Empowering Intrusion Detection Systems: A Synergistic Hybrid Approach with Optimization and Deep Learning Techniques for Network Security

Ramya Chinnasamy
Department of Computer Science and Engineering, Anna University, India
ramya.me@gmail.com

Malliga Subramanian
Department of Computer Science and Engineering, Anna University, India
mallinishanth72@gmail.com

Nandita Sengupta
Department of Information Technology
University of Bahrain, Bahrain
ngupta@ucb.edu.bh

Abstract: *Over last decade, there is a rapid advancement in networking and computing technologies that produced large volume of sensitive data. Clearly, protecting those data from intrusions and attack is of paramount importance. Researchers have proposed many cyber security solutions and tools to protect the data. One such technique for safeguarding data is the Intrusion Detection System (IDS). This research introduces a hybrid optimization-based Feature Selection (FS) and deep learning-driven categorization namely Honey Badger Optimization-Artificial Neural Network (HBO-ANN) to identify intrusions. The Honey Badger Optimization (HBO) is an optimization technique that is utilized to choose the dataset's most important features. The Artificial Neural Network (ANN) receives reduced features dataset and classifies it as benign or attack. Additionally, a well-known CIC-IDS 2017 dataset is employed to construct and validate the suggested system. Performance metrics for assessing the effectiveness of the suggested system are the false alarm rate, Mean Squared Error (MSE), precision, accuracy and recall. The testing and training MSEs are 0.009 and 0.00317, respectively. The model's accuracy is 97.66%. The model has a precision of 98.03% and a recall of 97.18%. There is a 1.97% false alarm rate. The outcomes have been compared with bench mark models such as Grey Wolf Optimizer-Support Vector Machine (GWO-SVM), Particle Swarm Optimization-Support Vector Machine (PSO-SVM), Fuzzy Clustering-Artificial Neural Network (FC-ANN), Bidirectional Long-Short-Term-Memory (BiDLSTM) and Feed-Forward Deep Neural Network (FFDNN). As demonstrated by the experimental results, the suggested model outperforms the benchmark algorithms.*

Keywords: *Artificial neural network, deep learning, honey badger optimization, intrusion detection system.*

Received July 3, 2024; accepted October 30, 2024
https://doi.org/10.34028/iajit/22/1/6

1. Introduction

A preliminary version of this paper appeared in IEEE ITIKD 2023 [12], March 08-09, Manama, Bahrain. This version includes a detailed methodology, comprehensive analysis of CIC-IDS2017 dataset, complete description of data pre-processing, elaborate details of Honey Badger Optimization (HBO) Feature Selection (FS), detailed experiments, through analysis of the results and result comparison with benchmark algorithms. In recent timeframe, there has been substantial development in computing system hardware, software, and networking technologies. This rapid connection development consequently increases susceptibility, which opens the door for malicious attacks. Cybersecurity is therefore a crucial subject for research [12]. Typically, the fundamental requirements of any cyber security system are confidentiality, integrity, and availability [22]. Intrusion Detection Systems (IDSs) could be the most effective way for strengthening cyber safety in system and connected environments. Screening the computing and networking infrastructure for hazards is known as intrusion detection [33].

1.1. Intrusion Detection Systems (IDS)

A tool dedicated for identifying and flagging potential intrusions is normally referred to as an IDS [33]. An IDS can safeguard against threats in any computing environment, including the smart cities, Internet of Things (IoT) and wireless networks. Based on detection and deployment techniques, IDS come in two varieties. The classification of IDS is shown in Figure 1.

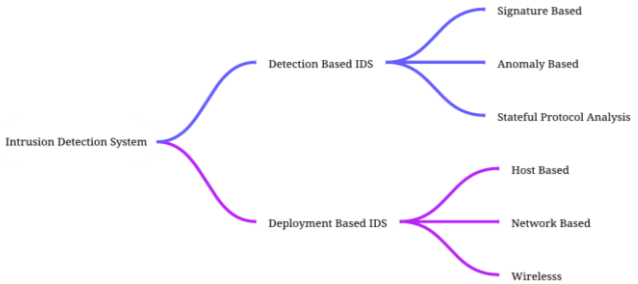


Figure 1. IDS categorization.

An IDS can be categorized as deployment-driven IDS based on where it is installed [18]. IDSs that are installed on individual device or host are considered to be host-based IDS. When the intended location is a distributed environment or network of computers, an IDS based on the network is utilized [1, 4]. Because most businesses are moving to distributed environments, network-driven IDSs are becoming more popular [11]. In addition, IDS is divided into three categories based on how it recognizes attacks: anomaly-based, signature-based and stateful protocol analysis [11]. A signature-based IDS determines potential attacks through the comparison of the traffic that comes in to established attacks recorded in a database of signatures [10, 19]. It is obvious that the IDS that relies on signatures can only detect known attacks. Also, it is ineffective to recognize fresh threats. The most recent transition to a distributed environment has significantly reduced the efficiency of signature-based IDS. The anomaly-based detection method, on the other hand, identifies potential risks by looking at the inbound network flow pattern [38]. Evidently, anomaly-driven IDS is best suited in detecting unidentified threats [38]. It recognizes unidentified threats by confirming patterns that separates typical and unusual behaviours. Both normal and abnormal actions are described by the network administrator [13, 21].

Certainly, effective anomaly detection can be achieved by applying artificial intelligence methods such as deep learning and machine learning [20, 32]. Different machine learning algorithms have been applied with various performance metrics, including Support Vector Machines (SVMs), decision trees, closest neighbour techniques and random forests for the development of an effective IDS.

1.2. Deep Learning

Deep learning, a subset of machine learning, involves neural networks with multiple layers that can automatically learn hierarchical representations from large amounts of data [38]. It is applied in the development of IDS because of its ability to handle complex and high-dimensional data, making it highly effective in detecting patterns of cyber threats. Unlike traditional methods, deep learning can adapt and improve over time, offering robust detection capabilities even in dynamic network environments. Moreover, deep learning models are capable of identifying both known and unknown attacks with higher accuracy, improving the overall security of networks. Long-Short Term Memory (LSTM), Convolutional Neural Networks (CNNs), Auto Encoders (AE), Artificial Neural Networks (ANN), and Recurrent Neural Networks (RNN) [37] are some recent deep learning techniques that are utilized to build IDS with varied degrees of performance.

1.2.1. Artificial Neural Networks (ANN)

An ANN is a computational model inspired by the human brain, consisting of interconnected neurons that learn from data through adaptive weights. It is utilized in this research to develop an IDS that automatically learn complex patterns and effectively detect anomalies or malicious behaviour in network traffic. By capturing non-linear relationships in high-dimensional data, the ANN improves classification accuracy, distinguishing between normal and attack traffic. This research leverages the ANN's robustness and flexibility to enhance the detection performance and scalability of the IDS.

1.3. Optimization Algorithms

Optimization algorithms refer to mathematical methods or algorithms used to find the best possible solution to a problem, often by minimizing or maximizing an objective function. In the context of IDS, these techniques are crucial for enhancing the system's performance by selecting the most relevant features and reducing dimensionality, which improves detection accuracy and reduces False Positives (FP). By optimizing FS, IDS models can process large datasets more efficiently, leading to faster detection of anomalies or attacks. Additionally, optimization helps balance between detection speed and computational resource usage, making IDS more scalable and adaptable to real-world environments. Overall, applying optimization techniques ensures that IDS systems are more effective, accurate, and resource-efficient [25, 36].

With better efficiency, many optimization algorithms are coupled with various artificial intelligence methodologies. Additionally, utilizing a merged or hybrid approach to solve computational issues outperforms standard techniques [4, 14].

1.3.1. Honey Badger Optimization (HBO) Algorithm

This research study utilizes a new HBO algorithm for FS in CIC-IDS2017. Major reason behind the selection of the HBO [17] is that,

1. It optimizes both discovery and extraction.
2. Having a great convergence speed.
3. There hasn't been much or any study on using it as an IDS [13].

1.4. CIC-IDS2017 Dataset

The CIC-IDS2017 dataset is used in this study to train and evaluate the model [29]. This dataset is an openly accessible dataset frequently utilized "in network IDS research" [29]. It was developed by "the University of New Brunswick's Canadian Institute for Cybersecurity (CIC)" [29]. The details of the "CIC-IDS2017 dataset" [29] are:

1. Collection method: the dataset was created by using the CIC's hybrid traffic generator tool to capture unprocessed network traffic in a controlled network environment.
2. Traffic types: the dataset contains a range of traffic types, including regular traffic, DoS assaults, port scans, DDoS attacks, botnet traffic, and web attacks.
3. Features: the dataset offers an extensive collection of features that include statistical flow-based features, payload-based features, and packet headers.
4. Labels: each network flow instance in the dataset is labelled as either benign or malicious.
5. File formats: the dataset is provided in two formats: Comma-Separated Values (CSV) and Packet Capture (PCap). In this research, CSV file is used.
6. Data set Size: It contains a total of approximately 2.5 million network flow instances.
7. Availability: "The CIC-IDS2017 dataset is available for download from the website (https://www.unb.ca/cic/datasets/ids-2017.html)" [29]. Figure 2 shows the various labels and their occurrences in the dataset.

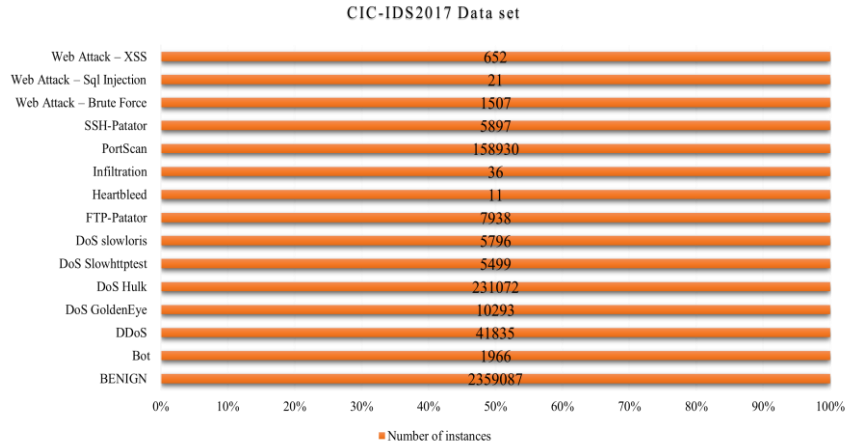


Figure 2. CIC-IDS2017 dataset description.

