IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received June 21, 2020, accepted June 27, 2020, date of publication July 2, 2020, date of current version July 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3006473

Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection

RAMI SIHWAIL[1], KHAIRUDDIN OMAR[1], KHAIRUL AKRAM ZAINOL ARIFFIN[1], (Member, IEEE), AND MOHAMMAD TUBISHAT[2]

[1]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[2]School of Technology and Computing, Asia Pacific University of Technology and Innovation, Kuala Lumpur 57000, Malaysia

Corresponding author: Rami Sihwail (p91206@siswa.ukm.edu.my)

This work was supported in part by the Universiti Kebangsaan Malaysia (UKM) under Grant GUP-2019-062 and Grant GP-2019-K005539, and in part by the Ministry of Education Malaysia under Grant FRGS/1/2018/ICT04/UKM/02/3.

ABSTRACT The rapid increase in data volume and features dimensionality have a negative influence on machine learning and many other fields, such as decreasing classification accuracy and increasing computational cost. Feature selection technique has a critical role as a preprocessing step in reducing these issues. It works by eliminating the features that may negatively influence the classifiers' performance, such as irrelevant, redundant and less informative features. This paper aims to introduce an improved Harris hawks optimization (IHHO) by utilizing elite opposite-based learning and proposing a new search mechanism. Harris hawks optimization (HHO) is a novel metaheuristic general-purpose algorithm recently introduced to solve continuous search problems. Compared to conventional HHO, the proposed IHHO can avoid trapping in local optima and has an enhanced search mechanism, relying on mutation, mutation neighborhood search, and rollback strategies to raise the search capabilities. Moreover, it improves population diversity, computational accuracy, and accelerates convergence rate. To evaluate the performance of IHHO, we conducted a series of experiments on twenty benchmark datasets collected from the UCI repository and the scikit-feature project. The datasets represent different levels of feature dimensionality, such as low, moderate, and high. Further, four criteria were adopted to determine the superiority of IHHO: classification accuracy, fitness value, number of selected features, and statistical tests. Furthermore, a comparison between IHHO and other well-known algorithms such as Generic algorithm (GA), Grasshopper Optimization Algorithm (GOA), Particle Swarm Optimization (PSO), Ant Lion Optimizer (ALO), Whale Optimization Algorithm (WOA), Butterfly Optimization Algorithm (BOA) and Slime Mould Algorithm (SMA) was performed. The experimental results have confirmed the dominance of IHHO over the other optimization algorithms in different aspects, such as accuracy, fitness value, and feature selection.

INDEX TERMS Harris Hawks optimization, optimization, feature selection, elite opposite based-learning, mutation, mutation neighborhood search.

I. INTRODUCTION

The growth in data volume and features dimensionality in the last few years has caused severe difficulties to researchers in many fields such as big data, data mining, data science and other fields. It is well-known that the analysis of high dimensional data suffers from problems of dimensionality, sparsity, and complexity [1]. Also, high dimensional data have a negative influence on machine learning classifiers, such as decreasing classification accuracy and increasing computational cost. The main reason for these issues is because the domain of features has expanded from tens to thousands in the last few years [2]. Therefore, performing feature selection technique is mandatory to reduce the number of features. The feature selection technique works by removing noisy and irrelative features from the dataset. Therefore, to deal with high dimensional features in machine learning, it is common to apply feature selection to select the most informative subset

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

of features [3]. Feature selection has been used in many other fields such as security systems, sentiment analysis, disease detection and classification, data mining and text classification, and many other fields [4].

Besides, feature selection aims to choose the best subset of features that can improve the learning model in terms of performance, simplicity, and speed [5]. Further, feature selection techniques are classified into two categories: filter-based and wrapper-based. In filter-based, features are evaluated independently from the classifier. It relies on the information content to weight the features and select the most informative subset [6]. This category includes Chi-Square Test, Variance Threshold, Pearson Correlation, Information Gain (IG), Mutual Information (MI), and Fisher Score. Although filter-based techniques are not optimized to match particular classifiers, they are fast and usually used as a preprocessing step for other feature selection techniques [7]. Unlike filter-based, wrapper-based techniques utilize machine learning classifier to train the subsets of features and choose the best subset that can retain the highest possible accuracy based on the used optimization algorithm. Recursive Feature Elimination, Genetic Algorithms, and Sequential Feature Selection are some examples of wrapper methods.

Moreover, there are several search techniques to allocate the best subset of features, including greedy search, random search, and meta-heuristic search. Greedy search works by evaluating all possible combinations of features in the dataset. Therefore, it is time-consuming. In contrast, random search follows the random strategy in exploring to find the best subset of features. However, it could be easily trapped in a local optimal solution [8]. On the other hand, meta-heuristic methods (wrapper-based) explore the search space by imitating physical or biological phenomena or even animal's behaviors in nature. Meta-heuristic search strategies have proved their success in dealing with significant scale problems in different areas such as data mining, data sciences, and machine learning [9].

One of the main categories of meta-heuristic algorithms is the nature-inspired algorithm (NIA), which is inspired by natural phenomena. NIA consists of two main subcategories: swarm intelligence (SI) and evolutionary algorithm (EA). EA algorithms, like genetic algorithm (GA), are influenced by natural selection and evolution, such as elitism, mutation and crossover. In contrast, SI algorithms mimic natural phenomena like the living style of birds, ants, whales and butterflies. The following are some examples of SI algorithms: GA [10], Grasshopper Optimization Algorithm (GOA) [11], Particle Swarm Optimization (PSO) [12], Ant Lion Optimizer (ALO) [13], Whale Optimization Algorithm (WOA) [14], Butterfly Optimization Algorithm (BOA) [15], Slime Mould Algorithm (SMA) [16] and Harris Hawk Optimization (HHO) [17]. Due to the stochastic nature of SI algorithms, it is commonly utilized to solve large space and complex optimization problems [18].

All meta-heuristic methods consist of two phases: exploration (diversification) and exploitation (intensification). Therefore, they explore the whole search space looking for a promising area. Hence, the selected promising area is exploited to find the best solution. Generally, meta-heuristic algorithms were showing excellent results when dealing with feature selection [19]. The more significant number of features makes more challenges for feature selection techniques to find the best solution. The number of possible solutions increases exponentially with the number of features in the search domain. For example, when the number of features is n features in the dataset, there will be 2^n combinations of features (solutions). Hence, the use of standard exhaustive search is time-consuming and inapplicable. However, meta-heuristic algorithms can solve these issues and return the optimal or close to the optimal solution within an acceptable time.

HHO is considered one of the most recent meta-heuristic proposed by Heidari *et al.* [17]. It is a fast, powerful, and high-performance population-based optimization algorithm. The algorithm mimics the style of Harris Hawk birds in searching and chasing the prey in nature. The prey, which is treated as the best solution, is symbolized by a rabbit in the algorithm. Based on the authors of HHO, the algorithm outperformed other well-known algorithms, including PSO, GA, GOA, ALO, WOA, BOA and SMA. Further, the algorithm was tested on 29 benchmark problems and other tasks that represent real-world engineering tasks. The experiments had shown very competitive results [17].

However, like most of the optimization algorithms, HHO has some limitations. First, it has limited solution diversity generated by its random function at the initialization phase. Second, there is no guarantee that the Harris hawks (solutions) will not end up in local optima instead of the optimal solution. Third, HHO depends on the rabbit energy to specify the type of search it will do. The rabbit energy approximately starts at 2 and gradually reduced to 0 with each iteration. Hence, the global search is performed in the first half of the iterations, when the rabbit energy is greater than 1 ($E \geq 1$). Accordingly, HHO does not perform a global search in the second half of iterations, although the currently selected area may not be the optimal one [20]. Premature convergence is another issue where the Harris hawks (population) converge to a local solution instead of the global solution [21].

Therefore, this paper introduced two improvement strategies to enhance the feature selection abilities of the standard HHO. As for the former, the Elite Opposition-Based Learning (EOBL) strategy is applied to improve the population diversity and the exploration phase of HHO. EOBL enhances the distribution of the initialized solutions in the search space. Unlike the random distribution used by the standard HHO, the use of EOBL strategy improves the computational accuracy of the algorithm and accelerates its convergence rate. As for the latter, a dynamic search is introduced based on the following three strategies: mutation technique, mutation neighborhood search, and rollback. The purpose of the dynamic search is to enhance the capabilities of both global and local searches of HHO. The proposed search strategies look for alternative promising areas in the search space to avoid the Harris hawks from being trapped in local optima.

Therefore, based on the reasons mentioned above, this research is motivated to improve the original HHO to suit the feature selection problem. The main contributions of this paper are summarized in the following:

1) Proposed an improved version of the conventional HHO, called improved Harris hawks optimization (IHHO), for feature selection in wrapper mode that can overcome its limitations.
2) EOBL strategy is utilized to improve the population diversity of HHO in the initialization phase and accelerate its convergence rate.
3) A new three search strategies mechanism is proposed to avoid the Harris hawks from trapping in local optima by exploring new promising areas in the search space.
4) The performance of IHHO is evaluated by comparing its accuracy, fitness value, and the number of selected features with a number of well-known optimization algorithms, including HHO, GA, GOA, PSO, ALO, WOA, BOA and SMA. IHHO outperformed other algorithms on 20 benchmark datasets representing various types of feature dimensionality: low, moderate, and high.

The rest of the paper is organized as follows: Section II presents the related work. Section III introduces some state-of-the-art about HHO, EOBL, and the three search strategies. The details of the improved IHHO are described in Section IV. Section V presents the details of the experiments and the benchmark datasets. In Section VI, discussion and analysis of experimental results are presented. Finally, Section VII concludes the paper.

II. RELATED WORK

EOBL was introduced in 2012 by Zhou *et al.* to enhance the quality of Opposition-Based Learning (OBL). The main idea of EOBL is to generate more promising solutions by evaluating the opposite solutions of elite solutions. The opposite solution is more likely to locate in a better position in which the global optimum is located [22]. EOBL strategy has been used to improve meta-heuristic optimization algorithms. In [23], the authors applied EOBL in the initialization phase of water wave optimization (WWO) to enhance convergence speed and precision calculation.

Similarly, in work by [24], EOBL is used to improve the computational accuracy and convergence speed rate of the spider optimization algorithm (SOA). Further, in research by [25], they applied the EOBL strategy to enhance the balance between exploration and exploitation ability in the Cuckoo search algorithm (CSA). Furthermore, the authors in [26] utilized EOBL to overcome the limitations of grey wolf optimizer (GWO) such as poor population diversity and slow convergence rate. Also, Zhou *et al.* applied EOBL to enhance the diversity of population and greedy strategy to enhance the exploitation ability of flower pollination algorithm (FPA) [27]. Likewise, in the research conducted by [28], the authors improved the whale optimization algorithm (WOA) and used it for feature selection. They improved the exploration phase by applying the EOBL strategy and introduced an advanced local search to improve the algorithm's exploitation phase.

Several kinds of research have applied standard HHO or modified it to solve general and specific problems. For example, in [29], the authors combined HHO with differential evolution (DE) for color image multilevel thresholding segmentation. Similarly, in [20], they modified the search mechanism of HHO for satellite image segmentation. Likewise, the authors in [30], applied HHO and improved adaptive generalized Gaussian distribution threshold to remove possible noise from the satellite image. Also, in work conducted by [31], the exploration phase of HHO has updated using a sine-cosine algorithm to optimize various engineering design problems.

Further, in work by [21], the authors improved the exploration phase of HHO for parameter identification of simulated annealing (SA) single-diode solar cell models. Additionally, In [32], the authors proposed to solve the job scheduling problem in cloud computing using modified HHO with a simulated annealing algorithm. Moreover, in their work [33], the authors predicted the slope stability by utilizing HHO and k-fold cross-validation. Yin *et al.* proposed a new control parameter method to HHO and random OBL to construct DNA storage [18]. Recently, the authors in [34] introduced a four strategies technique, including the OBL strategy, to improve HHO transition rules.

