Acta Informatica Pragensia
2025, Volume 14, Issue 1, 88–111
https://doi.org/10.18267/j.aip.254

PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

Article Open Access

Induced Partitioning for Incremental Feature Selection via Rough Set Theory and Long-tail Position Grey Wolf Optimizer

Said Al Afghani Edsa, Khamron Sunat

Department of Computer Science, College of Computing, Khon Kaen University, Khon Kaen, Thailand

Corresponding author: Khamron Sunat (skhamron@kku.ac.th)

Editorial Record

First submission received:
August 11, 2024

Revisions received:
October 12, 2024
November 14, 2024
November 25, 2024

Accepted for publication:
November 26, 2024

Academic Editor:
Zdenek Smutny
Prague University of Economics and Business, Czech Republic

This article was accepted for publication by the Academic Editor upon evaluation of the reviewers' comments.

How to cite this article:
Edsa, S. A. A., & Sunat, K. (2025). Induced Partitioning for Incremental Feature Selection via Rough Set Theory and Long-Tail Position Grey Wolf Optimizer. *Acta Informatica Pragensia*, 14(1), 88–111. https://doi.org/10.18267/j.aip.254

Abstract

Background: Feature selection methods play a crucial role in handling challenges such as imbalanced classes, noisy data and high dimensionality. However, existing techniques, including swarm intelligence and set theory approaches, often struggle with high-dimensional datasets due to repeated reassessment of feature selection, leading to increased processing time and computational inefficiency.

Objective: This study aims to develop an enhanced incremental feature selection method that minimizes dependency on the initial dataset while improving computational efficiency. Specifically, the approach focuses on dynamic sampling and adaptive optimization to address the challenges in high-dimensional data environments.

Methods: We implement a dynamic sampling approach based on rough set theory, integrating the Long-Tail Position Grey Wolf Optimizer. This method incrementally adjusts to new data samples without relying on the original dataset for feature selection, reducing variance in partitioned datasets. The performance is evaluated on benchmark datasets, comparing the proposed method to existing techniques.

Results: Experimental evaluations demonstrate that the proposed method outperforms existing techniques in terms of F1 score, precision, recall and computation time. The incremental adjustment and reduced dependence on the initial data improve the overall accuracy and efficiency of feature selection in high-dimensional contexts.

Conclusion: This study offers a significant advancement in feature selection methods for high-dimensional datasets. By addressing computational demands and improving accuracy, the proposed approach contributes to data science and machine learning, paving the way for more efficient and reliable feature selection processes in complex data environments. Future work may focus on extending this method to new optimization frameworks and enhancing its adaptability.

Index Terms

Optimizer; Rough set theory; Feature selection; Incremental; Data partitioning.

1 INTRODUCTION

The development of feature selection methods has been extensively pursued in both practical and academic contexts. This process is crucial due to the vast amount of attribute information that datasets may contain, which requires significant storage space for later analysis or related tasks. Not every attribute uniquely contributes to information; some samples and their attributes may contain redundant information, making the process inefficient.

Feature selection has evolved through various methodologies, including wrapper methods, filter-based methods and their combinations. Metaheuristic-based feature selection has also seen notable development (Abdel-Basset et al., 2020; Al Afghani Edsa & Sunat, 2023; Gu et al., 2018; Kwakye et al., 2024; Pan et al., 2023; Pichai et al., 2020; Sharma & Kaur, 2021; Shikoun et al., 2024; Tran et al., n.d.); however, its reliance on randomness and reassessment across all samples makes it time-consuming. Moreover, the features selected are not guaranteed to be unique or non-correlated.

