Intrusion Detection System based on Chaotic Opposition for IoT Network

Original Scientific Paper

Richa Singh
G.G.S.I.P.U.,
University School of Information, Communication, and Technology
Delhi, India
richa.singh081991@gmail.com

R.L. Ujjwal
G.G.S.I.P.U.,
University School of Information, Communication, and Technology
Delhi, India
ujjwal@ipu.ac.in

Abstract *– The rapid advancement of network technologies and protocols has fueled the widespread endorsement of the Internet of Things (IoT) in numerous domains, including everyday life, healthcare, industries, agriculture, and more. However, this rapid growth has also given rise to numerous security concerns within IoT systems. Consequently, privacy and security have become paramount issues in the IoT framework. Due to the heterogeneous data produced by smart IoT devices, traditional intrusion detection system doesn't work well with IoT system. The massive volume of heterogeneous data has several irrelevant, redundant, and unnecessary features which lead to high computation time and low accuracy of IDS. Therefore, to tackle these challenges, this paper presents a novel metaheuristic-based IDS model for the IoT systems. The chaotic opposition-based Harris Hawk optimization (CO-IHHO) algorithm is used to perform the feature selection of data traffic. The chosen features are subsequently inputted into a machine learning (ML) classifier to detect network traffic intrusions. The performance of the CO-IHHO based IDS model is verified against the BoT-IoT dataset. Experimental findings reveal that CO-IHHO-DT achieves the maximal accuracy of 99.65% for multiclass classification and 100% for binary classification, and minimal computation time of 31.34 sec for multiclass classification and 133.54 sec for binary classification.*

Keywords: *IoT, IDS, feature selection, machine learning, HHO*

1. INTRODUCTION

The Internet of Things (IoT) is a collection of several interconnected embedded devices that can communicate with each other through wireless or wired mediums [1]. Within the IoT system, numerous smart sensors collaborate to create intelligent environments. The advancement in IoT systems makes spectacular development in the everyday utilization of electronic services and appliances [2]. It has a profusion of applications and services in various domains including agriculture, healthcare, industry, military, smart homes, etc. However, the widespread adoption of IoT also makes these systems attractive targets for malicious actors aiming to carry out activities such as physical damage to devices, denial of service (DoS) attacks, and theft of information. Consequently, ensuring the security of IoT devices becomes paramount. [3]. Moreover, staying informed about contemporary vulnerabilities is essential to take appropriate measures for mitigation.

An intrusion detection system provides a security mechanism for protecting the IoT system from several malicious activities by analysing data packets received and generating responses when necessary. An IDS for the IoT system has to deal with rigorous conditions of high-volume data processing, rapid response, memory constraints, and low processing. Therefore, typical IDS are not suitable for IoT systems. The IDS are classified into major categories based on analysis strategy i.e. anomaly, signature [4], and hybrid. Signature IDS detects attack by analysing network traffic and matching attack signatures with signatures already stored in a database, generating an alarm if signatures are matched. While Anomaly-based IDS constructs user profiles by analyzing system usage patterns. Any deviation from established user behavior is treated as a potential intrusion. Hybrid combines the merits of both strategies.

The considerable volume of network traffic generated by IoT devices presents a significant challenge when

it comes to ensuring the security of IoT network traffic. Feature selection (FS) can resolve this issue by considering only the relevant and important features instead of all the features. It is an important technique that improves the performance of IDS for the IoT framework. FS [5] is intermediate phase of IDS that choose subset from the original features set without irrelevance, and redundancy. It helps in reducing the volume of training data, improving the classification accuracy, and reducing computation time [6]. FS is typically divided into three primary approaches: filter, wrapper, and hybrid. In the filter approach, statistical measures such as correlation and consistency are utilized for evaluation, and this process is conducted independently of any specific learning algorithm. In contrast, the wrapper approach entails the learning algorithm itself evaluating the feature subset, with the subset's efficiency determined by the error rate. While the wrapper approach is computationally more intensive, it often yields superior results compared to the filter approach because of its continuous interaction with the learning algorithm [7]. The hybrid approach leverages the strengths of both the filter and wrapper approaches.

Metaheuristic algorithms (MHA) are the wrapper approach of FS which gives a magnificent performance because of their global search capability. However, it is necessary to have equitable exploration and exploitation phase of MHA to avoid local optima [8]. Extreme exploitation and inadequate exploration cause premature convergence while extreme exploration and inadequate exploitation cause a slower convergence rate. Exploration refers to generating a candidate solution that leads to wider coverage of search space while exploitation refers to finding a near-optimal solution by focusing on the local area of the search process.

This paper introduces an IDS tailored for the IoT domian, employing a Metaheuristic Algorithm (MHA). The IDS model's feature selection process utilizes the proposed Chaotic Opposition-based Improved Harris Hawk Optimization (CO-IHHO) algorithm on the BoT-IoT dataset. CO-IHHO is employed to identify the most pertinent features, optimizing computational efficiency while upholding the accuracy of the IDS. Following feature selection, binary and multiclass classification tasks are carried out using machine learning (ML) classifiers. Consequently, the primary contributions of this paper encompass the following:

- Improving the population diversity of original HHO by applying opposition-based learning in the beginning of population selection of HHO. This helps in getting the best fitness solution in the early stages and leads to improving convergence speed.
- The chaotic map technique is employed for generating random numbers used in HHO.
- Non-linear target energy escaping form is used to have an equitable exploration and exploitation phase.
- The paper conducts classification tasks utilizing two distinct ML classifiers and subsequently assesses their performance. To address the issue of class imbalance, a sampling technique is employed as a resolution.
- The CEC-06 2019 Benchmark function is employed to evaluate the performance of proposed CO-IHHO.

The effectiveness of CO-IHHO is compared against seven other MHAs for the BoT-IoT dataset. The intrusion detection with CO-IHHO as FS algorithm achieves high accuracy with less computation time, compared to other MHAs used as FS algorithm. Further, the proposed work is also attaining better accuracy compared to other recent work.

The remaining paper unfolds as follows: section 2 describes a contribution summary of the latest IDS for the IoT system followed by a preliminary discussion about the original HHO, OB learning, and chaotic map techniques. Afterward, the proposed IDS model is described in section 3. Section 4 describes the proposed CO-IHHO algorithm in detail. Section 5 is dedicated to the implementation and analysis of results from the proposed work. Section 6 offers the paper's conclusion.

2. LITERATURE REVIEW

Researchers have developed several IDS for the IoT network using learning algorithms in recent years. This section summarizes the recent work done for the FS using MHAs, and other recent IDS for the IoT framework. Afterward, the original HHO, OB learning, and the chaotic map are discussed.

The authors in [9] proposed IoT based IDS using deep learning framework. They use MHA spider monkey optimization (SMO) for optimal FS. Afterwards, they employed stacked-deep polynomial network for the classification of intrusive traffic in IoT system. They used L2 regularization technique to avoid model overfitting. However, the proposed IDS is verified using NSL-KDD, which is an outdated dataset and doesn't include latest attacks. The random forest (RF) based smart IoT-based IDS is proposed in [10]. They combine elements of grey wolf optimization and particle swarm optimization to enhance the FS process for intrusive traffic. Furthermore, RF is used to perform multiclass classification. They performed oversampling to handle imbalanced data. The model is evaluated against outdated datasets including CICIDS-2017, KDDCup99, and NSL–KDD and attains accuracy above 99% for all three datasets. These datasets don't have IoT traces. The work in [11] proposed a hybridized MHA for the IoT system. The study combines the bird swarm algorithm with the gorilla troops optimizer to improve FS. The model's performance is assessed on various datasets, including CICIDS-2017, NSL-KDD, BoT-IoT, and UNSW-NB15, resulting in accuracies of 98.7%, 95.5%, 81.5%, and 81.5%, respectively. Authors in [12] proposed an smart botnet detection method for IoT system. They hybridized salp

swarm algorithm with an ant lion optimization algorithm for the FS of the N-BaIoT dataset. This hybrid approach leveraged the global search capabilities of ALO and the local search capabilities of SSA to obtain the optimal solution. The classification is performed using the KNN classifier. In paper [13], a lightweight IDS tailored for IoT systems was presented. This research involved a fusion of the genetic algorithm (GA) and the GWO for the FS. Their model's performance was evaluated against AWID dataset. Authors in [14] proposed MHA-based FS for the IoT system. They proposed three models based on simulated annealing and shuffled shepherd optimization algorithms i.e. SSO, SSO-SA1, and SSO-SA2 to perform FS. In SSA-SA1, SA is merged within SSO and in SSA-SA2, SA is used after SSO. Features obtained from the proposed system are classified using the KNN classifier. Experimental result shows that SSO-SA2 is better in contrast to all other algorithms. Hamed et al. [15] proposed an IDS to protect the edge layer of the IoT system from malware. The optimal feature selection of a dataset with opcodes and bytecodes is performed by using the GWO algorithm. They proposed a multi-kernel SVM approach for the classification of malware. The proposed approach is better than deep-RNN and fuzzy-based IDS. The authors in [16] proposed an IoT-based IDS with a deep neural network (DNN) framework. They used filter-based mutual information for feature selection. Features with a high MI score are selected as optimal features. The IoT traffic is classified as an anomaly or begins using DNN. The proposed system attains an accuracy of 99.01%. The authors in [17] proposed a hybrid IDS designed specifically for IoT framework. Their approach involved the use of an enhanced shuffled frog leaping algorithm as a wrapper for FS. Following this, they employed a Light Convolutional Neural Network with Gated Recurrent Neural Network (LCNNGRNN) for the classification of intrusive network traffic. The performance of their proposed model demonstrated its effectiveness when benchmarked against other methods, especially when evaluated on the NSL-KDD dataset. The work in [18] proposed a NIDS for the medical IoT system. They employed a butterfly optimization algorithm (BOA) to get the best features. Afterward, ANN is used to categorize the network traffic based on optimal features obtained using BOA. The proposed system attains an accuracy of 93.27% over the NSL-KDD dataset. The authors in [19] proposed GA based anomaly detection system for the fog based IoT framework. The FS is performed using wrapper based GA and classification is performed using deep brief network. Proposed system achieves 99.73% accuracy and 0.06% false positive rate for the NSL-KDD dataset. However, existing IDS approaches performed well for the IoT framework still there are certain limitations, which includes:

Mostly datasets like UNSW-NB15, KDDcup99, CICIDS-2017, NSL-KDD, etc. are used for performance evaluation of IDS for IoT framework. However, such datasets become obscure and doesn't have IoT traces.

