A Robust Wrapper-Based Feature Selection Technique Using Real-Valued Triangulation Topology Aggregation Optimizer

Li Pan[1], Wy-Liang Cheng[2]*, Sew Sun Tiang[3], Kim Soon Chong[4],
Chin Hong Wong[5], Abhishek Sharma[6], Touseef Sadiq[7], Aasam Karim[8], Wei Hong Lim[9*]
Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur, 56000, Malaysia[1, 2, 3, 4, 9]
Zhengzhou Institute of Engineering and Technology, Zhengzhou, 450044, China[1]
Maynooth International Engineering College, Maynooth University, Maynooth, Co Kildare, Ireland[5]
Maynooth International Engineering College, Fuzhou University, Fujian University, 350116, China[5]
Department of Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun, 248002, India[6]
Centre for Artificial Intelligence Research (CAIR)-Department of Information and Communication Technology,
University of Agder, Grimstad, Norway[7]
Instytut Informatyki Uniwersytet Opolski 45-040 Opole Poland[8]

Abstract*—Feature selection is a critical preprocessing technique used to remove irrelevant and redundant features from datasets while maintaining or improving the accuracy of machine learning models. Recent advancements in this area have primarily focused on wrapper-based feature selection methods, which leverage metaheuristic search algorithms (MSAs) to identify optimal feature subsets. In this paper, we propose a novel wrapper-based feature selection method utilizing the Triangulation Topology Aggregation Optimizer (TTAO), a newly developed algorithm inspired by the geometric properties of triangular topology and similarity. To adapt the TTAO for binary feature selection tasks, we introduce a conversion mechanism that transforms continuous decision variables into binary space, allowing the TTAO—which is inherently designed for real-valued problems—to function efficiently in binary domains. TTAO incorporates two distinct search strategies, generic aggregation and local aggregation, to maintain an effective balance between global exploration and local exploitation. Through extensive experimental evaluations on a wide range of benchmark datasets, TTAO demonstrates superior performance over conventional MSAs in feature selection tasks. The results highlight TTAO's capability to enhance model accuracy and computational efficiency, positioning it as a promising tool to advance feature selection and support industrial innovation in data-driven tasks.

Keywords—Classification; exploration; exploitation; feature selection; metaheuristic search algorithm; machine learning; optimization; triangulation topology aggregation optimizer

I. INTRODUCTION

Recent advancements in data-driven methodologies such as machine learning approaches have demonstrated significant benefits in addressing diverse, complex real-world challenges. Nonetheless, the escalating complexity of datasets from various domains and sources increasingly burdens machine learning models with inefficiencies and elevated computational costs [1]. This burden is often exacerbated by the presence of excessive, irrelevant, and redundant features, particularly when datasets are laden with noise, inconsistencies, and non-contributory information. Such datasets do not enhance model performance and may even compromise system approximation accuracy due to overfitting. Additionally, the presence of a large number of features necessitates the use of more complex machine learning models. These models require extensive data to optimize learning parameters, which can degrade their ability to generalize effectively [1].

To address these challenges, it becomes crucial to select a relevant subset of features while eliminating redundancies from the original datasets. This process not only enhances the efficiency, accuracy, and generalization capability of machine learning models but also serves as a vital preprocessing strategy. Feature selection effectively improves model performance by minimizing redundant input during training. It aims to optimize model efficiency by identifying and employing an optimal subset of features, thus alleviating the detrimental effects associated with the “curse of dimensionality”. This is particularly important when dealing with input datasets that contain an excessively large number of primitive features. Feature selection is instrumental in addressing a wide range of real-world machine-learning challenges, such as food fraud detection, automatic modulation recognition, predictive maintenance, robot path planning, kitchen waste segregation etc. [2-8].

Feature selection techniques are primarily divided into three categories: filter, wrapper, and embedded methods [9]. Filter methods determine feature subsets using statistical techniques that evaluate data dependencies. These methods assign rankings based on criteria such as inter-feature distances, correlations, and consistency indices. Widely utilized filter methods include the correlation coefficient, F-score, and Gini index [10]. Although filter methods are computationally efficient due to their classifier independence, they often do not reflect true feature relevance within specific models, potentially leading to reduced predictive accuracy. Conversely, wrapper methods integrate with a specific classifier, employing classification accuracy to assess feature subset quality [11]. While these methods typically enhance classifier performance, they also increase the risk of overfitting and require extensive computational resources due to repeated classifier executions to ascertain the optimal subset. Embedded methods, merging the benefits of filter and wrapper approaches, interact directly with the classifier while managing dependencies more efficiently to reduce computational demands. Although embedded methods offer a compromise in computational load, they remain more resource-intensive than filter methods.

The feature selection technique proposed in this paper is classified as wrapper-based, typically involves three components [9]: classifiers employed, evaluation criteria for feature selection, and the search algorithms used to derive feature subsets from raw data. Conventional search strategies [11] such as backward elimination, forward selection, greedy search, and complete search often exhibit significant limitations within the wrapper-based framework. These limitations include poor global search abilities, entrapment in local optima, and high computational costs. To address these deficiencies, this study advocates the use of metaheuristic search algorithms (MSAs), which offer superior global search strength, stochastic behavior, simple implementation, and do not rely on gradient information, making them well-suited to address complex optimization challenges [12-16]. A review of how MSAs effectively overcome feature selection challenges is provided [9].

MSAs represent a varied collection of optimization techniques, categorized by their foundational inspirations and search mechanisms [17]. The first category of MSAs is the evolutionary algorithms that are influenced by Darwin's theory of evolution and natural selection. Swarm intelligence algorithms are the second category of MSAs and they are inspired by collective animal behaviors such as flocking and foraging. Human-based algorithms are the third category of MSAs and they mimic aspects of human cognition including learning and social interactions, whereas the last category of MSAS are physics-based algorithms that apply principles from physical sciences and mathematics. Although numerous MSAs have been developed in response to the No-Free-Lunch (NFL) theorem, which asserts that no single algorithm can optimally solve all types of problems, their validation has predominantly been confined to mathematical benchmarks.

While significant theoretical advancements have been made in the development of MSAs, their performance evaluations remain largely confined to theoretical benchmarks. This narrow focus limits our understanding of their practical effectiveness in solving real-world problems, emphasizing the need for more empirical studies. In particular, the practical application of many recently developed MSAs in addressing real-world optimization challenges, such as feature selection, remains insufficiently explored. Moreover, most MSAs have not been rigorously validated in complex, high-dimensional feature selection tasks involving binary decision variables. This gap highlights the pressing need for empirical research that assesses the performance of novel MSAs in feature selection tasks, extending beyond traditional continuous-variable optimization problems.

This paper introduces an advanced wrapper-based feature selection method leveraging the unique search mechanisms of the Triangulation Topology Aggregation Optimizer (TTAO), a novel physics-based MSA proposed by Zhao et al. in 2024 [18]. Inspired by the geometric properties of triangular topology and the principle of triangular similarity, TTAO utilizes the consistent shape but variable sizes of similar triangles to generate diverse triangular topological units that serve as dynamic evolutionary entities throughout the optimization process. This technique aims to enhance the performance of machine learning models by effectively eliminating irrelevant features from datasets. TTAO incorporates two primary aggregation strategies: generic aggregation and local aggregation. Generic aggregation enhances exploratory search by promoting information exchange across different triangular topological units, whereas local aggregation focuses on exploitation, refining the search within individual units. Although initially applied in limited real-world contexts such as transmission expansion planning [19], productivity prediction [20], and controller parameter adjustment [21], where decision variables are real-valued, the application of TTAO to feature selection tasks involving binary decision variables is an unexplored area of research. This study aims to fill this gap by demonstrating how TTAO can be adapted to binary feature selection, presenting a novel conversion mechanism that enables its application in this domain. By expanding TTAO’s utility to feature selection tasks with binary decision variables, this paper contributes to addressing the broader challenge of validating MSAs in real-world optimization problems.