The standard HHO was designed for continuous search space, and it needs to be modified to match binary feature selection. For this purpose, the authors in [19] proposed a binary quadratic algorithm for feature selection. The approach integrates the two transfer functions, S-shaped and V-shaped, to convert the continuous HHO into binaries. In their work, the authors engaged twenty datasets collected from the UCI machine learning repository to test the performance of the approach. The experimental results showed that the proposed algorithm could maintain high classification accuracy based on selecting a relatively small number of features.

Similarly, the authors in [8] introduced binary HHO to select the best subset of features while retaining the highest possible accuracy. The technique was applied on nine medical datasets representing high dimensional features and a low number of samples, using 5-neighbors KNN classifier. The experimental result has ranked HHO to be the first over the other compared algorithms. Another work performed by Menesy *et al.* applied ten chaotic functions to improve the HHO search capabilities and avoid falling in local optima. The authors utilized the algorithm to extract parameters of fuel cell stacks [35]. Recently, in research conducted by [36], the authors applied the OBL strategy to improve the technique used by HHO in selecting the optimal size of the feature subset with maximum accuracy.

III. STATE-OF-THE-ART

A. HARRIS HAWKS OPTIMIZATION

Harris Hawks Optimization (HHO) is a novel nature-inspired, gradient-free, and population-based optimization algorithm that imitates the chasing style of Harris Hawks' birds. HHO was introduced recently by Heidari *et al.* in 2019 [17]. The algorithm follows the attacking behaviors of Harris hawks on the prey in nature, such as preaching, predation, and surprise pounce strategies. Like other meta-heuristic algorithms, HHO includes two main phases: exploration and exploitation, which are shown in Figure 1. However, HHO has two stages for exploration and four for exploitation, which described in detail as follows.

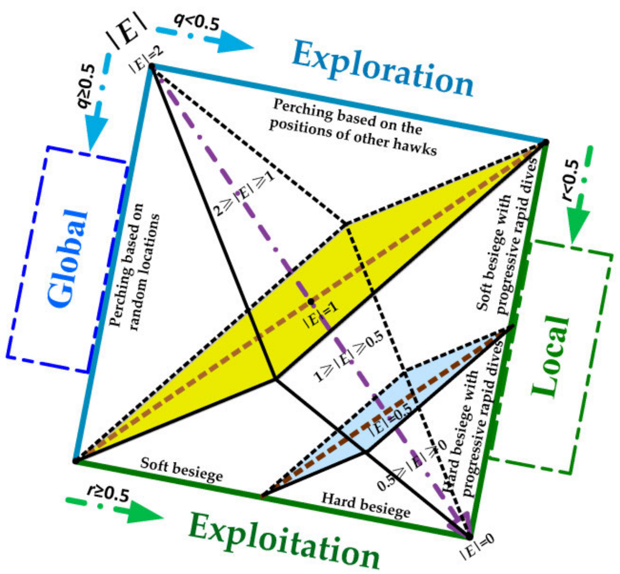

FIGURE 1. Exploration and exploitation phases of Harris hawks optimization (HHO).

Initialization Phase: In this phase, the objective function and its solution space are defined. In addition, the values for the parameters are assigned, and the initial population is created.

Exploration Phase: It is the phase where Harris Hawks search for the prey (the rabbit). The hawks have compelling eyes that can help them to detect and track the prey, but it is sometimes difficult to see the prey. In this case, the hawks wait and monitor the site hoping to observe the prey. Practically, in each iteration, all Harris hawks are the candidate solutions, and the fitness value is calculated for each of them based on the intended prey. After that, the Harris hawks may wait in some positions to detect the prey based on the following equation:

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1 \left| X_{rand}(t) - 2r_2 X(t) \right| & q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t)) - r_3(LB + r_4(UB - LB)) & q \leq 0.5, \end{cases} \tag{1}$$

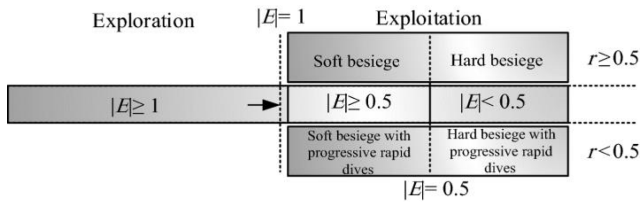

FIGURE 2. The exploration phase and the four types of exploitation phases in HHO. Note that transferring between the different phases depends on the rabbit escaping energy $|E|$.

where $X(t+1)$ is the position of hawks in the next iteration t, $X_{rabbit}(t)$ is the rabbit position, $X(t)$ is the current position vector of the hawks, $X_m(t)$ refers to the average position of the current population of hawks. The variables r_1, r_2, r_3, r_4, and q (wait) are random numbers over the interval [0, 1], and LB and UB represent the upper and lower bounds of the problem variables. HHO uses a straightforward way to calculate the average position of hawks $X_m(t)$ using the following equation:

$$X_m(t) = \frac{1}{N} \sum_{i=1}^{N} X_i(t), \tag{2}$$

where $X_i(t)$ refers to the position of the hawks in iteration t, and N represents the total number of hawks.

Transition From Exploration to Exploitation: It is critical to the performance of meta-heuristic algorithms to maintain the right balance between exploration and exploitation. In HHO, shifting between the exploration phase and exploitation phase, and between different exploitations depend on the prey escaping energy (E). HHO assumes that the energy of the rabbit is reducing during escaping from the hawks, which can be calculated as follows:

$$E = 2E_0 \left(1 - \frac{t}{T}\right), \tag{3}$$

where E is escaping energy, E_0 is the initial state of energy which its value randomly changes over the interval (-1, 1), and T is the maximum number of iterations. When the escaping energy of the rabbit $|E| \geq 1$, HHO redirects the hawks to explore different regions searching for the rabbit (exploration phase). However, when its energy is reduced $|E| < 1$, the hawks search the neighborhood for the solution during the exploitation phase.

Exploitation Phase: In this phase, Harris hawks attack the prey based on the position detected in the previous phase. However, the rabbit always attempts to escape, and the hawks follow the chasing strategy. Hence, HHO is designed based on four possible strategies of attacking techniques. Two variables indicate which strategy will be performed, r and $|E|$. While $|E|$ is the escaping energy of the rabbit, r refers to the probability of escaping, where $r < 0.5$ indicates a higher chance for the rabbit to escape successfully and $r \geq 0.5$ for failure to escape. The exploration phase and the four types of exploitation are illustrated in Figure 2. In the following, a summary of each attacking strategy is presented:

Soft Besiege: When $r \geq 0.5$ and $|E| \geq 0.5$, the rabbit still has some energy to escape. Hence, the hawks softly encircle around the rabbit to make it lose more energy before performing the surprise pounce. Soft besiege can be mathematically formulated as follows:

$$X(t+1) = \Delta X(t) - E\left|JX_{rabbit} - X(t)\right|, \quad (4)$$
$$\Delta X(t) = X_{rabbit} - X(t), \quad (5)$$

where $\Delta X(t)$ represents the difference between the rabbit location and the Harris Hawk location in iteration t. J denotes the random jump strength of the rabbit with the value of $J = 2(1 - r_5)$ as r_5 is a random variable.

Hard Besiege: When $r \geq 0.5$ and $|E| < 0.5$, then the rabbit is so tired with low escaping opportunity. In this situation, the hawk hardly encircles around the rabbit to perform the final surprise pounce. The next location of the hawk is updated using the following equation.

$$X(t+1) = X_{rabbit}(t) - E\left|\Delta X(t)\right|, \quad (6)$$

Soft Besiege With Progressive Rapid Dives: In the situation of soft besiege, |E| $\geq$ 0.5 but r $<$ 0.5, the prey still has the energy to escape from the attack. The hawk performs smart zigzag movement around the rabbit and several dives before the surprised pounce. This stage is considered more intelligent soft besiege, in which the position of the hawks is updated into two steps. In the first step, the hawks attempt to approach the rabbit by evaluating the next move based on the following equation:

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)|, \quad (7)$$

Based on the comparison between the last dive and the possible evaluated result, a decision is made to dive or not. If it is not, the hawks start performing rapid and irregular dive, based on the levy flight concept, when approaching the rabbit. Hence, this step can be formulated as:

$$Z = Y + S \times LF(dim), \quad (8)$$

where *dim* is the dimension of the optimization problem to be solved, S is a randomly generated vector of size $1 \times dim$, and LF denotes the Levy flight function that is calculated as follows:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}}, \quad \sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(\frac{\beta-1}{2})}}\right)^{\frac{1}{\beta}}, \quad (9)$$

where u and v are random values over the interval [0,1], and β is a constant equal to 1.5.

Therefore, the equation for updating the Harris hawks positions in the soft besiege phase can be mathematically formulated as follows:

$$X(t+1) = \begin{cases} Y & if\ F(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)), \end{cases} \quad (10)$$

where F is the calculated fitness function for Y and Z.

Algorithm 1 Standard HHO Algorithm

Input: N: the population size, T: the maximum number of iterations.
Output: The rabbit location and its fitness value
Initialize the population randomly $X_i (i = 1, 2, \ldots, N)$
While (maximum iteration not reached ($t < T$)) **do**
  Check the location boundaries and Evaluate the fitness of Harris hawks locations
  Set the rabbit best location to X_{rabbit}
  For (each hawk $X_i (i = 1\ to\ N)$) **do**
    Update the rabbit initial energy E_0
    Update the rabbit energy E using Equation (3)
    If ($|E| \geq 1$) **then** %**Exploration phase**
      Update the hawk's position using Equation (1)
    if ($|E| < 1$) **then** %**Exploitation phase**
      If ($r \geq 0.5$ *and* $|E| \geq 0.5$) **then** %*Soft besiege*
        Update the hawk's positions using Equation (4)
      Else if ($r \geq 0.5$ *and* $|E| < 0.5$) **then** %Hard besiege
        Update the hawk's positions using Equation (6)
      Else if ($r < 0.5$ *and* $|E| \geq 0.5$) **then** %*Soft besiege with progressive rapid dives*
        Update the hawk's positions using Equation (10)
      Else if ($r < 0.5$ *and* $|E| < 0.5$) **then** %Hard besiege with progressive rapid dives
        Update the hawk's positions using Equation (11)
Return the rabbit location (X_{rabbit})

Hard Besiege Progressive Rapid Dives: When |E| $<$ 0.5 and r $<$ 0.5, the prey does not have enough energy to escape. In this case, the Harris hawks attempt to decrease the distance by getting closer to the rabbit with rapid dives before performing surprise pounce to catch the rabbit. The movement of the hawks in the case of hard besiege is formulated as follows:

$$X(t+1) = \begin{cases} Y & if\ F(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)), \end{cases} \quad (11)$$

where

$$Y = X_{rabbit}(t) - E\left|JX_{rabbit}(t) - X_m(t)\right|, \quad (12)$$
$$Z = Y + S \times LF(dim), \quad (13)$$

Finally, calculation of fitness function involves classification error rate and a minimum number of selected features, which can be mathematically formulated as:

$$\downarrow Fitness = \alpha\gamma(R) + \beta\frac{|F|}{|N|}, \quad (14)$$

where $\gamma(R)$ is the classifier error rate, $|F|$ is the number of selected features, and $|N|$ represents the total number of features. $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$ are two factors. Furthermore, the pseudocode of the original HHO algorithm is presented in Algorithm 1.

B. ELITE OPPOSITION-BASED LEARNING (EOBL)

EOBL is an enhanced version of the OBL technique, which was proposed by Tizhoosh in 2005 [37]. OBL is a machine intelligence strategy that aims to enhance the performance of meta-heuristic optimization algorithms. Its strategy is based on finding a more effective solution between the current individuals, which is generally initialized randomly by the optimization algorithm, and its corresponding opposite solution. The fitness value is calculated for both solutions, and the best one is selected to proceed with the next iteration. However, it has been proved that OBL gives more opportunity to get closer to the optimal global solution for an objective function, enhancing the performance of optimization algorithms [37]. Therefore, the OBL technique has been successfully applied to enhance meta-heuristic search algorithms such as improving cuckoo optimization in [38], evolution algorithm in [39], PSO in [40], GOA in [41], WOA in [28], and salp swarm algorithm (SSA) for feature selection in [42]. OBL can be mathematically modeled as follows: Let $x = (x_1, x_2, \ldots, x_D)$ is a point in current population, D is the problem dimensional space and $x \in [a_i, b_i]$, $i = 1, 2, \ldots, D$. then, the opposition point $\breve{x} = (\breve{x}_1, \breve{x}_2, \ldots, \breve{x}_D)$ is defined as the following equation:

$$\breve{x}_i = a_i + b_i - x_i \tag{15}$$

EOBL strategy relies on the elite individual to lead the population towards the global solution. The elite individual most probably has more useful information than other individuals. Practically, EOBL works based on the elite individual from the current population to generate the complementary opposites of the current population located within the search boundaries. The population is guided then by the elite individual to reach eventually to the promising region, in which the global optimum may be found. Therefore, applying the EOBL technique will enhance the population diversity and improve the global search of the optimization algorithm [25]. As mentioned earlier in the literature, EOBL has been utilized to improve many optimization algorithms.