1.5. Novelty and Contribution

In summary, this research suggests an HBO algorithm to select essential features and deep learning technique namely ANN for classification in the development of effective IDS. The main contribution and novelty of this work is summarized below.

1. This research pioneers the use of the HBO algorithm for FS in IDS. HBO's adaptive search strategy, inspired by the foraging behaviour of honey badgers, is uniquely applied to select the most relevant features in CIC-IDS2017 dataset, which enhances detection performance.
2. This study presents a novel hybrid approach by integrating HBO for FS with ANN for classification. This combination leverages the strengths of both methods, where HBO reduces feature dimensionality and noise, improving ANN's ability to classify network intrusions with higher accuracy and efficiency.
3. The proposed Honey Badger Optimization-Artificial Neural Network (HBO-ANN) hybrid model significantly improves the classification accuracy, reduces the Mean Squared Error (MSE), and enhances the computational efficiency of the IDS. This work demonstrates how HBO-selected features lead to faster training times and better generalization on unseen attack data compared to classification without FS techniques.
4. Evaluating the performance of the proposed IDS by comparing it with GDMO, IFSO, GSO, SSO, and GOA in terms of MSE,accuracy, recall, precision, F-measure, the number of features, and execution time.

2. Related Studies

Ali *et al.* [3] introduced an innovative IDS named Particle Swarm Optimization-Fast-Learning Network (PSO-FLN), which integrates Particle Swarm Optimization (PSO) with a Fast-Learning Network (FLN) [3]. By utilizing an ANN and learning from previous attack examples, PSO-FLN demonstrates superior performance in detecting intrusions with higher testing accuracy compared to other meta-heuristic algorithms on the KDD99 dataset. Otair *et al.* [28] developed an IDS that leverages Grey Wolf Optimization (GWO) technique for feature reduction, enabling the identification of relevant attributes for effective detection of system attacks. Further, GWO is hybridized with PSO, which preserves each grey wolf's individual best position information for enhancing the GWO algorithm's performance and avoid local optima. Pingale and Sutar [30] suggested an IDS namely RWO-based combined deep model, which employs efficient CNN features along with normalization to detect network intrusions. By utilizing the hybrid optimization algorithm, RWO, "a combination of Whale Optimization Algorithm (WOA) and Remora Optimization Algorithm (ROA)" [30], the proposed

methodology achieves better performance with high testing precision, accuracy, recall, and F1-score values [30]. With the use of a Deep Neural Network (DNN) and Pretraining with a Deep Auto Encoder (PTDAE), a deep learning IDS was presented [23]. By employing automatic hyperparameter optimization, combining random and grid search techniques, the proposed model performs better in recognizing attacks compared to earlier approaches in multiclass classification, evaluated on the Communications Security Establishment and the Canadian Institute for Cybersecurity (CSE-CIC-ID2018) and, Neural Simulation Language-Knowledge Discovery in Databases (NSL-KDD). Sood *et al*. [34] suggested a new system for detecting network traffic anomalies in 5G networks [25], utilizing a binary-stage design that involves dimensionality reduction and a DNN technique [25].

With the UNSW-NB15 dataset along with the Telecommunications Standards Institute-Network Functions Virtualisation (ETSI-NFV) European architecture, the recommended methodology produced a detection accuracy of 98% at a dimensionality reduction factor of 81%, outperforming other recent methods and showcasing the architecture's overall merit. Li *et al*. [24] recommended an IDS model in the context of medical IoT systems utilizing the butterfly optimization algorithm to enhance the accuracy of NIDS. By selecting discriminative features for an ANN, the proposed method achieves an impressive 93.27% accuracy, surpassing the performance of other previously employed techniques like SVM, decision tree and ant colony optimization for the same purpose. Alwesha *et al*. [8] introduced an inventive wrapper FS model, incorporating the emperor penguin colony optimization algorithm, to advance intrusion detection capabilities within IoT environments. When tested on multiple IoT datasets, the suggested model outperforms existing techniques, including Multi-Objective Particle Swarm Optimization (MOPSO) and MOPSO-Lévy, with respect of accuracy and FS size, with an outstanding classification accuracy of 98%" [8]. Al and Dener [2] integrated Convolutional Neural Network (CNN) and LSTM to create a novel Hybrid Deep Learning (HDL) network for intrusion detection on large data sets [8]. The suggested system, called SMOTE+Tomek-Link-Hybrid Deep Learning (STL-HDL), produced 99.83% and 99.17% "for multi class and binary classification respectively" [2] outperforming current methods in the identification of network intrusion in unbalanced datasets. Alqahtani [7] created an innovative hybrid optimized LSTM method for identifying network intrusions in IoT networks [7]. Moreover, it combines a CNN to extract features and an optimized LSTM for predicting different attacks, outperforming other benchmark models with improved prediction performance and lower computational complexity.

A two-stage network intrusion detection model utilizing optimized deep learning model that employs the elastic net contractive auto encoder and generalized mean grey wolf algorithm for dimensionality reduction has been suggested [26]. The suggested model achieves high classification accuracy, outperforming benchmark IDSs and demonstrating effectiveness in learning from unlabelled data and generalizing to arbitrary test data. Ponmalar and Dhanakoti [31] designed a novel intrusion detection technique that integrates chaos game optimization algorithm with ensemble SVM to handle big data complexities in network traffic analysis. The proposed methodology achieves a higher classification accuracy of 96.29% compared to baseline models, demonstrating its effectiveness in reducing FPs on handling security incidents on large-scale data platforms. Nasir *et al*. [27] developed DF-IDS, by employing a feature engineering along with deep learning technique to protect edge IoT devices. DF-IDS demonstrates superior performance with an F1-score of 99.27% and an accuracy of 99.23% [27] surpassing previous research and comparative models in intrusion detection for edge IoT. Hajimirzaei and Navimipour [16] introduced a novel approach for detecting malicious network traffic using Artificial Neural Networks (ANNs), achieving 99% accuracy and an AUC-ROC of 0.99 through comprehensive testing with diverse data types. In addition, a hybrid IDS that combines a Multi-Layer Perceptron (MLP) with Artificial Bee Colony (ABC) optimization and fuzzy clustering has been proposed, which outperforms state-of-the-art methods when tested on the NSL-KDD dataset

In summary, several innovative approaches had been proposed for the development of NIDS to address the challenges of securing various environments, including medical IoT systems, smart networks, and edge IoT. These research works emphasize the significance of FS to enhance the learning process in ANNs, with methods like HDL networks, LSTM, and CNN integrated for improved performance. Additionally, the use of bio-inspired optimization algorithms demonstrates their potential to enhance ensemble SVM and deep learning models in big data platforms. Overall, "combining bio-inspired optimization techniques and deep learning methods to tackle the complexities of modern network security challenges", makes substantial strides in enhancing the integrity and security of critical systems [9]. Alohali *et al*. [5] proposed an AI-enabled Multi-Modal Fusion-based Intrusion Detection System (AIMMF-IDS) for Cognitive Cyber-Physical Systems (CCPS) in Industry 4.0, using an Improved Fish Swarm Optimization (IFSO) technique for FS and a weighted voting ensemble model combining RNN, Bi-LSTM, and DBN for enhanced detection. The results demonstrate superior performance compared to recent state-of-the-art techniques in terms of various metrics. Guezzaz *et al*. [15] proposed a hybrid IDS framework combining K-Nearest Neighbor (K-NN) and Principal

Component Analysis (PCA) for improved accuracy, achieving 99.10% accuracy and 98.4% detection rate on the NSL-KDD dataset and 98.2% accuracy on the Bot-IoT dataset. Tabash *et al*. [35] introduced a smart hybrid IDS model combining Genetic Algorithm (GA), Proportional K-Interval Discretization (PKID), Fisher Linear Discriminant Analysis (FLDA), and deep learning techniques to enhance feature selection and classification accuracy. The proposed model demonstrates superior performance with 99.93% classification accuracy, 99.97% detection rate, and a false alarm rate of 0.00093 on the NSL-KDD dataset.

From the related studies, the major challenges, further exploration and improvement to be considered for devising an effective IDS are,

1. Generalization to diverse environments.
2. Model's adaptability to dynamic changes in network behaviour.
3. Robustness against adversarial attacks.
4. Improved detection accuracy.
5. Real-World applicability.
6. Scalability and efficiency.
7. Privacy-preserving features.

3. Proposed Method

The major aim of an IDS is to identify and respond to unauthorized [22] or potentially malicious activities within computer networks, thereby enhancing network security. The principal aim of this investigation is to create a highly effective and robust IDS by leveraging the synergistic potential of an ANN [22] and the innovative HBO algorithm. The suggested HBO-ANN model has four modules.

1. Preparing the data set.
2. FS using HBO algorithm.
3. Classification using ANN.
4. Evaluate the effectiveness of the classifier utilizing various performance indicators.

Figure 3 is the diagrammatic illustration of the suggested method.

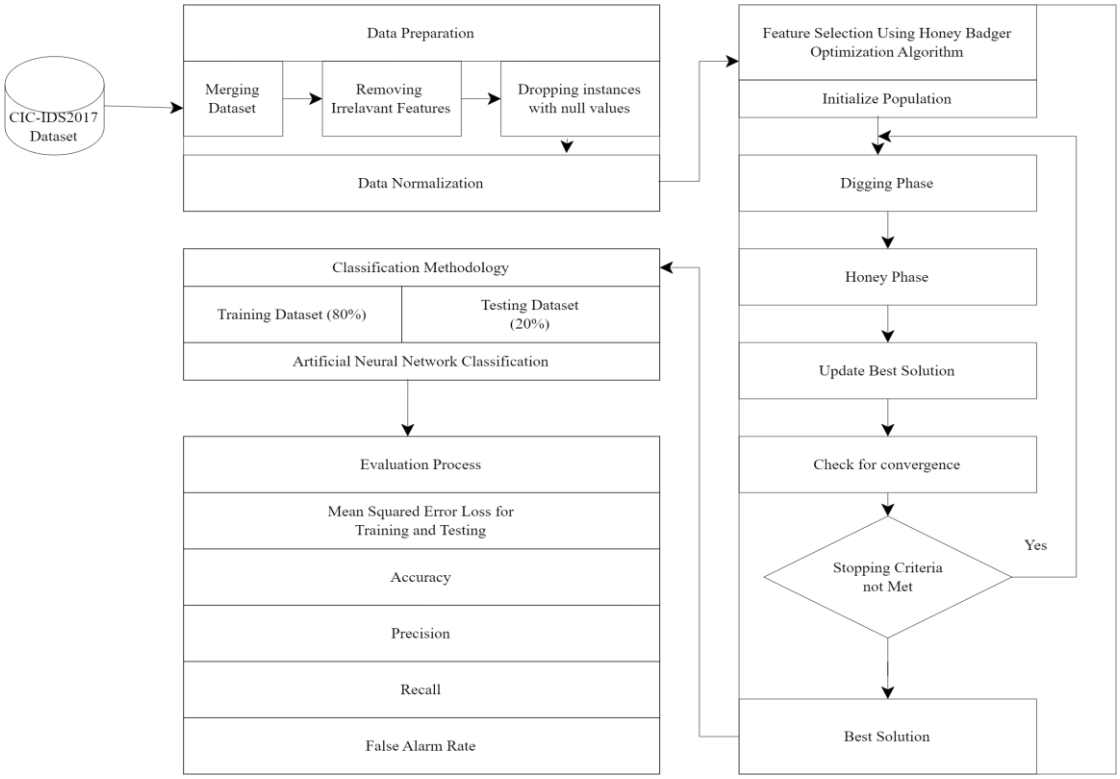


Figure 3. Proposed Model.