- Most of the proposed IDS used MHA for FS either incur high computation time or doesn't include any information about computation time. Furthermore, detecting intrusive traffic with less computation time, and high accuracy is important concern.
- The FS using HHO might leads to premature convergence and trapped in local optima.

2.1. HARRIS HAWK ALGORITHM (HHO)

The HHO algorithm is developed in 2019 by authors in [20]. HHO mimics the hunting behavior of Hawk birds required to catch the prey (rabbit). These birds perch in the air, search for prey, and then dive on it collectively. Each group of hawks contains two to seven members. The HHO algorithm includes two phases: exploration, which models hawks preaching behavior, and exploitation, which models different attacking styles of hawks.

Exploration Phase: During this phase, hawks are distributed randomly in the search area, waiting for their prey to arrive. They detect and trace prey with their powerful eyes. These birds can wait for several hours for a prey to arrive. If d≥0.5, hawks use family member position for hunting, while if d<0.5, then hawks use random positions for hunting. Using the value of d, hawks position is updated using following equations:

$$A(k+1) = \begin{cases} A_{rn}(k) - rn1|A_{rn}(k) - 2\,rn2\,A(k)|, & d \geq 0.5 \\ \left(A_{target}(k) - A_{mean}(k)\right) - rn3(LBU + rn4\,(UBU - LBU)), & d < 0.5 \end{cases} \quad (1)$$

where, d, $rn1$, $rn2$, $rn3$, and $rn4$ represents random numbers within range 0, and 1. d is used to toggle between two position-updating equations. LBU represents lower limit and UBU represents upper limit of search space. Target position is denoted by $A_{target}(k)$, and A_{rn} denotes the randomly selected hawks. $A(k)$ is the current hawks position, and $A_{mean}(k)$ signifies the average position computed based on the current population of hawks. This average position is determined through the following formula:

$$A_{mean}(k) = \frac{1}{k_{maxite}} \sum_{i=1}^{k_{maxite}} A_i(k) \quad (2)$$

Escaping Energy (En): The shift from exploration to the exploitation phase is contingent upon the energy levels of the prey, and the act of evading often results in a depletion of their energy. The energy is a time-varying variable defined by the equation:

$$En = 2E_{initial} * \left(1 - \frac{k}{k_{maxite}}\right) \quad (3)$$

$$E_{initial} = -1 + 2 \times rand() \quad (4)$$

where $E_{initial}$ denotes initial prey escaping energy. k denotes current iteration. k_{maxite} denotes the maximum iteration number. Depending on the E_n value exploration and exploitation phase happen. If $E_n \geq 1$ then the exploration part of HHO is executes while if $E_n < 1$ then the exploitation part of HHO executes.

Exploitation Phase: During this phase, hawks exploit prey using surprise dives. Based on escaping behavior of prey, the hawk selects their attacking strategy. The strategies include hard besiege (HB), hard besiege with progressive dive (HBPD), soft besiege (SB), and soft besiege with progressive dive (SBPD). Progressive dive strategies show the intelligent behavior of hawks. The strategy to be adopted by the hawks depends upon the value of escaping probability (pr), and escaping energy (En). If pr<0.5 then the prey escape successfully and if pr≥0.5 then the prey is unsuccessful in escaping before a surprise attack.

SB: Whenever the prey possesses sufficient energy to escape, the hawks employ a gentle encirclement strategy, which gradually tires out the prey, enabling them to execute surprise attacks effectively. The SB is determined using the following equation:

$$A(k+1) = \Delta A(k) - En\left|J\ A_{target}(k) - A(k)\right| \quad (5)$$

where, $A(k)$ is the current hawks position, En is energy level, $A_{target}(k)$ is target position, J is the jumping strength, which is determined using following equation:

$$J = 2(1 - rn5) \quad (6)$$

where, $rn5$ is the random number. Further, $\Delta A(k)$ refers to the disparity between the current hawks position and its target position. It is determined using following equation:

$$\Delta A(k) = A_{target}(k) - A(k) \quad (7)$$

HB: In this scenario, the prey lacks the necessary energy to escape from the hawks. The hard besiege is mathematically modeled as:

$$A(k+1) = A_{target}(k) - E|\Delta A(k)| \quad (8)$$

SBPD: In this method, the levy flight function (LFF) determines the zigzag maneuver of prey during an escape. The target has enough escaping power. Therefore, hawks try to distract the target so that it changes its path. This process continues until hawks perch their target. The hawk chooses their next favorable move using the equation:

$$B = A_{target}(k) - E\left|J\ A_{target}(k) - A(k)\right| \quad (9)$$

If the hawks find that target is trying to mislead hawks and tries to escape, then LFF function is used to perform dive. The position is determined using following equation:

$$C = B + X \times LFF \quad (10)$$

where, X is random vector of size 1 X DM (Dimension). LFF is determined using the following equation:

$$LFF = \frac{e \times \sigma}{|f|^{\frac{1}{\beta}}}, \text{ where } \sigma = \left(\frac{\Gamma(1+\beta)\times\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\times \beta\times 2^{\left(\frac{\beta-1}{2}\right)}}\right)^{\frac{1}{\beta}} \quad (11)$$

where, β is constant with a value 1.5, e and f are random number, lies in interval 0 and 1.

Therefore, in SBPD Hawks positon is determined using the following mathematical equation:

$$A(k+1) = \begin{cases} B\ if\ FF(B) < FF(A(k)) \\ C\ if\ FF\ (C) < FF(A(k)) \end{cases} \quad (12)$$

where, FF is the fitness function obtained using equation 24.

HBPD: In this method, prey has insufficient escaping energy. Hence, hawks strive to minimize the distance between their mean location and the location of their prey, which is determined using the following equation:

$$A(k+1) = \begin{cases} B'\ if\ FF(B') < FF(A(k)) \\ C'\ if\ FF(C') < FF(A(k)) \end{cases} \quad (13)$$

where,

$$B' = A_{target}(k) - E\left|J\ A_{target}(k) - A_{mean}(k)\right| \quad (14)$$

$$\text{and } C' = B' + X \times LFF \quad (15)$$

LFF is the levy flight function obtained from equation 11, $A_{target}(k)$ is the target position, and X defines a random vector of size 1× DM (dimension). $A_{mean}(k)$ is determined by equation 2.

The HHO is a population-based MHA that has several advantages and limitations as well. The HHO is popular due to its simpler structure. It requires few parameter settings compared to other MHA [21], and ease of implementation. The HHO is flexible, scalable, and robust. It provides a good convergence speed. However, it doesn't have any theoretical analysis, and also lacks mathematical analysis. Moreover, the existence of random variables in the different phases of HHO reduces its convergence speed and is stuck in local optima. Therefore, chaos theory is used to determine the value of these random variables used in HHO and helps to improve its performance.

2.2. CHAOTIC MAPS

The term chaotic refers to "state of chaos". The CM are functions that mathematically computes random values based on the seed value provided initially. From the last few decades, CM are widely adopted for optimizing MH algorithms due to their dynamic behaviour [22]. Searching search space becomes faster using CM than random number generator. They generate random numbers that lies between certain range. The chaotic map has various characteristic such as randomness, ergodicity, i.e. traverse all states without repetition, and highly sensitive to initial value [23]. These attributes of Chaotic Maps (CM) assist Metaheuristic (MH) algorithms in enhancing convergence speed and steering clear of local optima. The exploration and exploitation phases of HHO involve multiple random numbers, which could potentially lead to HHO becoming trapped in local optima. Therefore, in this paper, these random numbers are determined by using a chaotic map "Chossat-Golubitsky" [24]. The equation of CG-CM is given by:

$$A = a\ \times\ (x^2 + y^2) + b\ \times\ x\ \times (x^2 - 3y^2) + c \quad (16)$$

2.3. OPPOSITION BASED LEARNING

The OB learning concept was initially introduced by the authors in [25] and has been applied in various studies, including [26] and [27], to optimize Metaheuristic Algorithms (MHAs). Typically, many MHAs begin with the selection of random numbers as their initial solutions. Therefore, in this paper, the OB learning technique is employed to enhance the population diversity of the original HHO algorithm. Instead of selecting the hawk's population randomly from search space, the OB learning method is used to select random hawk positions i.e. $A_{rn}(k)$. This approach navigates the search space in dual directions, taking into consideration both the original solution and its reverse counterpart. By doing so, it offers a more comprehensive coverage of the search space [28]. The convergence speed of the MHA is slower in most cases. Therefore, OB learning resolves this problem by by taking into account both randomly generated solutions and their opposites [29]. The authors in [30] shows that the opposite solution is more capable to reach global optima compared to the original solution.

Definition: In general, let $x \in [l,u]$ be a real number. The opposite number ($\bar{x}$) is evaluated using the following equation:

$$\bar{x} = (lbu + ubu - x)$$

where lbu is lower bound. ubu is upper bound.