In this paper, the EOBL technique is used to improve the global search ability of HHO. The opposition point is defined as follows: for the individual $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,D})$ in the current population $X_e = (x_{e,1}, x_{e,2}, \ldots, x_{e,D})$, then the elite opposite point $\breve{X}_i = (\breve{x}_{i,1}, \breve{x}_{i,2}, \ldots, \breve{x}_{i,D})$ can be mathematically modeled as:

$$\breve{x}_{i,j} = S \times (da_j + db_j) - x_{i,j} \tag{16}$$

where $x \in [a_i, b_i]$, $S \in U(0, 1)$, S is a generalized factor. da_j and db_j are dynamic boundaries, which can be defined as:

$$da_j = min(x_{i,j}), \quad db_j = max(x_{i,j}) \tag{17}$$

However, the corresponding opposite can exceed the search boundary $[a_i, b_i]$. To solve this matter, the transformed individual is assigned a random value within $[a_i, b_i]$ as follows:

$$\breve{x}_{i,j} = rand(a_j, b_j), \quad if\ \breve{x}_{i,j} < a_j \parallel \breve{x}_{i,j} > b_j \tag{18}$$

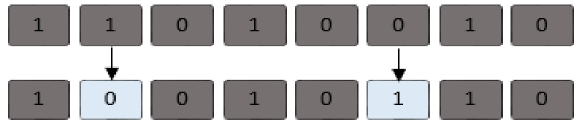

FIGURE 3. An example of bit string mutation, in which the 2nd and 6th features are flipped.

EOBL can enhance the search space by producing a new population from inverse solutions. Consequently, it can also improve the global search ability of the optimization algorithm by enhancing its population diversity.

C. THE THREE SEARCH STRATEGIES

HHO relies on the rabbit energy $|E|$ to shift from exploration to exploitation and to choose the current type of exploitation. It also uses the rabbit energy to prevent the hawks from falling in local optima. However, the rabbit escaping energy may rapidly change its convergence towards the optimal solution, which may cause the hawks to trapped in local optima [21]. In this subsection, we explain the proposed three search strategies (TSS) to enhance both of the global and local search mechanisms of the HHO algorithm. Besides, solving, to some extent, the problem of being trapped into local optima.

1) MUTATION

The purpose of mutation in the Genetic algorithm (GA) is to enhance the diversity into the sampled population. Mutation operators are used for preventing the population of chromosomes from falling in local optimum by preventing them from becoming too similar to each other.

There are various types of mutations based on the adopted technique. However, in this method, we utilized bit string mutation, which is performed by flipping features at random positions. For the solution $X = (x_1, x_2, \ldots, x_D)$, then the bit string mutation can be mathematically modeled as:

$$M(y) = |1 - X(y)| \tag{19}$$

where M is the solution after applying bit string mutation, $y = 1, 2, \ldots, D$ is a matrix of randomly selected positions (features) to be flipped in solution X. In solution X, for example, the second and sixth features are flipped, as illustrated in Figure 3.

Based on several numbers of trial and error experiments, we selected the mutation size randomly between 10% and 50% in the exploration phase, and from 1% to 9% in the exploitation phase. HHO relies on the rabbit escaping energy $|E|$ to shift from exploration to exploitation phase. The value of $|E|$ indicates the selection of the exploration phase when it is greater than 1, and the exploitation phase when it is less than 1. Based on Equation (3), the value of $|E|$ depends on E_0 and t. Hence, the value of $|E|$ is fluctuated between [0, 2] in the first half of iterations and between [0, 1] in the second half. Therefore, HHO is able to perform exploration and exploitation in the first half of iterations. However, it can only perform exploitation in the second half, as illustrated in Figure 4.

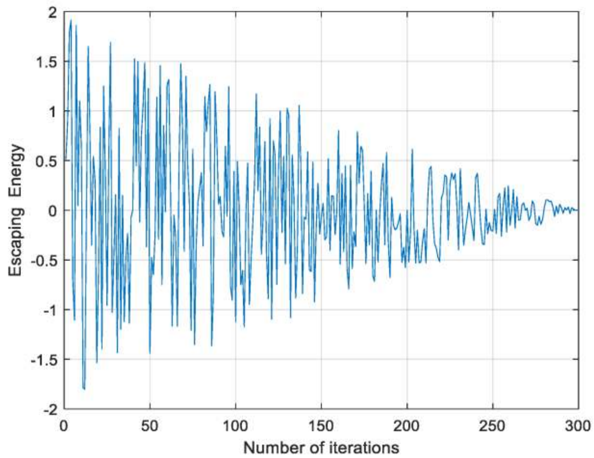

FIGURE 4. **Escaping rabbit energy $|E|$ indicates the HHO status, either exploration or exploitation.**

In IHHO, we adopted the $|E|$ strategy to select the size of features to be flipped. Basically, in the exploration phase, more features of the current best location need to be flipped to enhance the ability of the global search. However, in the exploitation phase, the hawks are assumed to be close to the rabbit location (optimal solution). Therefore, few features are flipped to improve the local search. The mutation size is modeled as the following:

$$Mutation_{size} = \begin{cases} Number\ of\ Features * \dfrac{10 * rand\,[1,5]}{100} & if\ \ |E| \geq 1 \\ Number\ of\ Features * \dfrac{rand\,[1,9]}{100} & if\ \ |E| < 1, \end{cases} \tag{20}$$

2) MUTATION NEIGHBORHOOD SEARCH (MNS)

The idea of the neighbor search was used by Das *et al.* in 2009 to balance between exploration and exploitation phase in DE [43]. The purpose of the neighbor search is to search the small area surrounding the current best solution rather than the entire population. In this work, we propose the mutation neighborhood search (MNS). The use of the MSN search is controlled by updating the current best solution caused by the mutation strategy. In other words, MNS is applied whenever there is a change in the location of the current best solution (the rabbit location) by the mutation. Hence, the fitness value is calculated each time after applying mutation on the current best location. If the fitness of the new position is better than the current position, then the current best solution is replaced by the new mutated solution, and the neighborhood search is performed.

Basically, MNS considers the two adjacent features to the flipped feature. The feature to the right side is flipped, and the fitness values for the two solutions are compared. After that, the same procedure is applied to the left-side feature. Therefore, two more solutions are generated, and the best one is considered as the best solution. Moreover, the ring MNS is applied, in which the last feature is connected to the first one to make it possible for them to have adjacent neighbors from both sides. The ring MNS strategy is illustrated in Figure 5.

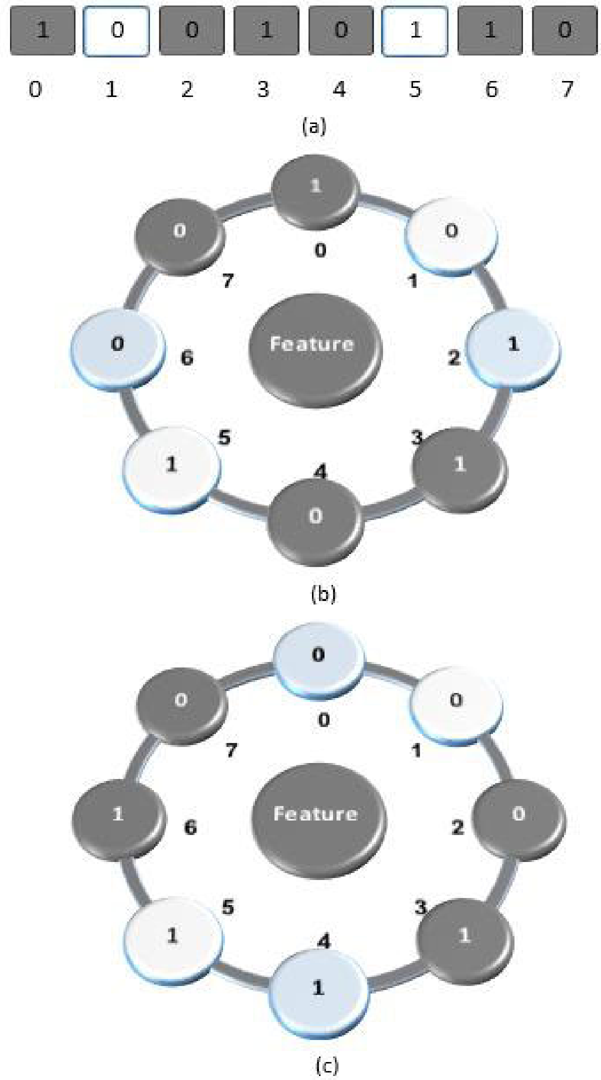

FIGURE 5. **An example of the mutation neighborhood search (MNS). (a) Mutation. (b) MNS applied to the right neighbors. (c) MNS applied to the left neighbors.**

3) ROLLBACK STRATEGY

A mutation is a robust strategy that can effectively enhance the global and local search. However, it may change the direction of the optimization algorithm and lead to local optima. In general, local optima is one of the most common problems for all optimization algorithms. Hence, the rollback strategy is followed in our proposed IHHO. Rollback strategy is a simple yet effective technique. The new mutated solution is not immediately considered as the current best solution, although it has a better fitness value compared to the current solution. It is temporarily saved as a potential solution. After the next iteration, the current best solution may be changed if the HHO algorithm has found a better solution. Consequently, the potential solution is compared to the current best solution, and this time the better solution is assigned as the best current solution. In other words, TSS accepts the new position generated by mutation or MSN, if it preserves the best fitness value for two subsequent iterations. The TSS strategies: mutation, MNS, and rollback are illustrated in Figure 6.

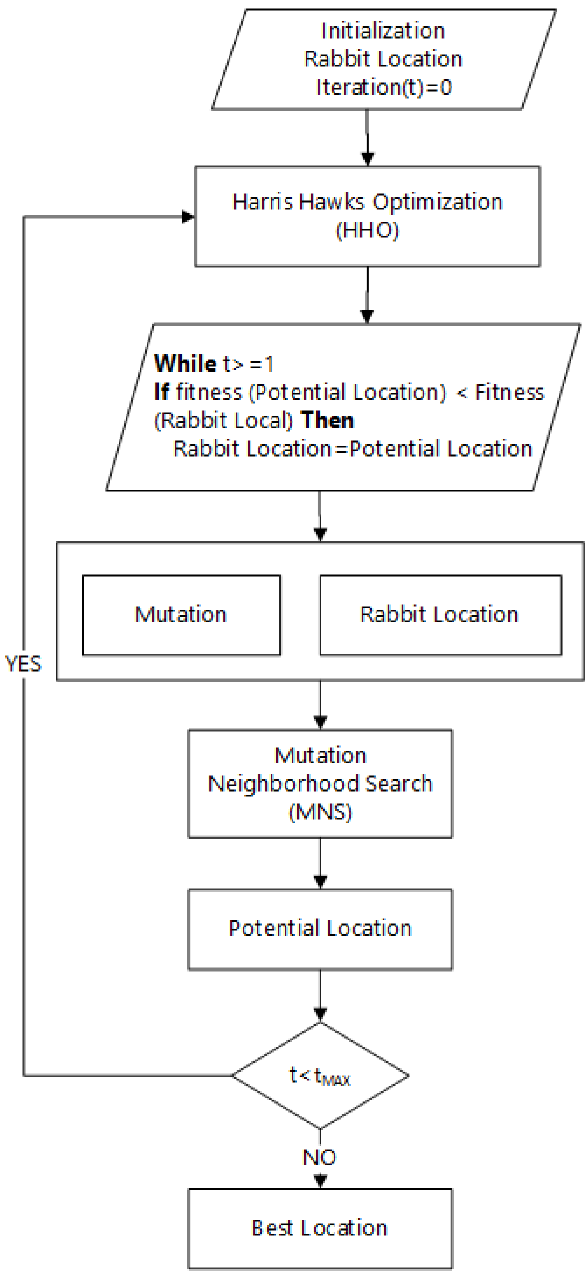

FIGURE 6. **The framework of the proposed three search strategies (TSS) to enhance the HHO search mechanism.**

IV. IMPROVED HARRIS HAWKS ALGORITHM (IHHO)

The standard HHO is a robust and high-performance optimization algorithm for solving practical engineering problems. However, based on the NFL theorem, no algorithm is perfect to handle all optimization problems [44]. Therefore, to avoid the limitations of HHO, to some extent, and to enhance its capabilities in handling feature selection, this paper proposed two main improvements for HHO. The first improvement includes utilizing the EOBL strategy at the initialization phase to enhance its population diversity. The second improvement aims to enhance the algorithm's global and local search abilities by applying the proposed TSS.