According to the literature, the principles of rough set theory have been widely applied to feature selection, primarily following two approaches: non-incremental and incremental. Non-incremental selection does not utilize previously acquired information about selected features, making it time-consuming. In contrast, incremental selection makes use of previously identified optimal features, improving computational efficiency. Incremental methods based on rough set theory have been further developed using various techniques such as credibility matrices, dependency measures, information theory and sample feature-based methods. Despite these advancements (F. Li et al., 2016; Yang et al., 2018, 2020), certain aspects, such as the determination of discernibility scores, sample usage in the process and feature assessment in existing rough set theory, still depend on static rules and the reassessment of original data, which can lead to higher computational costs.

Class imbalance must also be considered for feature selection as it can significantly affect model performance. Methods such as under-sampling, over-sampling and SMOTE aim to balance class samples but face challenges with extreme imbalances in multi-class classification, where synthetic data generation may introduce noisy samples. Rough set theory offers a promising approach to mitigate issues related to class imbalance and high-dimensional data, independent of reliance on original data.

Based on these considerations, this study proposes a feature selection method motivated by rough set theory and inspired by the sample and feature selection method introduced by Yang et al. (2022). To make incremental feature selection not rule-based by the user, we propose an adaptive procedure during the assessment of features. To gain significant features, we shuffle the samples by ordering them to help calculate the discernibility score and achieve optimal features. Unlike Yang et al. (2022), our proposed method does not rely on original data during feature assessment. Data partition is induced by a long-tail position grey wolf optimizer (GWO) to minimize variance in each part, enhancing data partition. The GWO is a strong optimizer with fast convergence but can easily get trapped in local optima, as introduced by Mirjalili et al. (2014). To address this, we introduce long-tail positions in the GWO, allowing less frequently assessed areas to be explored, enhancing the exploration and exploitation capabilities of the GWO. This scheme reduces computation time and provides more flexibility for incoming new data. The performance of the model with the selected features will be evaluated and compared to that of a model without feature selection and other relevant methods.

The contributions of this paper are as follows:

1. Feature selection using rough set theory with dynamic incremental feature assessment criteria and optimized data partition by long-tail GWO.
2. Ordered sample selection during feature assessment to achieve optimal selected features.
3. The proposed procedure does not rely on original data, saving computation time during feature assessment.
4. The proposed method is compared across experimental datasets and its performance is benchmarked against similar methods.

Subsequently, Section 2 discusses related work, Section 3 covers the proposed method, Section 4 presents the results and discussion and Section 5 addresses conclusions and future work.

2 RELATED WORKS

In this section, we will discuss related work on feature selection concerning high-dimensional data, imbalanced classes and noisy samples. The sources included in this overview were selected based on their relevance to these challenges and their contributions to addressing feature selection in various contexts, such as bio-inspired algorithms, feature extraction techniques and rough set approaches.

Several studies, including Pan et al. (2023), have explored bio-inspired or metaheuristic algorithms for solving feature selection problems in high-dimensional data. While these approaches are effective in searching for global or

near-optimal solutions, they have a drawback: recalculating the fitness function over the same samples that have already been computed in previous iterations. This process becomes inefficient when dealing with a large number of features, for instance, datasets with 300 or more features, or many samples. On the other hand, feature extraction techniques such as Principal Component Analysis (PCA) are commonly used to reduce dimensionality. However, this method struggles with high-dimensional data as it requires calculating a covariance matrix before determining the principal components through eigenvalues and eigenvectors. Additionally, feature extraction offers less interpretability since it combines the original features, making it difficult to identify which features affect the model.

Rough set theory provides another promising approach for handling high-dimensional, imbalanced and noisy data. For example, Y. Li et al. (2023) applied rough set theory to classification problems by using attribute grouping and dynamic neighbourhood rough sets to manage samples.

A significant body of research has aimed at developing effective feature selection methods in both supervised (Khan et al., 2017; Zhang et al., 2024) and unsupervised learning tasks (Dehghan & Mansoori, 2018; Hancer et al., 2020; Zhao et al., 2023). These studies address both high- and lower-dimensional data, considering feature count and sample size. Supervised learning-based feature selection methods often follow the wrapper approach, where the objective function of the learning algorithm is combined with a feature selection process to select only a subset of features—typically fewer than the original set. Metaheuristic algorithms, such as those in Jia et al. (2019) or Shikoun et al. (2024), have been particularly popular due to their capacity to search globally for optimal or near-optimal solutions. However, the iterative recalculation of samples remains a challenge for high-dimensional data, as the process is computationally expensive.