The technical contributions and novelty of this study are summarized as follows:

- We propose an advanced wrapper-based feature selection technique that utilizes the unique search mechanisms of the TTAO to identify optimal feature subsets. This approach aims to achieve high classification accuracy while maintaining low model complexity.
- To our knowledge, this is the first application of TTAO to address feature selection problems involving binary decision variables, which present more complex optimization challenges compared to those with continuous variables.
- A novel conversion mechanism is introduced, transforming continuous decision variables into binary ones, thus adapting the inherently real-valued TTAO for use in binary solution spaces.
- We provide a comprehensive evaluation of TTAO’s effectiveness as a wrapper-based method for feature selection, demonstrating its superior performance against other MSAs using diverse datasets from the UCI Machine Learning Repository.

The remainder of this paper is organized as follows: Section II reviews related work. Section III outlines the formulation of wrapper-based feature selection as an optimization problem and details the search mechanisms of TTAO. Section IV presents performance evaluations of various wrapper-based feature selection techniques. Section V concludes with a summary and future works.

II. Related Works of Using Different MSAs for Wrapper-Based Feature Selection

A wrapper-based feature selection technique incorporates three core components: the classifier types, the search algorithms for discovering optimal feature subsets, and the criteria for assessing the quality of these subsets. MSAs are often favored for wrapper-based feature selection due to their robust global search capabilities and straightforward implementation, as documented in study [9]. These MSAs are particularly effective in identifying feature subsets that optimize classification accuracy while minimizing the complexity of the machine learning model.

Recent developments in MSAs have significantly enhanced the robustness of feature selection methodologies. Various novel MSAs such as the Flow Direction Algorithm [22], African Vultures Optimization Algorithm [23], Sperm Swarm Optimization [24], Grasshopper Optimization Algorithm [25], Artificial Butterfly Optimization [26], have been employed to tackle feature selection challenges. Researchers also continue to refine these algorithms, creating more efficient versions tailored to specific problem characteristics. For example, Zekeri and Hokmabadi [11] introduced a real-value Grasshopper Optimization Algorithm (GOFS), utilizing a mathematical model that leverages repulsion and attraction forces between grasshoppers to effectively navigate the feature space. They enhanced GOFS with an adaptive parameter that modifies the influence zones to improve feature exploration and exploitation. Additionally, they implemented a feature probability factor to eliminate redundant features each iteration. Mostafa et al. [27] developed a Modified Chameleon Swarm Algorithm (mCSA), incorporating a transfer operator and a randomization Levy flight control parameter to fine-tune search behaviors. They also hybridized mCSA with the consumption operator from Artificial Ecosystem-based Optimization to augment its global search capabilities.

Zhang et al. [10] developed a novel wrapper-based feature selection method utilizing the Return-Cost-Based Binary Firefly Algorithm (Rc-BBFA), enhanced with three key modifications to address premature convergence. This version replaces traditional distance-based attractiveness with a return-cost metric to gauge each firefly's appeal. Additionally, a Pareto dominance strategy selects the most attractive firefly based on cost and return values. A new binary movement operator, driven by return-cost attractiveness and supplemented by an adaptive jump, updates each firefly's position within Rc-BBFA. Ma et al. [28] introduced the Multi-Strategy Binary Hunger Games Search (MS-bHGS) to tackle feature selection across 20 benchmark datasets. MS-bHGS incorporates chaotic maps, a vertical crossover scheme, and a greedy selection strategy, enhancing the balancing of exploration and exploitation. Wu et al. [29] enhanced a wrapper-based feature selection method using the Sparrow Search Algorithm, augmented by Quantum Computation and Multi-Strategy Enhancement (QMESSA). This approach integrates an improved circle chaotic map with a quantum gate mutation mechanism to diversify the initial population. Adaptive T-distribution and a novel position update formula were also embedded in QMESSA to boost its convergence speed. capabilities.

Zhong et al. [30] introduced the Self-Adaptive Quantum Equilibrium Optimizer with Artificial Bee Colony (SQEOABC) for feature selection, incorporating quantum theory and a self-adaptive mechanism to improve its convergence. Additionally, SQEOABC utilizes updating mechanisms from the Artificial Bee Colony to enhance the selection of effective feature subsets. Khafaga et al. [31] proposed a novel wrapper-based feature selection method using the Adaptive Squirrel Search Optimization Algorithm (ASSOA), paired with a KNN classifier. This method was applied to ten datasets from the UCI Machine Learning Repository. ASSOA was enhanced with new relocation equations and various movements (vertical, horizontal, exponential, and diagonal) to improve its search capabilities. Furthermore, various feature selection techniques were advanced by combining the Dipper Throated Optimization Algorithm with the Grey-World Optimizer [32] and Sine Cosine Algorithm [33]. These hybrid methods were tailored to identify superior feature subsets, contributing to higher accuracy and reduced model complexity in handling publicly available datasets.

Image Analysis Society (MIAS), the selected features were evaluated using the XGBoost classifier. In a follow-up study, they developed an adaptive binary TLBO with an ensemble classifier combining XGBoost and Random Forest, aimed at the early detection of breast cancer using mammograms from MIAS and the Digital Database for Screening Mammography (DDSM) [23].

III. Wrapper-Based Feature Selection Using TTAO

A. *Solution Representation of TTAO in Feature Selection*

In the context of feature selection, consider a dataset where $|F_o|$ denotes the total number of input features. Within the framework of TTAO, each search agent or vertex of the n-th triangular topological unit is defined by a position vector $X_n = [X_{n,1}, \ldots, X_{n,d}, \ldots, X_{n,D}]$, with D equating to $|F_o|$, representing the dimensionality of the problem. Each dimensional index d corresponds directly to a feature index l. Initially, the decision variables for each search agent are continuous. However, the binary nature required for feature selection dictates that these variables must be converted to binary values – 0 or 1.

To facilitates this conversion, the proposed wrapper-based feature selection technique based on TTAO implements a threshold parameter γ. This parameter is used to transform continuous decision variables into binary decisions by evaluating each real-valued decision variable $X_{n,d}$ against γ:

$$S_{n,l} = \begin{cases} 0, & \text{if } X_{n,d} < \gamma \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

Here, the binary value $S_{n,l}$ determines the inclusion status of each *l*-th feature, where a value of 1 indicates inclusion and 0 indicates exclusion. For example, a status of $S_n = [0,1,1,1,0]$ implies that features at indices *l* = 2 to 4 are selected, while those at indices *l* = 1 and 5 are excluded. This mechanism effectively transforms continuous input values encoded in the search agent into the discrete decisions crucial for effective feature selection.

B. Fitness Evaluation of TTAO Search Agent in Feature Selection

Feature selection plays a crucial role in machine learning by facilitating the identification of an optimal subset of features. This subset not only enhances classification accuracy but also reduces the numbers of utilized features, addressing a twofold challenge: lowering the classifier's error rate and minimizing the ratio of selected features to the total available.