If x is a multi-dimensional vector then, all elements of $\bar{x}$ is defined by:

$\bar{x}=(lbu_w+ubu_w-x_w)$ where, w=1,2,3, 4, ….., d and d is the multi-dimension.

3. PROPOSED SYSTEM

With the growing security concerns surrounding IoT devices, the need for an efficient IDS that can accurately detect intrusions while minimizing processing time becomes crucial. The complete architecture of the suggested system is illustrated in Fig. 1. The IDS is organized into multiple stages, encompassing data collection, data pre-processing, feature selection using the CO-IHHO method, and the classification phase. The proposed IDS is verified using BoT-IoT [31] dataset. The data undergoes pre-processing to eliminate ambiguities and address missing values. Subsequently, CO-IHHO is applied for feature selection on the pre-processed data, followed by binary and multiclass classification of intrusive traffic. The section aims to provide a comprehensive overview of each of these phases within the IDS.

3.1. DATA COLLECTION

During this phase various logging tools are used for used for preparing dataset, created dataset can be used to train model for identifying intrusive traffic. In this paper, publicly available BoT-IoT dataset [31] is used as collected data, described briefly in later section. It is feed as an input to data-processing phase of the IDS model.

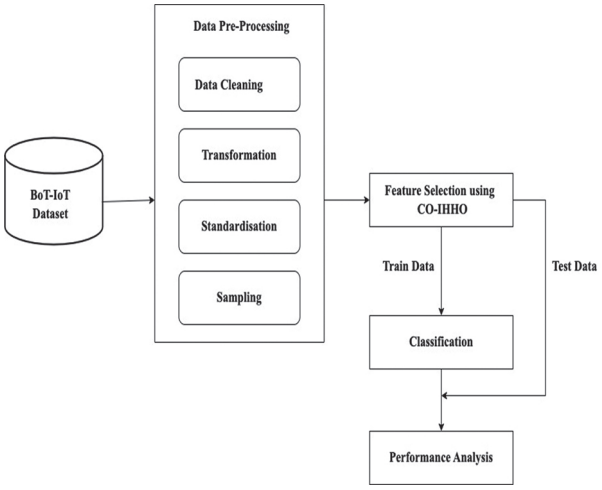

Fig. 1. Proposed System Design

3.2. PRE-PROCESSING

From the review of the literature conducted above, it becomes apparent that the majority of the datasets utilized for performance evaluation exhibit characteristics such as noise, missing values, irrelevant information, and redundancy. Additionally, it's worth noting that the data generated by IoT devices is inherently heterogeneous in nature. Hence, it is necessary to handle these redundant and irrelevant data and further, alter the data into a uniform form. The pre-pressing phase helps in improving data quality and providing effective, and accurate result. The data from the data collection module is provided as an input to the pre-processing phase. The pre-processing task includes, cleaning, transformation, standardization, and sampling.

During cleaning, the quality of the collected data is improved by eliminating undesired, and redundant attributes. Undesired attributes are those whose value doesn't impact the performance of intrusion detection while redundant attributes value can be derived from other attribute value. Furthermore, attributes with missing values are also eliminated. In this paper, redundant, and irrelevant attributes such as 'ltime', 'daddr', 'saddr', 'stime', 'flgs', 'state', 'proto', 'pkSeqID', 'sport', 'dport', and 'seq' are eliminated.

During transformation and standardization, several ML classifiers takes only numerical value as processing input. Consequently, categorical values are converted into numerical values using a method called Label encoding. Furthermore, the data is standardized to ensure that data values fall within the range of [0,1]. Failure to standardize data can impact the performance of the IDS due to varying data value ranges. To achieve this, the Standard Scaler function is employed.

Sampling: The BoT-IoT dataset exhibits imbalanced data with a significantly higher number of intrusive

traffic instances compared to normal instances. To address this disparity, sampling techniques are applied to balance the instance counts between these two categories. Imbalanced dataset makes the classifier biased towards majority class (intrusive traffic, in this case) and reduces the possibility to detect minority class (normal instances). In this context, random sampling is utilized to mitigate the imbalanced nature of the BoT-IoT dataset. After preprocessing, the FS is performed using CO-IHHO, described in later section.

3.3. CLASSIFICATION PHASE

During this phase, the task involves identifying intrusive traffic, and it encompasses both binary and multiclass classification. Binary classification distinguishes between normal and intrusive traffic, while multiclass classification goes further by identifying specific attack categories within the intrusive traffic, alongside normal traffic. To carry out these classification tasks, the selected features obtained from the feature selection phase are used as input for two machine learning classifiers: Decision Tree (DT) and K-Nearest Neighbor (KNN). Subsequently, an analysis of their performance is conducted. These ML algorithms offer several benefits, including interpretability, ability to handle mixed data types, resilience to noise and irrelevant features, and ease of implementation. These advantages make them well-suited for tasks involving classification and detection.

4. FEATURE SELECTION USING CO-IHHO

The fine-tuned is provided as an input to the FS phase. FS eliminates unwanted, and irrelevant features from dataset and extracts best features out of it [32]. The random number used in MHA has crucial effect on their performance of determining global and local search capability. HHO structure has several random parameters which prevent them in obtaining global optima. Therefore, CM techniques is embedded in the structure of HHO for determining random numbers. The hawks position in the exploration phase of HHO is determined either by family members or randomly selected population of hawks. Therefore, in this paper, OB learning technique is employed to intensify the population update of hawks instead of random population selection. Furthermore, the sine and cosine function are used to enhance the capability of hawk's exploration. The exploitation phase of HHO is also enhance by introducing dynamic capability using S function, which is inspired by [33] , and introducing random parameters to the advanced dive phase. This helps the hawks to exploit local regions rigorously. Here, CO-IHHO is used for the FS task, which overcomes the drawback of original HHO. The algorithm 1 depicts the pseudocode of proposed CO-IHHO.

Algorithm 1. Chaotic Opposition based Improved HHO (CO-IHHO).

Input: N is population size, and k_{maxite} is maximum iterations

Output: Optimal Feature subset

Assign the population s=1, 2,….,n

While(k<=k_{maxite})

Initialize "Chossat-Golubitsky" chaotic map to identify random numbers using equation 16

Compute population of fittest hawks $A_{OBL}(k)$

Compute hawks mean position using equation (2)

Compute escaping energy using equation 23

if ($|En|$ >=1.5) *Exploration phase

  revise hawks positon using equation 17

else if ($|En|$ <1.5) * Exploitation phase

if ($|En|$ ≥1 and pr ≥ 0.5)

  revise hawks position using equation 18 #SB

if ($|En|$ <1 and pr ≥ 0.5)

  revise hawks position using equation 19 #HB

else if ($|En|$ ≥1 and pr <0.5)

  revise hawks position using equation 20 #SBPD

else if ($|En|$ <1 and pr <0.5)

  revise hawks position using equation 21 #HBPD

end if

return optimal selected features

end while

end

The CO-IHHO is the result of following enhancements made to original HHO-

Chaotic Map: The incorporation of a chaotic map to generate random numbers plays a pivotal role in both the exploration and exploitation phases of the HHO algorithm. This deliberate inclusion serves as a safeguard against HHO becoming ensnared in local optima, leading to a substantial improvement in its convergence speed.

OB Learning: The OB learning is used for enhancing the population diversity of HHO. Instead of selecting hawks population randomly in equation (14), OB learning is used for selecting hawks population. OB Learning with CO-IHHO:

- Hawks position A is initialized as a_s, where s=1,2,3,….,n
- Calculate opposite position of hawks as $\bar{a}_s$, where j=1,2,3,…,n.
- Choose the n fittest hawks from $(a_s \cup \bar{a}_s)$ which represent the new initial population of hawks i.e. $A_{OBL}(k)$.

Exploration Phase (En≥1.5): CO-IHHO enhances the exploration capability of HHO by using sine and cosine functions in updating the position of Hawks. Inertia weight (w) is also introduced. The Hawks position is calculated as:

$$A(k+1) = \begin{cases} w \times A_{OBL}(k) - \sin(rd1) \times T \times |A_{rn}(k) - 2\,rd2\,A(k)|, & d \geq 0.5 \\ w \times \left(A_{target}(t) - A_{mean}(k)\right) - \cos(rd3) \times T \times (LBU + rd4\,(UBU - LBU)), & d < 0.5 \end{cases} \quad (17)$$

where, $w = \left(1 - \frac{k}{k_{maxite}}\right)^{\sqrt{\frac{k}{k_{maxite}}}}$ and $T = \left(1 - \frac{k^{\frac{1}{p}}}{k_{maxite}^{\frac{1}{p}}}\right)$ where p=6, $rd1$, $rd2$, $rd3$, and $rd4$ are the random vari-

ables computed using CM. UBU is upper search space limit. LBU is the upper search space limit. Harris hawks population mean position i.e. $A_{mean}(k)$ is calculated using equation 2, and $A_{target}(k)$ is target positon. Also, k is current iteration. $rd1$, $rd2$, $rd3$, and $rd4$ are random number determined using CM. k_{maxite} is maximum iteration.

***Exploitation Phase* (*En*<1.5):** The CO-IHHO enhances the exploitation capability of HHO by adding dynamic capability using value of S and random number. Depending upon the value of En and p (random number), the hawks position is updated using the following equations:

SB: This phase executes when $|En|$ >=1 and p >= 0.5. The position is determined using

$$A(k+1) = \Delta A(k) \times rd5 - En\left|J\ A_{target}(k) - A(k)\right| + S \quad (18)$$

where, $rd5$ is random number determined using CM. $\Delta A(k)$ is obtained using equation 7. En is escaping energy obtained using equation 23. Jumping strength is obtained using equation 6. $A_{target}(k)$ is target positon. $A(k)$ is current position.