3.1. Data Preparation

Preparation of data is a crucial part in data analysis and machine learning. It ensures data quality, enhances model performance, improves interpretability, and enables more accurate and reliable insights and predictions. Investing time and effort in data pre-processing leads to more robust and effective data analysis outcomes. The data preparation steps that are applied in this research are,

1. Merging dataset.
2. Removing irrelevant features.
3. Normalization.

3.1.1. Merging Dataset

The CIC-IDS2017 has totally eight. CSV files [20]. Since all the files have same features, they have been combined using .concat function of pandas data frame to create single dataset for further processing. After concatenating all the files, the data frame contains 2830743 instances of data and 79 features in total as depicted in Table 1.

Table 1. Pre-processing of dataset.

Number of instances before pre-processing	Features count before pre-processing	Pre-processing method	Number of instances after pre-processing	Features count after pre-processing
692703				
445909				
288602				
170366				
529918				
22574				
286467				
191033	79	Merge dataset	2830743	79
2830743	79	Removing irrelevant features	2830743	71
2830743	71	Dropping instances with null values	2829385	71
2829385	71	avoiding class imbalance problem	2827425	71

3.1.2. Removing Irrelevant Features

The features that have completely homogenous values are ineffective in for a machine to learn. So, they can be removed from the dataset to make the model more efficient. In this research, the features that are having homogenous features and removed from the data frame for further processing are shown in Figure 4. Undoubtfully, eight features are identified as completely homogenous and are removed. After removing the ineffective feature there are 2830743 instances and 71 features in the dataset as displayed in Table 1.

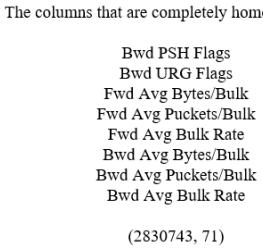


Figure 4. Removed irrelevant features.

3.1.3. Dropping Instances with Null Values

Dropping instances with null values is a common data pre-processing step. The null values in a dataset represent missing or unknown values. They may lead to biased or incorrect results if not handled properly. There are many advantages of dropping Null value instances in a dataset. Firstly, one can ensure that the model is based on complete and reliable data. Secondly, the elimination of unreliable and incomplete datapoints improves the overall quality of the dataset. Finally, most machine learning algorithms cannot handle null values and hence it is important to remove null values "to make the dataset suitable for training and testing the model" [1]. In this research, the dataset contains 1358 instances with Null values and are dropped. After dropping the Null values there are 2829385 records and 71 features as seen in Table 1.

3.1.4. Avoiding Class Imbalance Problem

A typical situation in a classification problem, where classes are not represented equally in a dataset is known as class imbalance problem. It occurs when one class has significantly larger or smaller number of instances comparing to other classes. Due to class imbalance problem, the model may be biased towards the majority class and have difficulty in learning minority class. This research deletes the class instances that have less than 40 samples to handle class imbalance problem. Also, "it deletes missing values to tackle class imbalance problem. The count of samples removed to tackle the class imbalance is seen in Table 1. Evidently, 1960 samples are deleted to tackle missing values and class imbalance problem.

3.1.5. Normalization

Normalization is a pre-processing technique, often referred to as data scaling or feature scaling which is commonly used in machine learning to transform the features (input variables) of a dataset to a similar scale. The purpose of normalization is to bring the features onto a comparable level and prevent any particular feature from dominating or biasing the learning algorithm due to differences in their scales or units. This research uses min-max normalization to scale the features uniformly. Min-Max scaling transforms the feature values to specific values [9], typically between zero and one. It is achieved by the Equation (1).

$$x_{scaled} = \frac{(x - x_{min})}{x_{max} - x_{min}} \quad (1)$$

3.2. Optimization Algorithms for Feature Selection

An optimization algorithm is a computational method or procedure used to determine the most suitable solution to an optimization problem [3]. Firstly, the optimizer has a set of decision variables, which are the parameters that can be adjusted to determine the best possible solution. Secondly, the optimizer has bounds on decision variables, which are the limits that define the feasible region of the search space. Thirdly, constraints, which are additional conditions or requirements that must be satisfied by a solution. Finally, an objective function, which is a mathematical expression that evaluates the quality of a solution based on the decision variables [3]. Deep learning algorithms are enhanced by a methodology named FS that decreases the total count of input features of a classifier. Optimization algorithms can help select a subset of relevant features from a large pool [26], minimizing the input space's dimensionality. By focusing on the most informative features, the model can achieve better generalization and predictive performance. Removing irrelevant or redundant features can prevent overfitting and meliorate the model's ability to extract meaningful patterns from the

data. FS with optimization algorithms can substantially decrease the number of input features, leading to faster training and inference times. With the black-box nature of deep learning, FS can help in identifying the most important features, providing insights into which aspects of the data are crucial for the model's decision-making process. In this research, "HBO algorithm is employed" [17] for FS.

3.2.1. HBO Algorithm

The HBO algorithm was developed by Hashim *et al*. [17] and was motivated by the foraging practices of the honey badger. The honey badger identifies and locates the food by two ways. In the first mode also called digging, it uses its smelling ability to approximately find the prey. After that, it wanders around the prey to find the best spot for digging and capturing it. In the second mode also known as honey phase, honeyguide bird directs the honey badger to find beehives quickly [17]. The following Equations (2) to (11) summarize the mathematical foundation of the HBO algorithm [17]. HBO can be referred to as a global optimization method, as it integrates exploration and exploitation stages. The population of possible solutions in HBO is [17] represented by the following Equations (2) and (3).

$$Population\ of\ candidate\ solution = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1D} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & x_{n3} & \dots & x_{nD} \end{bmatrix} \quad (2)$$

Honey badger at position i is given as

$$x_i = [x_i^1, \quad x_i^2, \quad \dots, \quad x_i^D] \quad (3)$$

The sequence [7] of the HBA algorithm is given below.

- *Stage* 1. Initialization

Set the initial values for the population size (N) and the position of the honey badgers based on Equation (2).

$$x_i = lb_i + r_1(ub_i + lb_i), r_1 \quad (4)$$

is a random number ranging from 0 to 1.

Where x_i is the honey badger at location i representing a possible solution having N population, and ub_i and lb_i are the upper and lower boundaries of the exploration domain" [17].

- *Stage* 2. Describing intensity (I)

The distance between the combined strength of the prey and the i^{th} honey badger is known as intensity. The smell intensity I_i of the prey" [17] is given by the Equation (5).

$$I_i = r_2 \times \frac{S}{4\pi d_i^2}, r_2 \quad (5)$$

is a random number ranging from 0 to 1 [17].

$$S = (x_i - x_{i+1})^2 \quad (6)$$

$$d_i = x_{prey} - x_i \quad (7)$$

Where S is the concentration strength of the prey while d_i denotes the distance between the prey and the i^{th} honey badger" [17].

- *Stage* 3. Adjusting density factor

"Time varying randomization is regulated by the density factor" [17] α which facilitates seamless move from exploration to exploitation. To reduce randomization with time, the density factor α is reduced in each iteration as described by the Equation (8).

$$\propto = C \times exp\left(\frac{-t}{t_{max}}\right) \quad (8)$$

Where t_{max} "is the maximum number of iterations and C≥1 is a constant" [17].

- *Stage* 4. Avoiding local optimum

Stages 4 and 5 describe the avoiding local optima trap. This technique gives agents the "best opportunity to thoroughly inspect the search field by using the flag F" [17] for altering the search direction.

- *Stage* 5. Agent position revision

As mentioned previously, the "digging phase" and the "honey phase" are the two separate stages of the process for updating the HBA position (x_{new}) [17]. The following is an explanation of it.

a) Digging phase. The actions performed by a honey badger [17] is modelled after Cardioid shape. The simulation of Cardioid motion is given by the Equation (9).

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_3 \times \alpha \times d_i \times [cos(2\pi r_4) \times [1 - cos(2\pi r_5)] \quad (9)$$

where, x_{prey} is the current position of the prey which is considered as global best position found so far. $\beta \geq 1$ is the capability of the honey badger to acquire food whose default value is 6. d_i is the distance between i^{th} honey badger and the prey. r_3, r_4 and r_5 are various random number ranging from 0 to 1" [17]. The search direction is altered by the flag F and is calculated as follows.

$$F = \begin{cases} 1 & if\ r_6 \leq 0.5 \\ -1 & otherwise \end{cases} \quad (10)$$

Where, r_6 denotes the arbitrary number ranging from 0 to 1.

A honey badger largely depends on the prey's smell intensity I, distance from prey and the badger d_i, and time-sensitive search influence factor α, during the digging phase. Furthermore, a badger may feel any disturbance F when digging, which helps it find its prey even more efficiently [17].

b) Honey phase. As discussed earlier, the honey badger identifies the beehive under the direction of honey bird [17]. This can be simulated with the following equation.

$$x_{new} = x_{prey} + F \times r_7 \times \alpha \times d_i \quad (11)$$

Where r_7 is an arbitrary number ranging from 0 to 1, x_{new} is the latest honey badger position, x_{prey} is the position of the prey, α and F are calculated using Equations (4)

and (8), respectively and d_i is the distance between the badger and the prey [17].