Define ξ_{Error} as the classifier's error rate and $|F_s|$ as the count of features selected for the subset, with $|F_s| \leq |F_o|$. The fitness value, which assess the quality for each search agent of the *n*-th triangular topological unit via the feature status vector $S_n = [S_{n,1}, \ldots, S_{n,l}, \ldots, S_{n,|F_O|}]$, is given by:

$$F(X_n) = \omega \times \xi_{Error} + \mu \times \frac{|F_s|}{|F_o|} \quad (2)$$

Here, ω is a coefficient ranging from 0 to 1, and μ is defined as $1 - \omega$. These parameters are designed to weigh the impacts of classification error and feature proportionality, respectively. The optimal feature subset minimizes the fitness function outlined in Eq. (2), achieving a balance between high classification accuracy and reduced feature set complexity, thereby simplifying and enhancing the efficacy of machine learning models.

In the wrapper-based feature selection framework using TTAO, the fitness evaluation process, denoted as Algorithm 1, employs the KNN classifier to measure each n-th search agent's performance based on the feature selection status S_n. Feature normalization is applied to scale the selected features between 0 and 1, followed by performance evaluation using K-fold cross-validation with the KNN classifier. A lower $F(X_n)$ value signifies superior fitness, indicative of higher classification accuracy and a smaller number of selected features.

Algorithm 1: Fitness Evaluation Process of Wrapper-Based Feature Selection Using TTAO

Inputs: X_n, F_o, D, γ
01: Convert X_n into S_n using Eq. (1);
02: Determine $|F_s|$ from S_n and train KNN classifier to get ξ_{Error};
03: Calculate $F(X_n)$ using Eq. (2) based on $|F_s|$ and ξ_{Error};
Outputs: $F(X_n)$

C. Mechanisms of TTAO to Identify Optimal Feature Subsets

1) Conceptual ideas of TTAO: The search mechanisms of TTAO draw inspiration from the fundamental properties of triangular topology in mathematics. The triangle, recognized as the most basic yet stable shape in planar geometry, serves as a cornerstone in both finite and infinite dimensional spaces. It functions as a graph within its two-dimensional subspace. Due to its inherent simplicity and robustness, the triangular topology is extensively employed as a structural unit in model representation and analysis across a variety of real-world applications. These applications span multiple disciplines, including computational geometry, structural engineering, digital image processing, etc.

The concept of triangular similarity is pivotal in geometry and plays a key role in the search mechanisms of TTAO. The principles of triangular similarity are covered in four theorems:

- Theorem 1: A new triangle formed by drawing a line parallel to one side of an original triangle and intersecting the extensions of the other two sides is similar to the original.
- Theorem 2: Two triangles are similar if their corresponding sides and angles are proportional.
- Theorem 3: A triangle is similar to another if the ratios of their corresponding sides are equal.
- Theorem 4: Triangles that have identical corresponding angles are similar.

TTAO employs these theorems of triangular similarity to direct its search strategy. Throughout its iterative search process, the algorithm continuously generates new vertices in the solution space to construct triangles of varying sizes, each considered an evolutionary unit with three external vertices and one internal random vertex. Additionally, the TTAO utilizes the concept of aggregation to merge vertices with superior traits, enhancing the information exchange within and across different topological units. All triangles within the TTAO framework are equilateral, maintaining geometric consistency by adhering to the second theorem of similarity. The optimization process of TTAO consists of two primary stages: aggregation between and within units, streamlining the exploration and exploitation phases.

2) Initialization phase of TTAO: The initialization phase of TTAO involves randomly generating a diverse set of potential solutions across the solution space. Let N and D represent the population size and problem dimensionality of TTAO, respectively. Each vertex within a triangular topological unit is treated as a search agent or potential solution. Using the floor rounding operator $\lfloor \cdot \rfloor$, the population set of N search agents is organized into $\lfloor N/3 \rfloor$ triangular topological units. Any additional search agents, arising when N is not divisible by 3, are randomly generated within the solution space.

The lower and upper boundary limits of decision variables are denoted as $X^L = [x_1^l, \ldots, x_d^l, \ldots, x_D^l]$ and $X^U = [x_1^u, \ldots, x_d^u, \ldots, x_D^u]$, respectively. Let r_0 be a random number between 0 and 1. For each *n*-th triangular topological unit, where $n = 1, \ldots, \lfloor N/3 \rfloor$, the position of the first search agent (vertex) is randomly determined within the feasible regions of solution space as follows:

$$X_{n,1} = X^L + r_0(X^U - X^L) \quad (3)$$

3) Construction of triangular topological unit: In addressing multi-dimensional optimization challenges, TTAO constructs equilateral triangles within each two-dimensional projection of a higher-dimensional space. The TTAO leverages transformations between polar and Cartesian coordinate systems to establish the vertices of each triangular topological unit.

For every *n*-th triangular topological unit, a direction vector, denoted as $lf(\cdot)$, is calculated and applied to the first vertex ($X_{n,1}$) to determine the second vertex ($X_{n,2}$) as follows:

$$X_{n,2} = X_{n,1} + lf(\theta) \tag{4}$$

The third vertex $(X_{n,3})$ is then generated by rotating the direction vector $lf(\cdot)$ by $\pi/3$ radians anticlockwise:

$$X_{n,3} = X_{n,1} + lf(\theta + \pi/3) \tag{5}$$

Here, l signifies the length of the edges of the triangular topology unit, given by:

$$l = 9e^{-\frac{t}{T^{max}}} \tag{6}$$

where t is the current iteration numbers, and T^{max} represents the maximum iteration numbers. According to Eq. (6), l decreases as the number of fitness evaluations increases. This adaptive strategy enables broader exploratory moves in the initial stages and more focused exploitation in the latter phases to refine the search in promising regions. The exponential decay ensures l remains positive, preventing excessive exploitation and potential premature convergence.

Moreover, the vectors $f(\theta)$ and $f(\theta + \pi/3)$, directing the edges from the first vertex, are defined respectively as:

$$f(\theta) = [\cos\theta_1, \dots, \cos\theta_d, \dots, \cos\theta_D] \tag{7}$$

$$f(\theta + \pi/3) = [\cos(\theta_1 + \pi/3), \dots, \cos(\theta_d + \pi/3), \dots, \cos(\theta_D + \pi/3)] \tag{8}$$

where θ_d for $d = 1, \dots, D$ is a randomly generated angle ranging from 0 to π.

Within each *n*-th triangular topological unit, a fourth vertex $X_{n,4}$ is derived through an internal aggregation process using a linear combination of the first three vertices, weighted by randomly generated coefficients:

$$X_{n,4} = r_1 X_{n,1} + r_2 X_{n,2} + r_3 X_{n,3} \tag{9}$$

where r_1, r_2 and r_3 are randomly numbers between 0 to 1, ensuring $r_1 + r_2 + r_3 = 1$.

In each iteration, a new triangular topological unit is generated from a vertex and two sides of equal lengths *l*, which dynamically change throughout optimization process. Within each *n*-th triangular topological unit, the vertex exhibiting the best fitness during the current iteration is designated as the lead vertex. This lead vertex plays a crucial role in guiding the search process of the other vertices within the same unit. As detailed in subsequent sections, vertices within and across different triangular topological units employ two pivotal search mechanisms: generic aggregation and local aggregation. These mechanisms enable exploration and exploitation, respectively.

