtained and considered from these results for the upcoming experiments. The IHHO with a 75% injection ratio is the best performed one, so from now we will call it the IHHO.

4.6. Comparison with other meta-heuristics

This subsection will present a comparison between the proposed IHHO and well-known meta-heuristic algorithms. The proposed multi-layers framework was used for all the compared algorithms to show the model's strength. Table 8 shows the results of the proposed IHHO, standard HHO, GA, and GWO algorithms in terms of classification accuracy, F1-score, G-mean, and CER.

The improved HHO (IHHO) outperforms the other algorithms in all measures. This improvement in the results is referred to using the smart initialization technique, which ensures the best fitness values in early iterations and uses a robust searching technique to enhance the searching. The IHHO wins in 5 types of attacks regarding classification accuracy, F1-score, and G-mean measures. Furthermore, IHHO performs better CER in 6 types of attacks and shows the same CER as GA and GWO due to the shellcode attack.

The used improvements show an enhancement in the speed of convergence, Fig. 8 shows the convergence curves of the IHHO compared with the standard HHO for all types of attacks. On the other hand, the smart initialization technique suffers from obtaining the best fitness in early iterations. And the use of the X-shaped transfer function makes the convergence more stable with a superior reduction rate than the standard HHO. The exact values for the average best fitness values and the average number of selected features are illustrated in Table 9.

4.7. Comparison with other multi-class classifiers

In order to show the strength of the proposed IHHO against multi-class classifiers, various classification algorithms were experimented with and compared. Table 10 shows the results of the

Table 8
Comparison between IHHO, HHO, GA, and GWO in terms of average Classification Accuracy, F1-Score, G-mean, and CER over 30 runs.

Classes	Mesaure	BHHO	GA	GWO	IHHO
analysis	Accuraccy	0.9986	**0.9992**	0.9968	**0.9992**
	F1-Score	0.8777	0.9399	0.7577	**0.9414**
	G_Mean	0.8920	0.9438	0.7813	**0.9451**
	CER	0.1666	0.1034	0.3884	**0.1013**
backdoor	Accuraccy	0.9977	0.9995	0.9957	**0.9996**
	F1-Score	0.7438	0.9645	0.5415	**0.9701**
	G_Mean	0.7812	0.9653	0.6087	**0.9706**
	CER	0.3248	0.0678	0.6134	**0.0579**
dos	Accuraccy	0.9900	0.9927	0.9894	**0.9928**
	F1-Score	0.8883	**0.9198**	0.8795	0.9075
	G_Mean	0.8939	**0.9231**	0.8863	0.9163
	CER	0.2007	0.1472	0.2137	**0.1456**
exploits	Accuraccy	0.9887	0.9895	0.9908	**0.9958**
	F1-Score	0.9554	0.9585	0.9635	**0.9829**
	G_Mean	0.9569	0.9599	0.9646	**0.9838**
	CER	0.0832	0.0774	0.0684	**0.0307**
fuzzers	Accuraccy	0.9868	0.9894	0.9902	**0.9968**
	F1-Score	0.8952	0.9079	0.9249	**0.9776**
	G_Mean	0.9023	0.9122	0.9292	**0.9780**
	CER	0.1795	0.1444	0.1330	**0.0434**
generic	Accuraccy	0.9939	**0.9962**	0.9957	0.9951
	F1-Score	0.9865	**0.9916**	0.9906	0.9891
	G_Mean	0.9866	**0.9917**	0.9906	0.9891
	CER	0.0266	**0.0166**	0.0187	0.0216
normal	Accuraccy	0.9396	0.9662	0.9538	**0.9784**
	F1-Score	0.9373	0.9641	0.9513	**0.9769**
	G_Mean	0.9435	0.9686	0.9570	**0.9800**
	CER	0.1097	0.0615	0.0839	**0.0393**
reconnaissance	Accuraccy	0.9838	**0.9997**	0.9955	0.9992
	F1-Score	0.7399	**0.9968**	0.9372	0.9887
	G_Mean	0.7736	**0.9968**	0.9420	0.9894
	CER	0.3804	**0.0064**	0.1063	0.0197
shellcode	Accuraccy	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	F1-Score	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	G_Mean	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	CER	0.0001	**0.0000**	**0.0000**	**0.0000**
worms	Accuraccy	**1.0000**	**1.0000**	0.9998	**1.0000**
	F1-Score	**0.9767**	0.9727	0.7580	0.9740
	G_Mean	**0.9770**	0.9731	0.7820	0.9752
	CER	**0.0455**	0.0530	0.3864	0.0470
		BHHO	GA	GWO	IHHO
Ranking (W|T|L)	Accuraccy	0|2|8	2|3|5	0|1|9	5|3|2
	F1-Score	1|1|8	3|1|6	0|1|9	5|1|4
	G_Mean	1|1|8	3|1|6	0|1|9	5|1|4
	CER	1|0|9	2|1|7	0|1|9	6|1|3
Ranks (F-test)	Accuraccy	3.35	2	3.05	**1.6**
	F1-Score	3.25	2.05	3.05	**1.65**
	G_Mean	3.25	2.05	3.05	**1.65**
	CER	3.4	2.1	3	**1.5**

Table 9
The average number of the selected features and the Best fitness values over 30 runs of the proposed IHHO compared with the standard HHO.