The following are the steps of the proposed IHHO algorithm:

Step 1: The population X is initialized, using the random function, with a size of N.

Step 2: Apply the EOBL technique and generate opposite solutions, then select the fittest N solutions.

Step 3: Perform the HHO algorithm to update the position of each individual in the population and find the Rabbit Location (best current location) according to the best fitness value.

Step 4: Apply the mutation strategy to improve the rabbit location. If the fitness of the new location is better than the current location, then set the new location as a potential rabbit location and perform the MNS strategy to further enhance its location. Lastly, set the best location to potential rabbit locations.

Step 5: Perform the next iteration in HHO. Compare the current rabbit location to the potential rabbit location. If the rabbit location is better than the potential location, then apply rollback strategy. Otherwise change the rabbit location to be equal to the potential rabbit location.

Step 6: Continue with the iterations until the termination condition is satisfied.

Note that the pseudocode of the proposed IHHO is presented in Algorithm 2.

However, HHO is designed for continuous solution search space and it needs to be modified to match binary feature selection. Therefore, the position of every Harris haws is converted into binary solutions by applying the following equation:

$$X_{i,j} = \begin{cases} 1 & if \ \dfrac{1}{1+e^{-X_{i,j}}} \geq 0.5 \\ 0 & otherwise, \end{cases} \tag{21}$$

Therefore, the features corresponding to ones in the dataset are selected as relevant features, while features corresponding to 0's are ignored.

V. EXPERIMENTS

A. PLATFORM

All of the algorithms and comparisons are implemented using Matlab R2020a software, and experiments are performed on a PC running Intel i5 processor with 2.2 GHz, 8 GB of RAM, and Windows 8 operating system.

Algorithm 2 IHHO Algorithm

Input: N: population size, T: maximum number of iterations.
Output: The rabbit location (X_{rabbit}), potential rabbit location ($X_{potential}$) and their fitness values
Initialize the population randomly $X_i (i = 1, 2, \ldots, N)$
Find the best N opposite solutions based on EOBL, then select the fittest N solutions, according to Equation (16), Equation (17) and Equation (18).
While (maximum iteration not reached ($t < T$)) **do**
  Check the location boundaries and Evaluate the fitness of Harris hawks locations
  Set the rabbit best location to X_{rabbit}
  For (each hawk $X_i (i = 1\ to\ N)$) **do**
    Update the rabbit initial energy E_0
    Update the rabbit energy E using Equation (3)
    If ($|E| \geq 1$) **then** **%Exploration phase**
      Update the hawks' position using Equation (1)
    If ($|E| < 1$) **then** **%Exploitation phase**
      If ($r \geq 0.5\ and\ |E| \geq 0.5$) **then** *%Soft besiege*
        Update the hawks' positions using Equation (4)
      Else if ($r \geq 0.5\ and\ |E| < 0.5$) **then** *%Hard besiege*
        Update the hawks' positions using Equation (6)
      Else if ($r < 0.5\ and\ |E| \geq 0.5$) **then** *%Soft besiege with progressive rapid dives*
        Update the hawks' positions using Equation (10)
      Else if ($r < 0.5\ and\ |E| < 0.5$) **then** %Hard besiege with progressive rapid dives
        Update the hawks' positions using Equation (11)
  For (i=1 to 10) **do** **%TSS**
    If ($Fitness\ X_{potential} < Fitness\ X_{rabbit(t+1)}$) **then**
      $X_{rabbit} = X_{potential}$
    Else
      $X_{rabbit} = X_{rabbit(t+1)}$ % Rollback
      Apply mutation strategy to rabbit location (X_{rabbit}) using Equation (19) and Equation (20)
    If rabbit location ($X_{mutation} < X_{rabbit}$) **then**
      Apply *MNS* search on $X_{mutation}$
      Set $X_{potential} = X_{mutation}$
Return the rabbit location (X_{rabbit})

B. BENCHMARK DATASETS

To verify the effectiveness of the proposed IHHO algorithm, we selected twenty benchmark datasets from the UCI datasets repository and scikit-feature project, which is an open-source feature selection repository at Arizona State University. The datasets are used to determine the capabilities of the IHHO algorithm. Further, to confirm the stability of IHHO, we used datasets with various feature dimensionality, including low, moderate and high dimensionality. The datasets' details are presented in Table 1.

TABLE 1. Details of the 20 benchmark datasets.

Dataset	Number of Selected Features	Number of Samples	Dimensionality
QSAR	9	546	Low
Lymphography	18	148	Low
Exactly	13	1000	Low
m-of-n	13	1000	Low
Vehicle	18	946	Low
Credit	20	1000	Low
Waveform	21	5000	Low
Spect	22	267	Low
HCV	29	1385	Moderate
Ionosphere	34	351	Moderate
Dermatology	34	366	Moderate
Spambase	57	4610	Moderate
Sonar	60	208	Moderate
LSVT	309	126	Moderate
Isolet	617	1560	High
CNAE	857	1080	High
warpAR10P	2400	130	High
RELATHE	4322	1427	High
TOX-171	5748	171	High
ALLAMAL	7129	72	High

C. PARAMETER SETTING

For the parameter setting, it is noted that the performance of algorithms can be improved by a fine-tuning of control parameters. Therefore, the choice of parameter setting is critical that should be selected carefully. In this work, we have set the parameters after many experimental comparisons as follows:

For the experiments, we used 10-fold cross-validation to evaluate the performance of the algorithms. The validation splits and shuffles the dataset into ten equal folds. While nine of them are utilized for the training phase, the last fold is left for testing. Further, the fitness function in Equation (14) was applied with parameter α was set to 0.99 and β to 0.01 [17]. In addition, to assure fairness comparison for all the algorithms, the maximum iterations for each algorithm was set to 50 iterations, and the population size was set to 10. Further, the experiments were repeated for 30 times; these settings are recommended by [8] and [45]. Therefore, the results were obtained from the average of 30 trials. Furthermore, IHHO was compared to the standard HHO and other state-of-the-art optimization algorithms such as GA, GOA, PSO, ALO, WOA, BOA and SMA. All the algorithms have been transferred to fit binary feature selection using Equation (21). TABLE 2 displays the general parameter settings for the utilized algorithms.

In this work, the proposed TSS was set to run for ten iterations. Also, we used classification accuracy, fitness function and selected features to evaluate the performance of optimization algorithms, mainly the KNN classifier with k was set to 5.

TABLE 2. General parameter settings of optimization algorithms.

Algorithm	Parameter	Value	Reference
HHO	Rabbit energy	[2,0]	[8]
GA	Crossover ratio	0.8	[46]
	Mutation ratio	0.2	
PSO	Acceleration-constants C1	1.5	[47]
	Acceleration-constants C2	2	
	Inertia-Weight W1	1	
	Inertia-Weight W2	0.9	
GOA	cMax	1	[41]
	cMin	0.00004	
ALO	K	500	[16]
WOA	A	[2,0]	[28]
BOA	Power exponent a	[0.1-0.3]	[15]
	Modular modality c	0.01	
	p	0.8	
SMA	z	0.03	[16]

D. COMPUTATIONAL COMPLEXITY

The computational complexity of IHHO depends on the following four factors: initialization, updating the Harris hawks, fitness function, and the TSS strategy. The complexity of the initialization process is $O(N)$, where N is the number of Harris hawks. Note that the computational complexity of updating mechanism, which consists of updating the Harris hawks' positions and finding the best location, is $O\,(T \times N)+O(T \times N \times D)$, where T and D represent the maximum number of iterations and the dimension of features respectively. Finally, the computational complexity of applying TSS strategy can be calculated as $O(T \times L \times S)$, where L is the number of TSS iterations and S is the TSS search strategies, including mutation and MNS. Thus, the computational complexity of IHHO is $O(N \times (T + TD + 1) + TLS)$.

VI. RESULTS AND ANALYSIS

In this section, the results of the two main experiments we have performed are outlined. In the first comparison, we compared the proposed IHHO to the standard HHO. In the second, we compared the IHHO with other well-known optimization algorithms like PSO, GA, GOA, ALO, WOA, BOA and SMA. In all experiments, each algorithm is applied on all the datasets to determine the stability of the algorithm over various feature dimensionality. Further, the results are reported based on calculating the average of 30 runs for each experiment.

A. COMPARISON OF HHO AND IHHO

In this experiment, the improved IHHO is compared with the original HHO. The comparison has been made based on the following four metrics: classification accuracy, number of selected features, fitness value, and performing Wilcoxon rank-sum test as a statistical test. The experimental results are shown in TABLE 3. For the statistical test, the improvement is considered significant if the p-value is less than 0.05; otherwise, it is not. The statistical test is used to determine whether the improvement in the classification accuracy of IHHO is significant or not.

Based on the results, IHHO has outperformed HHO in all datasets in terms of classification accuracy. Therefore, it is evident that the use of EOBL and TSS has improved the performance of IHHO. We have also found that the proposed algorithm has raised the average of classification accuracy by almost 3.2% and lower the average of fitness value by 3.1%. In terms of the number of selected features, IHHO outperformed the original algorithm by reducing the number of selected features in 14 datasets with 5.4% less in total average. In addition, the results of the fitness values support the previous discussion since IHHO outperformed on all the datasets. However, as a result of applying the TSS search strategy to the standard HHO, IHHO required extra time to optimize the datasets. Approximately an average of 20% is added to the time consumed by HHO. A comparison between IHHO and HHO is shown in Figure 7. Finally, the statistical test results showed that P-value is less than 0.05 for 18 datasets. Therefore, the IHHO improvement over HHO is significant. Hence, IHHO significantly enhanced the classification accuracy, fitness function value, and feature selection over various sizes of datasets.

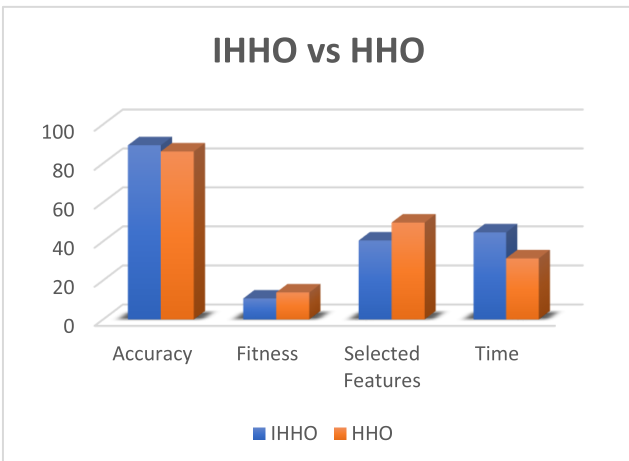

FIGURE 7. A comparison between IHHO and HHO in terms of accuracy, fitness value, number of selected features and the consumed time.