HB: This phase executes when ($|En|$ >=1 and p >= 0.5). The position is determined using:

$$A(k+1) = A_{target}(k) - En|\Delta A(k)| + S \quad (19)$$

where, $\Delta A(k)$ is obtained using equation 7. $A_{target}(k)$ is target position.

SBPD: This phase executes when ($|E_n|$ >=1 and p <0.5). The position is determined using

$$A(k+1) = \begin{cases} B \times rd6 + A(k) \times rd7 + S & if\ FF(B) < FF(A(k)) \\ C \times rd8 + A(k) \times rd9 + S & if\ FF\ (C) < FF(A(k)) \end{cases} \quad (20)$$

where, B and C are obtained using equation 9 and equation 10, respectively. FF is the fitness function. $rd6$, $rd7$, $rd8$, and $rd9$ are random number between 0 and 1, determined using CM. $A(k)$ is current hawks position.

HBPD: This phase executes when (|En| <1 and p <0.5). The position is determined using:

$$A(t+1) = \begin{cases} B' \times rd10 + A(k)\ \times\ rd11 + S & if\ FF(Y') < FF(A(k)) \\ C'\ \times\ rd12 + A(k) \times rd13 + S & if\ FF(Z') < FF(A(k)) \end{cases} \quad (21)$$

where,

$$S = randInt\ \times \left[\sin\left(\pi \times \frac{k}{2 \times k_{maxite}}\right) + \cos\left(\pi \times \frac{k}{2 \times k_{maxite}}\right) - 1\right] \quad (22)$$

$rd10$, $rd11$, $rd12$, and $rd13$ are random number determined using CM between [0,1]. FF is fitness function. $A(k)$ is current hawks position. B' and C' are obtained using equation 14 and equation 15 respectively.

Escaping Energy: The CO-IHHO modifies the escaping energy equation, shifting it from a linear to a non-linear form in order to achieve a more balanced exploration and exploitation phase. This transformation is precisely defined by the following equation:

$$En = rd0\ \times E_o \times e^{\left[\left(\frac{k_{maxite}-k}{k_{maxite}+k}\right)\times 1.5\right]} \quad (23)$$

where, E_o=0.75, and $rd0$ is random number lying between (0,1).

5. IMPLEMENTATION AND RESULT ANALYSIS

This section describes the experimental setup, and CEC-06 2019 benchmark function to evaluate the performance of proposed CO-IHHO. Furthermore, dataset used for model evaluation, and performance metrics considered for evaluating the model are described. Afterward, a comparison with other MHAs and recent IoT-based IDS is done.

5.1. EXPERIMENTAL SETUP

The experiment conducted in this paper is implemented using Python 3.2 on Mac OS Catalina with 8 GB RAM. For an accurate comparison, each implemented algorithm is given a standard situation. The number of iterations i.e. 50, and the population used by all algorithms is the same. Table 1 presents the parameter setting of each MHA used as FS.

Table 1. Parameter Setting.

S. No.	Method	Parameter Setting
1.	CO-IHHO	Threshold= 0.5 β = 1.5 E_o=0.75
2.	ISSA	Maximum iteration for local search algorithm ($maxLt$) = 10
3.	ISCA	Elites number (Ne) = 10 α= 2
4.	TMGWO	Mutation Probability (Mp) = 0.5
5.	SSA	Threshold = 0.5
6.	GWO	Threshold = 0.5
7.	HHO	Threshold = 0.5 β= 1.5
8.	WOA	b = 1 (constant)
9.	Chossat-Golubitsky	a = -1.0 b = 0.1 c = 1.52 d = -0.8 x = 0.1 y = 0.1

5.2. BENCHMARK CEC-06 2019 EVALUATION

The proposed CO-IHHO algorithm's performance is assessed using the CEC-06 2019 benchmark functions, which consist of 10 single-objective optimization problems [34]. These functions, labeled CEC F01 to CEC F10, present diverse challenges with shifted and rotated configurations for some functions. The dimensions of CEC F1, CEC F2, and CEC F3 are 9, 16, and 18, respectively, while the rest are 10-dimensional. The evaluation involves running all MHAs, including the original HHO, proposed CO-IHHO, SSA, GWO, WOA, and PSO [35], for 100 iterations on each function. Table 2 provides the names and ranges of the CEC-06 benchmark functions, while Table 3 displays the evaluation results for the minimization function.

The effectiveness of exploration and exploitation phases is evaluated using function assessment. The competitive outcomes indicate that CO-IHHO maintains a balanced trade-off between exploitation and exploration, outperforming other algorithms in terms of best value, average value, and standard deviation for CEC F1, CEC F2, CEC F3, CEC F6, CEC F7, CEC F8, and CEC F10. The results reveal that CO-IHHO consistently outperforms other MHAs in various instances. The performance of PSO is better for CEC F4 as compared to other algorithms, and GWO performs better for CEC F5. The original HHO shows superior performance for best value and average value in CEC F9, however, the convergence of CO-IHHO is better compared to original HHO for CEC F9, as shown in Fig. 10. It is noteworthy that PSO achieves zero standard deviation for CEC F9, indicating no further improvement can be made with this algorithm. Fig. 2 demonstrates that CO-IHHO exhibits better convergence compared to the HHO algorithm for CEC F1. Similarly, Fig. 3 demonstrates that CO-IHHO achieves better convergence in comparison to the original HHO for CEC F2. Moreover, Fig. 4, Fig. 5, and Fig. 6 illustrate the convergence behaviour of CO-IHHO and HHO for CEC F3, CEC F4, and CEC F5 benchmark functions, respectively. Notably, Fig. 7, Fig. 8, and Fig. 9, provide evidence that CO-IHHO consistently outperforms HHO for CEC F6, CEC F7, and CEC F8, respectively. In the case of CEC F10, as shown in Fig. 11, the CO-IHHO demonstrates better results compared to the performance of original HHO. These results suggest that CO-IHHO performs well in terms of convergence, best value, average value and standard deviation for most of the tested functions, outperforming the HHO and other MHAs in many cases.