4) Generic aggregation of TTAO: Generic aggregation facilitates exploration by enabling the information exchange between the best search agent (vertex) in each triangular topological unit and the best vertex from a randomly selected unit. This mechanism draws inspiration from the crossover operator in genetic algorithm, which creates a new offspring solution by merging genetic information from two parent solutions.

Let $X_{n,best}^t$ denote the best vertex of the *n*-th triangular topological unit at iteration *t*, and $X_{n_{rand},best}^t$ represents the best vertex from a randomly selected unit at the same iteration. For each *n*-th triangular topological unit, a new vertex $X_{n,new1}^{t+1}$ is generated through generic aggregation by linearly combining the dimensional variables of these two superior vertices with different weights:

$$X_{n,new1}^{t+1} = r_4 X_{n,best}^t + (1 - r_4) X_{n_{rand},best}^t \tag{10}$$

where r_4 is a random number between 0 to 1.

The fitness value of the newly generated vertex $X_{n,new1}^{t+1}$ is evaluated as $F(X_{n,new1}^{t+1})$, and compared against the fitness values of the current optimal and suboptimal vertices in the *n*-th triangular topological unit, represented as $F(X_{n,best}^t)$ and $F(X_{n,sbest}^t)$, respectively. Here, the suboptimal vertex $X_{n,sbest}^t$ is defined as the search agent with the second-best fitness in the n-th unit. For minimization problems, updates to the optimal and suboptimal vertices of n-th triangular topological unit for the subsequent iteration $(t + 1)$ are made according to the conditions below:

$$X_{n,best}^{t+1} = \begin{cases} X_{n,new1}^{t+1}, & \text{if } F\left(X_{n,new1}^{t+1}\right) < F(X_{n,best}^t) \\ X_{n,best}^t, & \text{otherwise} \end{cases} \tag{11}$$

$$X_{n,sbest}^{t+1} = \begin{cases} X_{n,new1}^{t+1}, & \text{if } F\left(X_{n,new1}^{t+1}\right) < F(X_{n,sbest}^t) \\ X_{n,sbest}^t, & \text{otherwise} \end{cases} \tag{12}$$

5) Local aggregation of TTAO: Local aggregation within the TTAO is pivotal for exploitation, refining searches within promising areas previously identified by the generic aggregation's exploratory processes. This strategy operates within each triangular topological unit, optimizing based on the best available internal information to enhance solution quality. Following generic aggregation, a temporary triangular topological unit is formed among the updated optimal or suboptimal vertex and two other vertices with relatively good fitness. Notably, this temporary unit may not necessarily form an equilateral triangle.

Within each *n*-th triangular topological unit, the optimal vertex's position is locally perturbed to refine the vicinity around the best current solution, based on the differences between the optimal and suboptimal vertices, thus ensuring the new search direction leverages the promising information. The new vertex generated through local aggregation is given by:

$$X_{n,new2}^{t+1} = X_{n,best}^{t+1} + \alpha X_{n,best}^{t+1} \tag{13}$$

where α is a decreasing parameter regulating the local aggregation's scope, defined as:

$$\alpha = \ln\left(\frac{e - e^3}{T^{max} - 1} t + e^3 - \frac{e - e^3}{T^{max} - 1}\right) \tag{14}$$

The parameter α progressively narrows the search area across iterations to emphasize exploitation in algorithm's later stages.

After local aggregation, it is crucial that the lead vertex of the triangular topological unit is the optimal within that unit. To assure convergence towards the most promising directions, the fitness of the newly aggregated vertex $X_{n,new2}^{t+1}$, denoted as $F(X_{n,new2}^{t+1})$, is compared against the current optimal vertex's fitness $F(X_{n,new1}^{t+1})$. For minimization problems, the updates of optimal vertex during local aggregation is determined as:

$$X_{n,best}^{t+1} = \begin{cases} X_{n,new2}^{t+1}, & \text{if } F\left(X_{n,new2}^{t+1}\right) < F(X_{n,best}^{t+1}) \\ X_{n,best}^{t+1}, & \text{otherwise} \end{cases} \quad (15)$$

Following this update, new similar triangular topological units are constructed for the subsequent iteration based on these updated optimal vertices, employing Eqs. (4) to (9) to refine the search space further and focus on previously promising areas.

6) Optimization workflow of TTAO for wrapper-based feature selection: Algorithm 2 delineates the workflow of the proposed TTAO-based wrapper feature selection technique. The process commences by loading the dataset and establishing dimensionality D equal to the number of input features, $|F_o|$. Initial setting of TTAO include resetting the iteration counter t and determining the number of triangular topological units as $\lfloor N/3 \rfloor$.

To deploy TTAO in searching for optimal feature subsets, initial positions for the first vertices, $X_{n,1}$ for $n = 1, \dots, \lfloor N/3 \rfloor$, of all triangular topological units are randomly generated as per

Algorithm 2: Proposed Wrapper-Based Feature Selection Using TTAO

Inputs: N, γ, T^{Max}, F_o

Load dataset containing $|F_o|$ input features and set the total dimensional size as $D \leftarrow |F_o|$;
Initialize $t \leftarrow 0$, number of triangular topological unit $\leftarrow \lfloor N/3 \rfloor$;
for n = 1 to $\lfloor N/3 \rfloor$ **do** */*Random initialization of the first vertices*/*
  Randomly generate the first vertex $X_{n,1}$ of each n-th triangular topology unit using Eq. (3);
end for
while $t \leq T^{Max}$ **do** */*Iterative search process*/*
  Update the value of parameter l using Eq. (6);
  */*Construction of each n-th triangular topological unit*/*
  for n = 1 to $\lfloor N/3 \rfloor$ **do**
    Determine $f(\theta)$ and $f(\theta + \pi/3)$ using Eqs. (7) and (8), respectively;
    Calculate second vertex $X_{n,2}$ of each n-th triangular topology unit using Eq. (4);
    Calculate third vertex $X_{n,3}$ of each n-th triangular topology unit using Eq. (5);
    Boundary checking for $X_{n,2}$ and $X_{n,3}$ to ensure solution feasibility;
    Calculate fourth internal vertex $X_{n,4}$ of each n-th triangular topology unit using Eq. (9);
    Boundary checking for $X_{n,4}$ to ensure solution feasibility;
    Fitness evaluation of all vertices (i.e., $X_{n,1}$ to $X_{n,4}$) using **Algorithm 1**;
    Identify the vertices with best and second-best fitness as $X_{n,best}^{t}$ and $X_{n,sbest}^{t}$, respectively.
  end for
  */*Generic aggregation*/*
  for n = 1 to $\lfloor N/3 \rfloor$ **do**
    Calculate new vertex $X_{n,new1}^{t+1}$ of each n-th triangular topology unit using Eq. (10);
    Boundary checking for $X_{n,new1}^{t+1}$ to ensure solution feasibility;
    Fitness evaluation of $X_{n,new1}^{t+1}$ using **Algorithm 1**;
    Update $X_{n,best}^{t+1}$ and $X_{n,sbest}^{t+1}$ along with their fitness using Eqs. (11) and (12), respectively;
  end for
  */*Local aggregation*/*
  for n = 1 to $\lfloor N/3 \rfloor$ **do**
    Update the value of parameter α using Eq. (14)
    Calculate new vertex $X_{n,new2}^{t+1}$ of each n-th triangular topology unit using Eq. (13);
    Boundary checking for $X_{n,new2}^{t+1}$ to ensure solution feasibility;
    Fitness evaluation of $X_{n,new2}^{t+1}$ using **Algorithm 1**;
    Update $X_{n,best}^{t+1}$ along with its fitness using Eq. (15);
  end for
  */*To check if the population size N is divisible by 3*/*
  if $N - \lfloor N/3 \rfloor \neq 0$ **then**
    $N^{Remain} = N - \lfloor N/3 \rfloor$;
    for i = 1 to N^{Remain} **do**
      Randomly generate the i-th remaining search agent X_i^{Remain} using Eq. (3);
      Fitness evaluation of X_i^{Remain} using **Algorithm 1**;
    end for
    Compare the fitness value of $X_{n,best}^{t+1}$ for n = 1 to $\lfloor N/3 \rfloor$ and X_i^{Remain} for i = 1 to N^{Remain};
    Extract the top $\lfloor N/3 \rfloor$ search agents with better fitness to be lead vertices in next iteration;
  end if
  Record the best solution X^{Best} and its fitness $F(X^{Best})$ found in each iteration;
end while