To show the effectiveness of the two proposed improvements on HHO, we repeated the experiment and compared the results from HHO with each improvement separately as shown in TABLE 4. First, in HHO-EOBL, the EOBL technique was applied only. The results show a slight improvement in the performance of HHO in all criteria after applying the EOBL in 15 datasets. Second, we applied the

TABLE 3. A comparison between HHO and IHHO based on classification accuracy, fitness value, time and the number of selected features.

Dataset	Accuracy		Selected Features		Fitness		Time		HHO P-Value
	HHO	IHHO	HHO	IHHO	HHO	IHHO	HHO	IHHO	
QSAR	0.911	**0.93**	**16.9**	18.2	0.092	**0.074**	**8.9**	11.1	6.37E-03
lymphography	0.685	**0.751**	6.9	**6.7**	0.315	**0.25**	**8.5**	10.6	1.99E-02
Exactly	0.946	**1**	6.6	**6.3**	0.059	**0.005**	**10.1**	12.6	1.44E-02
m-of-n	0.992	**1**	6.9	**6.1**	0.013	**0.005**	**9.9**	12.2	3.43E-04
Vehicle	0.776	**0.8**	8.6	**8.3**	0.227	**0.202**	**10.6**	12.7	1.90E-03
Credit	0.808	**0.831**	8.1	**8**	0.194	**0.171**	**9.7**	12.1	3.35E-04
Waveform	0.844	**0.854**	14.9	**13.5**	0.162	**0.151**	**33.2**	41.2	1.17E-03
Spect	0.850	**0.901**	**7**	8.6	0.152	**0.102**	**8.3**	10.3	9.23E-04
HCV	0.344	**0.361**	**7.4**	7.9	0.652	**0.634**	**11.1**	13.8	4.63E-03
Ionosphere	0.957	**0.978**	7.2	**7.1**	0.045	**0.023**	**8.5**	10.6	9.33E-04
Dermatology	0.977	**0.992**	12.6	**12.4**	0.005	**0.003**	**8.4**	10.4	7.51E-04
Spambase	0.932	**0.95**	30.7	**28.6**	0.073	**0.054**	**46**	57.9	1.25E-07
Sonar	0.968	**0.992**	**16**	17.1	0.034	**0.01**	**7.8**	9.7	9.61E-04
LSVT	0.866	**0.912**	35.3	**30.2**	0.133	**0.088**	**7.7**	9.6	3.44E-02
Isolet	0.918	**0.949**	292.9	**278**	0.086	**0.055**	**58.6**	72.8	1.75E-05
CNAE	0.892	**0.921**	545.7	**481**	0.114	**0.084**	**59**	73.6	3.09E-04
warpAR10P	0.834	**0.888**	**342.8**	406.8	0.165	**0.112**	**8.9**	11.1	**9.33E-02**
RELATHE	0.882	**0.932**	2039.9	**2027.6**	0.122	**0.072**	**288**	379	1.01E-08
TOX-171	0.909	**0.961**	**1392.8**	1478.7	0.093	**0.041**	**13.8**	16.8	2.16E-03
ALLAMAL	0.934	**0.957**	339.9	**337.7**	0.032	**0.009**	**8.2**	10.1	**6.89E-01**
IHHO Rank (W|T|L)	20|0|0		14|0|6		20|0|0		0|0|20		18|0|2

TSS strategy on the HHO without the EOBL technique. In this case, the results indicate a significant improvement in the performance of HHO on all the datasets. Finally, the IHHO, which contains the two improvements, has been ranked the first outperforming HHO, HHO-EOBL, and HHO-TSS in three criteria except for the consumption time.

From the results reported in the same table, we can see how the use of EOBL, achieved by equation (16), has enhanced the selection of the solutions instead of using the random method in the standard algorithm. A possible explanation is that EOBL chooses the best available solutions. The opportunity to select weak solutions is less compared to the solutions generated by the random method. In addition, the use of TSS has enhanced the algorithm's capabilities to balance exploration and exploitation. The algorithm uses the Harris hawk's best position to update the positions of the other search agents. Hence, the use of the proposed TSS has increased the algorithm's exploration ability in locating the promising region. It also prevents the algorithm from falling into a local solution by applying the mutation mechanism in Equation (20). Furthermore, both of the proposed neighborhood search and mutation mechanisms have improved the algorithm's exploitation ability in searching for the rabbit in the indicated local region. Therefore, the superiority of IHHO is proved in three aspects: classification accuracy, number of selected features, and fitness value.

B. COMPARISON OF IHHO ALGORITHM WITH OTHER OPTIMIZATION ALGORITHMS

The previous experiment has demonstrated a superior enhancement for IHHO, especially in classification accuracy and fitness value over the standard HHO. These improvements are the result of enhancing population diversity, proper balancing between exploration and exploitation, and the ability to avoid local optimum. Therefore, to confirm the superiority of IHHO, another comparison has been made between IHHO and other optimization algorithms such as GA, PSO, GOA, ALO, WOA, BOA and SMA. Like the first experiment, the second experiment utilized the four evaluation metrics to evaluate the performance of IHHO compared to other optimization algorithms. Classification accuracy was measured for all the algorithms, as shown in TABLE 6. Based on the results, the classification accuracy of IHHO has outperformed other algorithms in all of the datasets. The average of IHHO accuracy is 11.5% higher than GA, 7.3% than PSO, 5.9% than GOA, 5.5% than ALO, 6% than WOA, 11% than BOA and 10% than SMA algorithm. A comparison of the classification accuracy results for IHHO and the other algorithms are shown in TABLE 6.

The statistical test has been applied to determine the significance of classification accuracy, as shown in TABLE 7. As per results, with P-value is less than 0.05 for 18 datasets except for warpAR10P and ALLAMAL datasets. Therefore, we can detect that there is a significant difference between

TABLE 4. A comparison between HHO, HHO-EOBL, HHO-TSS and IHHO based on classification accuracy, fitness value and the number of features.

Dataset	Accuracy				Selected Features				Fitness			
	HHO	HHO EOBL	HHO TSS	IHHO	HHO	HHO EOBL	HHO TSS	IHHO	HHO	HHO EOBL	HHO TSS	IHHO
QSAR	0.898	0.899	0.928	**0.931**	15.9	16.1	14.9	**15.4**	0.105	0.103	0.074	**0.071**
lymphography	0.704	0.654	**0.759**	0.737	6.7	**6.6**	7.2	6.8	0.296	0.346	**0.241**	0.263
Exactly	0.932	0.928	0.998	**1**	6.35	6.4	6.2	**6.1**	0.072	0.076	0.005	**0.004**
m-of-n	0.986	0.990	0.999	**1**	6.6	6.5	6.1	**6**	0.019	0.015	0.005	**0.004**
Vehicle	0.763	0.773	0.793	**0.799**	8.7	8.3	8.1	**8**	0.238	0.229	0.208	**0.203**
Credit	0.793	0.806	0.821	**0.822**	**7.7**	8.2	8	8.2	0.208	0.195	0.181	**0.179**
Waveform	0.840	0.843	**0.853**	0.851	13.6	14.4	13	**12.4**	0.164	0.162	**0.151**	0.153
Spect	0.843	0.847	0.889	**0.892**	8.1	8	8.2	**7.9**	0.158	0.154	0.112	**0.110**
HCV	0.341	0.343	**0.358**	0.357	**6.7**	6.8	7.1	7.9	0.654	0.652	**0.637**	0.638
Ionosphere	0.956	0.965	0.981	**0.982**	7.3	7.2	7.2	**7.1**	0.044	0.035	0.020	**0.018**
Dermatology	0.996	0.996	0.997	**0.998**	12	11.7	**11.2**	11.8	0.006	0.006	0.005	**0.004**
Spambase	0.929	0.932	0.953	**0.954**	30.1	30.7	27.4	**26.6**	0.075	0.072	0.051	**0.049**
Sonar	0.966	0.964	0.991	**0.995**	17.1	16.5	16.5	**16.4**	0.035	0.037	0.009	**0.006**
LSVT	0.883	0.870	0.928	**0.943**	28.1	**27.1**	29.6	27.7	0.115	0.128	0.071	**0.057**
Isolet	0.915	0.916	0.945	**0.946**	298.2	306	274.4	**269.6**	0.087	0.087	0.059	**0.058**
CNAE	0.890	0.903	0.951	**0.963**	525	540.6	**415.1**	431.6	0.116	0.103	0.053	**0.042**
warpAR10P	0.827	0.835	0.841	**0.864**	**236.6**	303.7	419.3	371.3	0.171	0.163	0.157	**0.134**
RELATHE	0.88	0.883	0.939	**0.940**	2236	2189	2079	**1938**	0.117	0.121	0.065	**0.063**
TOX-171	0.904	0.914	**0.955**	0.947	1767	1486	1650	**1216**	0.096	0.086	**0.046**	0.054
ALLAMAL	0.947	0.948	**0.957**	0.949	328.3	385.9	**251.2**	315.2	0.018	0.017	**0.009**	0.016
HHO Rank (W|T|L)	-	4|1|15	0|0|20	0|0|20	-	10|0|10	6|0|14	4|0|16	-	4|1|15	0|0|20	0|0|20

IHHO and other algorithms. The results indicate the ability of IHHO to balance between global search and local search. Besides, it has a better opportunity to escape trapping in local optimum and avoid immature convergence, which results in a significant improvement in IHHO classification accuracy eventually.

TABLE 8 shows the average number of selected features by each algorithm over 30 runs. We can notice that IHHO outperformed 60% of the cases in terms of selected features. Moreover, it has been ranked first with selecting fewer features in 12 datasets out of 20, followed by WOA in 5 datasets, GA in two datasets, and PSO outperformed in one dataset only. GOA, ALO, BOA, SMA came last without outperforming any dataset. These results demonstrate the effectiveness of applying EOBL and TSS in reducing the number of selected features as well as improving the classification accuracy. It is also evident that IHHO focuses on informative regions in the search space to select the essential features and avoid irrelevant ones.

In TABLE 9, the results related to measuring fitness value is presented. As per the results, we can notice the dominance of IHHO over the rest of the algorithms. IHHO outperformed all of the algorithms in all the datasets, which indicates the superiority of IHHO. A comparison between IHHO and optimization algorithms based on average of fitness function value is shown in Figure 8. That means IHHO has minimum classification error, among other algorithms. The superiority

TABLE 5. A summary of IHHO improvements (based on median).

Algorithm	Accuracy	Features	Fitness
HHO	3.2 %↑	5.4 %↓	3.1 %↓
GOA	5.5 %↑	21 %↓	11 %↓
GA	11 %↑	17 %↓	7.5 %↓
PSO	5.9 %↑	20 %↓	5.9 %↓
ALO	5.5 %↑	23 %↓	5.5 %↓
WOA	6.0 %↑	7.7 %↓	5.7 %↓
BOA	11 %↑	23 %↓	12 %↓
SMA	10 %↑	22 %↓	11 %↓
Average	7.3 %↑	17.4 %↓	7.7 %↓

in fitness values indicates a strong ability of IHHO. In addition, the TSS search is dynamic and effective in searching for a promising area and best solution.

From the results reported in TABLE 6 - TABLE 9, it can be noticed that the datasets have multiple local optima, which are challenging for all optimization algorithms. Thus can discriminate the capabilities of the algorithms in balancing exploration and exploitation. For example, the classification accuracy of the "Exactly" dataset has shown variated results through the algorithms. While the highest accuracy achieved by IHHO, followed by HHO and PSO, with accuracy values

TABLE 6. A comparison of classification accuracy between IHHO and other optimization algorithms.