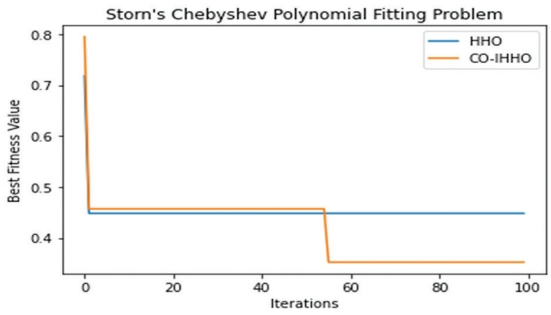

Fig. 2. Convergence curve for CEC F1

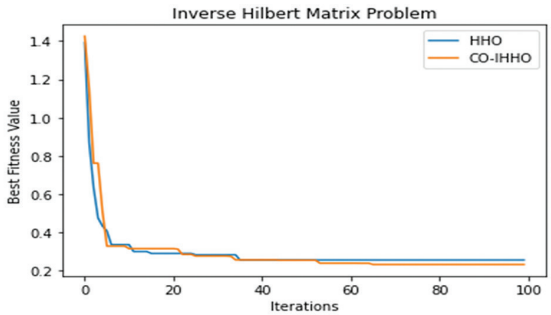

Fig. 3. Convergence curve for CEC F2

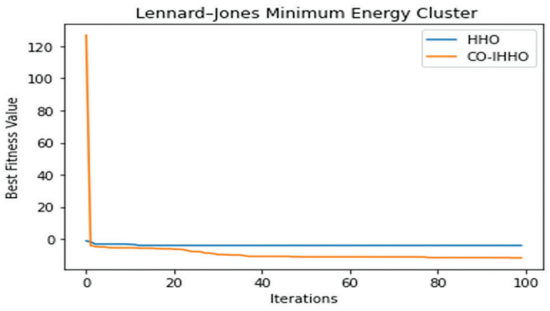

Fig. 4. Convergence curve for CEC F3

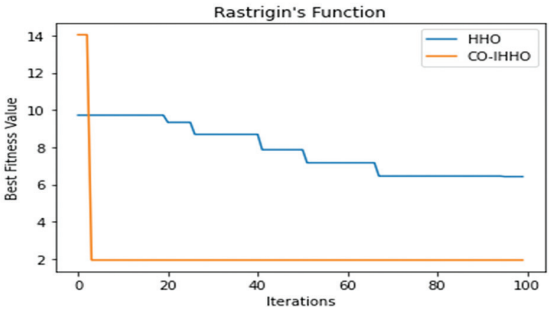

Fig. 5. Convergence curve for CEC F4

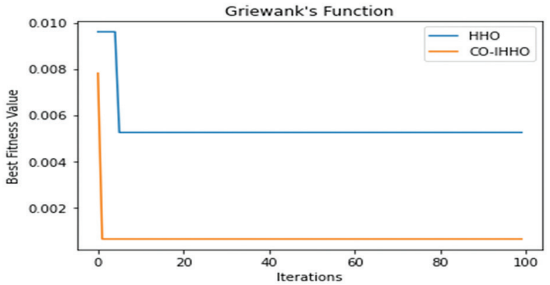

Fig. 6. Convergence curve for CEC F5

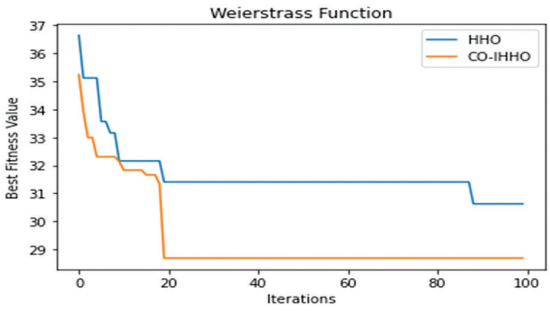

Fig. 7. Convergence curve for CEC F6

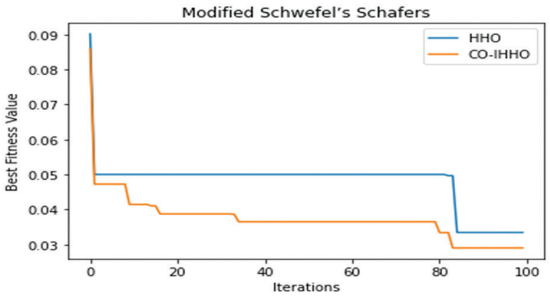

Fig. 8. Convergence curve for CEC F7

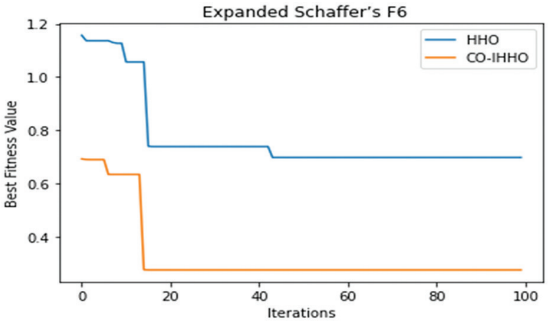

Fig. 9. Convergence curve for CEC F8

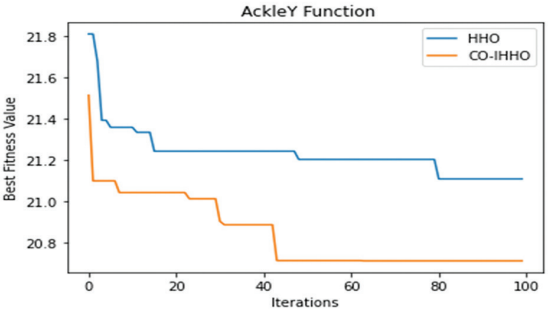

Fig. 11. Convergence curve for CEC F10

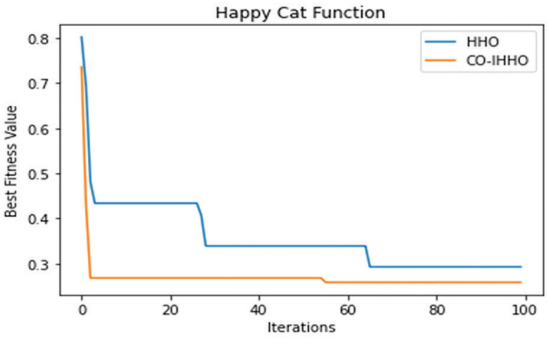

Fig. 10. Convergence curve for CEC F9

Table 2. CEC-06 2019 Benchmark Functions

S. No.	Function Name	Range
F1.	Storn's Chebyshev Polynomial Fitting Problem	[–8192,8192]
F2.	Inverse Hilbert Matrix Problem	[–16384,16384]
F3.	Lennard–Jones Minimum Energy Cluster	[–4,4]
F4.	Rastrigin's Function	[–100,100]
F5.	Griewanck's Function	[–100,100]
F6.	Weierstrass Function	[–100,100]
F7.	Modifed Schwefel'Schafers Functions	[–100,100]
F8.	Expanded Schafer's F6 Function	[–100,100]
F9.	Happy Cat Function	[–100,100]
F10.	Ackley Function	[–100,100]

Table 3. Evaluation Result for CEC-06 2019 Benchmark Functions.

Function	Metric	HHO	CO-IHHO	SSA	GWO	WOA	PSO
F1	Bt.	5.6458408963 636614e+57	4.050347176 1952476e+57	6.180403739 291002e+64	4.405029997 8216904e+57	9.425512550 977296e+64	5.55378945 8284711e+57
	Avg.	1.8001192737 876405e+59	1.449694038 9680123e+58	2.66903878 5340499e+65	4.405049361 695653e+57	2.433135768 798515e+65	5.55379419 7713616e+57
	Sd.	4.829127865 299456e+59	1.353562095 4277987e+57	9.80178069 2383188e+64	1.679703863 5079378e+58	9.085970285 252852e+64	6.70361779 8945432e+57
F2	Bt.	3.9916671 51716069	3.9973917 809429644	3.9973753 40600083	3.9951018 016288677	3.0854374 139651153	3.2763628 543348684
	Avg.	3.99952455 08433546	3.999797 00315296	3.9999312 267444482	3.9993981 76739008	3.9693092 439720545	3.7681114 29602772
	Sd.	0.001525954 7174125503	0.00053572 37019455018	0.000888591 7717980132	0.000968825 4637632288	0.16413144 511627908	1.4020140 157900978
F3	Bt.	-1.06258406 64408444	-0.5581271 752712231	-2.3814881 067413767	-0.8097890 269036659	-5.975875268 3271024e-18	-3.99152881 15300274
	Avg.	-0.021219745 050436466	-0.55812 71 752712231	-1.84750543 29449694	-0.729051427 8242319	-9.032367865 749103e-08	-2.63740153 32542218
	Sd.	0.6594785 017613516	0.0	0.25801905 44014521	0.05075578 2735039646	3.92214594 5550951e-07	0.8153980 301270428
F4	Bt.	4.4268217 33805227	1.00974297 73793607	7.957355181 876517e+55	3387003.90 6655918	82.90815 749102207	3.91957374 33495026e-08
	Avg.	4.4800847 56471913	1.2561787 419150254	1.156520407 7536015e+56	3387003.90 66559565	160.52086 235529387	3.9606395 51880973e-08
	Sd.	0.03837515 684925129	0.04593711 910053741	2.811800251 441804e+55	1.999977733 5956407e-08	77.405704 03960551	1.823048010 1262743e-10
F5	Bt.	0.7918726 025294613	0.08753646 705113927	6.4556002 50692994	2.009503674 5715333e-14	1.321126972 4866526e-05	0.6175348 701316968
	Avg.	0.7918726 032381472	0.5274561 64202598	9.1630051 76576055	3.809119686 3376806e-13	0.06835713 316697926	0.617557 7735425246
	Sd.	1.33732948 4646872e-10	0.2868547 368291001	1.48168001 13561526	6.14194083 8294567e-13	0.08538731 723710603	1.57485623 32045633e-05

F6	Bt.	33.150370660918725	19.99998 0926513693	28.695110 52244359	20.5198762 60083844	19.999980 926513675	19.99998 0926513746
	Avg.	39.33966 281894489	19.99998 092651373	39.621649 84271977	27.878543 87037633	27.942487 589997018	19.99998 0926513743
	Sd.	2.559311 914314841	1.486206 760386614	3.3426729 300780225	5.4533643 67130319	11.793876 127713025	3.55271367 8800501e-15
F7	Bt.	1266.8341 381373614	9.9507944 87333901	31142.4287 94015377	10636.4259 13333696	9.498185 76031255	72.58942 85819002
	Avg.	1267.116 2811961126	9.962486 072328463	68873.145 57159516	10636.425 9133337	16112.375 314667242	73.003582 99789353
	Sd.	0.17258945 10405593	0.002321930 3880354495	19268.516 53842171	3.637978807 091713e-12	22475.7060 53732516	0.6543264 666468622
F8	Bt.	7.1694108 77512084	2.2699551 600740326	6.7234936 74627376	15.493013 232370153	12.678902 550412865	3.9108997 237343455
	Avg.	7.1694108 77515147	4.5774542 11793856	8.3200261 5763271	26.963196 056684094	24.673701 286726512	6.291320 842737323
	Sd.	9.8337508 2019743212	1.6065586 525065239	1.8309652 794532914	5.8955512 06415807	6.3829279 60600941	1.735756 3057153931
F9	Bt.	-1.169939433 4559265e+66	-4.334776244 689166e+69	-2.5e+76	-2.5e+76	-5.7704303 10355887e+72	-2.5e+76
	Avg.	-9.3301679 53480409e+64	-4.33483332 6484094e+68	-5.515302187 0255146e+75	-1.9985345611 38153e+76	-5.02659225 28784734e+73	-2.5e+76
	Sd.	2.862540782 266106e+65	1.30043097 07285165e+69	9.13560888 6381215e+75	3.078090653 543935e+75	1.935246513 4230854e+74	0.0
F10	Bt.	20.52698 445657453	19.99993 579171894	19.999999 993888967	21.423167 174440493	20.050742 57169523	19.999999 993888967
	Avg.	21.060908 317329346	19.999996 789276956	19.99999 999388897	21.453085 896275244	21.1843797 83145335	19.9999 9999388897
	Sd.	0.22800868 237094374	0.01531973 776709719	3.5527136 78800501	0.2596558 779276892	0.32581475 455351727	3.5527136 78800501

*Sd=Standard Deviation (minimum), Bt. = Best value (minimum), and Avg. = Average Value (minimum).

5.3. DATASET DESCRIPTION

The proposed IDS performance is evaluated using the BoT-IoT dataset [31]. This realistic dataset was developed in Cyber Range Lab of UNSW in 2018, and published in 2019 for the IoT environment [34]. It includes both intrusive traffic with different attack categories and normal traffic. The intrusive traffic dataset comprises four primary categories: data exfiltration, DoS, reconnaissance, and DDoS. A summary of the dataset in Table 4 indicates that it is predominantly composed of over 99% intrusive traffic instances, with less than 1% representing normal traffic instances. Moreover, to improve the performance of ML classifiers additional attributes are also added. The BoT-IoT realistic testbed includes a network platform, which includes attacking and normal virtual machines, simulated IoT services, which include five IoT devices simulated using the Red-Node tool, feature extraction using the Agrus tool, and forensic analytics using ML algorithms. The five IoT devices used for simulation include a smart thermostat, smart door, smart fridge, smart lights, and weather station. This dataset has about 72 million labeled records in 74 .csv files with 46 features, out of which 14 are additionally generated from the original feature set. In the process of assessing the performance of the proposed IDS model, a subset amounting to 5% of the entire dataset is taken into account, which corresponds to four .csv files.