Outputs: X^{Best} and S^{Best}

Eq. (1). During iterative searching, subsequent vertices of each n-th triangular topological unit ($X_{n,2}$, $X_{n,3}$ and $X_{n,4}$) are constructed using Eqs. (4), (5) and (9), respectively, to form equilateral triangle. Boundary conditions for $X_{n,2}$, $X_{n,3}$ and $X_{n,4}$ are checked to maintain solution feasibility. In the fitness evaluation process using Algorithm 1, continuous decision variables in each vertex of the n-th triangular topological unit are converted into binary selection status vectors (i.e., $S_{n,1}$, $S_{n,2}$, $S_{n,3}$ and $S_{n,4}$) using Eq. (1), and their respective fitness values ($F(X_{n,1})$, $F(X_{n,2})$, $F(X_{n,3})$, $F(X_{n,4})$) are calculated using Eq. (2). Given these fitness values, the optimal and suboptimal vertices for each n-th unit at iteration t are identified as $X_{n,best}^{t}$ and $X_{n,sbest}^{t}$, respectively.

During the generic aggregation, a new vertex $X_{n,new1}^{t+1}$ is formulated for each *n*-th triangular topological unit via Eq. (10). The updates of position vector and fitness of $X_{n,best}^{t}$ and $X_{n,sbest}^{t}$ are performed using Eq. (11) and (12) if $X_{n,new1}^{t+1}$ demonstrates superior fitness. Similarly, during local aggregation, another new vertex $X_{n,new2}^{t+1}$ is generated for each *n*-th triangular topological unit using Eq. (13), with updates to $X_{n,best}^{t+1}$ conducted via Eq. (15) if $X_{n,new2}^{t+1}$ exhibits enhanced fitness. For population where N is not divisible by three, remaining search agents are randomly generated within the solution space per Eq. (3) and evaluated using Eq. (2). After completing both generic and local aggregations, the fitness values of the optimal vertices across all $\lfloor N/3 \rfloor$ triangular topological units and the randomly generated search agents are compared, selecting only the best $\lfloor N/3 \rfloor$ agents as lead vertices for the subsequent iteration.

This iterative search process persists until reaching the predetermined termination criterion, typically the maximum iteration number T^{max}. Upon termination, the decision variables in the best solution X^{Best} are translated into a binary feature subset S^{Best}, employed to train machine learning models that are both more accurate and less complex.

IV. Results

A. Datasets Used and Simulation Settings

This performance evaluation study employs ten benchmark datasets from the UCI Machine Learning Repository to assess the efficacy of the proposed wrapper-based feature selection technique utilizing the TTAO. These datasets were chosen based on their diverse characteristics, including varying numbers of input features, instances, and output classes, which represent a wide range of feature selection challenges. The datasets cover different problem domains such as medical diagnosis, survival analysis, signal processing, etc., providing a comprehensive assessment of the algorithm's versatility and robustness.

The number of input features influences the dimensionality of the problem, which is crucial in testing the capability of the algorithm to handle high-dimensional spaces. For instance, datasets like Multiple Features (649 features) and Arrhythmia (279 features) represent high-dimensional feature selection challenges, whereas datasets such as Diabetes and Haberman's Survival provide lower-dimensional tasks. The diversity in the number of instances, ranging from small datasets like Lung Cancer (27 instances) to larger datasets such as Maternal Health Risk (1014 instances)**,** ensures that the algorithm's performance is evaluated under varying data sizes. Additionally, the datasets include both binary and multiclass classification problems, further demonstrating the algorithm's adaptability to different problem types. Table I presents the detailed characteristics of the ten datasets, including the number of instances, features, and output classes. This diversity in datasets allows for a thorough evaluation of TTAO's performance in feature selection tasks across various real-world applications.

TABLE I. THE CHARACTERISTICS OF 10 BENCHMARK DATASETS USED IN PERFORMANCE EVALUATION

No	Dataset	No. of Instances	No. of Features	No. of Classes
DS1	Lung Cancer Data Set	27	56	10
DS2	Multiple Features Data Set	2000	649	2
DS3	Ionosphere Data Set	351	34	2
DS4	Arrhythmia Data Set	452	279	13
DS5	Echocardiogram Data Set	61	8	2
DS6	Haberman's Survival Data Set	306	3	2
DS7	Diabetes Data Set	768	8	2
DS8	Wine Data Set	178	13	3
DS9	Maternal Health Risk Data Set Data Set	1014	6	3
DS10	Zoo Data Set	101	16	7

This study evaluates the performance of the TTAO in feature selection tasks relative to seven other state-of-the-art MSAs: Bezier Search Differential Evolution (BeSD) [34], Coronavirus Herd Immunity Optimization Algorithm (CHIO) [35], Chaotic Oppositional based Hybridized Differential Evolution with Particle Swarm Optimization (CO-HDEPSO) [36], Differential Squirrel Search Algorithm (DSSA) [37], Flow Direction Algorithm (FDA) [38], Generalized Normal Distribution Optimization (GNDO) [39], and Oppositional and Social Learning with Enhanced Operator with Particle Swarm Optimization (ODSFMFO) [40]. Optimal parameters for these algorithms are adopted as per the specifications in their respective foundational publications.

To facilitate the conversion of real-valued decision variables within the TTAO and other MSAs to binary values for feature selection, the threshold parameter γ is set at 0.5. A KNN classifier with $k = 5$ is utilized to evaluate classification accuracy based on the selected feature subsets. Each dataset is split into two segments, with 80% of the instances designated as the training set and the remaining 20% as the testing set. The population size and the maximum iteration number for all MSAs are standardized at $N = 20$ and $T^{Max} = 200$, respectively. Given the stochastic nature of MSAs, each algorithm undergoes 30 simulation runs to ensure robustness in addressing the feature selection challenges across different datasets.

