Adaptive Neighborhood Adjustment Strategy Based On MOHHO and NSGA-III Algorithms for Feature Selection

Anusha Papasani , Revathi Durgam and Nagaraju Devarakonda

Abstract- **The efficiency of multi-objective evolutionary algorithms (MOEAs) in tackling issues with multiple objectives is examined. However, it is noted that current MOEA-based feature selection techniques often converge towards the center of the Pareto front due to inadequate selection forces. The study proposes the utilization of a novel approach known as MOEA/D, which partitions complex multi-objective problems into smaller, more feasible single-objective sub-problems. Each sub-problem may then be addressed using an equal amount of computational resources. The predetermined size of the neighborhood used by MOEA/D may lead to a delay in the algorithm's merging and reduce the effectiveness of the failure. The paper proposes the Adaptive Neighbourhood Adjustment Strategy (ANAS) as a novel approach to improve the efficiency of multi-objective optimisation algorithms in order to tackle this issue. The ANAS algorithm allows for adaptive adjustment of the subproblem neighborhood size, hence enhancing the trade-off between merging and variety. In the following section of the study, a novel feature selection technique called MOGHHNS3/D-ANA is introduced. This technique utilizes ANAS to expand the potential solutions for a particular subproblem. The approach evaluates the chosen features using the Regulated Extreme Learning Machine (RELM) classifier on sixteen benchmark datasets. The experimental results demonstrate that MOGHHNS3/D-ANA outperforms four commonly employed multi-objective techniques in terms of accuracy, precision, recall, F1 score, coverage, hamming loss, ranking loss, and training time, error. The APBI approach in decomposition-based multi-objective optimization focuses on handling constraints by adjusting penalty parameters to guide the search towards feasible solutions. On the other hand, the ANA approach focuses on dynamically adjusting the neighborhood size or search direction based on the proximity of solutions in the detached space to adapt the search process. The proposed approach achieves convergence by minimizing redundancy, preserving diversity in the decision space, and simultaneously enhancing classification accuracy.**

Index Terms—Non-dominated sorting genetic algorithm III, Adaptive Neighborhood Adjustment Strategy, wrapper Based Feature selection Method, Harris Hawks Algorithm, multi-objective optimization.

Anusha Papasani is an Assistant professor of Computer Science and Engineering Department, Koneru Lakshmaiah Education Foundation, Vaddeswaram 522502, AndhraPradesh, India (Corresponding Author email:anoosha.papasani@gmail.com).

Revathi Durgam is an Assistant professor of Computer Science and Engineering Department, AVN Institute of Engineering and Technology Hyderabad, Telangana 501510, India (e-mail: revathidurgam@gmail.com).

Nagaraju Devarakonda is a professor of Computer Science and Engineering Department, VIT-AP University, Amaravati AP, 522237, India. (e-mail: dnagaraj_dnr@yahoo.co.in)

I. INTRODUCTION

The increasing growth of data in fields including online education, healthcare, bioinformatics, and manufacturing makes effective information management difficult [1]. Characterizing the hidden knowledge hidden in the mountains of data requires a combination of ML and DM [2]. Classification is one way to organize data in a database by label. All classifiers are most precious by the huge feature size. A vital step in the classification process is feature selection (FS), which involves the removal of redundant or superfluous attributes from the dataset. [3]. the feature selection method takes a large dataset and chooses the most useful features from among it. It's not hard to complete the FS assignment if you already know the most important and useful aspects. Otherwise, it's hard to pick out what's most important and valuable [4]. The FS type informs the evaluation of the generated subset of features. Wrapper and filter are the two FS varieties. Filter-based FS methods perform admirably on big data with low computational overhead [5]. One major shortcoming of filter-based FS approaches is the lack of connections between the features and feature dependencies. A classifier employing the wrapper technique [6] determines the accuracy for each of the feature subsets chosen. Using this method raises computing costs for datasets with several attributes [7]. To minimise the selected feature selected subset while simultaneously increasing classification accuracy, the wrapper feature selection can be viewed as a multi-objective optimization model. [8].

A variety of meta-heuristic optimization algorithms, such as Particle Swarm Optimization, Genetic Algorithm (GA), Bat Algorithm (BA), Whale Optimization Algorithm (WOA), and Grey Wolf Optimization (GWO), have been proposed as wrapper solutions by experts. Both PSO [9-11] and GA have seen extensive application in the relevant academic literature. Better global search results can be obtained by using single-objective evolutionary algorithms like PSO [14] and GA [12, 13], but these algorithms are limited in their ability to make use of recognized regions. If these methods aren't constrained to their own local maxima, they can't effectively explore the vast search space. While most evolutionary algorithms strive for early convergence, the Harris Hawks Algorithm (HHO) is able to meet earlier and locate the best region in the exploration space. Newly proposed optimization method HHO[16] outperforms other evolutionary algorithms in terms of both survey and exploitation. Both the second version of

the Strength Pareto based evolutionary algorithm [17] (SPEA2) and the third version of the non-dominated sorting genetic algorithm (NSGA-III) have shown to be highly effective MOEAs for dealing with multiple objective functions. In high-dimensional domains, NSGA-III, an enhanced version of NSGA-II, is the most popular and influential evolutionary algorithm. This is especially true when compared to SPEA-II and PESA.

To lower error rates and eliminate unnecessary features in feature selection, multi-objective optimization algorithms [15] are advocated. An Evolutionary Algorithm with Multiple Goals[18,19]. Most people are interested in (MOEA/D), which is built on the idea of failure. Implementations of MOEA/D frequently employ scalarizing techniques such as weighted Tchebycheff (TCH) and penalty-based boundary intersection (PBI). Decomposition-based multi-objective optimization feature selection is the primary focus of this effort. Multi-objective optimization makes use of decomposition techniques like Adaptive Penalty Boundary Intersection (APBI) and Adaptive Neighborhood Adjustment (ANA). We break down each approach below.

Adaptive Penalty Boundary Intersection (APBI) is a technique for handling constraints in multi-objective optimization problems. Using penalty functions, it converts a limited optimization problem into an unconstrained one. Penalty settings are dynamically adjusted during optimization based on the seriousness of constraint violations. The unconstrained problem is solved using the APBI[20,21] method, which involves iteratively adjusting the penalty values. Finding a set of Pareto-optimal solutions that work within the constraints is the goal. By adjusting the penalty settings, the approach makes an effort to focus the search on feasible subsets of the Pareto front. Using the PBI decomposition technique to determine the feature that provides stable selection pressures for the population can be challenging when working with complex Pareto fronts. In contrast, when using a reasonable value for the penalty parameter, the PBI function shows promising results.

• Adaptive neighborhood adjustment is a method for dynamically modifying the neighbourhood size or search direction surrounding each solution in a multi-objective optimization process. It adjusts the search strategy depending on how closely solutions are clustered in the objective space. The scale of the neighborhood determines which alternatives are examined and which, if any, may be made better. Based on factors such as the number of solutions, the convergence rate, and the depth of exploration of the Pareto front, Adaptive Neighborhood Adjustment adjusts the neighborhood size. Equal time and effort must be devoted to both exploration and exploitation in order to maximize returns on investment.

Researchers have made numerous changes and improvements to the MOEA/D[22] algorithm. For instance, Chen et al. [23] presented the reference vector guided evolutionary algorithm (RVEA), which modifies the distribution of label weight vectors in the target space given the current Pareto solution set. Using differential evolution and polynomial mutation to produce progeny, MOEA/D-DE

[24], proposed by Huili et al., prioritizes sub-problems. MOEA/D-STM was developed by Li et al. [25] and relies on MOEA/D's stable matching model to evenly distribute solutions across all sub problems. Wang et al. [26] presented MOEA/D-GR, which uses a replacement technique to balance convergence and diversity in MOEA/D to solve MOPs with complex Pareto fronts. Yuan et al. [27] suggested MOEA/D-DU, which use the modernized Tchebycheff function and weight vector vertical distance, to strike a better balance between meeting and diversity. The last recommendation offered by Zhang et al. was to use the multi-objective evolutionary algorithm MOEA/D-DRA, which calculates the function value for each sub-utility problem.

If you want to improve your odds of evolving, maximize your utility function. ENS-MOEA/D was initially proposed by Zhao et al. [28], who studied the impact of neighborhood size on the effectiveness of the system. MOEA/D-DNS [29] by Zhou et al. improves upon some subproblems while mitigating others by focusing on the boundaries and near-boundaries. The MOEA/D-DN proposal by Wu et al. [30] considered the resources required for each individual subproblem. Striking a balance between convergence and diversity is difficult when optimizing for multiple, distinct subproblems at once [31]. This is because search ability reduces with increasing objective dimension. This work proposes MOGHHNS3/D-ANA, an innovative feature selection method based on a wrapper. It takes advantage of the adaptive neighbourhood adjustment (ANA) decomposition process and is based on a combination of Multi-Objective Guided HHO and NSGA-III.

The proposed wrapper method employs a classifier to evaluate the specified features. Scientists have used a wide variety of reliable and accurate classification methods, such as Naive Bayes, Decision trees, SVM, Artificial Neural Networks, and Convolutional Neural Networks. One such trustworthy and rapid learning algorithm is the Extreme Learning Machine (ELM) [32]. In this research, the Regulated Extreme Learning Machine (RELM) is used to analyze the selected characteristics. When applied to large, multi-labeled datasets, RELM may provide respectable classification results, which is not always the case with traditional classifiers [33]. The proposed MOGHHNS3/D-ANA method is compared to the multi-objective PSO (MO-PSO) [35], the multi-objective PSO with an adaptive operator selection mechanism (MOBGA-AOS) [34], the modified whale optimization algorithm (MWOA) [37], and the multi-objective PSO (MO-PSO) [36].The key benefits of the proposed feature selection approach MOGHHNS3/D-ANA are as follows:

- The multi-objective feature selection algorithm uses the ANA decomposition method to preserve archive variety. MOGHHNS3/D-ANA is a combination of the decomposition-based MO-GHHO and the NSGA-III algorithm that boosts HHO performance.
- The repository is an external archive where the non-dominated solutions are stored. By selecting

the leaders in accordance with the population density, a directed population archive guides the primary population to an accurate approximation.

- The selected features improve the classification performance of the Regulated Extreme Learning Machine. An ensemble of ELMs, SVMs, KNNs, CNNs, and RELMs are used to evaluate the effectiveness of the proposed feature selection technique.
- The suggested method is evaluated against the output of four well-known multi-objective evolutionary algorithms (MOBGA-AOS, BCNSG3, MO-PSO, MWOA) on a total of sixteen benchmark datasets.

Sections 2 and 3 provide a thorough history of the methods used in the proposed strategy. The proposed MOGHHNS3/D-ANA feature selection approach and classification strategy is elaborated forth in Section 4. The results and analysis of the experiments are presented in Section 5. The report concludes with several recommendations in Section 6.