	Average Best Fitness Values		Average No. of Selected Features	
Classes	IHHO	HHO	IHHO	HHO
analysis	**0.0042**	0.0081	**15.50**	19.67
backdoor	**0.0465**	0.0729	**13.73**	19.47
dos	**0.0595**	0.0820	**15.57**	20.00
exploits	**0.0145**	0.0391	**12.83**	19.77
fuzzers	**0.0431**	0.0670	**7.67**	21.10
generic	**0.0162**	0.0211	**10.20**	20.00
normal	**0.0110**	0.0130	**13.10**	21.00
	0.0133	0.0244	**12.40**	17.10
reconnaissance				
shellcode	**0.0017**	0.0048	**7.00**	19.73
worms	**0.0223**	0.0425	**12.47**	18.00

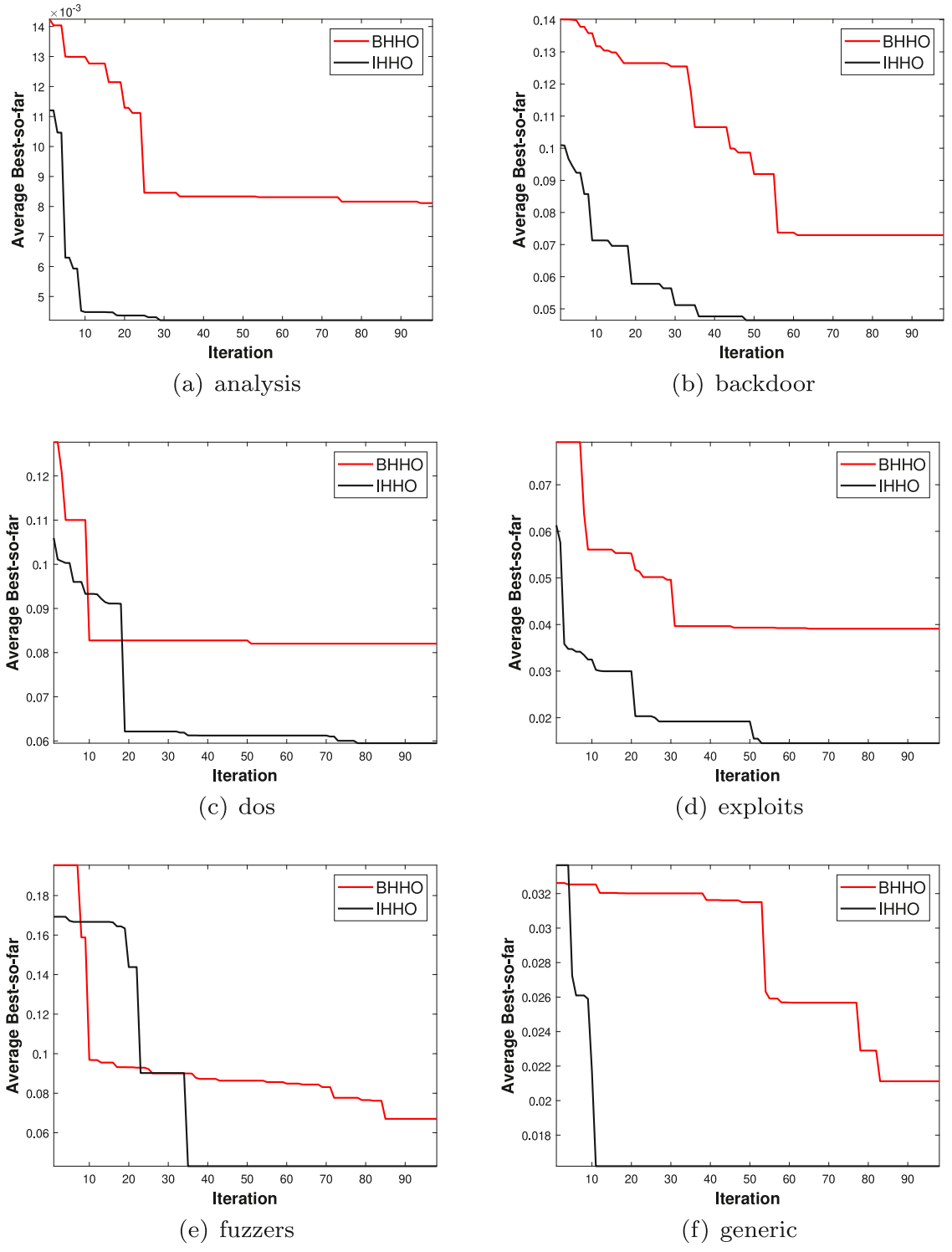

Fig. 8. Convergence curves for the standard BHHO and IHHO for different types of attacks.

proposed IHHO in comparison to ELM, K-nearest neighbors (KNN), and Decision Trees (DT) classifiers. These classifiers were used as a multi-class classifier on the same dataset. The results show the superior performance of the IHHO in the classification of each type of attack.