The sequence of HBO algorithm is given by Figure 5.

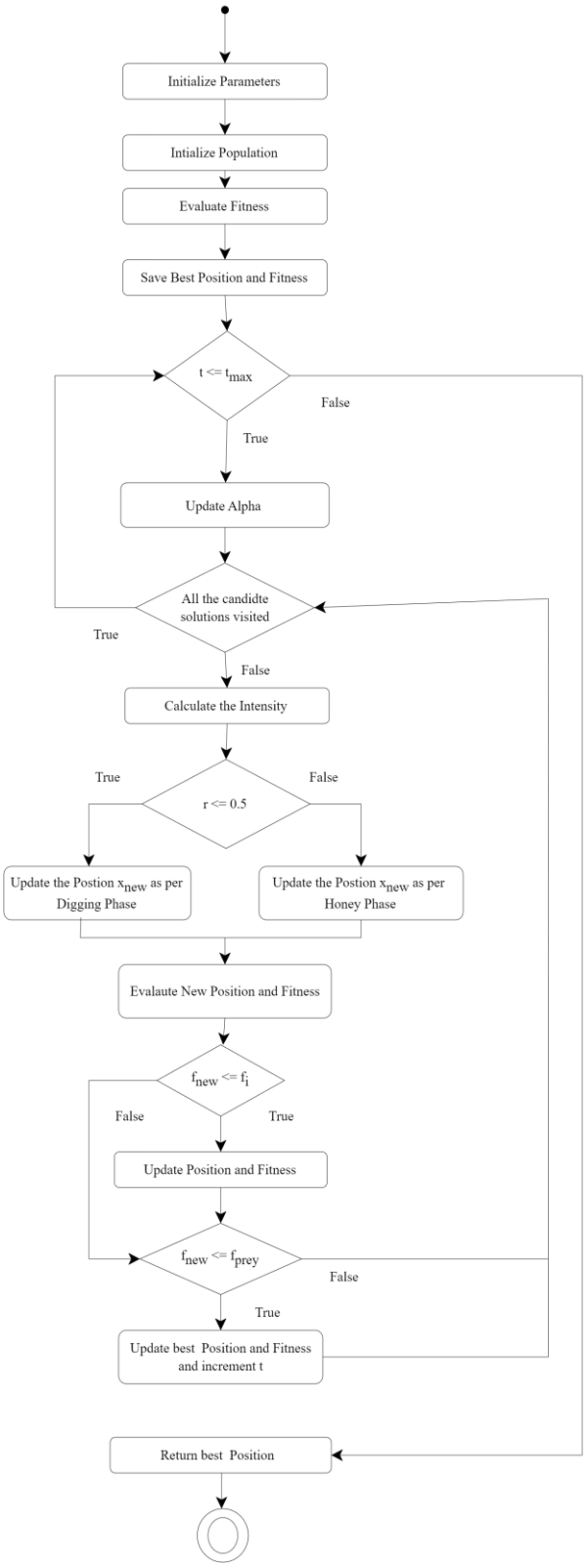


Figure 5. HBO sequence.

3.2.2. Working Principle of HBO

The HBO algorithm works by mimicking the foraging behaviour of the honey badger, specifically its hunting strategies and adaptability. The algorithm is divided into two main phases:

- Digging phase (exploration): in the digging phase, the honey badger uses its strong sense of smell to locate prey. Mathematically, this phase corresponds to the exploration of the solution space, where the honey badger moves randomly but strategically towards the global best position, or prey. During this phase, the search space is explored broadly to avoid getting trapped in local optima. The movement of the honey badger in this phase is influenced by the intensity of the smell (prey proximity), distance from the prey, and randomization factors, ensuring efficient exploration.
- Honey phase (exploitation): in the honey phase, the honey badger follows a guide (the honeyguide bird) directly to a food source (such as a beehive). In this phase, the algorithm focuses on exploitation, which means narrowing the search space to hone in on the best solution found so far. The movement of the honey badger becomes more concentrated near the identified solution, refining the search in the most promising areas of the solution space.

The combination of exploration (digging) and exploitation (honey phase) helps the HBO algorithm effectively solve complex optimization problems with multiple local minima, maintaining a balance between global exploration and local exploitation. This approach ensures that the algorithm performs well in both locating promising regions of the solution space and fine-tuning solutions to achieve optimal performance.

3.2.3. Suitability of HBO for Feature Selection in IDS

The dual behaviour of honey badgers allows HBO to efficiently explore the solution space and exploit the best regions, making it highly effective for selecting relevant features in high-dimensional datasets. HBO's adaptive mechanisms ensure quick convergence, which is crucial for FS tasks where time complexity and computational efficiency are important. This makes it particularly useful for large intrusion detection datasets like CIC-IDS2017. The balance between exploration and exploitation ensures that HBO avoids being trapped in local optima, a common problem in FS algorithms. This allows HBO to identify a globally optimal subset of features, leading to improved classification performance when used in conjunction with deep learning models like ANN.

3.2.4. Feature Selection in CIC-IDS2017 Using HBO Algorithm

The dataset with 2827425 records and 71 features is fed as an input to the HBO stage to perform FS. The HBO algorithm outputs 30 most relevant features for creation and testing of classification model. The selected features

by HB0 are shown in Table 2.

Table 2. Features selected by HBO Algorithm.

Feature number	Feature name	Feature number	Feature name
78	Idle min	55	Avg Bwd segment size
75	Idle mean	18	Flow IAT Std
77	Idle max	40	Max packet length
24	Fwd IAT max	48	ACK flag count
19	Flow IAT max	41	Packet length mean
23	Fwd IAT Std	53	Average packet size
44	FIN flag count	29	Bwd IAT max
42	Packet length Std	28	Bwd IAT Std
47	PSH flag count	31	FWD PSH flags
14	Bwd packet length Std	45	SYN flag count
21	FWD IAT total	49	URG flag count
11	Bwd packet length max	76	Idle Std
2	Flow duration	20	Flow IAT min
43	Packet length variance	22	Fwd IAT mean
13	Bwd packet length mean	1	Destination port

3.2.5. Comparison with Benchmark Algorithms

Figure 6 compares the number of features selected by various FS algorithms for IDS. The Quantum Dwarf Mongoose Optimization (QDMO-FS) [4] method selects 45 features, while Improved Fish Swarm Optimization (IFSO-FS) [9] selects 47, showing a slight increase. Group Search Optimization-FS (GSO-FS) [16] chooses 56 features, which is higher than both QDMO and IFSO. Grasshopper Optimization Algorithm-FS (GOA-FS) [25] and Social Spider Optimization-FS (SSO-FS) [25] select 61 and 64 features respectively, indicating a larger feature set compared to the previous methods. In contrast, the proposed HBO-FS method significantly reduces the number of selected features to just 30. This demonstrates that the HBO-FS approach is more efficient in reducing the feature set, potentially leading to improved model performance and lower computational cost.

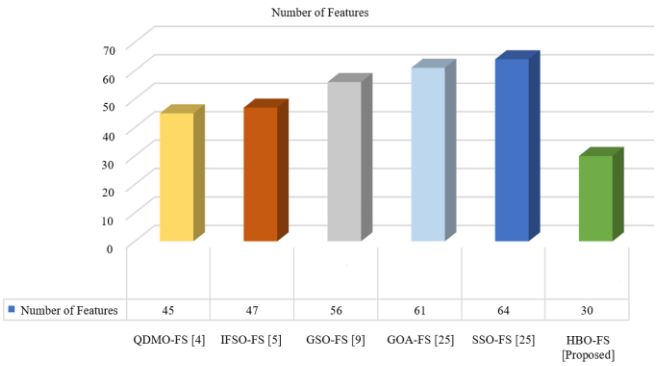


Figure 6. Features selected by various algorithms.

3.3. Classification Using Artificial Neural Network

ANN is a computational framework that draws inspiration from the structure and operations of the human brain [6]. An ANN comprises interconnected nodes, often referred to as neurons or units, arranged in layers. The three primary types of layers in an ANN [6] are:

1. Input Layer: the first layer of the network acts as the recipient for input data, where each node in this layer corresponds to a distinct feature or attribute present in the input data [6].
2. Hidden Layers: between the input and output layers, there are hidden layers [6], play a pivotal role in processing information. These concealed layers enable the network to acquire an understanding of intricate patterns and representations embedded within the input data.
3. Output Layer: the ultimate layer of the network generates the model's predictions or outputs by leveraging the acquired representations from the hidden layers. The quantity of nodes within the output layer in varies according to the specific task being addressed [6]. The weighted sum of inputs, often referred to as the weighted sum or linear combination, is a basic operation ANNs. It is the first step in calculating the output of a neuron in the network. The weighted sum can be computed with the Equation (12).

$$z = x.w + b \tag{12}$$

Where,

- x is the input vector which contains the values of input features.
- w is the weight vector which contains the weights with respect to each input features.
- b is the bias term, which is an additional learnable parameter added to the weighted sum.

The weighted sum z represents the net input to the neuron, which is then passed through the activation function to add non-linearity and generate the final output of the neuron [6].