II. RELATED WORK

Engineers and researchers use MOPs. The ideal solution may involve multiple competing goals. Recent MOPs have been solved with MOEAs [38]. Pareto dominance divides MOEAs into three categories: indicators [39, 41], decomposition [45, 46], and preference [47, 48]. Most early multi-objective evolutionary algorithms use Pareto dominance. Diego Oliva accepted the manuscript using a set-based technique. SetGA, a genetic algorithm, added a set-based Pareto dominance relation to NSGA-II's rapid non-dominance sorting method to solve optimization problems with three or more unknown objectives [49].Liu et al. presented the many-objective evolutionary algorithm, which uses one-to-one selection [50], convergence indicators, and population geographical diversity. The number of solutions covering the Pareto Front (PF) raises exponentially with objective dimensions, which can decrease the algorithm's search capability or stagnate it as the number of non-dominated solutions increases. Zhang et al. introduced MOEA/D in 2007. MOEA/D coevolves multi-objective issues using an aggregation function instead of Pareto dominance [51]. MOEA/D decreases computation complexity and improves convergence over Pareto-dominated. Label dependencies are ignored while choosing features. [52] Uses mutual information to value features. Mutual information is used to analyze labels and candidate features. Other features are not redundant or reliant. MDDM creates a lower-dimensional feature space from the original space to maximize label-feature connections.

The exhaustive search algorithm FOCUS [53] selects all feature subsets. This algorithm's comprehensive search takes longer on massive data. MIFS [54] classifies redundant and important information in supervised neural networks. Features with high redundancy and minimal information are redundant. Again, greedy searching limits MIFS [55]. [56] suggests supervised filter-FS. This fisher score algorithm ranks

characteristics independently. This algorithm's selected characteristics are unrelated and redundant after feature selection. [57] Creates quick discrete and continuous FS. CFS [40] selected features using heuristics. [58] Suggests a different feature selection strategy because the target class is important [42]. Most target class features share information. This strategy increased feature size and accuracy.

A redundancy and relevance approach is introduced. Conditional informative feature extraction reduces redundancy to improve feature set information. [60] Suggests Pearson correlation. The algorithm chooses subsets with the lowest validation error. We anticipated M nested subsets. Easily implemented and affordable. [61] Provides dynamic MI feature selection [43]. In [62], a filter-based Conditional Subset Assessment (CSA) method using entropy and MI was suggested to analyze high-dimensional datasets. Unlabeled features' mutual information (MI) was quantified. A BPSO method was proposed for MI and entropy assessment. A filter-based multi-objective feature selection (FS) method that uses entropy and MI and is compatible with SPEA2 and NSGAII was introduced in [63]. Additionally, [64] devised a filter-based CSA approach employing the generic filter algorithm. Most multi-objective optimization algorithms use NSGA. NSGAII was used to create a multi-objective FS framework [65].

[66] Introduces multi-objective selection feature for classification using a binary evolutionary algorithm and adaptive crossover operator. Altered search characteristics and five crossover operators are used. Every crossover operator's probability depends on evolution. WOA uses modified whale optimization. Early local optimal solution convergence in WOA. Combining genetic algorithm operators improves WOA. [67] A filter-based FS method using binary CSA and NSGA III is shown. Four gain ratio and mutual information-based multi-objective FS methods were proposed. They don't compare their evolutionary algorithm to others to see which delivers the best feature subsets.

In numerous ways, Adaptive Neighbourhood Adjustment (ANA) can help multi-objective optimization's APBI technique overcome constraints:

The APBI technique uses penalty functions to address limitations, but precise tuning is needed to balance feasibility and optimality. ANA adapts neighborhood size and search direction to limits. Without penalty parameters, constraint handling is flexible.

☐ APBI optimizes solutions along the feasible region boundary to locate feasible options. However, this may exclude interesting Pareto front regions from solution space exploration. To balance exploration and exploitation, ANA dynamically adjusts neighborhood size. While assuring solution feasibility, the algorithm may search diverse Pareto front regions.

☐ Penalty functions guide APBI to practical solutions. However, poorly calibrated penalty parameters may produce premature convergence or poor solutions. Dynamic neighborhood size modification using ANA enhances convergence. This allows finer solution exploration, which

may improve convergence to true Pareto-optimal solutions.

- Handling changeable Constraint Domains: Optimization issues may involve changeable domain constraints. Predetermined penalty parameters may hinder the APBI technique in such cases. ANA can adapt its neighborhood size to changing constraint domains. Optimizers can explore changing viable zones and maintain feasibility with this adaptability.

ANA may help the algorithm outperform APBI during optimization.

III. PRELIMINARIES

A. Harris Hawks Optimizer (HHO)

A robust HHO algorithm is developed according to Harris falcons' life. Harris falcon behavior is based on the way this type of falcon lives and hunts in the natural environment [7]. Using the HHO algorithm, two exploration strategies and four exploitation strategies are listed in the table below. In the exploration stage of HHO, two strategies are used. As part of the exploration phase, q, a random value between 0 and 1, is generated to determine which approach to take. The first method, which is a random approach, is used to hunt close to one of the other falcons if q is more than or equal to 0.5; however, if q is less than 0.5, the second method is employed in Eq. (1).

$$X(t + 1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2_{r_2}X(t)|q \geq 0.5 \\ (X_{rabbit}(t) - X_m(t) - r_3(LB + r_4(UB - LB)))q < 0.5 \end{cases}$$
(1)

$X_m(t)$ in Eq.(1) is calculated based on Eq.(2).

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

An alternative well balanced search strategy between exploration and exploitation phases is used in the HHO algorithm. In the exploration phase, algorithm optimization operations are performed effectively and promising solutions are employed while the number of repetitions and optimization operations are increased until the optimal answer is reached. Eq. (3) for modeling mathematics:

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

The algorithm starts the exploration phase if we considered the response is $|E| \geq 1$ and if the answer is $|E| < 1$ then we consider the algorithm starts the exploitation phase and the E value drops as the number of repeats increases. During the productivity phase, HHO uses four main tactics. Depending on the value of 'E', it uses soft besiege or soft besiege with progressive rapid dives techniques; if we consider r < 0.5 and $|E| \geq 0.5$ then it will be taken as soft besiege with progressive rapid dives, if we consider r < 0.5 and $|E| < 0.5$ then it will be taken as hard besiege with progressive rapid dives. Additionally, since the rabbit is fully awake at this point, it is easily able to flee, and the hawks have a hard time catching it, according to Eq. (4).

$$X(t + 1) = C(t) - E|JX_{rabbit}(t) - X(t)|$$
(4)

$$\Delta X(t) = X_{rabbit}(t) - X(t)$$
(5)

In Eq. (4) X stands for the distance that can be obtained by Eq. (5), In Eq. (5), we calculate the distance from the chosen hawk to the rabbit and E is that distance as calculated by Eq. (3). The equation $J = 2(1 - r_5)$ can be used to determine J, which is also the rabbit's escape energy. According to the mathematical model of this motion, rabbits lose their ability to flee and are preyed upon by hawks using Eq. (6).

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

According to Eq. (7), rabbits can escape while still being under a soft siege when E < 0.5 but r < 0.5. However, this strategy is more intelligent than the previous one.

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

The two modes are compared to the existing result in Eq. (9) and LF stands for levy flight function which is used to enhance performance.

$$Z = Y + S \times LF(D)$$
(8)

According to Eq. (8), levy flight is defined in the problem dimensions as LF(D), and random numbers between 0 and 1 are defined as 'S' shown in Eq. (9).

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

In Eq. (9), there is a fixed value of 1.5 for β and the random numbers u and v are between 0 and 1.

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

Eq. (6) is superior to the current solution, according to Eq.(7). If the answer derived using Eq. (10) seems superior to others, it replaces the current solution; otherwise, it is compared with the current solution. When _E_ 0.5 and r 0.5, a severe besiege is carried out before the surprise pounces to trap and kill the rabbit because the rabbits lack the stamina to flee. Eq. (7) and Eq. (8) are applied based on Eq. (10).

In this approach, if the result of solving Eq. (7) takes the place of the existing solution since it is more effective. In any other case, if the answer derived from Eq. (8) takes the place of the existing solution since it is more effective. By modelling the behavior of Harris falcons mathematically and reproducing the lifestyle and hunting of this species of falcon in the natural environment, the HHO algorithm has developed a reliable algorithm.

The approach must be explained using the symbols that are presented in TABLE I, along with the explanations that are provided for each symbol. An explanation of this algorithm is provided in the following paragraphs.

TABLE I
HHO ALGORITHM PARAMETER'S DESCRIPTION

DESCRIPTION	SYMBOL
Hawks' locations, position vectors	X_i, X
Hawk's position	X_m
Position of a random hawk	X_{rand}
Random numbers inside (0,1)	$r_1, r_2, r_3, r_4, r_5, q$
Escaping energy, the initial state of energy	E, E_0
Position of rabbit (best agent)	X_{rabbit}
Dimension, upper and lower bounds of variables	D, LB, UB
Swarm size, iteration counter, the maximum number of iterations	N, t, T

B. *NSGA*-III

The NSGA II is enhanced for optimizations with multiple objectives. NSGA III chooses the best solutions based on a reference point method rather than crowding distance as it did in NSGA II. The NSGA III is composed similarly to the NSGA II. To generate the initial population P_t of size N and the subsequent population Q_t of the same size, mutation and cross operators are employed. Q_t and P_t are mutual to create the new population R_t, which has a size of 2N. The best N individuals are picked to demonstrate that optimal solutions may resist evolution. Once the initial level of non-dominated individuals is determined, the non-dominated solutions are added to S_t until S_t contains at least N individuals for the first time. The layer in this case that caused the population size to surpass N is F_L. The NSGA-III links people together and normalizes each aim using reference points. The line joining the reference point and the coordinate's origin is known as the reference line. If the chosen person is connected to the current reference point, the reference point's niche number is updated. A random person from F_L is chosen to join S_t.

The NSGA II is enhanced for optimizations with multiple objectives. NSGA III chooses the best solutions based on a reference point method rather than crowding distance as it did in NSGA II. The NSGA III is composed similarly to the NSGA II. To generate the initial population P_t of size N and the subsequent population Q_t of the same size, mutation and cross operators are employed. Q_t and P_t are combined to create the new population R_t, which has a size of 2N. The best N individuals are picked to demonstrate that optimal solutions may resist evolution. Once the initial level of non-dominated individuals is determined, the non-dominated solutions are added to S_t until S_t contains at least N individuals for the first time. The layer in this case that

caused the population size to surpass N is F_L. The NSGA-III links people together and normalizes each aim using reference points. The line joining the reference point and the coordinate's source is known as the position line. If the chosen person is connected to the current reference point, the reference point's niche number is updated. A random person from F_L is chosen to join S_t.