The results conclude that the proposed model based on the one-vs-all approach works well on the multi-class classification problems, and the IHHO improves the performance of the IDS.

4.8. Comparison with other algorithms reported in the literature

Various experiments were conducted on the UNSW-NB15 dataset using different techniques. As such, (Sharma et al., 2019) proposed a one-vs-all approach based on ELM and Weighted ELM (WELM), with an Extra-trees classifier for features selection. (Gharaee and Hosseinvand, 2016) presented an anomaly-based IDS using GA and a support vector machine (SVM) to deal with multi-class classification in the dataset.

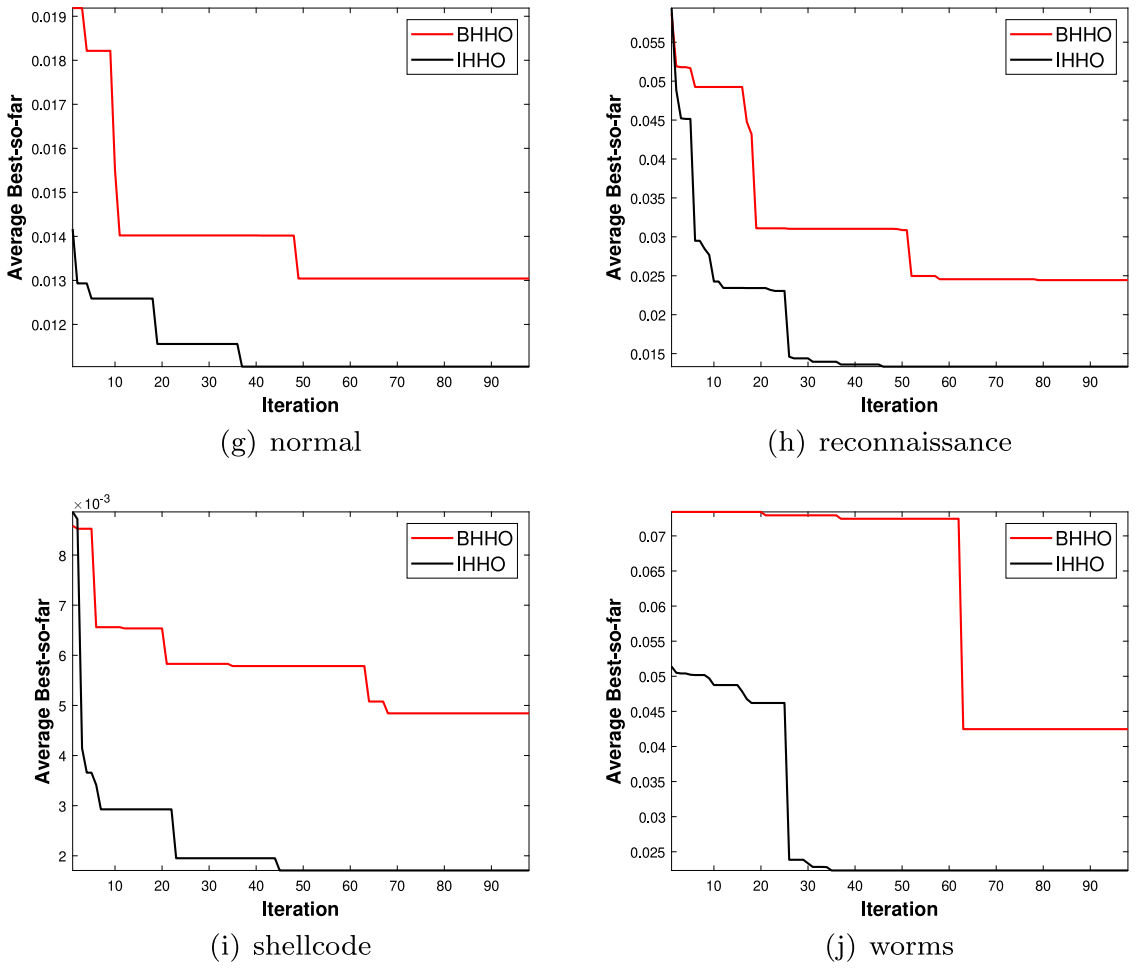

Fig. 8. Continued

(Salman et al., 2017) used Linear Regression (LR) and Random Forest (RF) for detecting and categorization multi-attacks in the cloud environment. While (Moustafa et al., 2018) suggests a new beta mixture technique (BMM-ADS) as a one-class classification by training the model to detect the normal traffic and considering all others are attacks.