Table 4. Dataset Instances.

S.No.	Category	Instances
1.	DDoS	1926624
2.	Normal	477
3.	Information Theft	79
4.	Reconnaissance	91082
5.	DoS	1650260

5.4. EVALUATION METRIC

The evaluation of performance utilizes the following metrics.

Fitness function: The FF is determined using following equation:

$$FF = \propto \times er + \beta_2 \times \left(\frac{Sel\_Fea}{Max\_Fea}\right) \quad (24)$$

where α=0.99 and β_2=1-α

Max_Fea = maximum number of features, Sel_{Fea} = selected features length, and er = 1-Accuracy

Accuracy: This metric defines the identification of data instances from complete traffic data correctly. It is mathematically defined as:

$$Accuracy = \frac{TpS+TnS}{TpS+TnS+FpS+FnS} \quad (25)$$

Precision (P): This metric defines the correctly identifies data instances as positive. It is mathematically defined as:

$$Precision = \frac{TpS}{TpS+FpS} \quad (26)$$

Recall (R): This metric defines the correctly identifies data traffic instances. It is defined mathematically as:

$$Recall = \frac{TpS}{TpS+FnS} \quad (27)$$

F-Score: This metric is calculated as the harmonic mean of recall and precision. Mathematically, it is defined as:

$$F - Score = \left(\frac{R \times P}{R+P}\right) \times 2 \quad (28)$$

Time: The time defines the overall time taken by the algorithm for identifying intrusion.

Features Selected: This indicates the length of features selected by MHA from overall features.

where,

- True Positive (TpS): Data traffic instances correctly classified as positive
- False Positive (FpS): Data traffic instances misclassified as positive
- False Negative (FnS): Number of data traffic instances misclassified as negative
- True Negative (TnS): Data traffic instances correctly classified as negative

5.5. RESULT DISCUSSION AND PERFORMANCE ANALYSIS

Comparison with Other MHA: The result obtained by the proposed work is compared against other wrapper-based MHAs such as ISSA [37], ISCA [38], TMGWO [39], SSA [40], GWO [41], HHO [20], and WOA [42] for FS of the intrusive network traffic for the IoT system. Furthermore, the performance of two ML classifiers: DT, and KNN with the FS methods are compared for identifying intrusive traffic.

Binary Classification: Fig. 12 depicts that CO-IHHO achieves the highest accuracy of 100% using the DT classifier, while ISCA and GWO achieve an accuracy of 99.98% for the BoT-IoT dataset. With KNN as a classifier, CO-IHHO achieves an accuracy of 99.97% while ISSA and SSA achieve an accuracy of 99.95%. Furthermore, Fig. 13 shows that CO-IHHO, ISCA, GWO, and HHO select the lowest number of features i.e. 3 with the DT classifier while ISCA and GWO select the minimum feature length i.e. 2 with KNN classifier for the BoT-IoT dataset.

Fig. 14 depicts that the computation time of CO-IHHO is lowest with both classifiers DT, and KNN in contrast to all other MHA used for FS task. Moreover, TMGWO takes the highest computation time with DT, and KNN classifiers compared to all other MHA used for FS tasks. The convergence curve of CO-IHHO-DT and HHO-DT is shown in Fig. 21 for the BoT-IoT dataset. While the convergence curve of CO-IHHO-KNN and HHO-KNN is shown in Fig. 23. It has been observed that CO-IHHO converges better than HHO for both classifiers.

The overall performance of CO-IHHO-DT is better compared to CO-IHHO-KNN for the accuracy, features selected, and computation time. Furthermore, for binary classification, the performance of CO-IHHO is better among all other MHA used for FS task.

Multiclass Classification: CO-IHHO achieves the highest accuracy of 99.65%, and 98.1% for DT, and KNN classifiers, respectively as depicted in Fig.15. Similarly, Fig. 17, Fig. 18, and Fig. 19. depict that CO-IHHO achieves the highest precision, recall, and F-score each of 100% and 98% with DT and KNN classifier, respectively for the BoT-IoT dataset. As seen in Fig. 16, CO-IHHO with DT classifier selects the minimum number of features i.e. 8. While SSA with DT classifier selects the maximum length of features i.e. 15. Moreover, CO-IHHO with KNN classifier selects the minimum length of features i.e. 5. While HHO selects the maximum length of the feature i.e. 22.

Furthermore, Fig. 20 depicts the CO-IHHO taking the lowest computation time i.e. 31.34 sec. and 42.87 sec. using DT, and KNN classifiers, respectively for the BoT-IoT dataset. TMGWO takes the maximum computation time among all other MHA with DT, and KNN classifier. Convergence curve of CO-IHHO-DT and HHO-DT is shown in Fig. 22. Convergence curve of CO-IHHO-KNN and HHO-KNN for the BoT-IoT dataset is shown in Fig. 24. It has been observed that CO-IHHO converges better than HHO for both classifiers.

The overall performance of CO-IHHO-DT is better compared to CO-IHHO-KNN in terms of number of features selected, accuracy, F-score, precision, computation time, and recall. Furthermore, for multiclass classification, the performance of CO-IHHO is better among all other MHA used for FS task.

Comparison with other recent work: The presented research is also benchmarked against other recent studies that share similarities. These recent investigations employ the BoT-IoT dataset as the basis for evaluating their performance.

Binary Classification: Table 5 depicts the accuracy comparison of CO-IHHO-DT with other recent work for the binary classification. It has been observed that the accuracy of the proposed CO-IHHO-DT is higher compared to the other similar approaches such as BD-PSO-V [43], DNN [44], BGWO-NB [45], AQUa [46], and RSA [47].

Multiclass Classification: Table 6 shows the accuracy comparison of CO-IHHO-DT with other similar approaches for the multiclass classification. The proposed CO-IHHO-DT attains the maximum accuracy of 99.65% compared to other approaches such as GbFS [48], AQUa [44], and RSA [47].