C. *Regulated Extreme Learning Machine (RELM)*

ELM has received a greater amount of attention in recent years as the supervised learning methodologies have progressed. When it comes to ELM, there are certain learning criteria that may be utilized to accomplish output weights, and there is no need for learning for the hidden layer. The bias values of single-hidden layer feed forward networks (SLFNs) and the parameters of input weights can be derived in a random fashion throughout the training process. In comparison to previous methods of training, the ELM has the potential to greatly improve both performance and the rate at which one learns. Due to the efficacy of ELM, it may be utilized in a wide variety of applications, including face recognition as well as medical analysis and diagnosis. SLFNs are believed to be generalized versions of the ELM model, which does not require the hidden layer parameters to be tuned. It is possible to define the generalized output function of ELM using the equation (11) that is described below.

$$f_L(\mathrm{x}) = \sum_{i=1}^{L} \beta_i h_i(x) = \mathrm{h}(\mathrm{x})\,\beta \qquad (11)$$

Between the output node and hidden layer of L nodes, the output weight's vector can be expressed as given in Equation (12)

$$\beta = [\beta_1, \beta_2, \dots \beta_L]^T \qquad (12)$$

With respect to input x, the output vector of the hidden layer can be expressed as given in Equation (13).

$$h(x) = [h_1(x), h_2(x), \dots, h_L(x)] \qquad (13)$$

From input space of D-dimension to hidden layer feature map H of L-dimension, the dataset is mapped according to h(x). Here, a feature map is represented by h(x). Based on the Bartlett's theory, the norm of weight is smaller and the network's generalization performance is better as the feed forward neural network's training error is smaller. To minimize output weight norm and training error, ELM model can be written as in Equation (14),

$$min \, \|H\beta - T\|^2 \text{ and } \|\beta\| \qquad (14)$$

In Equation (15), the H representation of the former is the output matrix of the hidden layer, and it may be written in the same way as the previous matrix. For the purpose of enabling the SLFN to learn from data, numerous algorithms have been created. There is a reputation for the RELM's superior and lightning-fast performance. In accordance with the ridge regression theory, the RELM can be represented by utilizing Equation (16).

$$Hd = \begin{bmatrix} hd(x_1) \\ hd(x_2) \\ hd(x_3) \end{bmatrix} =$$

$$\begin{bmatrix} hd_1(x_1) & \dots & hd_L(x_1) \\ . & . & . \\ . & . & . \\ . & . & . \\ hd_1(x_N) & \dots & hd_L(x_N) \end{bmatrix} \quad (15)$$

$$\beta = \left(\frac{l}{\lambda} + H^T H\right) H^T Y \quad (16)$$

Where $Y = H\beta$ and λ represents the regularization coefficient.

IV. PROFFERED METHODOLOGY

A. Fitness value Evaluation and Objective Function

The primary goal of the fitness function should not be to maximize classification accuracy but rather to decrease classification error. The size of the answer is considered by the second fitness function. In order to assess the solutions, the k-nearest neighbors (k-NN) classifier is employed. With k-NN, n-fold cross validation is used. To determine the starting fitness function, one can apply the following equation (17).

$$\min(f_1) = \left(\frac{1}{n}\sum_{l=1}^{n} \frac{N_{Error}}{N_{All}}\right) \times 100\% \quad (17)$$

Where 'X' is the feature, where 'N_{Error}' value represents the number of incorrectly predicted instances and 'N_{All}' value represents as the total number of occurrences Equation can be used to calculate the second fitness function (18).

$$\min(f_2) = \sum_{i=1}^{D} x_i \quad (18)$$

Where 'X' is the ith value in the X-feature. 'D' represents the number of original features.

a. External Archive Updating based on ANA Strategy

According to the DN technique provided by Wu Feng et al. [23], the size of the neighborhood should be determined by whether the sub-problem is a) the same across multiple time periods or b) different sub-problems at the same time. Adaptive neighbourhood adjustment (ANA) is advised due to the difficulty of solving sub-problems across many time periods and the requirement for varied neighbourhood sizes. The ability of the method to dynamically modify the neighbor size for each sub-problem is depicted in Fig. 2. An external archive is modified using a multi-objective optimization algorithm using the ANA (Adaptive Neighborhood Adjustment) method. The external repository houses the Pareto front approximation, a collection of high-quality, non-dominated solutions obtained through optimization. Using a method called "adaptive neighborhood adjustment," methods

based on ANA are able to keep the external archive current. In the repository, solutions are organized into "neighborhoods," each of which represents a group of similar solutions in objective space. By dynamically adjusting the size of the neighborhood and picking solutions based on proximity and quality, the archive is kept up-to-date. The size of a neighborhood is dependent on its density, convergence rate, and the level of exploration thus far. This modification guarantees that the neighborhood completely encompasses the distribution of solutions in the objective space. The repository chooses winning strategies depending on how they compare to the status quo. Only non-dominated or Pareto-optimal solutions are included to ensure the archive remains diverse and representative. The ANA strategy is used to refresh the external archive and modify the neighborhood size to improve the quality and coverage of the Pareto front approximation. During optimization, the adaptive neural network update algorithm for external archives efficiently explores and utilizes the search space, providing a large and evenly distributed pool of high-quality solutions.
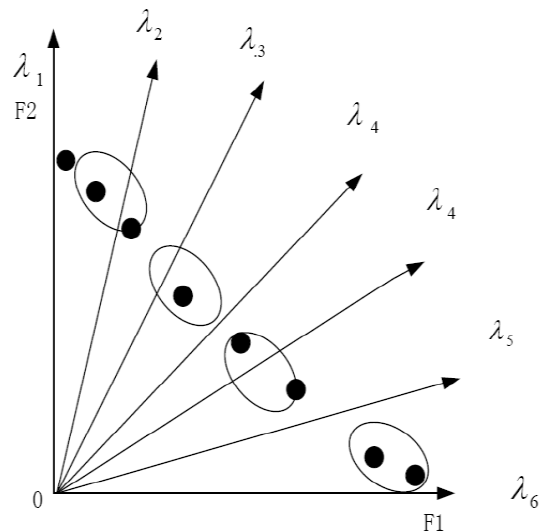
.

Fig 1. Standard Neighborhood Adjustment (NA) Strategy

Figure 2 illustrates how MOEA/D computes the utility of the objective function by aggregating the results of the sub-problem functions. Instead of focusing on the diversity of the sub problems, it takes convergence into account. In an ideal world, each individual should only be dealing with a single sub-problem. There are a number of persons in the vicinity of them. People are used as substitutes for solutions. This causes time to be wasted on refining ineffective solutions to sub problems while ignoring effective ones. It will impede both the accuracy and performance of the algorithm. In the beginning, we are going to assume that each of the sub problems has the same neighborhood size, also known as the standard neighborhood size.

$$T_i = T_{max} - T_{max} * \frac{1}{1 + e^{(-8*(\frac{gen}{Maxgen})-\beta}}$$

(19)

$$T_i = T_{max} - T_{max} * \frac{1}{1 + e^{(-8*(\frac{gen}{Maxgen})-\beta}} * \frac{\theta_i}{\theta_{max}}$$

(20)

During the evolutionary process, sub-problems that lack people have a restricted search space, which makes it harder to locate good candidate solutions. These sub-problems would benefit from having wider neighborhoods. On the other hand, sub problems involving numerous individuals can frequently be effectively solved, although this would call for neighborhoods to be scaled down. We propose certain improvements in order to solve these difficulties and increase the effectiveness as well as the accuracy of the MOGHHNS3/D-ANA. These adjustments will assist detect and address poor solutions in a more efficient manner.
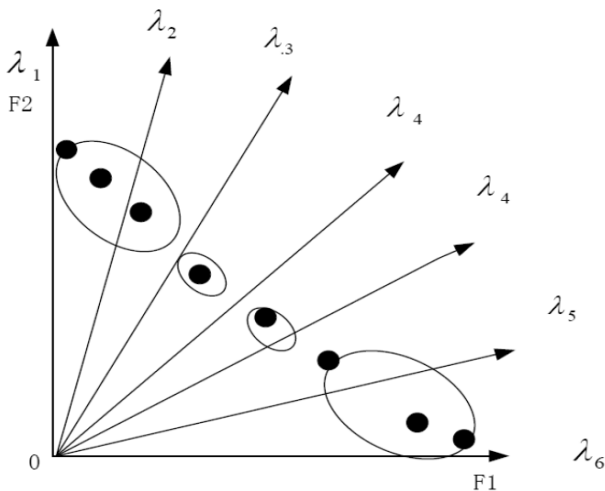
Fig 2. Adaptive Neighborhood Size Adjustment (ANA) Strategy

We consider a person to be a component of a problem's sub-issues if their location is near a sub-issue of that problem. Since this vector is identical to the sub-weight problem's vector, we can use it to calculate the vertical distance between the person and the sub-problem. We can conclude that no one is participating in the sub-problem if it does not affect any individuals. Right now, we're going to broaden the scope of our search by making our immediate neighborhood much larger. On the other hand, if there isn't much of a vertical disparity between any one person and this sub-problem, we can assume that there are probably other people linked to this sub-problem and thereby reduce the scope of our search. The NA (Neighborhood Adjustment) method is used, as shown in Figure 1. To account for the population's continuous change, we use Eq.(19) to adjust the neighborhood scale of each sub problem. Figure 2 also shows how the ANA (Adaptive Neighborhood Adjustment) method uses Eq. (20) to change the neighborhood size for every generation. You can find this equation in the reference [26]. In the same way that the

differentiation parameter controls how much a sub-problem's neighborhood scale changes over time, the angle I between the weight vectors of different sub-problems and the center vector controls how much neighbourhood scale changes across different sub-problems. Algorithm 1 provides a thorough explanation of this method.

Algorithm 1. Adjusting the size of a neighborhood adaptively(ANA)

Input: The largest possible number of neighborhoods -T_{max}

A group of constant reference vectors $W \leftarrow \{\lambda_1, \lambda_2, \lambda_3 ..... \lambda_N\}$

The differentiating factor-β;

Output: $T = \{T_1, T_2, T_3 ..... T_N\}$

For gen=1 to Maxgen do

$$T_i = T_{max} - T_{max} * \frac{1}{1 + e^{(-8*(\frac{gen}{Maxgen})-\beta}} * \frac{\theta_i}{\theta_{max}}$$

$T = Ceil (T_i)$;

end for

B. The Guided HHO's method for choosing leaders

The leaders have a vital role in directing the populace towards evenly distributed areas and gaining an accurate estimate of the genuine PF. In order to improve the algorithm's efficiency, a suitable leader selection technique is implemented, which can be executed in following phases.

- The crowding distance of each solution is calculated in the external archive.
- Effective leader selection can enhance the algorithm's performance by guiding the population towards evenly distributed regions and providing a dependable prediction of the genuine Pareto front.
- To accomplish this, follow these steps: sort the archive members in descending order based on their crowding distance, identify the topmost section of the archive as containing less crowded solutions, and randomly choose one solution from the predetermined upper portion of the organized archive to act as a leader for all solutions towards the least congested location. This approach ensures the maintenance of spread and Pareto front.

The HHO approach begins with a random generation of search agents and continues with an assessment of their fitness. The solutions that are not dominated are kept in a separate archive. The calculation of the crowding distance for each solution starts the primary iteration, which begins once the archive is built. The crowding distance is used to sort the non-dominated solutions in descending order, while Equations (7) and (9) are used to update the parameters. An external repository containing the final Pareto front is produced by the approach. The number of targets is M, and

the number of individuals in the external archive and primary population is N.