B. Performance Comparisons on Average Classification Accuracies

Table II presents the average classification accuracy achieved by the TTAO and seven other MSAs, each employed as wrapper-based feature selection techniques. The results reflect the mean values across 30 independent runs for each dataset, labeled as DS1 through DS10. Classification accuracy serves as a crucial validation measure, as it directly indicates how effectively the selected feature subsets enable the KNN classifier to distinguish between instances with high precision. The MSAs yielding higher average classification accuracies demonstrate superior capability in feature subset selection, contributing to improved overall performance. In this context, the MSAs that achieve the highest and second-highest accuracies are highlighted in bold and underlined, respectively, to facilitate a clear comparison. Furthermore, these accuracy results are juxtaposed with those from existing related studies to benchmark the efficacy of TTAO and verify its potential superiority in various classification tasks. This thorough comparison provides a more robust understanding of the advancements TTAO offers over previously proposed methods.

Table II shows that BeSD, ODSFMFO, and DSSA generally exhibit subpar performance when used as wrapper-based feature selection techniques across the ten benchmark datasets. The performance deficits are particularly pronounced in datasets with a large number of features or output classes. Specifically, BeSD recorded the lowest classification accuracies in three datasets (DS1, DS2, and DS6) and the second lowest in another (DS4). ODSFMFO displayed the poorest performance in two datasets (DS3 and DS4) and was second poorest in another two (DS8 and DS9). DSSA consistently ranked as having the second-worst average classification accuracy in four datasets (DS1, DS2, DS6, and DS7). Conversely, CO-HDEPSO and FDA demonstrated moderate performance, with their classification accuracies neither exceptionally high nor low across the majority of the datasets.

The wrapper-based feature selection techniques utilizing the three MSAs, including TTAO, have exhibited exemplary performance across the ten evaluated datasets. Specifically, CHIO recorded the highest average classification accuracy in two datasets (DS1 and DS5) and the second highest in four others (DS2 to DS4 and DS7). GNDO showed robust performance, achieving the highest classification accuracy in three datasets (DS2, DS5, and DS8) and the second highest in two (DS1 and DS9). However, both CHIO and GNDO demonstrated limitations in certain datasets, such as DS8 for CHIO and DS2 and DS10 for GNDO, indicating a need for improved robustness in diverse feature selection scenarios. TTAO emerged as the most effective MSA, securing the highest accuracy in eight datasets (DS1, DS3 to DS8, and DS10) and the second highest in DS9. Its superior performance, especially in datasets with a large number of features or classes (DS1, DS4, and DS10), underscores TTAO's capability to adeptly handle complex feature selection tasks prevalent in real-world applications.

C. *Performance Comparisons on Average Numbers of Selected Features*

While high classification accuracy is essential for effectively solving the given datasets, minimizing the size of the selected feature subset is equally important to prevent unnecessary complexity in the resulting machine learning models. Reducing feature subsets without compromising accuracy leads to simpler, more interpretable, and computationally efficient models, a critical goal in real-world applications. Achieving an optimal balance between classification accuracy and model simplicity is thus a fundamental validation criterion for evaluating feature selection techniques. To assess this trade-off, the average number of features selected by the KNN classifier is used as an additional performance metric in this study.

TABLE II. COMPARISON OF AVERAGE CLASSIFICATION ACCURACIES OF ALL MSAS FOR FEATURE SELECTION

No	BeSD	CHIO	CO-HDEPSO	DSSA	FDA	GNDO	ODSFMFO	TTAO
DS1	5.670E-01	**1.000E+00**	**1.000E+00**	6.130E-01	8.920E-01	*9.130E-01*	8.600E-01	**1.000E+00**
DS2	9.680E-01	*9.800E-01*	*9.800E-01*	9.710E-01	9.780E-01	**9.810E-01**	9.740E-01	9.750E-01
DS3	9.310E-01	*9.510E-01*	9.380E-01	9.290E-01	9.440E-01	9.270E-01	9.000E-01	**9.840E-01**
DS4	6.210E-01	*7.360E-01*	7.040E-01	6.620E-01	6.730E-01	6.730E-01	6.050E-01	**7.440E-01**
DS5	**1.000E+00**	**1.000E+00**	**1.000E+00**	**1.000E+00**	*9.790E-01*	**1.000E+00**	**1.000E+00**	**1.000E+00**
DS6	6.440E-01	8.280E-01	7.950E-01	7.540E-01	*8.300E-01*	8.010E-01	7.870E-01	**8.360E-01**
DS7	*7.810E-01*	*7.810E-01*	7.570E-01	7.520E-01	7.420E-01	7.700E-01	7.650E-01	**8.220E-01**
DS8	9.930E-01	9.310E-01	*9.990E-01*	9.560E-01	9.540E-01	**1.000E+00**	9.460E-01	**1.000E+00**
DS9	**7.680E-01**	7.370E-01	7.460E-01	7.430E-01	7.500E-01	*7.670E-01*	7.270E-01	*7.670E-01*
DS10	*9.970E-01*	9.550E-01	9.950E-01	9.820E-01	9.950E-01	8.900E-01	9.870E-01	**1.000E+00**

TABLE III. COMPARISON OF AVERAGE NUMBERS OF SELECTED FEATURES BY ALL MSAS FOR FEATURE SELECTION

No	BeSD	CHIO	CO-HDEPSO	DSSA	FDA	GNDO	ODSFMFO	TTAO
DS1	26.30	20.33	10.90	*9.33*	15.03	14.23	28.93	**5.23**
DS2	325.23	*289.13*	305.93	441.23	309.70	302.17	446.93	**243.30**
DS3	16.80	13.80	*7.83*	12.47	10.27	10.63	16.87	**4.33**
DS4	135.80	112.17	119.13	*83.70*	131.47	126.97	155.87	**49.93**
DS5	3.83	*2.60*	**1.00**	7.10	**1.00**	**1.00**	3.63	**1.00**
DS6	6.67	6.13	4.57	*2.03*	5.33	6.53	2.30	**2.00**
DS7	4.67	4.07	5.03	**2.27**	4.97	4.80	6.20	*3.87*
DS8	6.23	5.60	4.03	5.37	3.67	**2.97**	9.00	*3.00*
DS9	3.70	3.57	3.30	*3.17*	**3.00**	4.00	5.00	**3.00**
DS10	8.53	6.27	*4.97*	12.13	5.00	6.80	9.97	**3.07**

Table III details the average number of features selected by all MSAs, implemented as wrapper-based feature selection techniques across 30 simulation runs for each dataset. The MSAs achieving the smallest and second-smallest feature subset sizes for each dataset are highlighted in bold and underlined text, respectively. In addition to their poor performance in terms of average classification accuracy, the results also reveal that ODSFMFO and BeSD are notably ineffective in minimizing the number of selected features, often yielding the largest and second-largest feature subsets across most datasets. Specifically, ODSFMFO consistently produced the largest feature subsets in seven datasets (DS1 to DS4, DS7 to DS9) and the second largest in one dataset (DS10). Meanwhile, BeSD was frequently associated with the second-largest feature subsets in six datasets (DS1, DS3 to DS6, and DS8). In contrast, CO-HDEPSO and FDA exhibited moderate performance, maintaining an average number of selected features that was neither particularly high nor low across most datasets.

Moreover, certain MSAs demonstrate inconsistent performance across both evaluation metrics, highlighting their inability to effectively balance the trade-off between model accuracy and complexity. For instance, while CHIO and GNDO achieve competitive average classification accuracies across most datasets, they fall short in consistently identifying smaller feature subsets that could reduce machine learning model complexity. Conversely, DSSA successfully identified the smallest feature subset size for one dataset (DS7) and the second smallest for four others (DS1, DS4, DS6, and DS9). However, DSSA ranks among the poorest in terms of average classification accuracy, as detailed in Table II. Additionally, DSSA was found to select the largest feature subsets for two datasets (DS5 and DS10) and the largest subset for another (DS2), indicating its potential inconsistency in handling datasets with diverse characteristics.