3.4.1. Activation Function

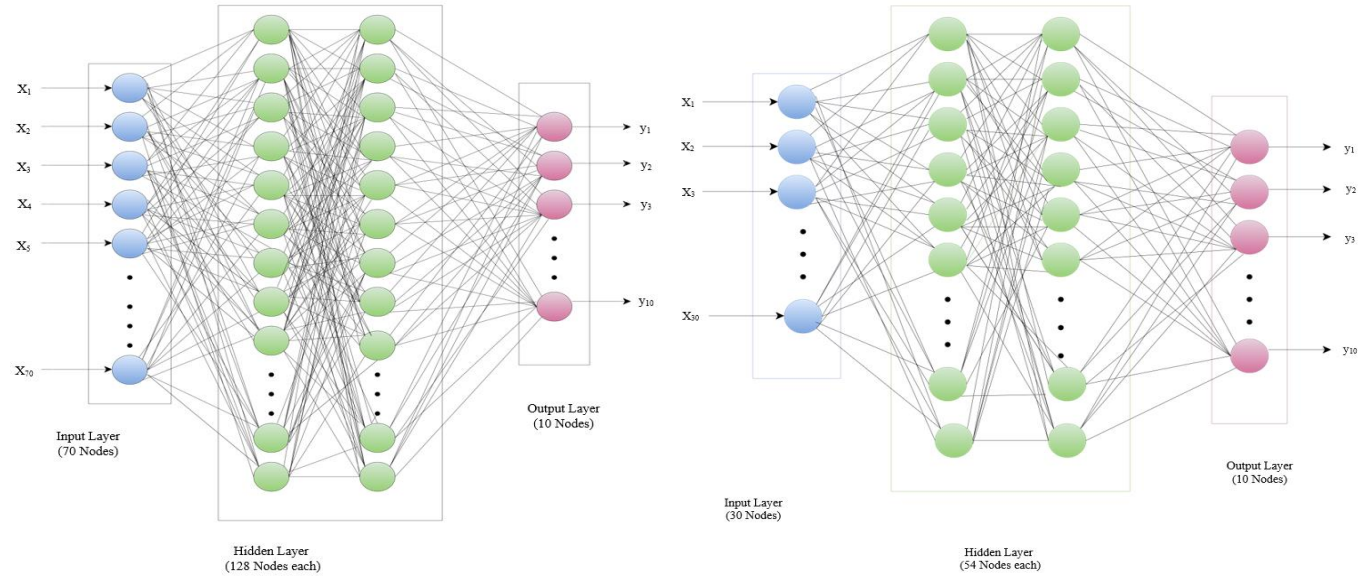
The activation function, represented as activation (z), is employed on the weighted sum (z) to infuse non-linearity into the model. The presence of an activation function is crucial, as it prevents the neuron's output from being a linear function of the inputs. This non-linearity is essential for the model to effectively grasp intricate patterns within the data. In this paper Rectified Linear Unit (ReLU) activation function has been applied in hidden layer given by the Equation (13). Softmax activation function is applied to the output layer as described [28] by Equation (14).

$$ReLU(z) = max(0, z) \tag{13}$$

$$Softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{14}$$

where, i is the input (logit) to the i^{th} node in the output layer, and k is the total number of classes in the multi-class classification problem [6]. The hidden layers learn feature representations, while the *Softmax* activation produces the probabilities for each class, which makes predictions for multiple classes simultaneously. The

schematic diagram of ANN without using any FS is shown in Figure 7-a). It has a 70 nodes input layer, 128 nodes of two hidden layers and 10 nodes output layer describing multiclass classification. Figure 7-b) displays the schematic diagram of ANN with FS using honey badger algorithm. It has 30 nodes input layer, two hidden layers of 54 nodes each and 10 nodes output layer describing multiclass classification.



a) ANN without HBO-FS. b) ANN with HBO-FS.

Figure 7. Schematic diagram of ANN.

4. Experimental Results and Discussion

This section will detail the experimental findings of the suggested approach. The investigation was conducted using a Dell Latitude laptop equipped with an Intel Core processor i7-1265U CPU, 16 GB of RAM, and a 64-bit version of Windows 11 Pro [11]. Python3.7 was utilized as a programming language along with the Anaconda IDE development libraries for the deep learning programs Keras3 and TensorFlow4. Microsoft Excel 2021 has been used to make comparison charts. Fotor Pro has been utilized to improve the image quality from the programming tools. To assess the performance of suggested system, we conduct experiments in a variety of scenarios and compared it to a number of comparable systems, including Fuzzy Clustering-Artificial Neural Network (FC-ANN), CNN, and Long-Short-Term-Memory-Auto Encoders (LSTM-AE) and DNN.

4.1. Testing and Analysis

A comprehensive evaluation and comparison of the suggested HBO-ANN and ANN without FS is given in this section. The Mean Square Error (MSE) is the metric used to assess the loss of the proposed model [10] with training and testing dataset. It is calculated with the Equation (15).

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2 \quad (15)$$

The accuracy, precison, recall and False Alram Rate (FAR) are taken as the performance measures to assess the performance of the suggested model [19]. The confusion matrix to compute the performance is depicted in Table 3.

Table 3. Confusion matrix.

Actual class	Anticipated class	
	Attack	**Normal**
Attack	True Negative (TN)	False Positive (FP)
Normal	False Negative (FN)	True Positive (TP)

4.2. Results

This part presents the empirical findings derived from the experimentation conducted on the proposed model.

4.2.1. Selection of Important Features and Processing Time

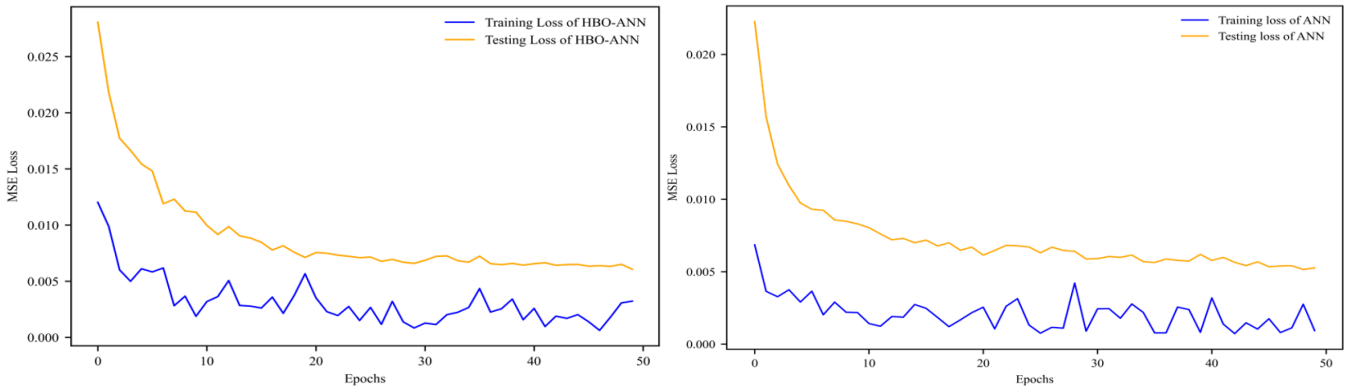
Table 4 shows the various feature numbers selected by the HBO algorithm and the features selected by pre-processing without HBO optimization, total number of features selected, percentage of selection and the training time of ANN. The HBO FS algorithm selects 30 features and is 38.46% of CIC-IDS2017 features. Further, it takes 4870.2932 seconds to train the ANN with 80% of dataset. By removing the redundant features, there are 70 features to be fed as input to the ANN without HBO optimization and is 91.02% of CIC-IDS2017 features. In addition, it takes 8735.2872 seconds to train the ANN with 80% of dataset. Clearly, the training time of ANN with HBO optimized features is nearly half of that of training ANN without HBO optimization. The chart in Figure 7 shows "the comparison analysis of the total number of selected features of HBO algorithm and the total number of selected features" [23] of the algorithms in the literature.

Table 4. Features selected by pre-processing with and without HBO algorithm.

Method	Selected feature numbers	Total	Percentage of selection	Training time
Pre-processing with HBO	78, 75, 77, 24, 19, 23, 44, 42, 47, 14, 21, 11, 2, 43, 13, 55, 18, 40, 48, 41, 53, 29, 28, 31, 45, 49, 76, 20,22, 1	30	38.46 %	4870.2932 seconds
Pre-processing without HBO	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 33, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78	70	91.02%	8735.2872 seconds

4.2.2. Training and Testing Loss

In IDS classification, MSE plays a critical role in measuring and minimizing classification errors. It ensures that the model accurately differentiates between normal and malicious traffic, aids in the model's optimization, and contributes to effective generalization. Low MSE values in IDS suggest a well-performing model that is both accurate and reliable in detecting intrusions with minimal errors. In the development of IDS, MSE represents how well the model is learning from the data.



a) MSE loss of HBO-ANN algorithm. b) MSE loss of ANN algorithm.

Figure 8. Comparison of MSE Loss Curves for HBO-ANN and ANN.

This subsection gives the details about the training and testing of the ANN model with and without HBO FS. The model has been trained up to 50 epochs. Figure 8-a) describes MSE loss for HBO-ANN algorithm. The training loss was around 0.012 and reduced 0.005 in about 4 epochs and has been consistently less than 0.005 up to 50 epochs. The testing loss was around 0.028 initially and is reduced to less than 0.010 in about 10 epochs. The testing loss was consistently less than 0.010 until 50 epochs. Since the training and testing losses are low and invariable after 10 epochs the training of the model has been stopped at 50 epochs.

Figure 8-b) shows MSE loss for ANN model without HB-FS. The training loss was about 0.007 and reduced to around 0.003 in 10 epochs. The trend of being less than 0.004 is maintained until 50 epochs. The testing loss was around 0.023 initially and is decreased to less than 0.007 at 10 epochs and is maintained until 50 epochs.

The average squared difference between the predicted and actual values during model training and testing are 0.00317 and 0.009, respectively. Low MSE values signify that the model is making very few errors in its predictions, meaning the predicted values are very close to the actual values.

Figure 9 shows a comparison of MSE with various benchmark models during training and testing, including the proposed HBO-ANN model. The suggested method, HBO-ANN, has the lowest MSE values for both training and testing, compared to other models like FC-ANN [18], Neural Network Intrusion Detection (NNID) [18], Selection of Relevant Features (SFR) [27], and Multi-Layer Perceptron-Artificial Bee Colony (MLP-ABC) [27]. This means that HBO-ANN model outperforms the others in minimizing prediction errors. The lower MSE indicates that HBO-ANN provides more accurate predictions with fewer errors.

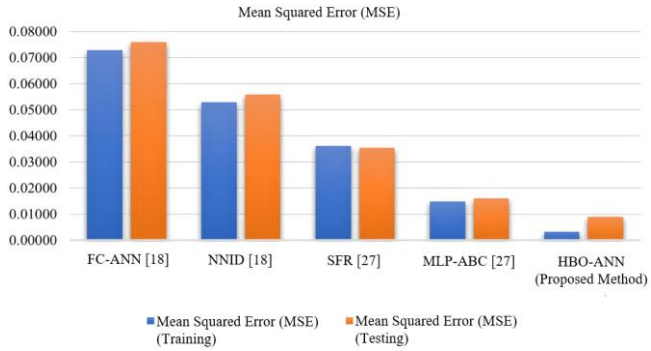


Figure 9. MSE comparison with benchmark algorithm.