C. NSGAIII with MOGHHO

a. Selection of Reference points

It is possible to improve exploitation ability by using local search with Levy Flight. The reference points are defined via NSGA-III on a regularised hyper-plane. This normalised hyper-plane is inclined equally to all objectives and has an intercept of one on each objective axis. Equation can be used to compute the total reference points R for M objectives (21)

$$R = \binom{M + d - 1}{d} \qquad (21)$$

Where the given integer value is represented by 'd'.

b. Solutions encoding

Each population answer represents a workable scaling plan for the current auto-scaling stage. Each solution is represented as a vector of size $3 \times n$, where n is the number of instance categories used for auto scaling. The positions show the total number of on-demand instances to be acquired for all n instances (1, n).

c. Crossover

The crossover operator is utilized to generate offspring encoded solutions by combining the encoded solutions of parent pairs. This operator is applied to all pairs, and it employs a crossover distribution index $'D_c'$ and probability $'P_c'$. Equations (22) and (23) can be used to create the offspring solutions p_1 and p_2 from a given parent pair (23).

$$p'_{1i} = 0.5[(1 + B_i) \, p_{1i} + (1 - B_i) \, p_{2i}] \qquad (22)$$
$$p'_{2i} = 0.5[(1 + B_i) \, p_{1i} + (1 - B_i) \, p_{2i}] \qquad (23)$$

Where, p'_{1i} and p'_{2i} are used to represent the values for the solutions in the ith position, p'_1 and p'_2, respectively. the number $i=1,...,n$. The values of position 'i' are represented by p'_{1i} and p'_{2i} in the solutions p_1 and p_2, respectively. The term B_i is calculated using the Equation (24) and polynomial probability distribution.

$$B_i = \begin{cases} (2u_i)^{\left(\frac{1}{D_c+1}\right)}, & if \ u_i \le 0.5, \\ \left(\frac{1}{2(1-u_i)}\right)^{(1/(D_c+1)}, & if \ u_i > 0.5 \end{cases} \qquad (24)$$

In this case, u_i represents a random real number with the range [0, 1]. Where $'D_c'$ Serves as a representation for the specified non-negative real number. A big value of $'D_c'$ permitting a focused search provides a better potential to produce offspring solutions that are similar to those of the parents. The child solutions produced by a small value of $'D_c'$ permit a wide range of searches.

d. Mutation

NSGA-III utilizes the polynomial mutation operator to modify the encoded individuals obtained from the crossover operator. Using a mutation distribution index D_m and mutation probability P_m, where 'i' value ranges from 1 to n, the mutation operator is applied to each point 'i' of the encoded solutions. The operator modifies the ith location of an encoded solution 'p' creating a new value p'_i based on Equation (25).

$$p'_i = \begin{cases} p_i + d(p_i - L_i), & if \ u_i \le 0.5, \\ p_i + d(U_i - p_i), & if \ u_i > 0.5, \end{cases} \qquad (25)$$

The real number u_i which is chosen at random, stands in for the range [0,1]. p'_i stands for the value of the ith position of p. L_i and U_i, respectively, stand for the lower and upper boundaries of p's ith position. Equation (26) uses the polynomial probability distribution to calculate the value of d.

$$d = \begin{cases} (2u_i)^{(1/(D_m+1))} - 1, & if \ u_i \le 0.5 \\ 1 - \left(2(1 - u_i)\right)^{(1/(D_m+1))}, & if \ u_i > 0.5 \end{cases} \qquad (26)$$

D. Computational Complexity

The term "computational complexity" refers to the amount of resources needed to systematically apply an algorithm to solve a particular type of problem. In this context, computational time denotes the time taken to identify the optimal characteristics. The proposed method has a computational cost of $O(mn^2)$, where n represents the size of the population and m represents the number of chosen features. Computing the model parameters gets simpler as the computational cost of a model lowers with the amount of characteristics. It is also possible to reduce the amount of data storage that is required in order to store the properties of the model. Detailed explanation of the pseudo-code for the suggested method may be found in Algorithm 2.

V. EXPERIMENTAL RESULTS AND DISCUSSION

A. Datasets Description

The suggested feature selection strategy is tested on sixteen datasets from the UCI Machine Learning Library [68]. TABLE I provides a description of the datasets.

B. Pre-processing

The suggested method's first step is data pre-processing. A simpler and more useful representation is created out of the raw data. The data is normalised using the min-max method. The normalisation method can reduce training time. The dataset is normalised using Equation (27).

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \qquad (27)$$

Where X_{min} and X_{max} for the least and maximum values of each feature, respectively.

Algorithm 2. Pseudo code for MOGHHNS3/D-ANA algorithm
Initialize the main parameters N, $\bar{N}$, β, C
Generate a newly-formed population randomly within the range of 0 to 1.
Initialize the Archive Ar.
While t is less than or equal to t_{max},
Update Equation (6) to incorporate the decreasing factor.
For each solution in Ar,
Update the Archive using the Adaptive Neighborhood Adjustment Strategy **//ANA**
Sort out the archive members in ascending order ($\overline{P_0}$)
Set $P_t = P_0 \cup \overline{P_0}$
.   Calculate the fitness of each feature ($\bar{P}_t$) in $\overline{P_{t+1}}$
If 'r' is less than 0.5,
$\overline{P_{t+1}}$ Position updated using Equation (7)   **//Guided HHO**
Else,
$\overline{P_{t+1}}$ Position updated using Equation (9).
End if.
End for.
Select candidates for the NSGA-III.   **//NSGA-III**
Apply the crossover and mutation methods.
End while. When the stop criteria are met, terminate the process.
Select the desired features and return them.
End.
End.

TABLE II
DATASET DESCRIPTION

S. N o	DATASETS	NO. OF INSTANCES	FEATU RES	CLASSE S
1	Dermatology	366	35	6
2	Glass	214	10	6
3	Audiology	226	70	24
4	Waveform	5000	41	3
5	Zoo	101	16	7
6	Vehicle	846	18	4
7	Mnist	70000	780	10
8	Covtype	581012	54	7
9	Letter	20000	16	26
10	Pendigits	10992	17	10
11	Iris	150	4	3
12	Daily Sports	9120	5625	19
13	Vowel	990	14	11
14	SEMG	3000	2500	6
15	Seeds	210	7	3
16	Satimage	6430	36	6

C. Parameter Settings

The optimization parameters for the proposed feature selection technique are detailed in Table II. The table also includes the parameters of other feature selection methods used for comparison.

TABLE III
COMPARATIVE OPTIMIZATION ALGORITHMS PARAMETERS

METHOD	PARAMETER
MOGHHNS3/D-ANA	No of Harris falcons : 'N' = 50
	β =6 and C=2
	Mutation rate=1/N
	Neighborhood Size T=0.1*1/N
	NS pool T= {0.5* T,T,1.5*T}
	number of update neighborhood's solutions - nr =0.01*N
MOBGA-AOS	Crossover = 0.69,
	LP=5,M=2.
	Mutation = 0.05,
	Q=5.
MWOA	C = 0.6,
	Mutation rate = 0.05
MO-PSO	w_{min}= 0.4
	w_{max}= 0.9
BCNSG3	β = 0.6,
	Crossover = 0.5

D. Performance Measures

The suggested feature selection (FS) method is evaluated

using the IGD measure, and its evaluation includes the calculation of the standard deviation and means values. Additionally, the accuracy, hamming loss, and ranking loss, recall, precision, F1 score are used to assess how well the suggested feature selection method works.

Accuracy: The accuracy of the measurement is calculable by applying Equation (28).

$$\text{Accuracy Performance} = \frac{1}{|Z|}\sum_{i=1}^{|Z|}\frac{|\gamma_i \cap P_i|}{|\gamma_i \cup P_i|} \tag{28}$$

Precision: The precision of the measurement is calculable by applying Equation (29).

$$\text{Precision Value} = \frac{1}{|Z|}\sum_{i=1}^{|Z|}\frac{\gamma_i \cap P_i}{|P_i|} \tag{29}$$

Recall: The recall of the measurement is calculable by applying Equation (30).

$$\text{Recall value} = \frac{1}{|Z|}\sum_{i=1}^{|Z|}\frac{\gamma_i \cap P_i}{|\gamma_i|} \tag{30}$$

F1 score: The F1 Score of the measurement is calculable by applying Equation (31).

$$micro\ F1\ Score = \frac{2 \times \sum_{i=1}^{|Z|}\|\gamma_i' \cap \gamma_i\|}{\sum_{i}^{|Z|}\|\gamma_i\| + \sum_{i}^{|Z|}\|\gamma_i'\|} \tag{31}$$

Hamming loss Calculation: The hamming loss of the measurement is calculable by applying Equation (32).

$$H_{loss}(Z) = \frac{1}{|Z|}\sum_{i=1}^{|Z|}\frac{1}{|L|}|\gamma_i \oplus P_i| \tag{32}$$

Ranking loss Calculation : The r used to assess how well the suggested feature selection method works. ranking loss of the measurement is calculable by applying Equation (33)

$$R_{loss}(Z) = \frac{1}{|Z|}\sum_{i=1}^{|Z|}\frac{|\{(a,b)a \in \gamma_i, b \in \overline{\gamma_i}, \varphi_{i,a} \leq \varphi_{i,b}\}|}{|\gamma_i||\overline{\gamma_i}|} \tag{33}$$

To obtain the set of projected labels for an instance "i", the intersection of γ_i and γ_i' prime" is calculated and represented by the symbol "∩" and denoted as φ_i. The number of labels in the multi-label dataset can be denoted by $|L|$. The dataset itself possibly represented as $Z=\{(Z_i, \gamma_i)|1 \leq i \leq |Z|\}$, where Z_i, indicates an instance $\gamma_i \in L$ represents the subset of labels associated with that instance. It is assumed that the predicted label subsets are symmetric to the true label set and are represented by the symbol $\oplus$. The set of labels predicted by a classifier for instance 'i' is represented as P_i .

E. Experimental Results

The performance of the suggested MOGHHNS3/D-ANA strategy for feature selection is shown to be superior to that of competing methods, as shown in Figure 4. This is the case across all sixteen datasets. The optimal size of the feature subset is signified along the x-axis, while the classification error is shown along the y-axis. According to the results of the Pareto front, the method that was suggested performs better than any other strategy in terms of reducing the number of classification errors across all sixteen datasets.

Figure 3 examines the differences and similarities between the IGD values of the ANA, APBI, Tchebycheff (TCH), MOBGA-AOS, MWOA, MO-PSO, and BCNSG3 algorithms. Penalty-based boundary intersection (PBI) and TCH are two scalarizing functions that are frequently utilized, and this comparison is carried out utilizing both of them. The creation of a wide variety of solutions is encouraged by higher values of the scalarizing function, whilst the convergence towards optimal solutions is facilitated by using smaller values. Under conditions in which the parameter falls within the range of 0 to 1, the ANA method demonstrates its highest level of performance. Multiple sample points, specifically 1000, 2000, 3000, 4000, and 5000 generations, were used to evaluate its performance across a range of generations. When the values of the scalarizing function increase, the ANA technique moves away from the estimated nadir point. This is in contrast to the TCH and PBI approaches, which use the estimated ideal point as a reference and converge solutions towards it. All aspects of performance are tested, and the ANA algorithm displays superior performance when compared to the TCH algorithm. On the other hand, when confronted with a large number of targets, WS arises as a more competitive alternative. Furthermore, when the objective number is between 0 and 1, ANA outperforms APBI in terms of performance, and the superiority of ANA becomes even more evident as the objective number increases. Across all sixteen datasets, the ANA approach displays improved performance in comparison to MOBGA-AOS, MWOA, MO-PSO, and BCNSG3, which ultimately results in a higher score.