Contrary to the other MSAs evaluated, TTAO has exhibited superior performance by consistently identifying the smallest average number of selected features in eight datasets (DS1 to DS6, DS9, and DS10) and the second smallest in two others (DS7 and DS8). This emphasizes the effectiveness of TTAO's inherent search mechanism in optimally selecting feature subsets across datasets with diverse characteristics, thereby reducing the complexity of the machine learning models. The results presented in Tables II and III affirm TTAO's excellence in harmonizing accuracy with model simplicity, effectively addressing the challenges associated with feature selection.

D. Discussion

A key strength of TTAO lies in its ability to achieve an optimal trade-off between classification accuracy and feature subset size, primarily due to the effective balance it strikes between exploration and exploitation. The presence of both generic and local aggregation mechanisms enables TTAO to explore the search space while refining promising solutions, thus ensuring a well-balanced search process. In feature selection, high classification accuracy alone is insufficient if the feature subset is excessively large, as it can lead to overly complex models that are difficult to interpret and computationally expensive. TTAO addresses this issue by selecting smaller feature subsets while maintaining high accuracy, making it valuable in applications where simplicity and efficiency are critical. This balance is crucial for developing robust machine learning models that generalize well to unseen data. TTAO's consistent ability to reduce feature subset size across diverse datasets without sacrificing accuracy demonstrates the efficacy of its search strategies, allowing it to perform effectively even in high-dimensional spaces or datasets with multiple output classes, where many other algorithms tend to struggle.

Another practical advantages of TTAO over other MSAs is its reduced reliance on extensive parameter tuning. Many MSAs require fine-tuning of algorithm-specific parameters to balance exploration (global search) and exploitation (local search) effectively. In contrast, TTAO's performance depends primarily on the population size N and a few stochastically generated random variables (i.e, r_0, r_1, r_1' r_1 and r_4), all of which require minimal adjustment. This reduction in parameter dependency simplifies the application of TTAO to different problem domains. By ensuring that the algorithm performs well without requiring extensive experimentation to find the optimal parameter settings, TTAO is highly adaptable and user-friendly. This makes it particularly attractive for real-world applications where tuning complex algorithmic parameters may not be feasible due to time constraints or lack of domain expertise.

The consistent performance of TTAO across a wide range of datasets indicates its versatility in tackling real-world feature selection problems. Its ability to handle datasets with high-dimensional features and varying class distributions highlights its robustness and generalizability. Furthermore, the efficiency of TTAO to reduce feature subsets without compromising accuracy can have significant practical implications. For example, in industries where computational resources are limited or where model interpretability is crucial (e.g., such as healthcare, finance, or sensor-based monitoring systems), TTAO's approach can lead to more efficient models with fewer features, ultimately reducing training time, memory requirements, and the risk of overfitting.

V. Conclusion

This paper introduces a novel wrapper-based feature selection technique utilizing the Triangulation Topology Aggregation Optimizer (TTAO), which is inspired by the geometric properties of triangular topology and principles of triangular similarity. Unlike its prior applications to real-valued decision variable problems, this study explores TTAO's adaptability to challenging real-world optimization problems with binary solution spaces. To facilitate this adaptation, a conversion mechanism is employed to transform continuous decision variables into binary ones, thus enabling the inherently real-valued TTAO for use in binary domains. TTAO generates diverse triangular topological units of consistent shape but varying sizes, serving as dynamic evolutionary entities throughout the optimization process. It incorporates two primary search strategies, generic and local aggregation, designed to balance exploration and exploitation effectively. Extensive simulations compare TTAO's performance in feature selection against seven other metaheuristic search algorithms (MSAs). The results indicate varied performances among the MSAs, with some underperform across both matrices (i.e., classification accuracy and feature subset size), while others fail to achieve a satisfactory balance. In contrast, the TTAO-based wrapper method excels, demonstrating an outstanding ability to achieve superior classification accuracy while minimizing feature subset size, thereby solving datasets with varied characteristics effectively.

While TTAO has shown excellent performance across the datasets used in this study, its scalability to extremely large datasets (in terms of both the number of features and instances) remains untested. Future research is needed to assess its computational efficiency and performance under more demanding, large-scale conditions. Additionally, the current study has focused on datasets with relatively balanced class distributions. TTAO's performance in highly imbalanced datasets, where classification bias might occur, has not been thoroughly explored and may require algorithmic adjustments to address such challenges. One promising direction for future research is to explore hybridization between TTAO and other MSAs to further improve performance. Combining the strengths of different algorithms could enhance the ability to balance exploration and exploitation, especially for more complex and dynamic datasets. Another potential extension of this work is applying TTAO in a multi-objective optimization framework, allowing the simultaneous optimization of multiple criteria (e.g., accuracy, computational cost, interpretability) to provide more comprehensive solutions for real-world feature selection tasks.

ACKNOWLEDGMENT

This study was funded by the Malaysian Ministry of Higher Education through the Fundamental Research Grant Scheme (FRGS/1/2024/ICT02/UCSI/02/1) and UCSI University Research Excellence & Innovation Grant (REIG-FETBE-2022/038).

REFERENCES

[1] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," Neural Computing and Applications, 2014, vol. 24, no. 1, pp. 175-186.

[2] T. Berghout and M. Benbouzid, "EL-NAHL: Exploring labels autoencoding in augmented hidden layers of feedforward neural networks for cybersecurity in smart grids," Reliability Engineering & System Safety, 2022, vol. 226, p. 108680.

[3] S. Mahmud Iwan *et al.*, "Spectroscopy data calibration using stacked ensemble machine learning," IIUM Engineering Journal, 2024, vol. 25, no. 1, pp. 208 - 224.

[4] W. Li, M. I. Solihin, and H. A. Nugroho, "RCA: YOLOv8-based surface defects detection on the inner wall of cylindrical high-precision parts," Arabian Journal for Science and Engineering, 2024.

[5] B. Jdid, W. H. Lim, I. Dayoub, K. Hassan, and M. R. B. M. Juhari, "Robust automatic modulation recognition through joint contribution of hand-crafted and contextual features," IEEE Access, 2021, vol. 9, pp. 104530-104546.

[6] Z. H. Ang, C. K. Ang, W. H. Lim, L. J. Yu, and M. I. Solihin, "Development of an artificial intelligent approach in adapting the characteristic of polynomial trajectory planning for robot manipulator," International Journal of Mechanical Engineering and Robotics Research, 2020.

[7] M. I. Solihin;, Y. Shameem;, T. Htut;, C. K. Ang;, and M. b. Hidayab', "Non-invasive blood glucose estimation using handheld near infra-red device," International Journal of Recent Technology and Engineering, 2019, vol. 8, no. 3, pp. 16-19.

[8] T. B. Hong et al., "Intelligent kitchen waste composting system via deep learning and Internet-of-Things (IoT)," Waste and Biomass Valorization, 2024, vol. 15, no. 5, pp. 3133-3146.

[9] O. O. Akinola, A. E. Ezugwu, J. O. Agushaka, R. A. Zitar, and L. Abualigah, "Multiclass feature selection with metaheuristic optimization algorithms: a review," Neural Computing and Applications, 2022, vol. 34, no. 22, pp. 19751-19790.