As the revolution of deep learning (DL), the work of (Ashiku and Dagli, 2021) shows the effect of using DL in the IDS. To identify the attacks in the UNSW-NB15 dataset, the authors used a convolutional neural network (CNN) with regularized multi-layer perceptron. In addition, they used the original imbalanced dataset and a stratified sampled dataset called a user-defined dataset.

Table 11 shows the comparison between the proposed IHHO and the rest of the experiments reported in the literature in terms of classification accuracy. The IHHO outperforms the others in all types of attacks. But the normal type because the IHHO aims to detect the attacks instead of the normal traffic. The results show significant differences in the obtained values.

The IHHO outperforms the work of (Sharma et al., 2019), which is also used the ELM and WELm. This is because the technique was utilized to overcome the imbalance of the dataset, the broad searching capabilities of the IHHO, and the tight integration of the ELM with the IHHO.

4.9. The most frequent-based feature analysis

The selected features for each attack were deeply analyzed to highlight the importance of each feature in detecting these attacks. The feature frequency is appearances over the number of runs used to rank the features. The IHHO determines the best features set to decide this attack for each type of attack and at each run. Here, the most frequent features overall the runs will be considered the essential features for that type of attack (Al-Daweri et al., 2020). The frequency of each feature was calculated using Eq. 22, Then, these features were ranked by dividing the feature's frequency over the total number of runs as Eq. 23. Where n is the number of runs, F_{freq}, and $Rank_f$ is the feature's frequency and the feature's Rank, respectively, and F_i is the appearance of the feature F in the i^{th} run.

$$F_{freq} = \sum_{i=1}^{n} F_i \tag{22}$$

$$Rank_f = \frac{F_{freq}}{n} \tag{23}$$

The overall sorted ranks for the features in the dataset are shown in Fig. 9. The overall ranks are obtained by dividing the total frequency of feature attacks by the total number of runs (10 classes and 30 runs for each, which produce 300 runs). It is clearly shown the is_ftp_login feature is the most informative one with a ranking weight of 59.3%, secondly, the *dloss* in the second rank, and so on.

5. Conclusion and future work

This research proposes an effective intrusion detection system that can deal with a dynamic, realistic environment. The proposed

Table 10
Comparison between IHHO and multi-class classifiers in terms of average Classification Accuracy, F1-Score, G-mean, and CER over 30 runs.

Classes	Mesaure	IHHO	ELM	KNN	DT
analysis	Accuraccy	**0.9992**	0.9426	0.9073	0.9647
	F1-Score	**0.9414**	0.0104	0.0754	0.0249
	G_Mean	**0.9451**	0.1440	0.6384	0.2014
	CER	**0.1013**	0.9001	0.4578	0.9202
backdoor	Accuraccy	**0.9996**	0.9782	0.9257	0.9234
	F1-Score	**0.9701**	0.0109	0.0392	0.0483
	G_Mean	**0.9706**	0.1112	0.4261	0.4874
	CER	**0.0579**	0.9509	0.7254	0.6349
dos	Accuraccy	**0.9928**	0.9188	0.9399	0.9318
	F1-Score	**0.9075**	0.2514	0.1442	0.1397
	G_Mean	**0.9163**	0.5084	0.3128	0.3273
	CER	**0.1456**	0.6557	0.8785	0.8570
exploits	Accuraccy	**0.9958**	0.8728	0.8937	0.8867
	F1-Score	**0.9829**	0.3825	0.5308	0.5108
	G_Mean	**0.9838**	0.5309	0.6529	0.6467
	CER	**0.0307**	0.6484	0.5209	0.5185
fuzzers	Accuraccy	**0.9968**	0.8399	0.8310	0.7802
	F1-Score	**0.9776**	0.2408	0.2912	0.2568
	G_Mean	**0.9780**	0.5419	0.6360	0.6368
	CER	**0.0434**	0.5102	0.3811	0.2893
generic	Accuraccy	**0.9951**	0.9225	0.9874	0.9482
	F1-Score	**0.9891**	0.8186	0.9721	0.8664
	G_Mean	**0.9891**	0.8615	0.9754	0.8792
	CER	**0.0216**	0.2073	0.0436	0.2167
normal	Accuraccy	**0.9784**	0.7401	0.7628	0.7545
	F1-Score	**0.9769**	0.6213	0.6449	0.6321
	G_Mean	**0.9800**	0.6722	0.6904	0.6806
	CER	**0.0393**	0.4805	0.5127	0.5180
reconnaissance	Accuraccy	**0.9992**	0.9375	0.9430	0.9416
	F1-Score	**0.9887**	0.1258	0.3777	0.1964
	G_Mean	**0.9894**	0.2668	0.6221	0.3960
	CER	**0.0197**	0.8343	0.5516	0.7881
shellcode	Accuraccy	**1.0000**	0.9065	0.9438	0.8960
	F1-Score	**1.0000**	0.0531	0.0949	0.0713
	G_Mean	**1.0000**	0.6211	0.7604	0.8345
	CER	**0.0000**	0.4450	0.3035	0.1172
worms	Accuraccy	**1.0000**	0.9165	0.9874	0.9867
	F1-Score	**0.9740**	0.0152	0.0584	0.0382
	G_Mean	**0.9752**	0.8459	0.8197	0.5446
	CER	**0.0470**	0.1724	0.2898	0.6772
		IHHO	ELM	KNN	DT
Ranking (W|T|L)	Accuraccy	10|0|0	0|0|10	0|0|10	0|0|10
	F1-Score	10|0|0	0|0|10	0|0|10	0|0|10
	G_Mean	10|0|0	0|0|10	0|0|10	0|0|10
	CER	10|0|0	0|0|10	0|0|10	0|0|10
Ranks (F-test)	Accuraccy	**1**	3.4	2.4	3.2
	F1-Score	**1**	3.8	2.2	3
	G_Mean	**1**	3.6	2.6	2.8
	CER	**1**	3.2	2.8	3