The results of the comparison are presented in Figure 4, which depicts the comparison between the length of the selected attributes and that of the existing techniques. Figure 4 also illustrates the characteristics of the selected attributes. It is possible that the implementation of the reduced feature selection method will result in an improvement in both the speed at which the classifier is trained and the accuracy of the classification. It is desirable to employ a technique that involves a smaller number of selected qualities in order to obtain a better level of precision. This is because it is more likely to produce accurate results. This is demonstrated in Figure 4, which shows that the suggested method makes use of a smaller number of features selected in a manner that is consistent across all sixteen examples. It is advised that the approach be used for the SEMG dataset since it has a maximum feature length of 225, which is a value that is shorter than the feature lengths of the strategies that came before it.
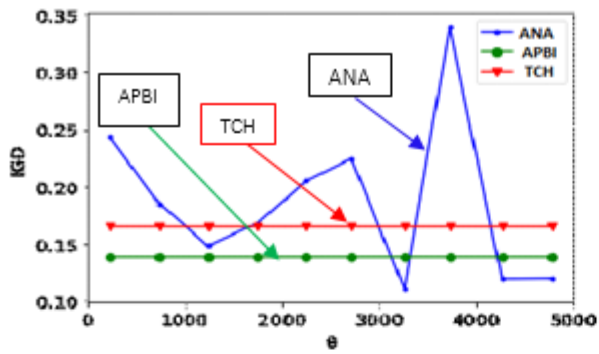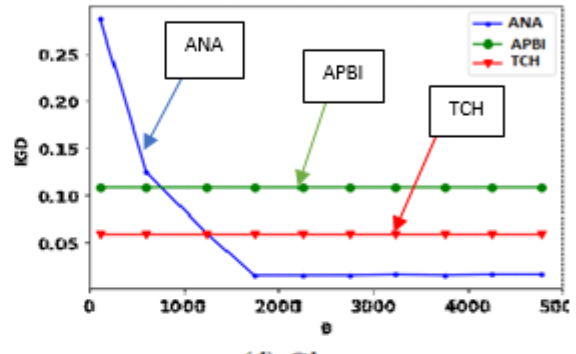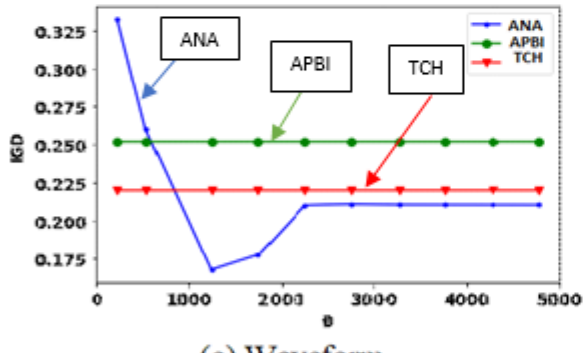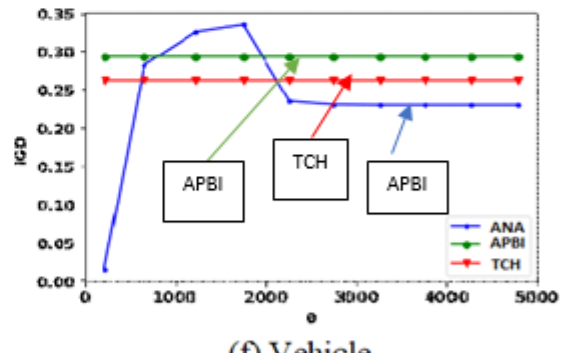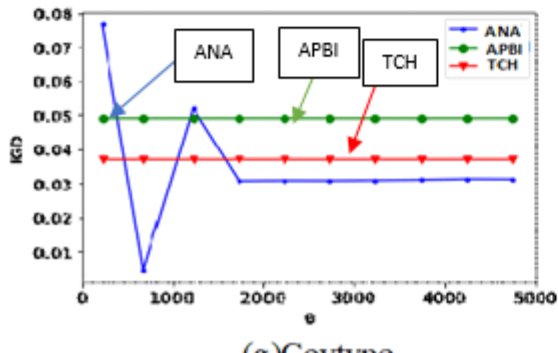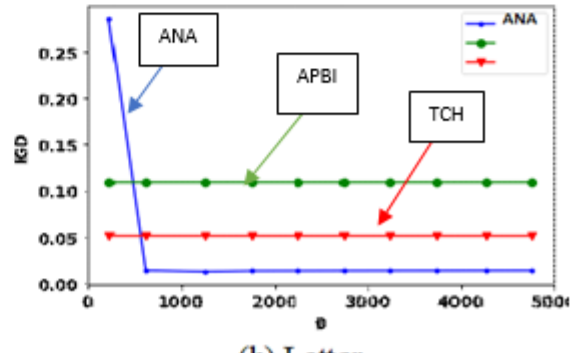
(a) Dermatology
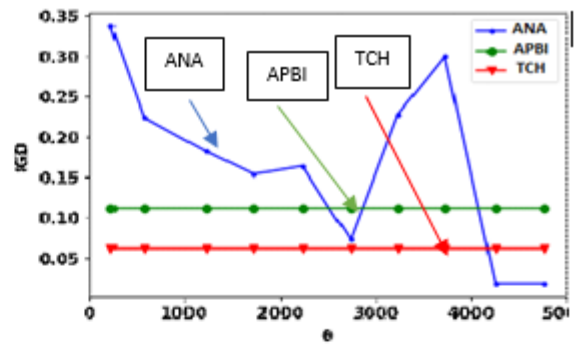
(b) Audiology
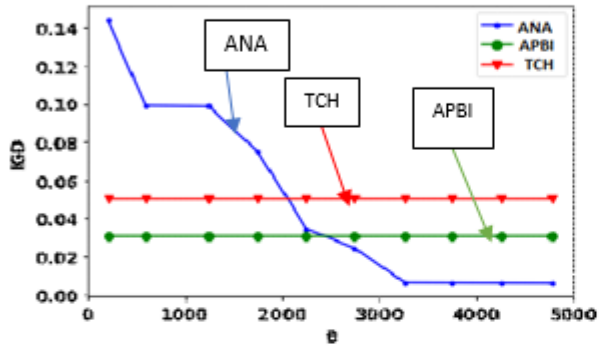
(c) Zoo

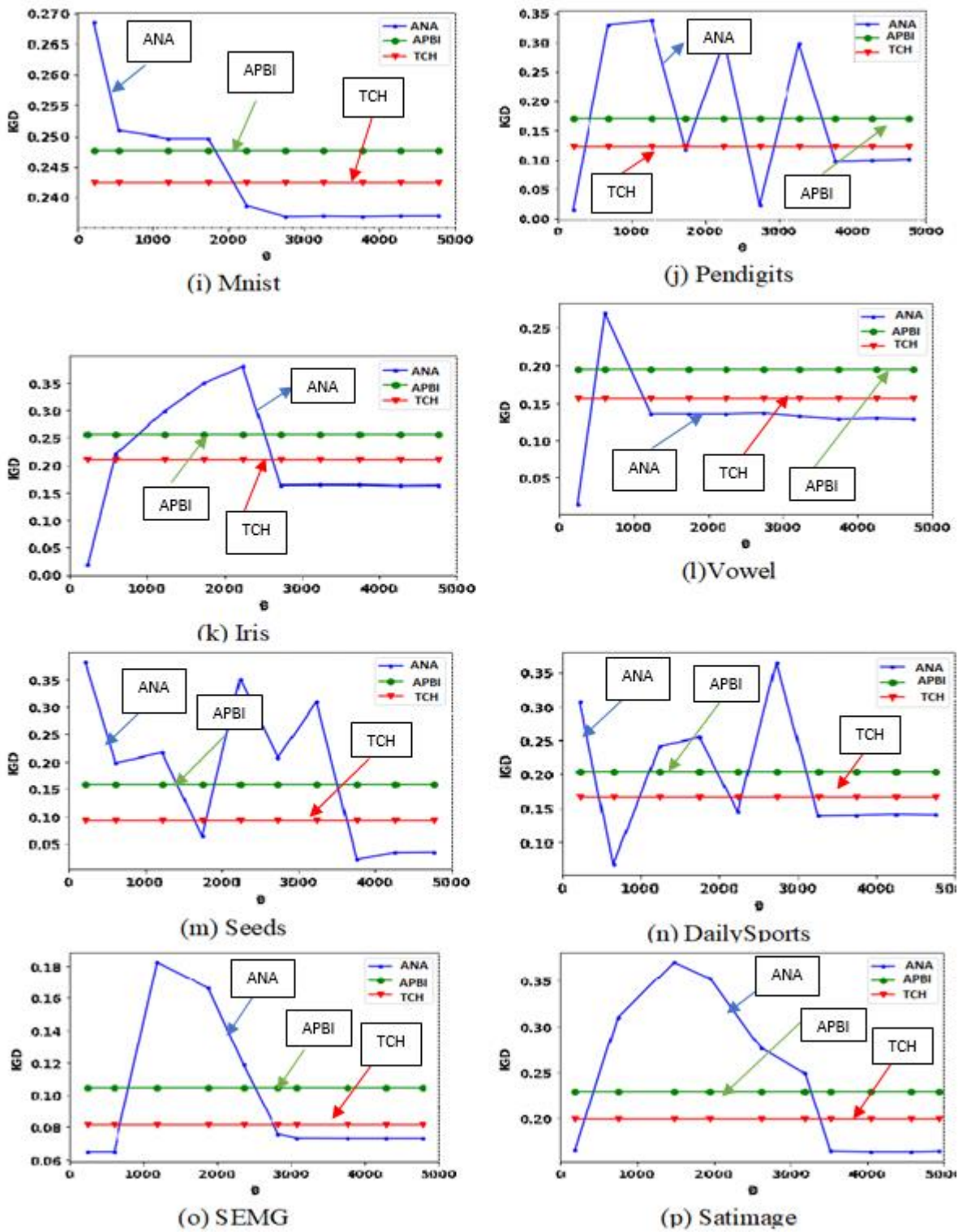(d) Glass

(e) Waveform

(f) Vehicle

(g) Covtype

(h) Letter

Fig 3. Pareto front analysis on all Datasets (a)Dermatology (b) Audiology (c) Zoo dataset (d) Glass (e) Waveform (f) Vehicle (g) Covtype (h) Letter (i) Mnist (j) Pendigits (k) Iris (l) Vowel (m) Seeds (n) DailySports (o) SEMG (p) Satimage

The BCNSG3 algorithm had the highest value of 232 for the selected qualities, followed by MO-PSO (283), MO-BGA-AOS (286), and MWOA (228). The suggested method chooses fewer features than previous methods. This suggests that the described method may be more precise. The MOGHHNS3/D-ANA approach evaluates all sixteen datasets using the IGD metric. We may evaluate the recommended method's dispersion and convergence using the IGD. Accuracy increases with IGD values below 0. Fig. 5(a) and Fig.5(b) shows the analysis of Standard Deviation value and

Mean value on different Datasets. The MOGHHNS3/D-ANA method beats MOBGA-AOS, MWOA, BCNSG3, and MO-PSO. The suggested feature selection method reduced standard deviations and mean values for all sixteen datasets when related to alternative methods. The recommended technique uses fewer attributes to improve accuracy with less effort. The current MOBGA-AOS, MWOA, BCNSG3, and MO-PSO algorithms determine the Standard Deviation values for 16 datasets in TABLE IV. Current approaches show 16 datasets in TABLE V.