The small gap between training and testing MSE in HBO-ANN suggests that the model generalizes well, meaning it performs consistently on both the training data (used for learning) and testing data (unseen data). Lower MSE during testing indicates that HBO-ANN can detect intrusions with higher precision and fewer false alarms, which is crucial for an IDS. HBO-ANN achieves near-zero MSE (close to 0.01), whereas the

other models range between 0.01 and 0.07. This is a clear advantage in terms of error reduction.

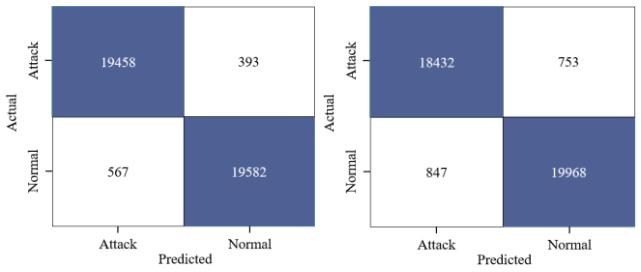
In Summary, lower MSE in both training and testing phases implies that the suggested model generalizes well to new, unseen data. This is crucial for an IDS, where the model must perform well on real-world network traffic. The gap between training and testing MSE is also minimal, demonstrating that the HBO-ANN model avoids overfitting and has strong generalization capabilities. This is a significant achievement, as it means the model is learning the underlying patterns of attacks and benign behaviour accurately. A low MSE indicates not only a high classification performance but also a minimization of errors in attack detection, which enhances the overall effectiveness of an IDS.

Initially, the model has been trained and tested for 30 epochs. Later the model was retrained for 50 epochs to check for any variance. However, the two models do not differ significantly. For the purpose of assessing various performance metrics, the model has therefore been selected and set.

4.2.3. Accuracy

The accuracy is calculated with the Equation (16) [12].

$$Accuracy = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (16)$$

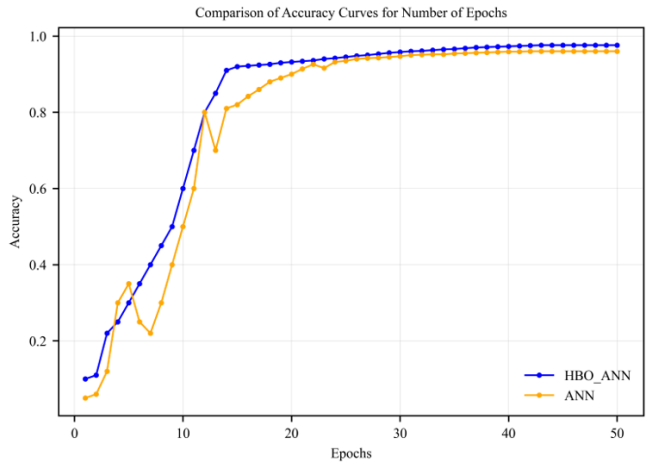


a) Confusion matrix for HBO-ANN. b) Confusion matrix of ANN algorithm.

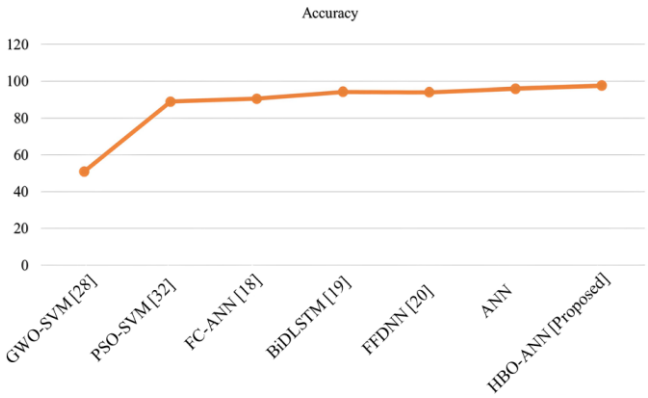
Figure 10. Comparison of confusion matrix for HBO-ANN and ANN.

Figure 10-a) illustrates the confusion matrix for HBO-ANN algorithm. The overall classification accuracy of HBO-ANN model is 97.6% as mentioned in Figure 10-a). It predicted 19458 instances as attack correctly which in turn makes 98.02% as TN. Also, the model predicted 393 instances as normal instead of attack which makes 1.98% as FP. In addition, the model predicted 19582 instances as normal correctly making 97.18% as TP and 567 instances as attack instead of normal making 2.82 % as FN.

Figure 10-b) shows the confusion matrix of ANN without any FS. It predicted 18432 instances as attack correctly which in turn makes 96.07% as TN. Also, the model predicted 753 instances as normal instead of attack which makes 3.93% as FP. In addition, the model predicted 19968 instances as normal correctly making 95.93% as TP and 567 instances as attack instead of normal making 4.07 % as FN.



a) Accuracy of HBO-ANN and ANN.



b) Accuracy comparison with benchmark algorithms.

Figure 11. Comparison of accuracy metrics for HBO-ANN, ANN, and benchmark algorithms.

Figure 11-a) shows the comparison of accuracy curve of the HBO-ANN model and ANN model for the number of epochs. Further, the accuracy of HBO-ANN model starts at 0.1 and gradually increasing and reached above 90% in around 14 epochs. Later on, the accuracy increased in small intervals and reached 97.6 % around 40 epochs and the accuracy remain the same for next 10 epochs.

Figure 11-b) describes the comparison of accuracy of the suggested HBO-ANN model, ANN method and the methods in the literature including "Grey Wolf Optimizer-Support vector machine (GWO-SVM) [28], Particle Swarm Optimization-Support Vector Machine (PSO-SVM)" [32], FC-ANN [18], Bidirectional Long-Short-Term-Memory (BiDLSTM) [19] and Feed-Forward Deep Neural Network (FFDNN) [20]. In terms of accuracy, it is evident that the recommended HBO-ANN model surpasses the previous models.

4.2.4. Precision and Recall

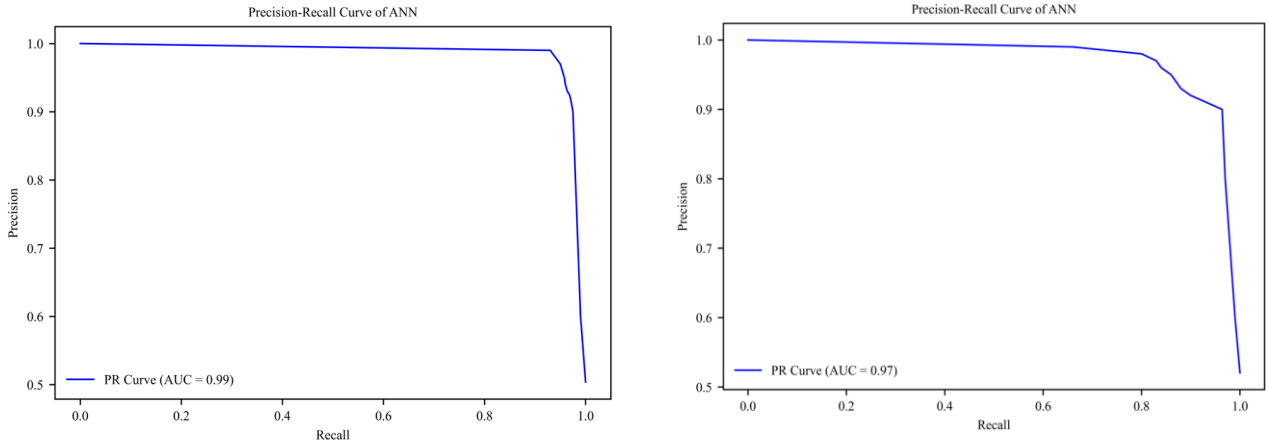
The Precision is calculated with the Equation (17) [12].

$$Precision = \frac{(TP)}{(TP + FP)} \quad (17)$$

The Recall is calculated with the Equation (18) [12].

$$Recall = \frac{(TP)}{(TP + FN)} \quad (18)$$

Recall is often used in combination with precision to assess the effectiveness of a classification model. Figure 12-a) gives the Precision Recall (PR) curve of the HBO-ANN model. The Area Under the Curve (AUC) of the PR curve is 0.99 [28]. Figure 12-b) gives the PR curve of ANN model. The AUC of the PR curve [28] is 0.97. Generally, high AUC in the context of PR curve indicates good performance. In addition, a model with high AUC-PR is able to achieve favourable balance between precision and recall across different decision thresholds. Undoubtfully, with Figure 12, the HBO-ANN model exhibits better performance when compared to the current benchmark ANN model.



a) PR curve of HBO-ANN model.

b) Precision-Recall Curve of ANN model.

Figure 12. Precision-recall curve comparison of HBO-ANN and ANN models.

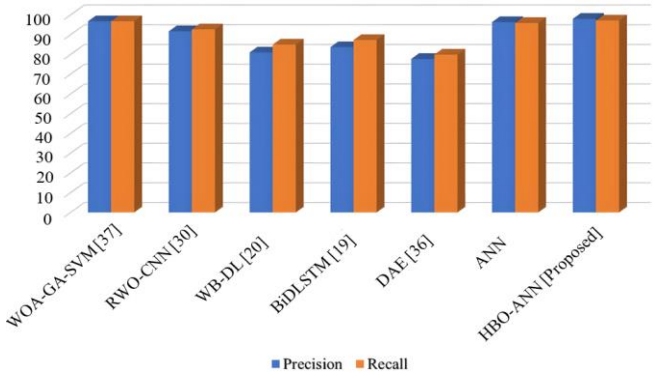


Figure 13. Comparison of PR values with state-of-the art algorithms.