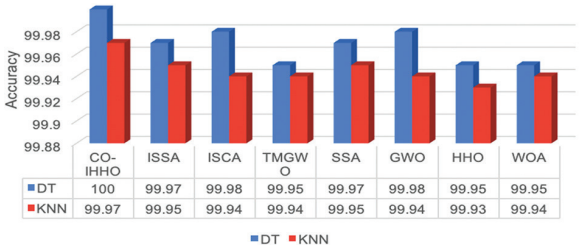

Fig. 12. Binary classification accuracy for BoT-IoT dataset

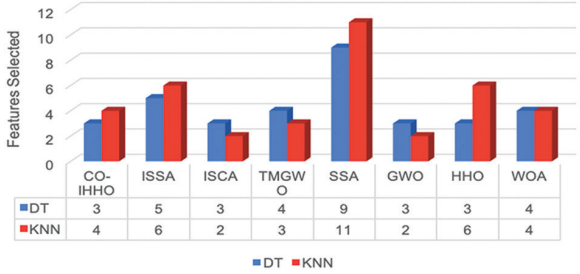

Fig. 13. Binary classification features selected for BoT-IoT dataset

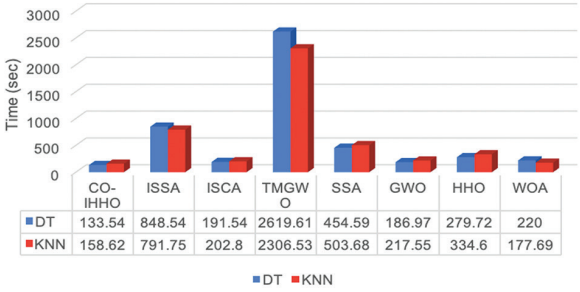

Fig. 14. Binary classification computation time for BoT-IoT dataset

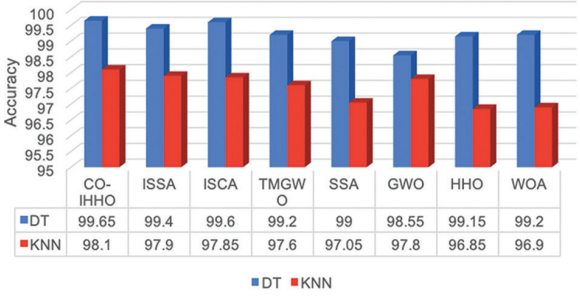

Fig. 15. Multiclass classification accuracy for BoT-IoT dataset

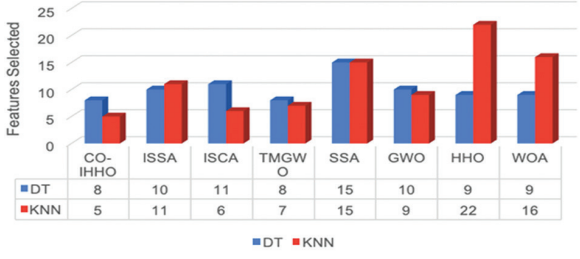

Fig. 16. Multiclass classification features selected for BoT-IoT dataset

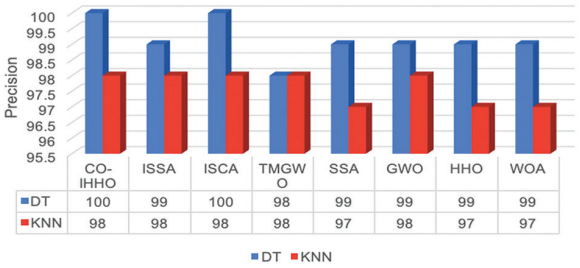

Fig. 17. Multiclass classification precision for BoT-IoT dataset

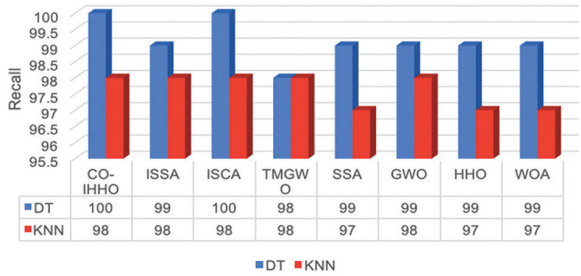

Fig. 18. Multiclass classification recall for BoT-IoT dataset

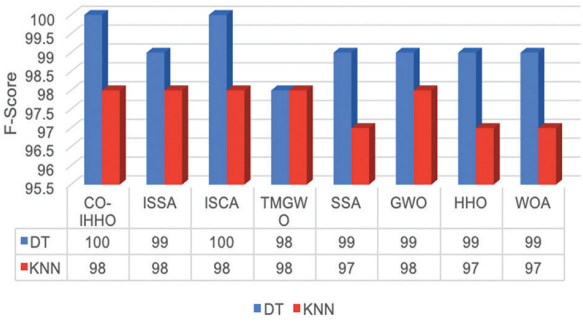

Fig. 19. Multiclass classification F-Score for BoT-IoT dataset

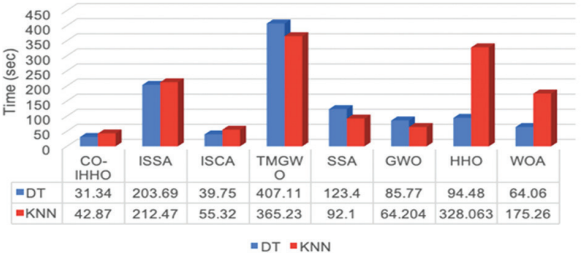

Fig. 20. Multiclass classification computation time for BoT-IoT dataset

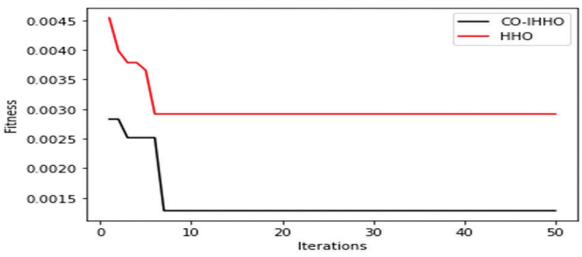

Fig. 21. Binary convergence curve with DT for BoT-IoT dataset

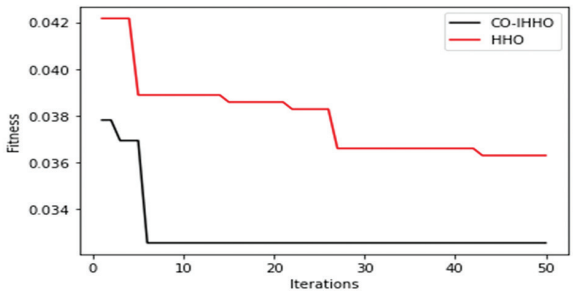

Fig. 22. Multiclass convergence curve with DT for BoT-IoT dataset

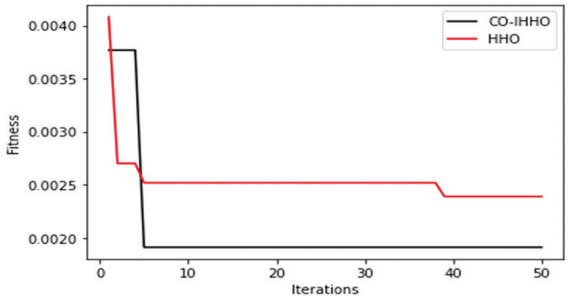

Fig. 23. Binary convergence curve with KNN for BoT-IoT dataset

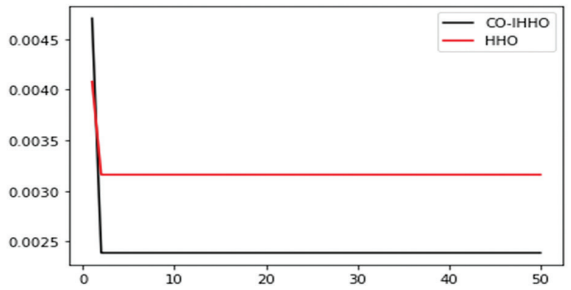

Fig. 24. Multiclass convergence curve with KNN for BoT-IoT dataset

Table 5. Binary classification

S.No.	Method	Accuracy
1.	BD-PSO-V [46]	99.91%
2.	DNN [47]	99.01%
3.	BGWO-NB [48]	99.15%
4.	CO-IHHO-DT (Proposed)	100%
5.	AQUa [44]	99.99%
6.	RSA [45]	99.99%

Table 6. Multiclass classification

S. No.	Method	Accuracy
1.	GbFS [48]	98.90%
2.	CO-IHHO-DT (Proposed)	99.65%
3.	AQUa [46]	98.90%
4.	RSA [47]	98.92%

5.6. REAL WORLD APPLICATIONS

The feature selection using CO-IHHO for IoT based IDS finds application in many real world scenarios, such as:

Smart Homes: Enhancing the security of smart homes by selecting relevant features for intrusion detection in IoT devices such as smart cameras, door sensors, and environmental sensors.

Industrial IoT (IIoT): Securing industrial processes and critical infrastructure by optimizing feature sets for intrusion detection in sensors, controllers, and communication networks.

Healthcare IoT: Protecting patient data and medical devices by employing MHA to select features for intrusion detection in healthcare IoT systems.

Supply Chain Management: Securing IoT-enabled supply chain systems by optimizing features for intrusion detection in RFID tags, sensors, and communication networks.

Energy Management: Improving the security of energy grids and IoT-enabled smart energy systems by selecting features for intrusion detection in smart meters, sensors, and communication networks.

Environmental Monitoring: Securing environmental monitoring systems by optimizing features for intrusion detection in sensors deployed for climate monitoring, pollution detection, and wildlife tracking.

Banking and Finance: Protecting IoT-enabled financial services and ATMs by optimizing intrusion detection features in devices connected to banking networks.

Telecommunications: Protecting IoT devices and networks within the telecommunications sector by optimizing intrusion detection features in routers, switches, and communication equipment.

Military and Defence: Securing military IoT systems by optimizing features for intrusion detection in surveillance equipment, communication networks, and unmanned aerial vehicles (UAVs).

6. CONCLUSION

With the advancement in IoT technology, the everyday utilization of smart IoT devices is increasing briskly. These smart IoT equipment are connected to the internet usually via a wireless network, which makes them vulnerable to several attacks. Hence, security is one of the major issues with the IoT framework. As a result, this paper introduces a metaheuristic-based Intrusion Detection System (IDS) designed for the IoT framework. The feature selection process for network traffic data is accomplished using CO-IHHO, which is an enriched version of the metaheuristic harris hawk optimization algorithm. CO-IHHO achieves a better convergence rate compared to HHO. Furthermore, the result of CO-IHHO is classified using two ML classifiers: DT and KNN. The experimental result shows that CO-IHHO-DT achieves better accuracy compared to CO-IHHO-KNN. The performance of the proposed system is assessed by conducting a comparative analysis with various MHAs used as FS methods. These MHAs include ISSA,

ISCA, TMGWO, SSA, GWO, HHO, and WOA. The result shows that the performance of CO-IHHO-DT attains maximum accuracy of 100%, and minimal computation time for binary classification. CO-IHHO-DT attains the highest accuracy of 99.65% among all other MHA used for FS task of multiclass classification of intrusive traffic. The proposed IDS model is further subjected to comparison with other contemporary approaches. The outcomes indicate that CO-IHHO-DT consistently achieves superior accuracy in both binary and multiclass classification scenarios when compared to these alternatives. The performance of CO-IHHO is further evaluated using CEC-06 2019 benchmark function. In the future, we can utilize deep learning techniques to achieve more refined classification results while addressing real-world applications.

7. REFERENCES

[1] O. Novo, N. Beijar, M. Ocak, J. Kjällman, M. Komu, T. Kauppinen, "Capillary networks - bridging the cellular and IoT worlds", Proceedings of the IEEE 2nd World Forum on Internet of Things, Milan, Italy, 14-16 December 2015.

[2] A. Duraisamy, S. Muthusamy, C. R. R. Robin, "An Optimized Deep Learning Based Security Enhancement and Attack Detection on IoT Using IDS and KH-AES for Smart Cities", Studies in Informatics and Control, Vol. 30, No. 2, 2021, pp. 121-131.

[3] M. F. Elrawy, A. I. Awad, H. F. A. Hamed, "Intrusion detection systems for IoT-based smart environments: a survey", Journal of Cloud Computing: Advances, Systems and Applications, Vol. 7, No. 21, 2018, pp. 1-20.

[4] A. Thakkar, R. Lohiya, "A Review on Machine Learning and Deep Learning Perspectives of IDS for IoT: Recent Updates, Security Issues, and Challenges", Archives of Computational Methods in Engineering, Vol. 28, 2021, pp. 3211–3243.

[5] S. Maza, M. Touahria, "Feature Selection Algorithms in Intrusion Detection System: A Survey", KSII Transactions on Internet and Information Systems, Vol. 12, No. 10, 2018, pp. 5079-5099.

[6] V. R. Balasaraswathi, M. Sugumaran, Y. Hamid, "Feature selection techniques for intrusion detection using non-bio-inspired and bio-inspired optimization algorithms", Journal of Communications and Information Networks, Vol. 2, No. 4, 2017, pp. 107-119.