TABLE IV
COMPARATIVE EVALUATION OF STANDARD DEVIATION

Datasets	Proposed Method	MWOA	MO-PSO	BCNSG3	MOBGA-AOS
Dermatology	0.0082	0.00951	0.0112	0.00992	0.01033
Audiology	0.0004	0.0008	0.0018	0.0013	0.0016
Zoo	0.0274	0.0326	0.0432	0.0357	0.0397
Glass	0.01529	0.01562	0.01998	0.01795	0.01918
Waveform	0.01206	0.01286	0.01437	0.01317	0.01325
vehicle	0.00405	0.00427	0.00457	0.00438	0.00447
Covtype	0.00582	0.00613	0.00687	0.00635	0.00658
letter	0.00567	0.00592	0.00625	0.00607	0.00613
mnist	0.00749	0.00768	0.00795	0.00772	0.00787
Pendigit	0.0057	0.0067	0.0085	0.0073	0.0078
iris	0.0003	0.0007	0.0018	0.0013	0.0017
Vowel	0.00276	0.00287	0.00335	0.00298	0.00305
Seeds	0.00245	0.00258	0.00285	0.00267	0.00278
Daily and Sports Activities	0.0053	0.0067	0.0073	0.0073	0.0075
SEMG	0.04531	0.04642	0.04796	0.04715	0.04744
Satimage	0.001313	0.001498	0.001775	0.001547	0.001683
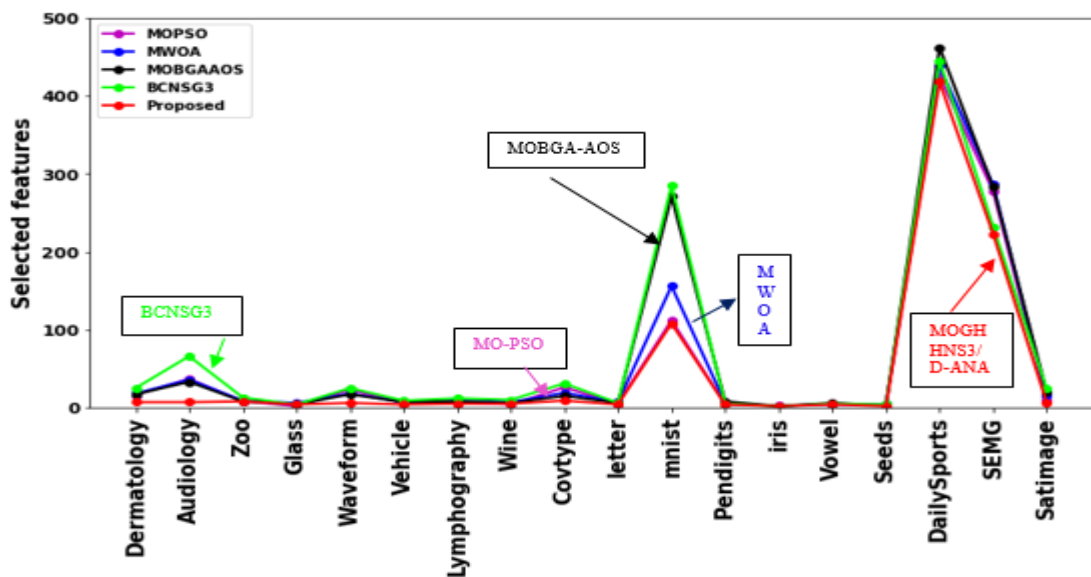
Fig 4. Comparison for Selected features

Comparative Evaluation of Standard Deviation

(a)   Standard deviation
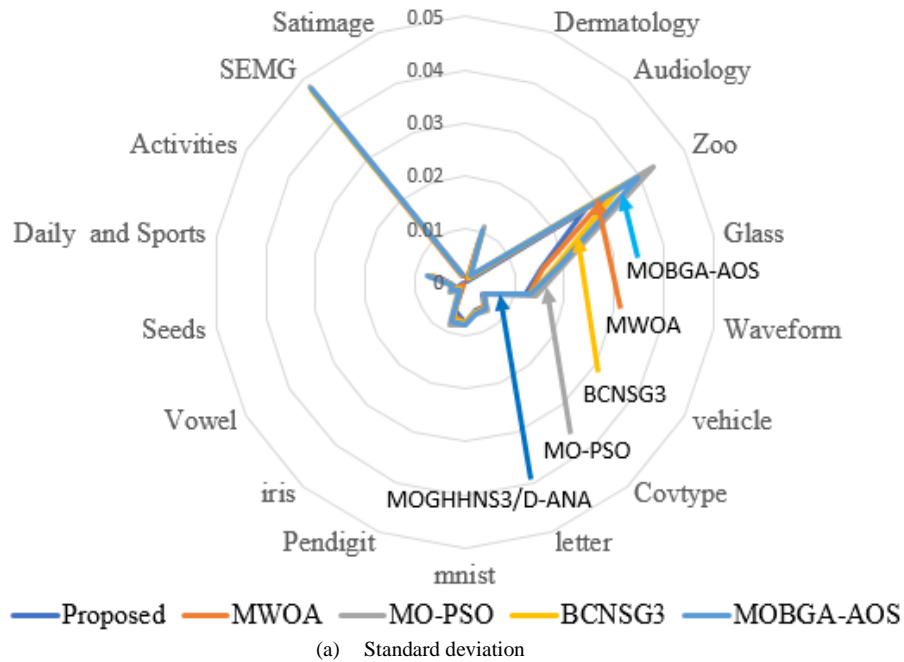
Comparative Evaluation of Mean Value
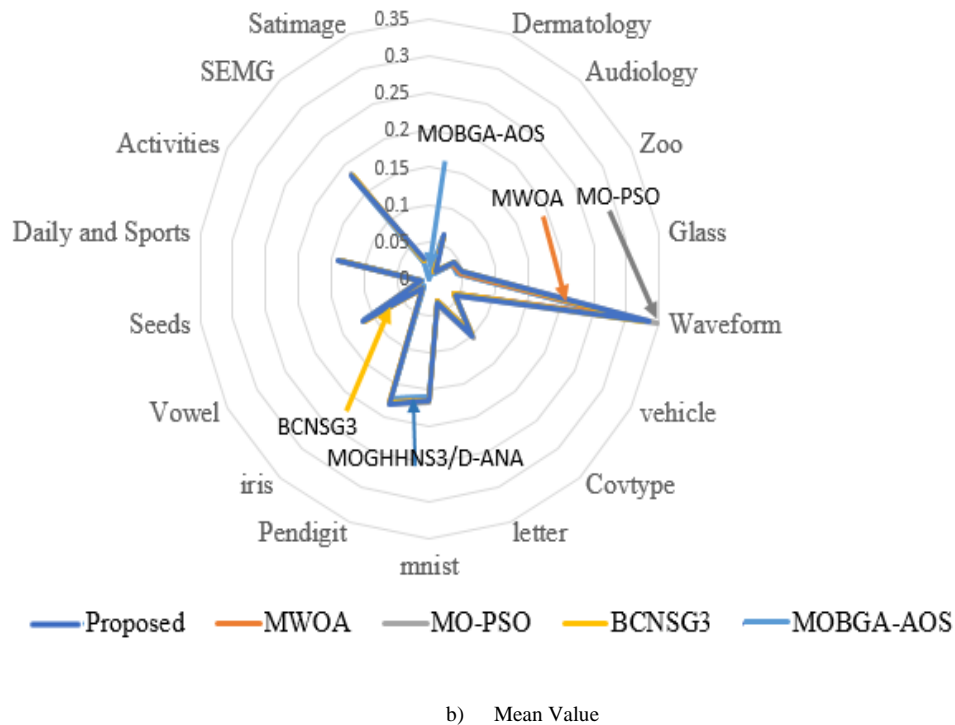

b)   Mean Value

Fig 5. (a) Standard deviation and b) mean value –Radar Graphs

TABLE V
COMPARATIVE EVALUATION OF MEAN VALUE

Datasets	Proposed Method	MWOA	MO-PSO	BCNSG3	MOBGA-AOS
Dermatology	0.05831	0.06078	0.0627	0.06124	0.0619
Audiology	0.00967	0.01082	0.0132	0.01156	0.0128
Zoo	0.03885	0.03954	0.04357	0.04152	0.04264
Glass	0.04262	0.04632	0.0501	0.04885	0.04974
Waveform	0.29815	0.30456	0.34586	0.32797	0.33453
vehicle	0.04102	0.04182	0.04463	0.04259	0.04387
Covtype	0.09551	0.09754	0.10452	0.09837	0.09971
letter	0.03135	0.03248	0.03679	0.03353	0.03532
mnist	0.16071	0.16345	0.16658	0.16448	0.16524
Pendigit	0.1734	0.1769	0.1804	0.1782	0.1796
iris	0.01452	0.01579	0.01824	0.01624	0.01763
Vowel	0.11247	0.11356	0.11683	0.11421	0.11542
Seeds	0.01152	0.01258	0.01593	0.01379	0.01437
Daily and Sports Activities	0.13681	0.13827	0.14068	0.13843	0.13984
SEMG	0.17895	0.17945	0.18248	0.18042	0.18163
Satimage	0.02245	0.024824	0.029124	0.026439	0.028213

Fig.6 illustrates that hamming and ranking loss values of all feature selection algorithms like MWOA, BCNSG3, MO-PSO, MOBGA-AOS, MOGHHNS3/D-ANA box plots. Upper and top percentiles of 0.61 and 0.61 for the suggested technique indicate minimal hamming and ranking losses. However, existing approaches have upper quartile values from 0.67 to 0.79, therefore the box plot's lowest value fits the suggested method. With smaller box median lines (0.04) and median lines (0.031), the recommended method outperforms earlier methods in ranking and hamming loss. Box plots show no outliers because all methods perform similarly. Reduce hamming and ranking losses to improve feature selection. Boxplots show data stability. Splitting the data evenly, the top half median is the third quartile. Check box lengths or interquartile ranges to see how samples distribute data. Shorter boxes have less data volatility, but longer boxes bias data. Boxes dictate numbers of data points per group. It is common practice to scale the box width to the standard error, which is equal to the square root of the data points. Notate the points for each group next to their names if the width of the box is not known.