Table 11
Comparison between the IHHO and other algorithms from previous works based on the average accuracy results.

Classes	IHHO	ELM (Sharma et al., 2019)	WELM (Sharma et al., 2019)	GF+SVM (Gharaee and Hosseinvand, 2016)	Step-wise RF(Salman et al., 2017)	BMM+outlier detection (Moustafa et al., 2018)	CNN original (Ashiku and Dagli, 2021)	CNN user-defined (Ashiku and Dagli, 2021)
analysis	**99.92**	98.96	99.26	-	2.00	83.40	89.50	90.60
backdoor	**99.96**	99.11	99.11	-	5.00	63.80	91.20	92.20
dos	**99.28**	94.75	94.90	91.24	20.00	89.60	94.60	93.80
exploits	**99.58**	89.13	90.12	79.19	99.50	79.40	94.20	93.80
fuzzers	**99.68**	91.30	91.47	96.39	-	52.80	88.60	90.10
generic	**99.51**	98.16	98.23	91.51	97.00	80.50	95.10	95.30
normal	97.84	91.26	93.54	97.45	**99.50**	93.40	97.20	97.90
reconnaissance	**99.92**	94.60	95.33	91.51	86.00	55.60	95.10	96.30
shellcode	**100.00**	99.40	99.40	99.45	80.00	48.70	91.60	91.50
worms	**100.00**	99.92	99.92	–	70.00	47.80	89.80	90.80
Ranking (W|T|L)	9|0|1	0|0|10	0|0|10	0|0|7	1|0|8	0|0|10	0|0|10	0|0|10