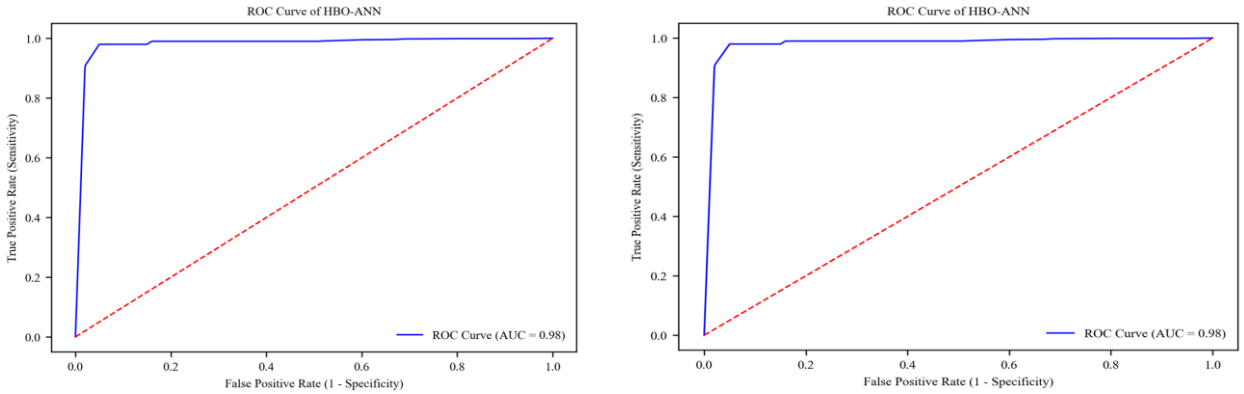
Figure 13 gives the comparison of precision and recall of HBO-ANN with the algorithms discussed in the literature. The Precision and recall of HBO-ANN is 98.03% and 97.18% respectively and that of ANN is 96.36% and 95.93%. Also, the precision and recall of Whale Optimization Algorithm-Genetic Algorithm-Support Vector Machine (WOA-GA-SVM) [37] is 96.77% and 96.76% and that of RWO-CNN [28] is 91.7% and 92.7%. In addition, the precision and recall of WB-DL [20], BiDLSTM [19], DAE [36] is 81% and 85%, 83.7% and 87.3, 77.7% and 79.9% respectively. Clearly, HBO-ANN outperforms other models with respect to recall and precision.

4.2.5. Receiver Operating Characteristic (ROC) Curve

ROC curves provide a visual representation of a model's performance, illustrating the trade-off between TP rate (sensitivity) and FP rate (1-specificity) [28].

The False Positive Rate (FPR) is the ratio of FP predictions among all actual negative instances [28]. The formula to calculate FPR is given in Equation (19).

$$Fals\ Positive\ Rate = \frac{(FP)}{(FP + TN)} \tag{19}$$



a) ROC curve of HBO-ANN.

b) ROC curve of ANN.

Figure 14. ROC curve comparison of HBO-ANN and ANN models.

Figure 14-a) gives the ROC curve of the HBO-ANN model. The AUC of the proposed model [3] is 0.98. Figure 14-b) gives the ROC curve of the ANN model. Higher AUC-ROC values indicate better discrimination ability of the system. The AUC of the proposed system is 0.96. It is evident that HBO-ANN model outperforms the ANN model. The FPR of a model must be low to minimize the count of false alarms. FPR of HBO-ANN model is 1.97% and of ANN model is 3.92%. Obviously, HBO-ANN model performs better than the ANN model in terms of FPR too.

4.3. Analysis and Discussion

The proposed hybrid approach combines HBO for FS with ANN for classification. This synergistic model addresses the challenge of selecting relevant features from high-dimensional intrusion detection dataset namely CIC-IDS2017, leading to a significant improvement in both detection accuracy and computational efficiency. HBO identifies the most critical features, while ANN effectively classifies network traffic as either benign or malicious, resulting in an improved performance.

4.3.1. Benefits of HBO for Feature Selection

FS is a crucial step in IDS, as it reduces the dimensionality of the dataset, removing irrelevant or redundant features that may confuse the classification algorithm. HBO is particularly effective for this task due to its ability to balance exploration and exploitation. HBO algorithm evaluates a wide range of possible feature subsets, thus avoiding local optima and ensuring that the most promising regions of the search space are thoroughly explored. HBO's honey phase allows it to intensively search the best areas identified during the exploration phase, fine-tuning the FS. This capability ensures that the algorithm converges on an optimal or near-optimal subset of features that have the most significant impact on improving the classification performance of the ANN.

4.3.2. Strength of ANN in Intrusion Detection

ANNs have proven to be highly effective in intrusion detection tasks because of their ability to model non-linear relationships and complex patterns in data. For IDS, where attacks may vary widely in terms of frequency, type, and signature, ANN's flexibility allows it to learn from diverse attack vectors and accurately differentiate between benign and malicious traffic. Some of the key benefits of ANN in the development of IDS are:

1. Pattern recognition: ANN excels at recognizing complex patterns in network traffic data, making it well-suited for identifying subtle anomalies that could indicate an intrusion.
2. Generalization: once trained, ANN can generalize well to unseen data, meaning it can effectively identify new and emerging threats.
3. Adaptability: ANNs can be fine-tuned or retrained with new data, making them highly adaptable to changes in network environments and evolving attack strategies.

4.3.3. Combining HBO and ANN as a Hybrid Approach

The hybrid HBO-ANN approach benefits from the synergy between the efficient FS of HBO and the robust classification capabilities of ANN. By reducing the number of features with HBO, the ANN can focus on the most informative and impactful features, leading to:

1. Improved classification accuracy: with less irrelevant data, ANN can classify the network traffic more accurately, as it is not confused by noise or redundant features.
2. Faster training and testing time: by reducing the dimensionality of the input, the computational complexity of ANN is lowered. This results in faster training times, which is particularly important when dealing with large-scale datasets, such as those used in IDS.
3. Reduced risk of overfitting: with fewer features, the ANN model becomes less prone to overfitting, ensuring better generalization to unseen data. Overfitting is a common issue in machine learning models trained on high-dimensional data, and the use of HBO helps mitigate this risk.

The complementary nature of these two techniques ensures that the hybrid model not only improves performance metrics such as accuracy precision and recall but also ensures efficiency in terms of computational resources.

4.3.4. Comparison with Traditional Feature Selection Approaches

Traditional FS methods, such as Genetic Algorithms (GA) or PSO, have been widely applied in IDS. However, these algorithms often struggle with balancing exploration and exploitation effectively [28], which can lead to suboptimal feature subsets. HBO, on the other hand, has been designed to maintain population diversity throughout the search process, ensuring that the algorithm explores a wider range of feature subsets before converging.

4.3.5. Impact on IDS Performance

The integration of HBO for FS and ANN for classification leads to significant improvements in overall system performance. The experiments conducted in this study have demonstrated several advantages:

1. Higher detection accuracy: the hybrid HBO-ANN model consistently achieves higher accuracy rates compared to models that do not employ FS. The results showed that HBO-ANN achieved an accuracy rate of 97.6%, which is a significant improvement over standalone ANN model.
2. Lower MSE: the HBO-ANN model achieved training MSE of .00317 and testing MSE of 0.009, which demonstrates that the model's predictions are highly accurate and that it has learned the patterns in the data effectively. This low error rate is particularly beneficial in detecting subtle and sophisticated attacks that may otherwise go undetected in high-dimensional data.
3. Improved precision: the proposed HBO-ANN model achieved a precision score of 98.03%, indicating a significant reduction in FPs. By carefully selecting the most relevant features through HBO, which allows the ANN to focus on the most informative aspects of the data yields this higher precision.
4. Enhanced recall: the HBO-ANN hybrid model achieved a recall score of 97.18%, demonstrating its ability to detect a wide range of attacks. A high recall rate ensures that the system does not miss malicious activities that could compromise the network.
5. Reduced FAR: the HBO-ANN approach achieved a FAR of 1.97%, making the system more reliable in real-world network environments, where false alarms are a common challenge.

4.3.6. Advantages of Proposed HBO-ANN Method

1. By applying HBO for FS, the hybrid system reduced the computational cost by narrowing the dataset to only the most relevant features.
2. The speed and efficiency of the ANN model is improved while maintaining high accuracy.

In conclusion, the combination of HBO for FS and ANN for classification in the hybrid model offers a novel and highly effective solution for intrusion detection. The synergy between HBO's optimization capabilities and ANN's classification strengths leads to significant improvements in detection accuracy, precision, recall with lower MSE and false alarm rates.

5. Conclusions and Future Tasks

This research has presented an effective approach called HBO-ANN to improve the functionality of IDS by combining ANNs with the cutting-edge HBO algorithm. Overall, HBO-ANN showed better performance in comparison with the benchmark algorithms with respect to accuracy, recall, precision and FPR. The model has been evaluated with CIC-IDS 2017 dataset. Through experiments, it has been proved that the HBO-ANN model converges faster in comparison with ANN without HBO feature reduction. The outcomes not only validated the effectiveness of the suggested system but also highlighted its potential to outperform existing approaches.

In future, the HBO algorithm can be improved by combining it with benchmark algorithms like PSO which is good in exploitation. Also, deep learning methods could be ensembled to utilize the advantages of each algorithm to achieve better performance. The ensemble model can be trained and tested with several bench mark datasets such as NSL-KDD, CIC-IDS 2018, CIC-IDS 2019, AWID and Bot-IoT and performance of the system on these datasets can be measured and compared.

References

[1] Aburomman A. and Reaz M., "A Novel SVM-KNN-PSO Ensemble Method for Intrusion Detection System," *Applied Soft Computing*, vol. 38, pp. 360-372, 2016. https://doi.org/10.1016/j.asoc.2015.10.011

[2] Al S. and Dener M., "STL-HDL: A New Hybrid Network Intrusion Detection System for Imbalanced Dataset on Big Data Environment," *Computers and Security*, vol. 110, pp. 102435, 2021. https://doi.org/10.1016/j.cose.2021.102435

[3] Ali M., Al Mohammed B., Ismail A., and Zolkipli M., "A New Intrusion Detection System Based on Fast Learning Network and Particle Swarm Optimization," *IEEE Access*, vol. 6, pp. 20255-20261, 2018. https://ieeexplore.ieee.org/document/8326489

[4] Almutairi L., Daniel R., Khasimbee S., Lydia E., Acharya S., and Kim H., "Quantum Dwarf Mongoose Optimization with Ensemble Deep Learning Based Intrusion Detection in Cyber-Physical Systems," *IEEE Access*, vol. 11, pp. 66828-66837, 2023. DOI:10.1109/ACCESS.2023.3287896

[5] Alohali M., Al-Wesabi F., Hilal A., Goel S., Gupta D., and Khanna A., "Artificial Intelligence Enabled Intrusion Detection Systems for Cognitive Cyber-Physical Systems in Industry 4.0 Environment," *Cognitive Neurodynamics*, vol. 16, no. 5, pp. 1045-1057, 2022. https://pubmed.ncbi.nlm.nih.gov/36237400/