(a) Ranking loss

(b) Hamming Loss

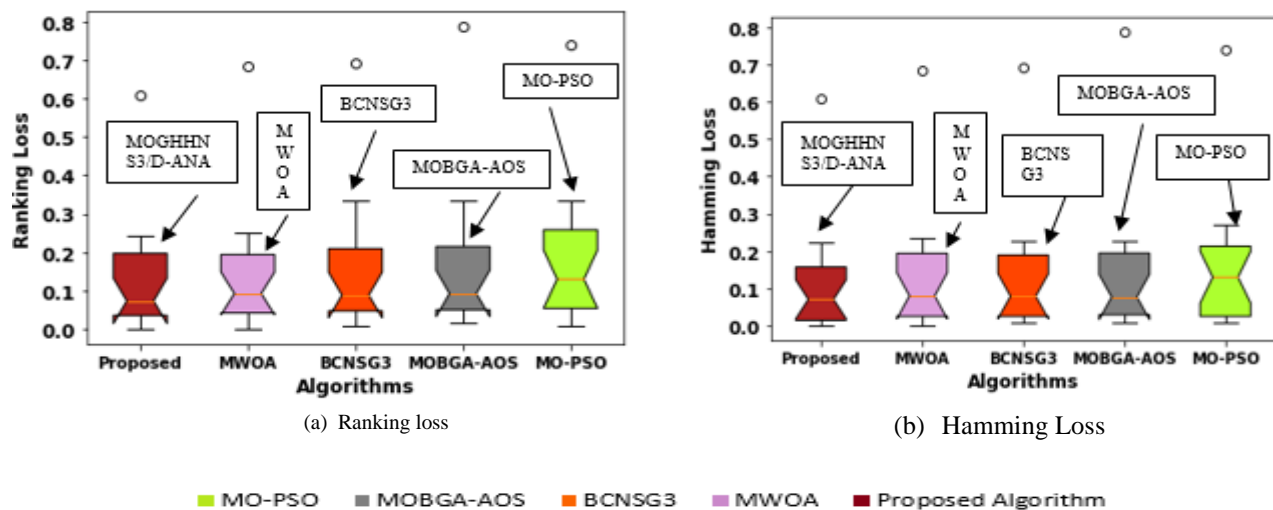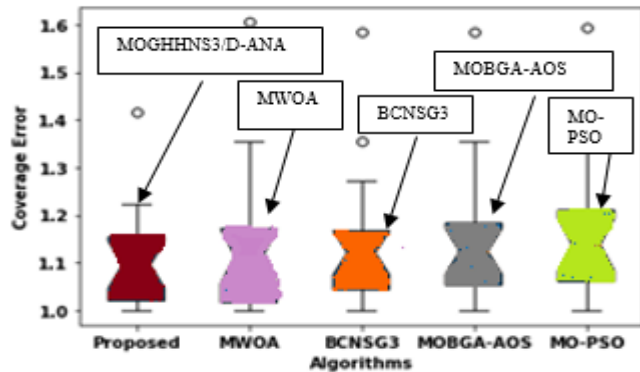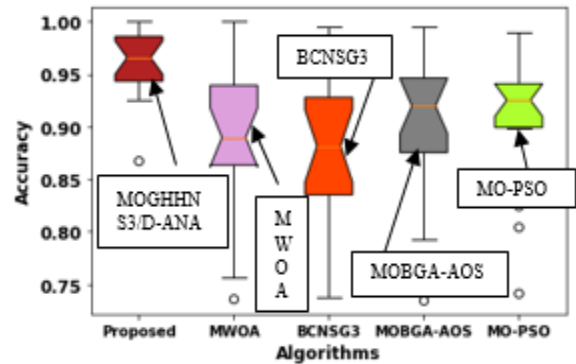■ MO-PSO ■ MOBGA-AOS ■ BCNSG3 ■ MWOA ■ Proposed Algorithm

Fig 6. Box plot analysis of Ranking loss and hamming loss

Fig. 7 compares the proposed approach to existing methods in accuracy, precision, recall, F1 score, coverage error, and runtime. As shown in Fig 7. (a), the suggested technique has a coverage error upper percentile of 1.72, lower than MOBGA-AOS, MWOA, MO-PSO, and BCNSG3 algorithms, which range from 1.91 to 1.98. The proposed method has the lowest upper quartile value in the box plot and a lower median line (1.1) than previous methods. From Fig. 7(b), which compares the proposed method to the Zoo dataset, the recommended method has 100% accuracy. Compares box plot analysis accuracy across all datasets to other approaches. The box

plot's top quartile value for the suggested approach is 100%, higher than MOBGA-AOS, MWOA, MO-PSO, and BCNSG3 algorithms' 99.5% to 99.8%. The proposed method also offers a higher median box line (98%) than previous methods. See also Fig. 7. (c),(d),(e) demonstrate how the proposed technique beats existing strategies in precision, F1 score, and recall. The box plot shows that all approaches perform better on average. The proposed strategy outperforms existing methods in runtime, coverage error, F1 score, accuracy, and precision.
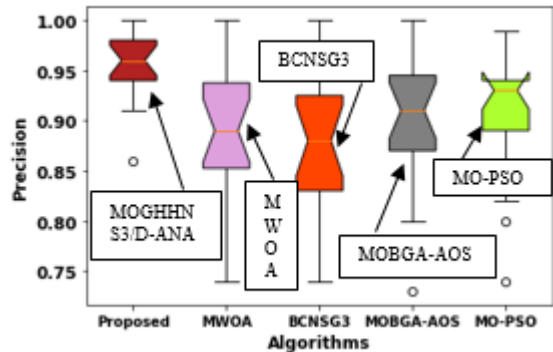

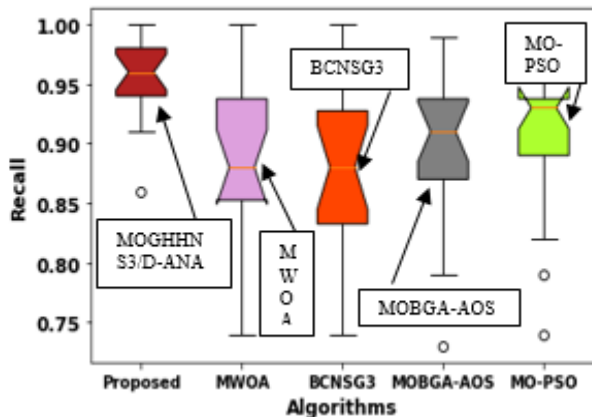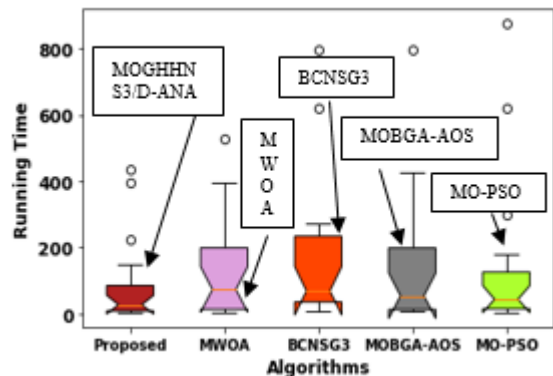(a) Box plot analysis -Coverage Error

(b) Accuracy Performance
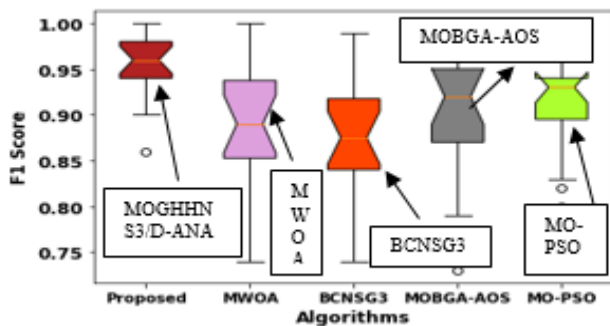
(c) Box plot analysis -F1 score

(d) Box plot analysis -Precision

(e) Box plot analysis -Recall

(f)   Graph for Computational complexity

Fig 7. Box plot analysis is used to compare performance

Fig.7.(f) represents the computational complexity of proposed algorithm MOGHHNS3/D-ANA compared with different existing algorithms using box plot analysis. MOGHHNS3/D-ANA takes 425.7 seconds, MOBGA-AOS and MWOA 535.8 seconds, MO-PSO 621.7 seconds, MO-SG3 785.3 seconds, and BCNSG3 864.2 seconds. The suggested feature selection method computes 13.31 seconds longer than previous methods. Calculation delay is in seconds. This proves that the suggested feature selection method may work with smaller features and still get good results. See how the suggested feature selection method stacks up against RELM, SVM, KNN, CNN, and an ensemble of ELMs in terms of accuracy and training time in Fig. 8 [69]. RELM is less accurate than ELM ensembles. The ensemble of ELMs is more accurate than RELM but takes longer to train.
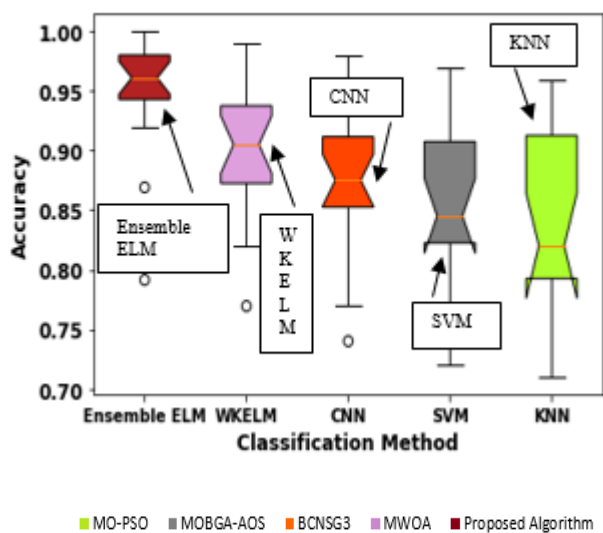

Fig 8. Comparison of classifier Accuracy

VI. CONCLUSION

This study uses the MOGHHNS3/D-ANA and the Multi Objective Guided Harris Hawks Algorithm, a novel feature selection approach based on wrapper method that updates the archive using the adaptive neighborhood adjustment (ANA) decomposition method, to optimize MOEA performance. A guided population archive selects leaders based on density to approximate the primary population more accurately. We assessed MOGHHNS3/D-ANA on sixteen benchmark datasets for precision, F1 score, accuracy, ranking loss, recall, hamming loss, training duration, and coverage error. Compared to popular FS methods, our method enhanced classification accuracy, feature selection redundancy, variety, and convergence rate. It takes longer than other ways but is better. The suggested technique includes computational complexity, parameter sensitivity, difficulties handling disconnected Pareto fronts, limited adaptability for dynamic environments, and no diversity preservation. Our multi-

objective feature selection approach will be used to test these constraints in future work.

REFERENCES

[1] Han, F., Chen, W. T., Ling, Q. H., & Han, H. (2021). Multi-objective particle swarm optimization with adaptive strategies for feature selection. Swarm and Evolutionary Computation, 62, 100847.
[2] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. ACM computing surveys (CSUR), 50(6), 1-45.
[3] Hancer, E. (2019). Differential evolution for feature selection: a fuzzy wrapper–filter approach. Soft Computing, 23(13), 5233-5248.
[4] Russell, S., & Norvig, P. (2002). Artificial intelligence: a modern approach.
[5] Labani, M., Moradi, P., & Jalili, M. (2020). A multi-objective genetic algorithm for text feature selection using the relative discriminative criterion. Expert Systems with Applications, 149, 113276. doi:10.1016/j.eswa.2020.113276.
[6] Xue, B., Zhang, M., & Browne, W. N. (2013). Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. IEEE Transactions on Cybernetics, 43(6), 1656–1671. doi:10.1109/tsmcb.2012.2227469.
[7] Bermejo, P., Gamez, Jose A., Puerta, J M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. Knowledge-Based Systems, 55(2014), 140–147. doi:10.1016/j.knosys.2013.10.016.
[8] Li, A.-D., Xue, B., & Zhang, M. (2020). Multi-objective feature selection using hybridization of a genetic algorithm and direct multisearch for key quality characteristic selection. Information Sciences, 523, 245–265. doi:10.1016/j.ins.2020.03.032.
[9] Banka, H., Dara, S. (2015). A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. Pattern Recognition Letters, 52(2014), 94–100. doi:10.1016/j.patrec.2014.10.007.
[10] Zhang, X., Zhang, Q., Chen, M., Sun, Y., Qin, X., Li, H (2017). A two-stage feature selection and intelligent fault diagnosis method for rotating machinery using hybrid filter and wrapper method. Neurocomputing, 275(C), 2426–2439. doi:10.1016/j.neucom.2017.11.016.
[11] Xue, B., Zhang, M., Browne, W N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing, 18(2014), 261–276. doi:10.1016/j.asoc.2013.09.018.
[12] Yilmaz Eroglu, Duygu; Kilic, Kemal (2017). A novel Hybrid Genetic Local Search Algorithm for feature selection and weighting with an application in strategic decision making in innovation management. Information Sciences, 405(), 18–32. doi:10.1016/j.ins.2017.04.009.
[13] Hamdani T M., Won J., Alimi A M., Karray F. (2011). Hierarchical genetic algorithm with new evaluation function and bi-coded representation for the selection of features considering their confidence rate. , 11(2), 2501–2509. doi:10.1016/j.asoc.2010.08.020.
[14] Han, M., Ren, W. (2015). Global mutual information-based feature selection approach using single-objective and multi-objective optimization. Neurocomputing, 168(), 47–54. doi:10.1016/j.neucom.2015.06.016.
[15] Xue, B., Fu, W., & Zhang, M. (2014). Multi-objective Feature Selection in Classification: A Differential Evolution Approach. Simulated Evolution and Learning, 516–528. doi:10.1007/978-3-319-13563-2_44.
[16] A. Abbasi, B. Firouzi, and P. Sendur, "On the application of Harris hawk's optimization (HHO) algorithm to the design of microchannel heat sinks," Engineering with Computers, Vol. 37, no. 2, 1409-1428, 2021.
[17] itzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm, EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, 95-100.
[18] L.-P. Wang, F. Wu, M.-Z. Zhang, and F.-Y. QiuY, ``Decomposition multiobjective evolutionary algorithm based on differentiated neighborhood strategy,'' Pattern Recognit. Artif. Intell., vol. 30, no. 12, pp. 1069_1082,Dec. 2017.
[19] Q. Zhang, W. Liu, and H. Li, ``The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances,'' in Proc. IEEE Congr. Evol. Comput., Trondheim, NO, USA, May 2009, pp. 203_208.