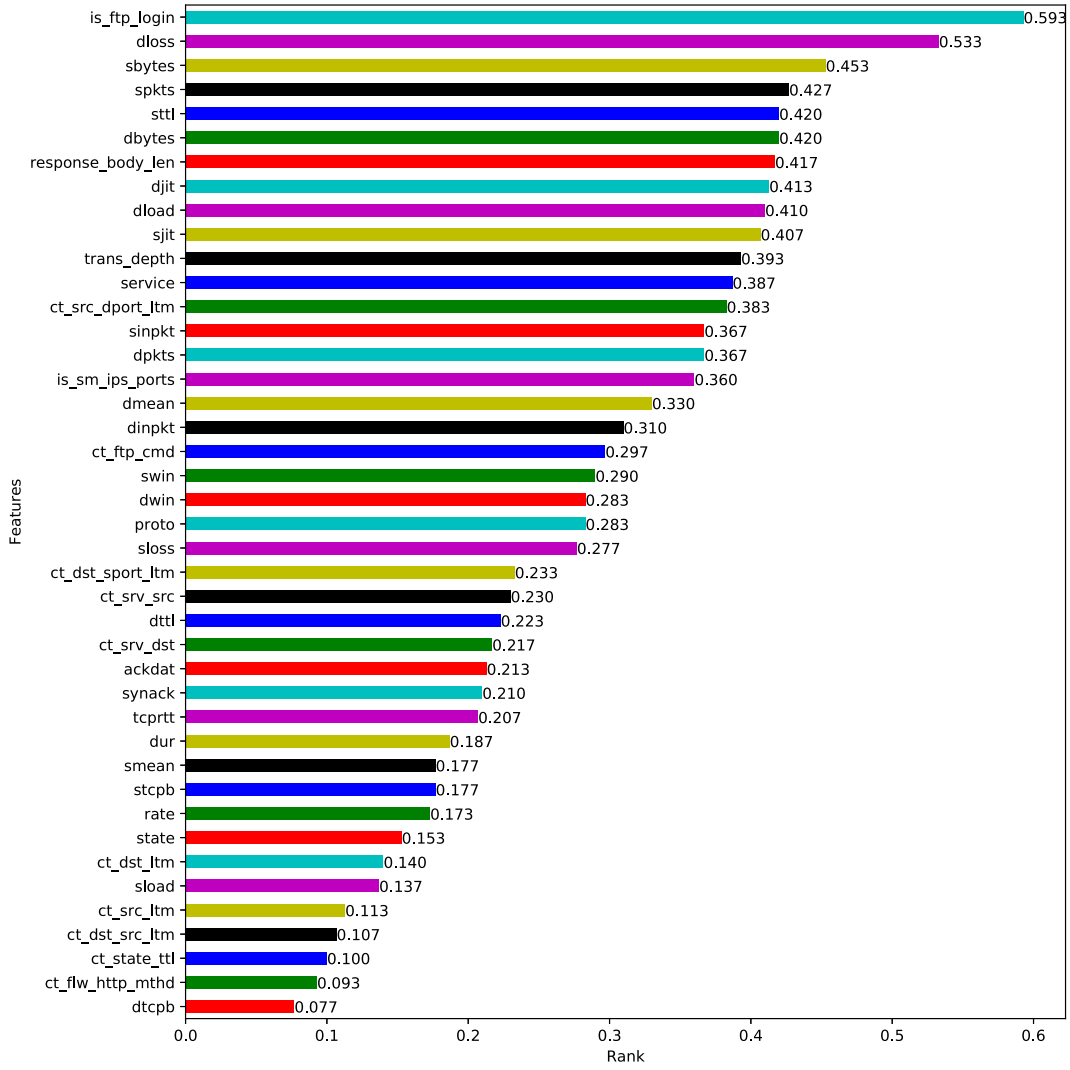

Fig. 9. Overall Ordered Ranks.

IDS can discriminate between multiple sorts of attacks simultaneously. Feature selection is a preprocessing technique that reduces the dataset's dimension by removing irrelevant, redundant, and noisy features. The best features subset is generated in this study by combining the capability of the extreme learning machine with an enhanced harris hawks optimization algorithm. Furthermore, the significance of the features varies depending on the attack. The model is trained for each type of attack using the one-versus-all strategy, which extracts the most significant features for that attack. Simultaneously, the optimization procedure focuses on picking the best features subset and optimizing the ELM's weights in each type.

The proposed modification in the HHO accelerates the algorithm's convergence by improving the initialization step. The x-shaped transfer function, on the other hand, is utilized to avoid trapping in local optima and improve the HHO's searching strategy. The results on the UNSWNB-15 dataset show the superior performance of the proposed work against other meta-heuristic algorithms and with state-of-the-art multi-class classifiers. For example, the crossover-error rate achieved the least values in 7 types of attacks. Furthermore, it performs the best G-mean values (greater than 94%) in 6 types. In all measures, the proposed work outperforms all other multi-class classifiers, ELM, KNN, and DT.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Abdullah Alzaqebah: Writing – original draft, Methodology, Software. **Ibrahim Aljarah:** Methodology. **Omar Al-Kadi:** Conceptualization, Writing – review & editing.

References

Acharya, N., Singh, S., 2018. An IWD-based feature selection method for intrusion detection system. Soft Comput. 22 (13), 4407–4416.
Al-Daweri, M.S., Zainol Ariffin, K.A., Abdullah, S., et al., 2020. An analysis of the kdd99 and unsw-nb15 datasets for the intrusion detection system. Symmetry 12 (10), 1666.
Al-Kadi, O., 2020. Defending Against Anomalies in Cloud Services and Live Migration. University of New South Wales, Sydney, Australia.
Alazzam, H., Sharieh, A., Sabri, K.E., 2020. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. Expert Syst. Appl. 148, 113249.
Alkadi, O., Moustafa, N., Turnbull, B., 2020. A review of intrusion detection and blockchain applications in the cloud: approaches, challenges and solutions. IEEE Access 8, 104893–104917.