[10] Y. Zhang, X.-f. Song, and D.-w. Gong, "A return-cost-based binary firefly algorithm for feature selection," Information Sciences, 2017, vol. 418-419, pp. 561-574.

[11] A. Zakeri and A. Hokmabadi, "Efficient feature selection method using real-valued grasshopper optimization algorithm," Expert Systems with Applications, 2019, vol. 119, pp. 61-72.

[12] A. Machmudah et al., "Design optimization of a gas turbine engine for marine applications: off-design performance and control system considerations," Entropy, 2022, vol. 24, no. 12, p. 1729.

[13] L. Yao and W. H. Lim, "Optimal purchase strategy for demand bidding," IEEE Transactions on Power Systems, 2018, vol. 33, no. 3, pp. 2754-2762.

[14] Z. Danin, A. Sharma, M. Averbukh, and A. Meher, "Improved moth flame optimization approach for parameter estimation of induction motor," Energies, 2022, vol. 15, no. 23, p. 8834.

[15] L. Yao, W. H. Lim, S. S. Tiang, T. H. Tan, C. H. Wong, and J. Y. Pang, "Demand bidding optimization for an aggregator with a genetic algorithm," Energies, 2018, vol. 11, no. 10, p. 2498.

[16] A. Machmudah;, E. A. Bakar;, R. Rajendran;, W. H. Nugroho;, M. I. Solihin;, and A. Ghofur, "Control system optimisation of biodiesel-based gas turbine for ship propulsion," IAES International Journal of Artificial Intelligence, 2024, vol. 13, no. 2, pp. 1990-2000.

[17] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: A recent review based on state-of-the-art works," Alexandria Engineering Journal, 2022, vol. 61, no. 5, pp. 3831-3872.

[18] S. Zhao, T. Zhang, L. Cai, and R. Yang, "Triangulation topology aggregation optimizer: A novel mathematics-based meta-heuristic algorithm for continuous optimization and engineering applications," Expert Systems with Applications, 2024, vol. 238, p. 121744.

[19] A. Almalaq, K. Alqunun, R. Abbassi, Z. M. Ali, M. M. Refaat, and S. H. E. Abdel Aleem, "Integrated transmission expansion planning incorporating fault current limiting devices and thyristor-controlled series compensation using meta-heuristic optimization techniques," Scientific Reports, 2024, vol. 14, no. 1, p. 13046.

[20] M. A. Elaziz, F. A. Essa, H. A. Khalil, M. S. El-Sebaey, M. Khedr, and A. Elsheikh, "Productivity prediction of a spherical distiller using a machine learning model and triangulation topology aggregation optimizer," Desalination, 2024, vol. 585, p. 117744.

[21] M. A. Zeidan, M. R. Hammad, A. I. Megahed, K. M. AboRas, A. Alkuhayli, and N. Gowtham, "Enhancement of a hybrid electric shipboard microgrid's frequency stability with triangulation topology aggregation optimizer-based 3DOF-PID-TI controller," IEEE Access, 2024, vol. 12, pp. 66625-66645.

[22] W.-L. Cheng et al., "Flow direction algorithm for feature selection," Singapore, 2023: Springer Nature Singapore, in Advances in Intelligent Manufacturing and Mechatronics, pp. 187-198.

[23] W.-L. Cheng et al., "Feature selection of medical dataset using african vltures optimization algorithm," Singapore, 2023: Springer Nature Singapore, in Advances in Intelligent Manufacturing and Mechatronics, pp. 175-185.

[24] W.-L. Cheng et al., "Wrapper-based feature selection using sperm swarm optimization: a comparative study," Singapore, 2024: Springer Nature Singapore, in Advances in Intelligent Manufacturing and Robotics, pp. 343-353.

[25] 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 Systems with Applications, 2019, vol. 117, pp. 267-286.

[26] D. Rodrigues, V. H. C. de Albuquerque, and J. P. Papa, "A multi-objective artificial butterfly optimization approach for feature selection," Applied Soft Computing, 2020, vol. 94, p. 106442.

[27] R. R. Mostafa, A. A. Ewees, R. M. Ghoniem, L. Abualigah, and F. A. Hashim, "Boosting chameleon swarm algorithm with consumption AEO operator for global optimization and feature selection," Knowledge-Based Systems, 2022, vol. 246, p. 108743.

[28] B. J. Ma, S. Liu, and A. A. Heidari, "Multi-strategy ensemble binary hunger games search for feature selection," Knowledge-Based Systems, 2022, vol. 248, p. 108787.

[29] R. Wu et al., "An improved sparrow search algorithm based on quantum computations and multi-strategy enhancement," Expert Systems with Applications, 2023, vol. 215, p. 119421.

[30] C. Zhong, G. Li, Z. Meng, H. Li, and W. He, "A self-adaptive quantum equilibrium optimizer with artificial bee colony for feature selection," Computers in Biology and Medicine, 2023, vol. 153, p. 106520.

[31] D.-S. Khafaga, E.-S.-M. El-kenawy, F. Alrowais, S. Kumar, A. Ibrahim, and A.-A. Abdelhamid, "Novel optimized feature selection using metaheuristics applied to physical benchmark datasets," Computers, Materials \& Continua, 2023 vol. 74, no. 2, pp. 4027--4041.

[32] D.-S. Khafaga et al., "Hybrid dipper throated and grey wolf optimization for feature selection applied to life benchmark datasets," Computers, Materials \& Continua, 2023, vol. 74, no. 2, pp. 4531--4545.

[33] A. A. Abdelhamid et al., "Innovative feature selection method based on hybrid sine cosine and dipper throated optimization algorithms," IEEE Access, 2023, vol. 11, pp. 79750-79776.

[34] P. Civicioglu and E. Besdok, "Bezier Search Differential Evolution Algorithm for numerical function optimization: A comparative study with CRMLSP, MVO, WA, SHADE and LSHADE," Expert Systems with Applications, 2021, vol. 165, p. 113875.

[35] M. A. Al-Betar, Z. A. A. Alyasseri, M. A. Awadallah, and I. Abu Doush, "Coronavirus herd immunity optimizer (CHIO)," Neural Computing and Applications, 2021, vol. 33, no. 10, pp. 5011-5042.

[36] Z. C. Choi et al., "Hybridized metaheuristic search algorithm with modified initialization scheme for global optimization," Cham, 2021: Springer International Publishing, in Advances in Robotics, Automation and Data Analytics, pp. 172-182.

[37] B. Jena, M. K. Naik, A. Wunnava, and R. Panda, "A differential squirrel search algorithm," Singapore, 2021: Springer Singapore, in Advances in Intelligent Computing and Communication, pp. 143-152.

[38] H. Karami, M. V. Anaraki, S. Farzin, and S. Mirjalili, "Flow Direction Algorithm (FDA): a novel optimization approach for solving optimization problems," Computers & Industrial Engineering, 2021, vol. 156, p. 107224.

[39] Y. Zhang, Z. Jin, and S. Mirjalili, "Generalized normal distribution optimization and its applications in parameter extraction of photovoltaic models," Energy Conversion and Management, 2020, vol. 224, p. 113301.

[40] Z. Li, J. Zeng, Y. Chen, G. Ma, and G. Liu, "Death mechanism-based moth–flame optimization with improved flame generation mechanism for global optimization tasks," Expert Systems with Applications, 2021, vol. 183, p. 115436.