[20] Z.Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, ``Adaptive replacement strategies for MOEA/D,'' IEEE Trans. Cybern., vol. 46, no. 2, pp. 474_486, Feb. 2016.

[21] D. A. Van Veldhuizen and G. B. Lamont, ``On measuring multiobjective evolutionary algorithm performance,'' in Proc Congr. Evol. Comput.,Washington, DC, USA, Feb. 2002, pp. 204_211.

[22] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, ``MOEA/D with adaptive weight adjustment,'' Evol. Comput., vol. 22, no. 2, pp. 231_264, Jun. 2014.

[23] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, ``A reference vector guided evolutionary algorithm for many-objective optimization,'' IEEE Trans. Evol. Comput., vol. 20, no. 5, pp. 773_791, Oct. 2016.

[24] H. Li and Q. Zhang, ``Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II,'' IEEE Trans. Evol. Comput.,vol. 13, no. 2, pp. 284_302, Apr. 2009.

[25] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, ``Stable matching-based selection in evolutionary multiobjective optimization,'' IEEE Trans. Evol. Comput., vol. 18, no. 6, pp. 909_923, Dec. 2014.

[26] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, ``A replacement strategy for balancing convergence and diversity in MOEA/D,'' in Proc. IEEE Congr. Evol. Comput. (CEC), Beijing, China, Jul. 2014, pp. 2132_2139.

[27] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, ``Balancing convergence and diversity in decomposition-based many-objective optimizers,'' IEEE Trans. Evol. Comput., vol. 20, no. 2, pp. 180_198, Apr. 2016.

[28] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, ``Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes,'' IEEE Trans. Evol. Comput., vol. 16, no. 3, pp. 442_446, Jun. 2012.

[29] H. Zhou and L.-P. Wang, ``Decomposition multiobjective evolutionary algorithm of dynamic neighborhood,'' J. Chin. Comput. Syst., vol. 38, no. 9, pp. 2039_2044, Sep. 2017.

[30] L.-P. Wang, F. Wu, M.-Z. Zhang, and F.-Y. QiuY, ``Decomposition multiobjective evolutionary algorithm based on differentiated neighborhood strategy,'' Pattern Recognit. Artif. Intell., vol. 30, no. 12, pp. 1069_1082,Dec. 2017.

[31] L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle, ``Comparing decomposition-based and automatically component-wise designed multiobjective evolutionary algorithms,'' in Proc. Int. Conf. Evol. MultiCriterion Optim., Cham, Switzerland, Mar. 2015, pp. 396_410.

[32] Xue, X., Yao, M., & Wu, Z. (2018). A novel ensemble-based wrapper method for feature selection using extreme learning machine and genetic algorithm. Knowledge and Information Systems, 57(2), 389-412.

[33] Zhang, N., & Ding, S. (2017). Unsupervised and semi-supervised extreme learning machine with wavelet kernel for high dimensional data. Memetic Computing, 9(2), 129-139.

[34] Xue, Y., Zhu, H., Liang, J., & Słowik, A. (2021). Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. Knowledge-Based Systems, 227, 107218. doi:10.1016/j.knosys.2021.107218.

[35] Paul, D., Jain, A., Saha, S., & Mathew, J. (2021). Multi-objective PSO based online feature selection for multi-label classification. Knowledge-Based Systems, 222, 106966. doi:10.1016/j.knosys.2021.106966.

[36] Usman, A M., Yusof, U K., Naim, S. (2020). Filter-Based Multi-Objective Feature Selection Using NSGA III and Cuckoo Optimization Algorithm. IEEE Access, 8, 76333–76356. doi:10.1109/ACCESS.2020.2987057.

[37] Vijayanand, R., & Devaraj, D. (2020). A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network. IEEE Access, 1–1. doi:10.1109/access.2020.2978035.

[38] R. Wang, P. J. Fleming, and R. C. Purshouse, ``General framework for localised multi-objective evolutionary algorithms,'' Inf. Sci., vol. 258, pp. 29_53, Feb. 2014.

[39] E. Zizler, M. Laumanns, and L. Thiele, ``SPEA2: Improving the strength Pareto evolutionary algorithm,'' in Proc. Evol. Methods Design, Optim. Control Appl. Ind. Prob. (EUROGEN), Athens, Greece, Sep. 2001,pp. 95_100.

[40] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, ``A fast and elitist multiobjective genetic algorithm: NSGA-II,'' IEEE Trans. Evol. Comput.,vol. 6, no. 2, pp. 182_197, Apr. 2002.

[41] K. Deb and J. Sundar, ``Reference point based multi-objective optimization using evolutionary algorithms,'' in Proc. 8th Annu. Conf.

[42] Genetic Evol.Comput. (GECCO), Washington, DC, USA:, 2006, pp. 635_642.

[42] N. Beume, B. Naujoks, and M. Emmerich, ``SMS-EMOA: Multiobjective selection based on dominated hypervolume,'' Eur. J. Oper. Res., vol. 181,no. 3, pp. 1653_1669, Sep. 2007.

[43] J. Bader and E. Zitzler, ``HypE: An algorithm for fast hypervolume-based many-objective optimization,'' Evol. Comput., vol. 19, no. 1, pp. 45_76,Mar. 2011.

[44] E. Zitzler and S. Künzli, ``Indicator-based selection in multiobjective search,'' in Proc. Int. Conf. Parallel Problem Solving From Nature,vol. 3242. Berlin, Germany: Springer, 2004, pp. 832_842.

[45] Q. Zhang and H. Li, ``MOEA/D: A multiobjective evolutionary algorithm based on decomposition,'' IEEE Trans. Evol. Comput., vol. 11, no. 6, pp. 712_731, Dec. 2007.

[46] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, ``A reference vector guided evolutionary algorithm for many-objective optimization,'' IEEE Trans. Evol. Comput., vol. 20, no. 5, pp. 773_791, Oct. 2016.

[47] D. Gong, J. Sun, and X. Ji, ``Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems,'' Inf. Sci.,vol. 233, pp. 141_161, Jun. 2013.

[48] L.-P. Wang, M.-L. Feng, and Q.-Q. Qiu, ``Survey on preference-based multi-objective evolutionary algorithms,'' Chin. J. Comput., vol. 42, no. 6, pp. 1289_1315, Jul. 2019.

[49] D. Gong, J. Sun, and Z. Miao, ``A set-based genetic algorithm for interval many-objective optimization problems,'' IEEE Trans. Evol. Comput.,vol. 22, no. 1, pp. 47_60, Feb. 2018.

[50] Y. Liu, D. Gong, J. Sun, and Y. Jin, ``A many-objective evolutionary algorithm using a one-by-one selection strategy,'' IEEE Trans. Cybern.,vol. 47, no. 9, pp. 2689_2702, Sep. 2017.

[51] W.-J. Kong, J.-L. Ding, and T.-Y. Chai, ``Survey on large-dimensional multi-objective evolutionary algorithms,'' Control Decis., vol. 25, no. 3,pp. 321_326, 2010.

[52] Lee, J., & Kim, D. W. (2013). Feature selection for multi-label classification using multivariate mutual information. Pattern Recognition Letters, 34(3), 349-357.

[53] Almuallim, H., & Dietterich, T. G. (1994). Learning boolean concepts in the presence of many irrelevant features. Artificial intelligence, 69(1-2), 279-305.

[54] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks, 5(4), 537-550.

[55] Kwak, N., & Choi, C. H. (2002). Input feature selection by mutual information based on Parzen window. IEEE transactions on pattern analysis and machine intelligence, 24(12), 1667-1671.

[56] Bishop, C. M. (1995). Neural networks for pattern recognition. Oxford university press.

[57] Hall, M. A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

[58] Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence, 27(8), 1226-1238.

[59] Lin, D., & Tang, X. (2006, May). Conditional infomax learning: An integrated framework for feature extraction and fusion. In European conference on computer vision (pp. 68-82). Springer, Berlin, Heidelberg.

[60] Benesty, J., Chen, J., Huang, Y., & Cohen, I. (2009). Noise reduction in speech processing (Vol. 2). Springer Science & Business Media.

[61] Liu, H., Sun, J., Liu, L., & Zhang, H. (2009). Feature selection with dynamic mutual information. Pattern Recognition, 42(7), 1330-1339.

[62] Moghadasian, M., & Hosseini, S. P. (2014). Binary cuckoo optimization algorithm for feature selection in high-dimensional datasets. In International conference on innovative engineering technologies (ICIET'2014) (pp. 18-21).

[63] Xue, B., Cervante, L., Shang, L., Browne, W. N., & Zhang, M. (2013). Multi-objective evolutionary algorithms for filter based feature selection in classification. International Journal on Artificial Intelligence Tools, 22(04), 1350024.

[64] Usman, A. M., Yusof, U. K., & Naim, S. (2018). Cuckoo inspired algorithms for feature selection in heart disease prediction. International Journal of Advances in Intelligent Informatics, 4(2), 95-106.

[65] Hamdani, T. M., Won, J. M., Alimi, A. M., & Karray, F. (2007, April). Multi-objective feature selection with NSGA II. In International conference on adaptive and natural computing algorithms (pp. 240-247). Springer, Berlin, Heidelberg.

[66] Xue, Y., Zhu, H., Liang, J., & Słowik, A. (2021). Adaptive crossover operator based multi-objective binary genetic algorithm for feature selection in classification. Knowledge-Based Systems, 227, 107218. doi:10.1016/j.knosys.2021.107218.

[67] Usman, A M., Yusof, U K., Naim, S. (2020). Filter-Based Multi-Objective Feature Selection Using NSGA III and Cuckoo Optimization Algorithm. IEEE Access, 8, 76333–76356. doi:10.1109/ACCESS.2020.2987057.

[68] Bache, K., Lichman, M. (2013), UCI machine learning repository.

[69] Abuassba, A. O., Dezheng, Z., Ali, H., Zhang, F., & Ali, K. (2022). Classification with ensembles and case study on functional magnetic resonance imaging. Digital Communications and Networks, 8(1), 80-86.