Dataset	IHHO	GOA	GA	PSO	ALO	WOA	BOA	SMA
QSAR	**0.93**	0.875	0.899	0.907	0.894	0.888	0.870	0.873
lymphography	**0.751**	0.563	0.623	0.669	0.613	0.586	0.525	0.567
Exactly	**1**	0.735	0.808	0.852	0.811	0.821	0.756	0.811
m-of-n	**1**	0.875	0.932	0.948	0.955	0.939	0.915	0.934
Vehicle	**0.8**	0.712	0.748	0.762	0.744	0.754	0.734	0.736
Credit	**0.831**	0.758	0.788	0.8	0.784	0.763	0.744	0.748
Waveform	**0.854**	0.813	0.838	0.838	0.84	0.839	0.835	0.831
Spect	**0.901**	0.779	0.833	0.846	0.822	0.812	0.777	0.785
HCV	**0.361**	0.303	0.317	0.327	0.329	0.31	0.301	0.311
ionosphere	**0.978**	0.904	0.936	0.949	0.939	0.940	0.896	0.900
dermatology	**1**	0.978	0.992	0.996	0.995	0.993	0.984	0.98
spambase	**0.95**	0.906	0.923	0.937	0.928	0.926	0.913	0.913
Sonar	**0.992**	0.893	0.941	0.967	0.938	0.932	0.887	0.887
LSVT	**0.912**	0.703	0.738	0.728	0.84	0.875	0.692	0.696
Isolet	**0.949**	0.898	0.923	0.93	0.913	0.910	0.894	0.894
CNAE	**0.921**	0.794	0.869	0.893	0.903	0.895	0.879	0.871
warpAR10P	**0.888**	0.689	0.707	0.716	0.815	0.81	0.629	0.642
RELATHE	**0.932**	0.86	0.882	0.911	0.883	0.874	0.856	0.856
TOX-171	**0.961**	0.778	0.846	0.864	0.893	0.881	0.774	0.80
ALLAMAL	**0.957**	0.84	0.84	0.84	0.929	0.917	0.822	0.808
Mean Rank (F-test)	**1.00**	6.80	4.30	2.75	3.15	3.95	7.20	6.40
Overall Rank	1	7	5	2	3	4	8	6

TABLE 7. p-values for the classification accuracy based on Wilcoxon rank-sum test.

Dataset	IHHO	GOA	GA	PSO	ALO	WOA	BOA	SMA
QSAR	2.12E-09	9.77E-05	3.18E-03	1.21E-05	2.12E-09	1.10E-08	4.18E-10	1.20E-10
lymphography	8.47E-07	1.36E-04	6.22E-03	3.69E-05	8.47E-07	7.09E-04	2.40E-06	5.60E-05
Exactly	1.54E-11	6.24E-11	2.71E-05	5.01E-10	1.54E-11	4.55E-12	1.20E-12	1.21E-12
m-of-n	1.32E-11	1.15E-10	4.46E-05	2.82E-11	1.32E-11	1.20E-12	1.20E-12	1.20E-12
Vehicle	3.98E-09	3.65E-07	1.24E-04	1.16E-07	3.98E-09	3.08E-06	2.60E-08	2.03E-09
Credit	1.47E-09	4.77E-07	2.52E-04	1.86E-07	1.47E-09	4.09E-07	2.22E-09	1.61E-09
Waveform	3.29E-10	3.15E-05	1.28E-04	3.05E-05	3.29E-10	5.45E-06	3.95E-08	3.04E-09
Spect	1.93E-09	2.85E-06	2.45E-04	9.84E-07	1.93E-09	8.84E-06	6.20E-09	2.18E-08
HCV	2.91E-09	1.01E-08	2.40E-06	9.50E-06	2.91E-09	7.07E-08	1.46E-10	3.15E-10
ionosphere	1.13E-09	2.31E-07	9.01E-06	4.26E-07	1.13E-09	2.08E-05	3.07E-10	1.37E-09
dermatology	2.29E-11	1.63E-09	7.36E-05	2.08E-10	2.29E-11	1.36E-10	2.74E-11	2.71E-11
spambase	3.87E-11	6.11E-10	1.41E-04	8.11E-09	3.87E-11	7.37E-11	3.02E-11	3.02E-11
Sonar	1.41E-10	1.97E-09	4.14E-08	6.31E-09	1.41E-10	5.33E-09	5.90E-11	9.66E-11
LSVT	8.86E-10	3.48E-09	1.69E-09	1.33E-03	8.86E-10	3.65E-04	3.01E-11	2.99E-11
Isolet	1.10E-08	2.25E-04	2.32E-03	4.44E-07	1.10E-08	9.06E-08	9.91E-11	3.15E-10
CNAE	3.02E-11	9.53E-07	5.57E-03	3.18E-03	3.02E-11	1.20E-10	1.09E-10	6.06E-11
warpAR10P	2.78E-07	1.29E-06	6.74E-06	**6.60E-02**	2.78E-07	2.50E-02	1.10E-08	2.60E-08
RELATHE	4.62E-10	2.02E-08	1.37E-03	5.95E-09	4.62E-10	3.15E-10	7.38E-11	3.68E-11
TOX-171	3.82E-10	5.97E-09	2.39E-08	7.30E-04	3.82E-10	4.03E-03	3.47E-10	5.57E-10
ALLAMAL	5.65E-09	4.73E-09	6.74E-09	4.46E-03	5.65E-09	**3.55E-01**	9.61E-09	5.67E-09

equal to 1, 0.946 (from TABLE 3) and 0.852 respectively, GOA could achieve a modest result of 0.735 only. The proposed IHHO is flexible that it keeps looking for new promising regions, achieved by mutating the best solution using Equation (20). This technique helps to prevent the algorithm from falling into local optima. In addition, the mutation

TABLE 8. A comparison between IHHO and other optimization algorithms based on average of number of selected features.

Dataset	IHHO	GOA	GA	PSO	ALO	WOA	BOA	SMA
QSAR	18.2	20.6	19.0	19.6	19.5	**17.7**	25.2	23.7
Lymphography	6.7	9.0	8.0	8.6	7.2	**6.2**	11.1	11.5
Exactly	**6.3**	7.2	7.8	6.8	8.8	7.5	11.2	9.7
m-of-n	**6.1**	7.4	8.1	6.7	9.3	9.2	11.2	10.7
Vehicle	**8.3**	9.5	9.2	9.1	10.2	8.4	11.6	11.2
Credit	**8.2**	9.8	9.5	9.8	8.8	8.5	12.5	11.5
Waveform	13.5	11.8	14.0	**11.5**	17.4	15.8	18.3	18.1
Spect	**8.6**	10.6	9.6	10.3	9.0	9.0	12.6	11.2
HCV	**7.9**	13.6	9.5	12.6	10.2	9.2	14.4	15.7
Ionosphere	**7.1**	16.2	13.1	15.3	10.1	7.7	16.7	15.2
Dermatology	**12.4**	17.0	13.8	15.8	16.7	16.2	20.8	18.3
Spambase	28.6	29.0	**28.0**	28.6	30.5	35.6	39.8	35.8
Sonar	**17.1**	29.3	27.0	27.7	23.3	20.1	32.7	29.2
LSVT	30.2	152.3	120.4	145.9	30.1	**14.0**	159.3	151.2
Isolet	**278.0**	306.0	296.3	305.5	339.8	316.6	449.7	353.3
CNAE	481.0	426.9	**414.6**	426.5	632.3	661.9	799.9	714.7
warpAR10P	406.8	1195.8	1108.2	1177.8	335.6	**267.6**	1296.0	1243.7
RELATHE	**2027.7**	2158.3	2141.7	2158.9	2511.1	2607.1	3201.5	2798.0
TOX-171	**1478.6**	2864.4	2815.8	2849.9	1623.8	1535.9	3234.1	3075.6
ALLAMAL	337.7	3433.9	3215.1	3400.5	372.6	**188.7**	3405.6	3398.4
Mean Rank (F-test)	**1.65**	5.15	3.35	3.95	4.10	3.00	7.85	6.80
Overall Rank	1	6	3	4	5	2	8	7

TABLE 9. A comparison between IHHO and other optimization algorithms based on average of fitness function value.

Dataset	IHHO	GOA	GA	PSO	ALO	WOA	BOA	SMA
QSAR	**0.074**	0.129	0.104	0.097	0.109	0.114	0.114	0.134
Lymphography	**0.250**	0.438	0.377	0.332	0.387	0.412	0.475	0.434
Exactly	**0.005**	0.268	0.196	0.152	0.195	0.183	0.249	0.194
m-of-n	**0.005**	0.130	0.074	0.057	0.052	0.067	0.092	0.072
Vehicle	**0.202**	0.290	0.255	0.241	0.259	0.247	0.269	0.267
Credit	**0.171**	0.245	0.215	0.202	0.218	0.238	0.259	0.254
Waveform	**0.151**	0.191	0.167	0.166	0.167	0.167	0.171	0.175
Spect	**0.102**	0.224	0.169	0.156	0.180	0.189	0.225	0.217
HCV	**0.634**	0.694	0.679	0.669	0.667	0.677	0.696	0.687
Ionosphere	**0.023**	0.099	0.068	0.054	0.062	0.060	0.107	0.103
Dermatology	**0.003**	0.027	0.012	0.007	0.010	0.011	0.021	0.017
Spambase	**0.054**	0.098	0.082	0.067	0.077	0.079	0.092	0.091
Sonar	**0.010**	0.110	0.063	0.036	0.065	0.069	0.116	0.115
LSVT	**0.087**	0.298	0.263	0.273	0.159	0.123	0.309	0.305
Isolet	**0.055**	0.106	0.081	0.074	0.092	0.094	0.112	0.109
CNAE	**0.084**	0.209	0.135	0.111	0.105	0.112	0.129	0.135
warpAR10P	**0.112**	0.313	0.295	0.286	0.18	0.188	0.372	0.359
RELATHE	**0.072**	0.143	0.122	0.093	0.122	0.130	0.149	0.148
TOX-171	**0.041**	0.225	0.157	0.140	0.108	0.119	0.229	0.196
ALLAMAL	**0.009**	0.130	0.129	0.129	0.037	0.048	0.147	0.161
Mean Rank (F-test)	**1.00**	6.95	4.20	2.60	3.25	3.90	7.30	6.45
Overall Rank	1	7	5	2	3	4	8	6

neighborhood strategy has improved the local search of the IHHO by digging inside the promising area looking for a better solution.

Considering the convergence behavior of an optimization algorithm is very important in evaluating its performance. Convergence shows the ability of the algorithm

TABLE 10. A comparison between IHHO and other optimization algorithms based on average time in seconds.

Dataset	IHHO	HHO	GOA	GA	PSO	ALO	WOA	BOA	SMA
QSAR	11.28	9.27	4.96	4.49	4.41	4.71	**4.19**	9.77	4.71
Lymphography	10.35	8.49	4.22	4.07	4.03	4.05	**3.57**	7.54	3.74
Exactly	12.41	9.72	4.95	4.89	4.82	4.65	**4.34**	8.62	4.36
m-of-n	12.54	10.1	5.19	5.01	4.88	4.82	**4.37**	8.63	4.50
Vehicle	12.40	9.60	4.91	4.70	4.70	4.67	**4.20**	8.57	4.27
Credit	12.23	9.81	4.81	4.88	4.70	4.72	**4.28**	8.69	4.34
Waveform	38.39	32.28	**15.13**	15.48	15.21	16.86	15.21	35.82	16.46
Spect	11.19	8.59	4.31	4.15	4.09	4.23	**3.71**	7.78	3.87
HCV	13.62	10.4	5.44	5.32	5.24	5.13	**4.54**	11.04	5.40
Ionosphere	12.66	8.58	4.27	4.01	3.99	4.20	**3.54**	7.75	3.85
Dermatology	11.88	8.32	4.40	4.05	4.05	4.19	**3.78**	8.17	4.01
Spambase	65.87	44.67	21.41	**20.58**	19.99	23.27	23.77	69.01	30.47
Sonar	9.73	8.32	4.75	4.03	4.02	4.30	**3.73**	8.01	4.04
LSVT	9.95	7.97	7.65	3.94	3.94	5.30	**3.21**	7.92	4.02
Isolet	73.88	53.63	35.97	28.53	**28.30**	35.56	28.31	104.11	43.62
CNAE	75.60	56.16	31.29	21.01	**20.81**	37.59	30.54	75.21	34.41
warpAR10P	12.04	8.93	35.41	5.46	5.48	15.54	**4.24**	13.42	7.10
RELATHE	365.82	317.2	202.9	152.7	**151.7**	216.4	181.0	579.7	246.8
TOX-171	17.42	13.73	82.77	9.80	9.77	36.77	**8.12**	27.75	14.83
ALLAMAL	11.89	8.14	92.18	**5.89**	5.93	37.30	6.27	14.10	8.47
Mean Rank (F-test)	8.40	7.10	5.75	3.35	2.45	4.85	**1.60**	7.50	3.75
Overall Rank	9	7	6	3	2	5	1	8	4

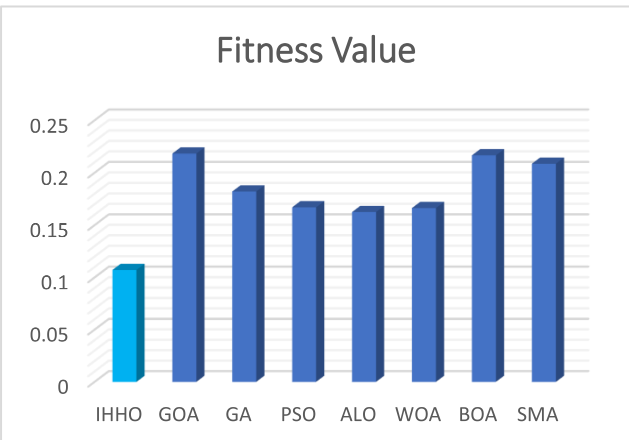

FIGURE 8. A comparison between IHHO and optimization algorithms based on average of fitness function value.

to escape local optima and immature convergence. If the optimization algorithm cannot balance between exploration and exploitation in all stages, it will probably be converged to local optima. A comparison between IHHO convergence and the other algorithms are shown in Figures 9, 10, and 11. From the convergence curves in the figures, we can observe that IHHO can achieve superior solutions faster than the other algorithms, which indicates the superiority of IHHO in dealing with all the datasets. Also, we can notice the effectiveness of the proposed TSS search, which transfers from global to local search in the middle of iterations (iteration number 25 in our experiments as maximum iteration was set to 50), in improving the convergence curves for all cases.