[6] Alpaydin E., *Machine Learning* (*Revised and Updated Edition)*, MIT Press, 2021. https://www.amazon.com.au/Machine-Learning-revised-updated-Alpaydin/dp/0262542528

[7] Alqahtani A., "FSO-LSTM IDS: Hybrid Optimized and Ensembled Deep-Learning Network-based Intrusion Detection System for Smart Networks," *The Journal of Supercomputing*, vol. 78, pp. 9438-9455, 2022. https://link.springer.com/article/10.1007/s11227-021-04285-3

[8] Alweshah M., Hammouri A., Alkhalaileh S., and Alzubi O., "Intrusion Detection for the Internet of

Things (IoT) Based on the Emperor Penguin Colony Optimization Algorithm," *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 5, pp. 6349-6366, 2023. https://link.springer.com/article/10.1007/s12652-022-04407-6

[9] Alzubi O., Alzubi J., Alazab M., Alrabea A., Awajan A., and Qiqieh I., "Optimized Machine Learning-based Intrusion Detection System for Fog and Edge Computing Environment," *Electronics*, vol. 11, no. 19, pp. 1-16, 2022. https://www.mdpi.com/2079-9292/11/19/3007

[10] Chinnasamy R. and Subramanian M., *Artificial Intelligence for Intrusion Detection Systems*, Chapman and Hall/CRC, 2023. https://www.taylorfrancis.com/chapters/edit/10.1201/9781003346340-3/detection-malicious-activities-smart-signature-based-ids-ramya-chinnasamy-malliga-subramanian

[11] Chinnasamy R., Malliga S., and Sengupta N., "Deep Learning-Driven Intrusion Detection Systems for Smart Cities-A Systematic Study," *in Proceedings of the 6th Smart Cities Symposium*, Hybrid Conference, Bahrain, pp. 79-84, 2022. DOI:10.1049/icp.2023.0341

[12] Chinnasamy R., Subramanian M., and Sengupta N., "Designing of Intrusion Detection System Using an Ensemble of Artificial Neural Network and Honey Badger Optimization Algorithm," *in Proceedings of the International Conference on IT Innovation and Knowledge Discovery*, Manama, pp. 1-6, 2023. https://ieeexplore.ieee.org/document/10100161

[13] Fatani A., Dahou A., Al-Qaness M., Lu S., and Abd Elaziz M., "Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System," *Sensors*, vol. 22, no. 1, pp. 1-20, 2022. https://www.mdpi.com/1424-8220/22/1/140

[14] Ferrag M., Maglaras L., Ahmim A., Derdour M., and Janicke H., "Rdtids: Rules and Decision Tree-Based Intrusion Detection System for Internet-of-Things Networks," *Future Internet*, vol. 12, no. 3, pp. 1-14, 2020. https://www.mdpi.com/1999-5903/12/3/44

[15] Guezzaz A., Azrour M., Benkirane S., Mohy-Eddine M., Attou H., and Douiba M., "A Lightweight Hybrid Intrusion Detection Framework Using Machine Learning for Edge-based IIoT Security," *The International Arab Journal of Information Technology*, vol. 19, no. 5, pp. 822-830, 2022. https://www.iajit.org/portal/images/Year2022/No.5/21353.pdf

[16] Hajimirzaei B. and Navimipour N., "Intrusion Detection for Cloud Computing Using Neural Networks and Artificial Bee Colony Optimization Algorithm," *ICT Express*, vol. 5, pp. 56-59, 2019. https://doi.org/10.1016/j.icte.2018.01.014

[17] Hashim F., Houssein E., Hussain K., Mabrouk M., and Al-Atabany W., "Honey Badger Algorithm: New Metaheuristic Algorithm for Solving Optimization Problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84-110, 2022. https://doi.org/10.1016/j.matcom.2021.08.013

[18] Imran M., Khan S., Hlavacs H., Khan F., and Anwar S., "Intrusion Detection in Networks Using Cuckoo Search Optimization," *Soft Computing*, vol. 26, no. 20, pp. 10651-10663, 2022. https://link.springer.com/article/10.1007/s00500-022-06798-2

[19] Imrana Y., Xiang Y., Ali L., and Abdul-Rauf Z., "A Bidirectional LSTM Deep Learning Approach for Intrusion Detection," *Expert Systems with Applications*, vol. 185, pp. 115524, 2021. https://doi.org/10.1016/j.eswa.2021.115524

[20] Kasongo S. and Sun Y., "A Deep Learning Method with Wrapper Based Feature Extraction for Wireless Intrusion Detection System," *Computers and Security*, vol. 92, pp. 101752, 2020. https://doi.org/10.1016/j.cose.2020.101752

[21] Khan I., Moustafa N., Pi D., Haider W., Li B., and Jolfaei A., "An Enhanced Multi-Stage Deep Learning Framework for Detecting Malicious Activities from Autonomous Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 25469-25478, 2021. https://ieeexplore.ieee.org/document/9519840

[22] Khraisat A., Gondal I., Vamplew P., and Kamruzzaman J., "Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges," *Cybersecurity*, vol. 2, no. 20, pp. 1-22, 2019. https://doi.org/10.1186/s42400-019-0038-7

[23] Kunang Y., Nurmaini S., Stiawan D., and Suprapto B., "Attack Classification of an Intrusion Detection System Using Deep Learning and Hyperparameter Optimization," *Journal of Information Security and Applications*, vol. 58, pp. 102804, 2021. https://doi.org/10.1016/j.jisa.2021.102804

[24] Li Y., Ghoreishi S., and Issakhov A., "Improving the Accuracy of Network Intrusion Detection System in Medical IoT Systems through Butterfly Optimization Algorithm," *Wireless Personal Communications*, vol. 126, no. 3, pp. 1999-2017, 2022. https://link.springer.com/article/10.1007/s11277-021-08756-x

[25] Mansour R., Abdel-Khalek S., Hilali-Jaghdam I., Nebhen J., Cho W., and Joshi G., "An Intelligent Outlier Detection with Machine Learning Empowered Big Data Analytics for Mobile Edge Computing," *Cluster Computing*, vol. 26, pp. 71-83, 2023.

https://link.springer.com/article/10.1007/s10586-021-03472-4

[26] Moizuddin M. and Jose M., "A Bio-Inspired Hybrid Deep Learning Model for Network Intrusion Detection," *Knowledge-Based Systems*, vol. 238, pp. 107894, 2022. https://doi.org/10.1016/j.knosys.2021.107894

[27] Nasir M., Javed A., Tariq M., Asim M., and Baker T., "Feature Engineering and Deep Learning-based Intrusion Detection Framework for Securing Edge IoT," *The Journal of Supercomputing*, vol. 78, pp. 8852-8866, 2022. https://link.springer.com/article/10.1007/s11227-021-04250-0

[28] Otair M., Ibrahim O., Abualigah L., Altalhi M., and Sumari P., "An Enhanced Grey Wolf Optimizer Based Particle Swarm Optimizer for Intrusion Detection System in Wireless Sensor Networks," *Wireless Networks*, vol. 28, no. 2, pp. 721-744, 2022. https://link.springer.com/article/10.1007/s11276-021-02866-x

[29] Panigrahi R. and Borah S., "A Detailed Analysis of CICIDS2017 Dataset for Designing Intrusion Detection Systems," *International Journal of Engineering and Technology*, vol. 7, no. 3, pp. 479-482, 2018. file:///C:/Users/user/Downloads/IJET-22797.pdf

[30] Pingale S. and Sutar S., "Remora Whale Optimization-Based Hybrid Deep Learning for Network Intrusion Detection Using CNN Features," *Expert Systems with Applications*, vol. 210, pp. 118476, 2022. https://doi.org/10.1016/j.eswa.2022.118476

[31] Ponmalar A. and Dhanakoti V., "An Intrusion Detection Approach Using Ensemble Support Vector Machine Based Chaos Game Optimization Algorithm in Big Data Platform," *Applied Soft Computing*, vol. 116, pp. 108295, 2022. https://doi.org/10.1016/j.asoc.2021.108295

[32] Safaldin M., Otair M., and Abualigah L., "Improved Binary Gray Wolf Optimizer and SVM for Intrusion Detection System in Wireless Sensor Networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 1559-1576, 2021. https://link.springer.com/article/10.1007/s12652-020-02228-z

[33] Scarfone K. and Mell P., *Guide to Intrusion Detection and Prevention Systems (IDPS)*, NIST Special Publication, 2007. https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-94.pdf

[34] Sood K., Nosouhi M., Nguyen D., Jiang F., Chowdhury M., and Doss R., "Intrusion Detection Scheme with Dimensionality Reduction in Next Generation Networks," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 965-979, 2023. https://ieeexplore.ieee.org/document/10004971

[35] Tabash M., Abd Allah M., and Tewfik B., "Intrusion Detection Model Using Naive Bayes and Deep Learning Technique," *The International Arab Journal of Information Technology*, vol. 17, no. 2, pp. 215-224, 2020. https://www.iajit.org/portal/PDF/Vol%2017,%20No.%202/17046.pdf

[36] Thakur S., Chakraborty A., De R., Kumar N., and Sarkar R., "Intrusion Detection in Cyber-Physical Systems Using a Generic and Domain Specific Deep Autoencoder Model," *Computers and Electrical Engineering*, vol. 91, pp. 107044, 2021. https://doi.org/10.1016/j.compeleceng.2021.107044

[37] Vijayanand R. and Devaraj D., "A Novel Feature Selection Method Using Whale Optimization Algorithm and Genetic Operators for Intrusion Detection System in Wireless Mesh Network," *IEEE Access*, vol. 8, pp. 56847-56854, 2020. https://ieeexplore.ieee.org/document/9022974

[38] Yan X., He Z., Huang Y., Xu X., Wang J., Zhou X., Wang C., Lu Z., "A Lightweight Pedestrian Intrusion Detection and Warning Method for Intelligent Traffic Security," *KSII Transactions on Internet and Information Systems*, vol. 16, no. 12, pp. 3904-3922, 2022. https://itiis.org/digital-library/38212

Ramya Chinnasamy is a Ph.D. Research Scholar, Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, India.

Malliga Subramanian is a Professor and Head of the Department, Department of Computer Science and Engineering, Kongu Engineering College, Perundurai, India.

Nandita Sengupta is a Professor, Head of Information Technology Department, University College of Bahrain (UCB), Manama, Kingdom of Bahrain