[7] B. Venkatesh, J. Anuradha, "A review of feature selection and its methods", Cybernetics and Information Technologies, Vol. 19, No. 1, 2019, pp. 3-26.

[8] J. D. Ser, E. Osaba, D. Molina, X.-S. Yang, S. Salcedo-Sanzf, D. Camacho, S. Das, P. N. Suganthan, C. A. C. Coello, F. Herrera, "Bio-inspired Computation: Where We Stand and What's Next", Swarm and Evolutionary Computation, Vol. 48, 2019, pp. 220-250.

[9] Y. Otoum, D. Liu, A. Nayak, "DL-IDS: a deep learning–based intrusion detection framework for securing IoT", Transactions on Emerging Telecommunications Technologies, Vol. 33, No. 3, 2022, p. e3803.

[10] P. K. Keserwani, M. C. Govil, E. S. Pilli, P. Govil, "A smart anomaly based intrusion detection system for the Internet of Things (IoT) network using GWO–PSO–RF model", Journal of Reliable Intelligent Environments, Vol. 7, No. 1, 2021, pp. 3-21.

[11] S. S. Kareem, R. R. Mostafa, F. A. Hashim, H. M. El-Bakry, "An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection", Sensors, Vol. 22, No. 4, 2022, p. 1396.

[12] R. A. Khurma, I. Almomani, I. Aljarah, "IoT Botnet Detection Using Salp Swarm and Ant Lion Hybrid Optimization Model", Symmetry, Vol. 13, No. 8, 2021, p. 1377.

[13] A. Davahli, M. Shamsi, G. Abaei, "Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks", Journal of Ambient Intelligence and Humanized Computing, Vol. 11, No. 11, 2020, pp. 5581-5609.

[14] M. Alweshah, S. Alkhalaileh, M. Beseiso, M. Almiani, S. Abdullah, "Intrusion detection for IoT based on a hybrid shuffled shepherd optimization algorithm", The Journal of Supercomputing, Vol. 78, No. 10, 2022, p. 12278-12309.

[15] H. Haddadpajouh, A. Mohtadi, A. Dehghantanaha, H. Karimipour, X. Lin, K.-K. R. Choo, "A Multikernel and Metaheuristic Feature Selection Approach for IoT Malware Threat Hunting in the Edge Layer", IEEE Internet of Things Journal, Vol. 8, No. 6, 2020, pp. 4540-4547.

[16] Z. Ahmad, A. S. Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, J. J. P. C. Rodrigues, "Anomaly Detection Using Deep Neural Network for IoT Architecture", Applied Sciences, Vol. 11, No. 15, 2021, p. 7050.

[17] R. A. Ramadan, K. Yadav, "A novel hybrid intrusion detection system (IDS) for the detection of internet of things (IoT) network attacks", Annals of Emerging Technologies in Computing, Vol. 4, No. 5, 2020, pp. 61-74.

[18] Y. Li, S. Ghoreishi, A. Issakhov, "Improving the Accuracy of Network Intrusion Detection System in Medical IoT Systems through Butterfy Optimization Algorithm", Wireless Personal Communications, Vol. 126, 2022, pp. 1999-2017.

[19] J. O. Onah, S. M. Abdulhamid, M. Abdullahi, I. H. Hassan, A. Al-Ghusham, "Genetic Algorithm based feature selection and Naïve Bayes for anomaly detection in fog computing environment", Machine Learning with Applications, Vol. 6, 2021, p. 100156.

[20] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, "Harris hawks optimization: Algorithm and applications", Future Generation Computer Systems, Vol. 97, 2019, pp. 849-872.

[21] H. M. Alaboo, D. Alarabiat, L. Abualigah, A. A. Heidari, "Harris hawks optimization: a comprehensive review of recent variants and applications", Neural Computing and Applications, Vol. 33, No. 15, 2021, p. 8939-8980.

[22] A. A. Dehkordi, A. S. Sadiq, S. Mirjalili, K. Z. Ghafoor, "Nonlinear-based Chaotic Harris Hawks Optimizer: Algorithm and Internet of Vehicles application", Applied Soft Computing, Vol. 109, 2021, p. 107574.

[23] A. A. Ewees, M. A. Elaziz, "Performance analysis of Chaotic Multi-Verse Harris Hawks Optimization: A case study on solving engineering problems", Engineering Applications of Artificial Intelligence, Vol. 88, 2020, pp. 1-16.

[24] P. Chossat, M. Golubitsky, "Iterates of maps with symmetry", SIAM Journal on Mathematical Analysis, Vol. 19, No. 6, 1988, pp. 1259-1270.

[25] H. R. Tizhoosh, "Opposition-Based Learning: A New Scheme for Machine Intelligence", Proceedings of the International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce, Vienna, Austria, 28-30 November2005.

[26] R. Hans, H. Kaur, N. Kaur, "Opposition-based Harris Hawks optimization algorithm for feature selection in breast mass classification", Journal of Interdisciplinary Mathematics, Vol. 23, No. 1, 2020, pp. 97-106.

[27] M. Tubishat, N. Idris, L. Shuib, M. A. Abushariah, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection", Expert Systems with Applications, Vol. 145, 2020, p. 113122.

[28] Z. Wang, H. Ding, Z. Yang, B. Li, Z. Guan, L. Bao, "Rank-driven salp swarm algorithm with orthogonal opposition-based learning for global optimization", Applied Intelligence, Vol. 52, 2022, p. 7922–7964.

[29] X. Zhao, F. Yang, Y. Han, Y. Cui, "An Opposition-Based Chaotic Salp Swarm Algorithm for Global Optimization", IEEE Access, Vol. 8, 2020, pp. 36485-36501.

[30] S. Rahnamayan, H. R. Tizhoosh, M. M. Salama, "Opposition versus randomness in soft computing techniques", Applied Soft Computing, Vol. 8, No. 2, 2008, pp. 906-918.

[31] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset", Future Generation Computer Systems, Vol. 100, 2019, pp. 779-796.

[32] P. Agrawal, A. F. Hattan, G. Talari, M. W. Ali, "Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019)", IEEE Access, Vol. 9, 2021, pp. 26766-26791.

[33] H. Jia, C. Lang, D. Oliva, W. Song, X. Peng, "Dynamic Harris Hawks Optimization with Mutation Mechanism for Satellite Image Segmentation", Remote Sensing, Vol. 11, No. 12, 2019, p. 1421.

[34] J. M. Abdullah, T. A. Rashid, "Fitness Dependent Optimizer: Inspired by the Bee Swarming Reproductive Process", IEEE Access, Vol. 7, 2019, pp. 43473-43486.

[35] J. C. Bansal, "Particle Swarm Optimization", Evolutionary and swarm intelligence algorithms, Springer, Berlin, 2019, pp. 11-23.

[36] J. M. Peterson, J. L. Leevy, T. M. Khoshgoftaar, "A Review and Analysis of the Bot-IoT Dataset", Proceedings of the IEEE International Conference on Service-Oriented System Engineering, Oxford, UK, 23-26 August2021.

[37] M. Tubishat, N. Idris, L. Shuib, M. A. Abushariah, S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection", Expert Systems with Applications, Vol. 145, 2020, p. 113122.

[38] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, M. Hariharan, "Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism", Neural Computing and Applications, Vol. 28, 2017, p. 2947-2958.

[39] M. Abdel-Basset, D. El-Shahat, I. El-henawy, V. H. C. Albuquerque, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection", Expert Systems with Applications, Vol. 139, 2020, p. 112824.

[40] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, "Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems", Advances in Engineering Software, Vol. 114, 2017, pp. 163-191.

[41] S. Mirjalili, S. M. Mirjalili, A. Lewis, "Grey Wolf Optimizer", Advances in Engineering Software, Vol. 69, 2014, pp. 46-61.

[42] S. Mirjalili, A. Lewis, "The Whale Optimization Algorithm", Advances in Engineering Software, Vol. 95, 2016, pp. 51-67.

[43] M. Asadi, M. A. J. Jamal, S. Parsa, M. Vahid, "Detecting Botnet by Using Particle Swarm Optimization Algorithm Based on Voting System", Future Generation Computer Systems, Vol. 107, 2020, pp. 95-111.

[44] Z. Ahmad, A. S. Khan, K. Nisar, I. Haider, R. Hassan, M. R. Haque, S. Tarmizi, J. J. P. C. Rodrigues, "Anomaly Detection Using Deep Neural Network for IoT Architecture", Applied Sciences, Vol. 11, No. 15, 2021, p. 7050.

[45] I. M. Nur, E. Ulker, "A Novel Hybrid IoT Based IDS Using Binary Grey Wolf Optimizer (BGWO) and Naive Bayes (NB)", European Journal of Science and Technology, No. 2020, 2020, pp. 279-286.

[46] F. Abdulaziz, A. Dahou, M. A. A. Al-qaness, S. Lu, M. A. Elaziz, "Advanced Feature Extraction and Selection Approach Using Deep Learning and Aquila Optimizer for IoT Intrusion Detection System", Sensors, Vol. 22, No. 1, 2021, p. 140.

[47] A. Dahou, M. A. Elaziz, S. A. Chelloug, M. A. Awadallah, M. A. Al-Betar, M. A. A. Al-qaness, "Intrusion Detection System for IoT Based on Deep Learning and Modified Reptile Search Algorithm", Computational Intelligence and Neuroscience, Vol. 2022, 2022.

[48] Z. Halim, M. N. Yousaf, M. Waqas, M. Sulaiman, G. Abbas, M. Hussain, I. Ahmad, M. Hanif, "An effective genetic algorithm-based feature selection method for intrusion detection systems", Computers & Security, Vol. 110, 2021, p. 102448.