The algorithms' consumption time is shown in TABLE 10. Based on the results, IHHO has consumed more time, with an increased rate of 20%, compared to the HHO algorithm. It can also be noticed that IHHO came last in the rank. A good explanation is that IHHO proposes a simple search strategy, which is added to HHO to enhance its exploration and exploitation capabilities. However, HHO is ranked in the 7th place in the same table, two places before IHHO. In other words, the consumed time of IHHO is considered relatively high because the time of the standard algorithm is high in the first place.

A summary comparing IHHO with the other algorithms by calculating the median of classification accuracy, selected features and fitness value for all experiments is shown in TABLE 5.

C. LIMITATIONS OF IHHO ALGORITHM

The proposed IHHO is a beneficial algorithm, which can solve large space and complex optimization problems. IHHO enhanced the standard HHO in many aspects, such as

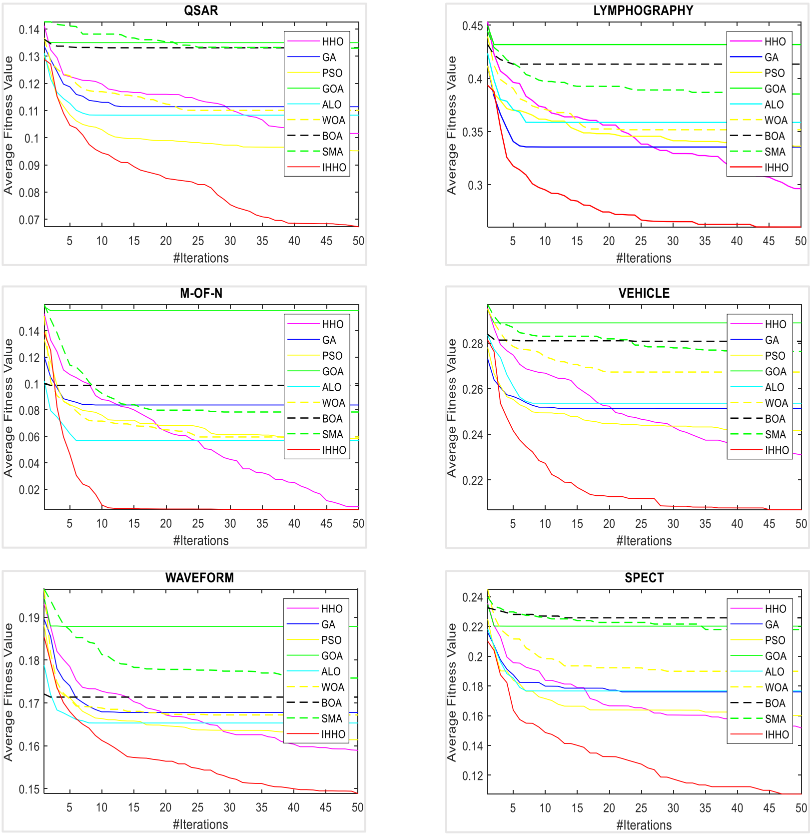

FIGURE 9. Convergence speed for low dimensional features datasets.

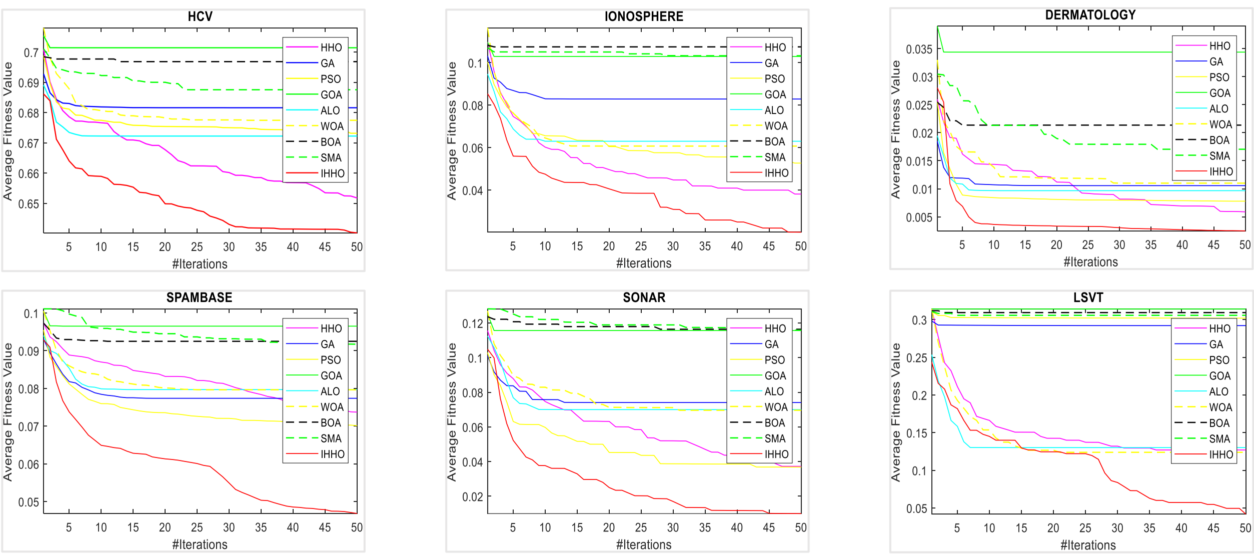

FIGURE 10. Convergence speed for moderate dimensional features datasets.

classification accuracy, fitness value, and the number of selected features. However, like other optimization algorithms, IHHO has some limitations. The main limitation is the relatively high time consumption compared to the other algorithms. However, the high consumption is not mainly caused by the proposed improvements, but it is caused by the high computational complexity of the standard HHO. Therefore, improving the complexity of HHO will result in improving the complexity of IHHO as well. Another limitation is related to the iterations of the proposed TSS; we believe that the time complexity of IHHO can be reduced by replacing the ten iterations of TSS with a less complicated solution.

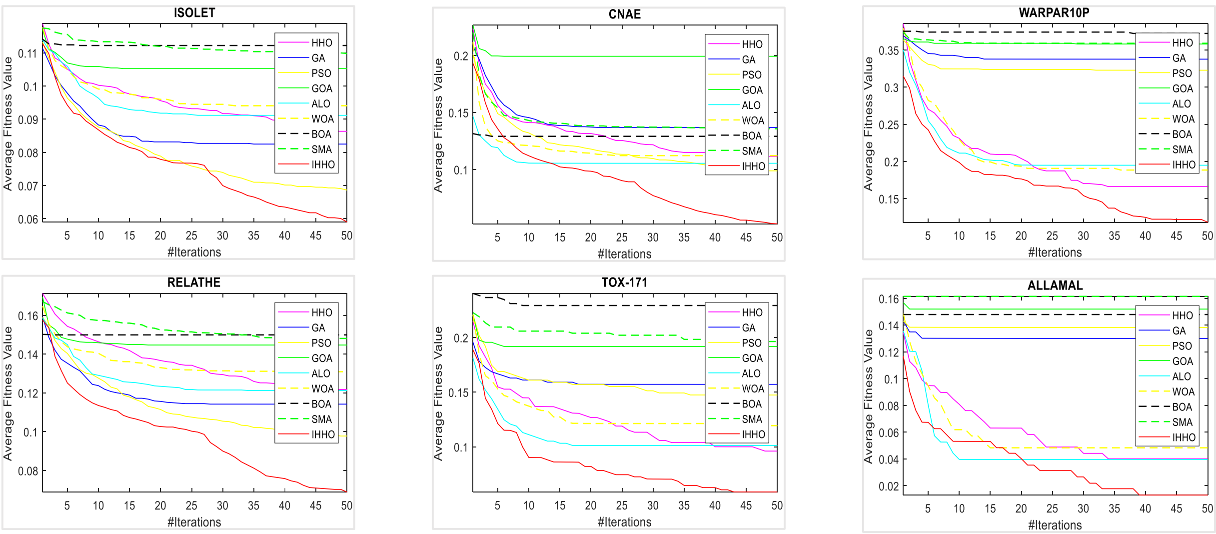

FIGURE 11. Convergence speed for high dimensional features datasets.

VII. CONCLUSION

Harris hawks optimization (HHO) is one of the latest meta-heuristic, population-based, and high-performance algorithms that mimic Harris hawks' style in searching and chasing the prey in nature. HHO has two phases devoted to the exploration and four phases for exploitation. In this work, we proposed an improvement of the original HHO algorithm called IHHO using the EOBL technique and the proposed TSS search. The TSS search uses mutation, mutation neighborhood search, and rollback strategies to improve the global and local searches of HHO. Further, the new IHHO has shown the right balance in transferring between exploration and exploitation. While the use of EOBL enhanced the population diversity of HHO, TSS search strategies have helped the algorithm in its search for the global optima and to avoid trapping in local optima.

We utilized twenty low, moderate, and high-dimensional benchmark datasets from the UCI repository and scikit-feature, which is a project introduced by Arizona State University, to evaluate the performance of IHHO. Besides, we compared IHHO with well-known optimization algorithms, such as HHO, GA, GOA, PSO, ALO, WOA, BOA and SMA. The comparison based on the following four evaluation metrics: classification accuracy, fitness value, number of selected features, and statistical tests. The results from the experiments have confirmed the superiority of IHHO over the other algorithms in all metrics. Moreover, its abilities to improve computational accuracy and accelerates convergence rate besides lowering the number of selected features have been proved for most of the twenty datasets.

The results from the conducted experiments suggest that IHHO can be applied as a promising technique to deal with real-world feature selection datasets that have low, moderate, and high dimensional features. It also works in different domains like data science, data mining, engineering problems, digital forensics analysis, sentiment analysis, and many more applications.

For the future work, we believe that there are several directions in which IHHO can be extended to tackle new real-world datasets such as applying IHHO to hybrid wrapper-filter feature selection techniques. Further, the performance of IHHO can be acquired using different classifiers such as Support vector machine (SVM) and neural networks. Additionally, time reduction is to be considered in future work as well. Furthermore, EOBL and the proposed TSS techniques can be utilized to improve other optimization algorithms.

REFERENCES

[1] A. Khanan, S. Abdullah, A. H. H. M. Mohamed, A. Mehmood, and K. A. Z. Ariffin, "Big data security and privacy concerns: A review," in *Smart Technologies and Innovation for a Sustainable Future* (Advances in Science, Technology & Innovation). Cham, Switzerland: Springer, 2019, pp. 55–61, doi: 10.1007/978-3-030-01659-3_8.