Alkadi, O., Moustafa, N., Turnbull, B., Choo, K.-K.R., 2020. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks. IEEE Internet Things J. 8 (12), 9463–9472.
Alzaqebah, A., Aljarah, I., Al-Kadi, O., Damaeviius, R., 2022. A modified grey wolf optimization algorithm for an intrusion detection system. Mathematics 10 (6). doi:10.3390/math10060999.
Alzaqebah, A., Smadi, B., Hammo, B.H., 2020. Arabic sentiment analysis based on salp swarm algorithm with s-shaped transfer functions. In: 2020 11th International Conference on Information and Communication Systems (ICICS). IEEE, pp. 179–184.
Alzubi, Q.M., Anbar, M., Alqattan, Z.N., Al-Betar, M.A., Abdullah, R., 2020. Intrusion detection system based on a modified binary grey wolf optimisation. Neural Comput. Appl. 32 (10), 6125–6137.
Ashiku, L., Dagli, C., 2021. Network intrusion detection system using deep learning. Procedia Comput. Sci. 185, 239–247.
Awasthi, L.K., Sikka, G., et al., 2020. Behavior-based approach for fog data analytics: an approach toward security and privacy. In: Fog Data Analytics for IoT Applications. Springer, pp. 341–354.
Basnet, R.B., Shash, R., Johnson, C., Walgren, L., Doleck, T., 2019. Towards detecting and classifying network intrusion traffic using deep learning frameworks. J. Internet Serv. Inf. Secur. 9 (4), 1–17.
Beheshti, Z., 2021. A novel x-shaped binary particle swarm optimization. Soft Comput. 25 (4), 3013–3042.
Emary, E., Zawbaa, H.M., Hassanien, A.E., 2016. Binary grey wolf optimization approaches for feature selection. Neurocomputing 172, 371–381.
Faris, H., Mafarja, M.M., Heidari, A.A., Aljarah, I., AlaM, A.-Z., Mirjalili, S., Fujita, H., 2018. An efficient binary salp swarm algorithm with crossover scheme for feature selection problems. Knowl. Based Syst. 154, 43–67.
Feng, Z.-k., Niu, W.-j., Tang, Z.-y., Xu, Y., Zhang, H.-r., 2021. Evolutionary artificial intelligence model via cooperation search algorithm and extreme learning machine for multiple scales nonstationary hydrological time series prediction. J. Hydrol. 595, 126062.
Gao, Z., Xu, Y., Meng, F., Qi, F., Lin, Z., 2014. Improved information gain-based feature selection for text categorization. In: 2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE). IEEE, pp. 1–5.
Gharaee, H., Hosseinvand, H., 2016. A new feature selection ids based on genetic algorithm and SVM. In: 2016 8th International Symposium on Telecommunications (IST). IEEE, pp. 139–144.
Ghosh, K.K., Singh, P.K., Hong, J., Geem, Z.W., Sarkar, R., 2020. Binary social mimic optimization algorithm with x-shaped transfer function for feature selection. IEEE Access 8, 97890–97906.
Hammouri, A.I., Mafarja, M., Al-Betar, M.A., Awadallah, M.A., Abu-Doush, I., 2020. An improved dragonfly algorithm for feature selection. Knowl. Based Syst. 203, 106131.
Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H., 2019. Harris hawks optimization: algorithm and applications. Future Gener. Comput. Syst. 97, 849–872.
Huang, G.-B., Zhu, Q.-Y., Siew, C.-K., 2004. Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541), Vol. 2. Ieee, pp. 985–990.
Hussien, A.G., Amin, M., 2022. A self-adaptive harris hawks optimization algorithm with opposition-based learning and chaotic local search strategy for global optimization and feature selection. Int. J. Mach. Learn. Cybern. 13 (2), 309–336.
Kardani, N., Bardhan, A., Roy, B., Samui, P., Nazem, M., Armaghani, D.J., Zhou, A., 2021. A novel improved harris hawks optimization algorithm coupled with elm for predicting permeability of tight carbonates. Eng. Comput. 1–24.
Kasongo, S.M., Sun, Y., 2019. A deep learning method with filter based feature engineering for wireless intrusion detection system. IEEE Access 7, 38597–38607.
Khalvati, L., Keshtgary, M., Rikhtegar, N., 2018. Intrusion detection based on a novel hybrid learning approach. J. AI Data Mining 6 (1), 157–162.
Krishnaveni, S., Vigneshwar, P., Kishore, S., Jothi, B., Sivamohan, S., 2020. Anomaly-based intrusion detection system using support vector machine. In: Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, pp. 723–731.
Liu, J., Yu, F.R., Lung, C.-H., Tang, H., 2009. Optimal combined intrusion detection and biometric-based continuous authentication in high security mobile ad hoc networks. IEEE Trans. Wireless Commun. 8 (2), 806–815.
Mafarja, M., Aljarah, I., Faris, H., Hammouri, A.I., AlaM, A.-Z., Mirjalili, S., 2019. Binary grasshopper optimisation algorithm approaches for feature selection problems. Expert Syst. Appl. 117, 267–286.
Mafarja, M., Mirjalili, S., 2018. Whale optimization approaches for wrapper feature selection. Appl. Soft Comput. 62, 441–453.
Mafarja, M.M., Eleyan, D., Jaber, I., Hammouri, A., Mirjalili, S., 2017. Binary dragonfly algorithm for feature selection. In: 2017 International Conference on New Trends in Computing Sciences (ICTCS). IEEE, pp. 12–17.
Malathi, R., Manimekalai, M., 2021. Ant colony–information gain based feature selection method for weather dataset. Ann. Rom. Soc. Cell Biol. 3838–3850.
Mirjalili, S., Lewis, A., 2013. S-shaped versus v-shaped transfer functions for binary particle swarm optimization. Swarm Evol. Comput. 9, 1–14.
Moustafa, N., Creech, G., Slay, J., 2017. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. In: Data Analytics and Decision Support for Cybersecurity. Springer, pp. 127–156.
Moustafa, N., Creech, G., Slay, J., 2018. Anomaly detection system using beta mixture models and outlier detection. In: Progress in Computing, Analytics and Networking. Springer, pp. 125–135.
Moustafa, N., Slay, J., 2015. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: 2015 Military Communications and Information Systems Conference (MilCIS). IEEE, pp. 1–6.
Moustafa, N., Slay, J., 2016. The evaluation of network anomaly detection systems: statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set. Inf. Secur. J. 25 (1–3), 18–31.
Moustafa, N., Slay, J., Creech, G., 2017. Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks. IEEE Trans. Big Data 5 (4), 481–494.
Piri, J., Mohapatra, P., 2021. An analytical study of modified multi-objective harris hawk optimizer towards medical data feature selection. Comput. Biol. Med. 104558.
Qaddoura, R., Al-Zoubi, M., Faris, H., Almomani, I., et al., 2021. A multi-layer classification approach for intrusion detection in IoT networks based on deep learning. Sensors 21 (9), 2987.
Salman, T., Bhamare, D., Erbad, A., Jain, R., Samaka, M., 2017. Machine learning for anomaly detection and categorization in multi-cloud environments. In: 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud). IEEE, pp. 97–103.
Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2020. Netflow datasets for machine learning-based network intrusion detection systems. arXiv preprint arXiv:2011.09144.
Sharma, J., Giri, C., Granmo, O.-C., Goodwin, M., 2019. Multi-layer intrusion detection system with extratrees feature selection, extreme learning machine ensemble, and softmax aggregation. EURASIP J. Inf. Secur. 2019 (1), 1–16.
Shushlevska, M., Efnusheva, D., Jakimovski, G., Todorov, Z., 2022. Anomaly detection with various machine learning classification techniques over UNSW-nb15 dataset. Appl. Innov. IT 21.
Tama, B.A., Rhee, K.H., 2015. 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, pp. 489–495.
Too, J., Abdullah, A.R., Mohd Saad, N., 2019. A new quadratic binary harris hawk optimization for feature selection. Electronics 8 (10), 1130.
Wolpert, D.H., Macready, W.G., 1997. No free lunch theorems for optimization ieee transactions on evolutionary computation. E997.
Zhou, Y., Cheng, G., Jiang, S., Dai, M., 2020. Building an efficient intrusion detection system based on feature selection and ensemble classifier. Comput. Netw. 174, 107247.