[2] M. A. Halim, A. Abdullah, and K. A. Z. Ariffin, "Recurrent neural network for malware detection," *Int. J. Adv. Soft Comput. Appl.*, vol. 11, no. 1, pp. 46–63, 2019.

[3] A. Arram, M. Ayob, G. Kendall, and A. Sulaiman, "Bird mating optimizer for combinatorial optimization problems," *IEEE Access*, vol. 8, pp. 96845–96858, 2020, doi: 10.1109/ACCESS.2020.2993491.

[4] R. Sihwail, K. Omar, and K. A. Z. Ariffin, "A survey on malware analysis techniques: Static, dynamic, hybrid and memory analysis," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 8, nos. 2–4, pp. 1662–1671, 2018, doi: 10.18517/ijaseit.8.4-2.6827.

[5] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam, and J. Yearwood, "Hybrids of support vector machine wrapper and filter based framework for malware detection," *Future Gener. Comput. Syst.*, vol. 55, pp. 376–390, Feb. 2016, doi: 10.1016/j.future.2014.06.001.

[6] R. Sihwail, K. Omar, K. Z. Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Appl. Sci.*, vol. 9, no. 18, p. 3680, Sep. 2019, doi: 10.3390/app9183680.

[7] K. O. Babaagba and S. O. Adesanya, "A study on the effect of feature selection on malware analysis using machine learning," in *Proc. ACM Int. Conf. Educ. Inf. Technol.*, 2019, pp. 51–55, doi: 10.1145/3318396.3318448.

[8] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris hawks optimizer for high-dimensional, low sample size feature selection," in *Evolutionary Machine Learning Techniques*. Singapore: Springer, 2020, pp. 251–272, doi: 10.1007/978-981-32-9990-0_12.
[9] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018, doi: 10.1016/j.asoc.2017.11.006.
[10] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.
[11] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017, doi: 10.1016/j.advengsoft.2017.01.004.
[12] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, vol. 4, 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.
[13] S. Mirjalili, "The ant lion optimizer," *Adv. Eng. Softw.*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.
[14] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.
[15] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.*, vol. 23, no. 3, pp. 715–734, Feb. 2019, doi: 10.1007/s00500-018-3102-4.
[16] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020, doi: 10.1016/j.future.2020.03.055.
[17] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019, doi: 10.1016/j.future.2019.02.028.
[18] Q. Yin, B. Cao, X. Li, B. Wang, Q. Zhang, and X. Wei, "An intelligent optimization algorithm for constructing a DNA storage code: NOL-HHO," *Int. J. Mol. Sci.*, vol. 21, no. 6, p. 2191, Mar. 2020, doi: 10.3390/ijms21062191.
[19] J. Too, A. R. Abdullah, and N. M. Saad, "A new quadratic binary Harris hawk optimization for feature selection," *Electronics*, vol. 8, no. 10, pp. 1–27, 2019, doi: 10.3390/electronics8101130.
[20] H. Jia, C. Lang, D. Oliva, W. Song, and X. Peng, "Dynamic Harris hawks optimization with mutation mechanism for satellite image segmentation," *Remote Sens.*, vol. 11, no. 12, p. 1421, Jun. 2019, doi: 10.3390/rs11121421.
[21] H. M. Ridha, A. A. Heidari, M. Wang, and H. Chen, "Boosted mutation-based Harris hawks optimizer for parameters identification of single-diode solar cell models," *Energy Convers. Manage.*, vol. 209, Apr. 2020, Art. no. 112660, doi: 10.1016/j.enconman.2020.112660.
[22] X. Zhou, Z. Wu, and H. Wang, "Elite opposition-based differential evolution for solving large-scale optimization problems and its implementation on GPU," in *Proc. Parallel Distrib. Comput. Appl. Technol. (PDCAT)*, 2012, pp. 727–732, doi: 10.1109/PDCAT.2012.70.
[23] X. Wu, Y. Zhou, and Y. Lu, "Elite opposition-based water wave optimization algorithm for global optimization," *Math. Problems Eng.*, vol. 2017, pp. 1–25, Jan. 2017, doi: 10.1155/2017/3498363.
[24] R. Zhao, Q. Luo, and Y. Zhou, "Elite opposition-based social spider optimization algorithm for global function optimization," *Algorithms*, vol. 10, no. 1, p. 9, Jan. 2017, doi: 10.3390/a10010009.
[25] K. Huang, Y. Zhou, X. Wu, and Q. Luo, "A cuckoo search algorithm with elite opposition-based strategy," *J. Intell. Syst.*, vol. 25, no. 4, pp. 567–593, Oct. 2016, doi: 10.1515/jisys-2015-0041.
[26] S. Zhang, Q. Luo, and Y. Zhou, "Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method," *Int. J. Comput. Intell. Appl.*, vol. 16, no. 2, pp. 1–37, 2017, doi: 10.1142/S1469026817500122.
[27] Y. Zhou, R. Wang, and Q. Luo, "Elite opposition-based flower pollination algorithm," *Neurocomputing*, vol. 188, pp. 294–310, May 2016, doi: 10.1016/j.neucom.2015.01.110.
[28] M. Tubishat, M. A. M. Abushariah, N. Idris, and I. Aljarah, "Improved whale optimization algorithm for feature selection in arabic sentiment analysis," *Int. J. Speech Technol.*, vol. 49, no. 5, pp. 1688–1707, May 2019, doi: 10.1007/s10489-018-1334-8.
[29] X. Bao, H. Jia, and C. Lang, "A novel hybrid Harris hawks optimization for color image multilevel thresholding segmentation," *IEEE Access*, vol. 7, pp. 76529–76546, 2019, doi: 10.1109/ACCESS.2019.2921545.
[30] N. A. Golilarz, H. Gao, and H. Demirel, "Satellite image de-noising with Harris hawks meta heuristic optimization algorithm and improved adaptive generalized Gaussian distribution threshold function," *IEEE Access*, vol. 7, pp. 57459–57468, 2019, doi: 10.1109/ACCESS.2019.2914101.
[31] V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, "An intensify Harris hawks optimizer for numerical and engineering optimization problems," *Appl. Soft Comput.*, vol. 89, Apr. 2020, Art. no. 106018, doi: 10.1016/j.asoc.2019.106018.
[32] I. Attiya, M. A. Elaziz, and S. Xiong, "Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm," *Comput. Intell. Neurosci.*, vol. 2020, Mar. 2020, Art. no. 3504642, doi: 10.1155/2020/3504642.
[33] H. Moayedi, A. Osouli, H. Nguyen, and A. S. A. Rashid, "A novel Harris hawks' optimization and k-fold cross-validation predicting slope stability," *Eng. Comput.*, Jul. 2019, pp. 1–11, doi: 10.1007/s00366-019-00828-8.
[34] S. Gupta, K. Deep, A. A. Heidari, H. Moayedi, and M. Wang, "Opposition-based learning Harris hawks optimization with advanced transition rules: Principles and analysis," *Expert Syst. Appl.*, vol. 158, Nov. 2020, Art. no. 113510, doi: 10.1016/j.eswa.2020.113510.
[35] A. S. Menesy, H. M. Sultan, A. Selim, M. G. Ashmawy, and S. Kamel, "Developing and applying chaotic Harris hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks," *IEEE Access*, vol. 8, pp. 1146–1159, 2020, doi: 10.1109/ACCESS.2019.2961811.
[36] R. Hans, H. Kaur, and N. Kaur, "Opposition-based Harris hawks optimization algorithm for feature selection in breast mass classification," *J. Interdiscipl. Math.*, vol. 23, no. 1, pp. 97–106, Jan. 2020, doi: 10.1080/09720502.2020.1721670.
[37] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling Control Automat. (CIMCA), Int. Conf. Intell. Agents, Web Technol. Internet*, vol. 1, 2005, pp. 695–701, doi: 10.1109/cimca.2005.1631345.
[38] X. Li and M. Yin, "A particle swarm inspired cuckoo search algorithm for real parameter optimization," *Soft Comput.*, vol. 20, no. 4, pp. 1389–1413, Apr. 2016, doi: 10.1007/s00500-015-1594-8.
[39] F. Zhao, J. Zhang, J. Wang, and C. Zhang, "A shuffled complex evolution algorithm with opposition-based learning for a permutation flow shop scheduling problem," *Int. J. Comput. Integr. Manuf.*, vol. 28, no. 11, pp. 1220–1235, 2015, doi: 10.1080/0951192X.2014.961965.
[40] J. Shang, Y. Sun, S. Li, J.-X. Liu, C.-H. Zheng, and J. Zhang, "An improved opposition-based learning particle swarm optimization for the detection of SNP-SNP interactions," *BioMed Res. Int.*, vol. 2015, pp. 1–12, Jul. 2015, doi: 10.1155/2015/524821.
[41] A. A. Ewees, M. A. Elaziz, and E. H. Houssein, "Improved grasshopper optimization algorithm using opposition-based learning," *Expert Syst. Appl.*, vol. 112, pp. 156–172, Dec. 2018, doi: 10.1016/j.eswa.2018.06.023.
[42] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection," *Expert Syst. Appl.*, vol. 145, May 2020, Art. no. 113122, doi: 10.1016/j.eswa.2019.113122.
[43] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009, doi: 10.1109/TEVC.2008.2009457.
[44] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.
[45] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Feature selection using chaotic salp swarm algorithm for data classification," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3801–3816, Apr. 2019, doi: 10.1007/s13369-018-3680-6.
[46] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019, doi: 10.1016/j.eswa.2018.09.015.
[47] G. I. Sayed, A. Tharwat, and A. E. Hassanien, "Chaotic dragonfly algorithm: An improved Metaheuristic algorithm for feature selection," *Appl. Intell.*, vol. 49, no. 1, pp. 188–205, Jan. 2019, doi: 10.1007/s10489-018-1261-8.

RAMI SIHWAIL received the B.Sc. degree in computer science from Al Al-Bayt University, Jordan, in 1999, and the M.Sc. degree in computer science from University Putra Malaysia, Malaysia, in 2002. He is currently pursuing the Ph.D. degree in machine learning with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM). His research interests include machine learning, optimization algorithms, security, malware analysis, memory forensics, feature selection, feature extraction, and data science and analytics.

KHAIRUDDIN OMAR received the bachelor's and master's degrees in computer science from Universiti Kebangsaan Malaysia, in 1986 and 1989, respectively, and the Ph.D. degree in philosophy from Universiti Putra Malaysia, in 2000. He is currently a Professor with the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia. His research interests include artificial intelligence (pattern recognition in decision making with uncertainty–Bayesian reasoning, neural networks, fuzzy logic, and fuzzy neural networks, and image processing–2D and 3D, edge detection, thinning, segmentation, feature extraction, image improvement, texture, resolution, and image transforms, such as Trace, Fourier, and Wavelet,) with applications to Jawi/Arabic manuscripts and biometric authentication.

KHAIRUL AKRAM ZAINOL ARIFFIN (Member, IEEE) received the bachelor's and master's degrees (Hons.) in system engineering with computer engineering from the University of Warwick, U.K., in 2008 and 2009, respectively, and the Ph.D. degree in information system from the Universiti Teknologi PETRONAS (UTP). During his time in UTP, a number of journal articles and conference papers have been produced and published internationally. Then, he was appointed as a Researcher with the Digital Forensic Department, CyberSecurity Malaysia, and had been entrusted with the research on embedded systems and live forensic. He is currently a member of the Technology and Information Science Faculty, National University of Malaysia, to pursue his passion in research towards cyber security, digital forensics, algorithms, and embedded systems. He is GCFA certified and a member of IET.

MOHAMMAD TUBISHAT received the B.Sc. degree in computer science and the M.Sc. degree in computer and information sciences from Yarmouk University, in 2002 and 2004, respectively, and the Ph.D. degree in computer science (artificial intelligence–natural language processing) from the University of Malaya, in 2019. He is currently working as a Lecturer with the Asia Pacific University of Technology and Innovation. His research interests include natural language processing, data mining, artificial intelligence, machine learning, optimization algorithms, data science, and sentiment analysis.