Abdullah Alzaqebah is a Ph.D. candidate at the computer science department in the University of Jordan. He is also a System Analyst and Software Developer and Assistant of Computer Center Director at the World Islamic Sciences & Education University. His PhD research is concerned with improving the performance of security applications by employing nature-inspired techniques for anomaly detection. His research interest include Machine Learning, Evolutionary Computations, Artificial Intelligence, Natural Language Processing.

Ibrahim Aljarah is an Associate Professor in the Department of Information Technology at the University of Jordan, Amman, Jordan. He was born in a small Jordanian village called Almazar located in Irbid Southwest, in 1981. He received a High school degree in science from Almazar School, Irbid-Jordan. He obtained a bachelor's degree in Computer Science from Yarmouk University - Jordan, 2003. Ibrahim also obtained a master's degree in Computer Science and Information Systems from the Jordan University of Science and Technology - Jordan in 2006. After graduation, he was worked in University of Jordan as Online Exams Administrator. Through this period, he was granted a scholarship from University Of Jordan, Amman to complete his PhD degree. He received his PhD in Computer Science from the North Dakota State University (NDSU), USA, in May 2014. He is especially interested with Data mining, Big Data, MapReduce, Hadoop, Swarm intelligence, Evolutionary Computation, and large scaledistributed algorithms. He loves all sporting and outdoor events, particularly soccer, and enjoys listening to music.

Omar Al-Kadi is a Professor in Computational and Machine Intelligence at King Abdullah II School for Information Technology - University of Jordan. He has many contributions to the development of Artificial Intelligent-based solutions and decision support systems for improve learning; with particular interest in clinical diagnosis and disease management. Other research achievements include applying natureinspired algorithms, by learning from the social behaviour of real-world examples of organisms, for designing more intelligent algorithms. Moreover, data mining and machine intelligence techniques have been employed for identifying personalised learning styles in Web-based Educational frameworks. These works have been documented in scholarly papers published in numerous scientific journals and proceedings of world conferences. He is also a senior member of the Jordan Engineers Association (JEA), the Institute of Electrical and Electronic Engineers (IEEE) and Engineers Australia (EA).