On the other hand, imbalanced data containing noise also affect the effectiveness of predictive models, necessitating a method to assist machine learning models in making accurate predictions. In the case of data balancing, problems can arise, particularly in extreme multi-class scenarios where the class ratio is highly disproportionate. Therefore, a method that does not solely rely on original data or data augmentation is needed. One approach is to examine the characteristics of the samples and their features related to class attributes. Rough set theory is considered suitable for addressing these issues, especially in high-dimensional datasets. The following table provides selected recent research studies related to feature selection in high-dimensional datasets, addressing imbalanced data in both multi-class and binary class cases:

Table 1. *Summary of related studies.*

Reference	Proposed method/framework	Strengths	Opportunities for improvement
Yang et al. (2022)	Incremental feature selection by sample selection and feature-based accelerator	The proposed method updates feature subsets without forgetting previous knowledge, avoids redundant calculations and reduces computational and memory usage. The sample selection scheme filters out irrelevant data and the feature-based accelerator incrementally selects the best features while removing redundant ones, ensuring an efficient and effective feature selection process.	It still relies on original data during feature assessment, data are neither shuffled nor reordered and the incremental threshold is rule-based.
Asniar et al. (2022)	SMOTE-LOF for noise identification in imbalanced data classification	The proposed SMOTE-LOF method strengthens the standard SMOTE approach by using the Local Outlier Factor (LOF) to remove noise from synthetic samples.	Not suitable for small samples and extreme multi-class ratios.
Pan et al. (2023)	High-dimensional feature selection method based on modified grey wolf optimization	Utilizes advanced sigmoid principles in feature selection, suitable for relatively high-dimensional data.	Recalculation for feature selection consumes significant time, especially in high-dimensional data.
Ma et al, (2024)	Membership-based resampling and cleaning algorithm for multi-class imbalanced overlapping data	The MC-MBRC algorithm addresses class imbalance, noise and overlapping classes in multi-class datasets by categorizing samples based on membership degrees and applying targeted operations such as noise removal and oversampling.	Limited to specific types of noise where some samples have maximal noise levels.
Gilal et al. (2019)	Rough-Fuzzy Model for Early Breast Cancer Detection	The study employs a Rough-Fuzzy hybrid technique to create an early breast cancer detection model,	Limited to specific datasets.

Reference	Proposed method/framework	Strengths	Opportunities for improvement
		providing a cost-effective alternative to traditional detection methods.	
C. Wang et al. (2021)	Attribute reduction with fuzzy rough self-information measures	This study introduces four uncertainty measures that integrate fuzzy-rough approximations with self-information concepts, addressing both lower and upper approximations of fuzzy decisions. The fourth measure, relative decision self-information, is highlighted for its superior attribute reduction capability compared to others. These measures generalize conventional methods based on fuzzy rough sets and are validated through experimental results that show improved efficiency and accuracy in attribute reduction compared to three other algorithms.	Exploring fuzzy self-information across multiple granularities.
Meng & Shi (2016)	Quick attribute reduction in decision-theoretic rough set models	Developing an efficient reduction algorithm specifically designed for Decision-Theoretic Rough Set (DTRS) models.	Efficiency in terms of computation time.
Raza & Qamar (2018)	Feature selection using rough set-based direct dependency calculation by avoiding the positive region	The strength lies in the Direct Dependency Calculation (DDC) approach for assessing attribute dependency in feature selection.	The effectiveness of DDC in wrapper techniques, where the classification algorithm feedback measures the selected feature quality, remains untested. Future work will focus on integrating DDC with wrapper-based algorithms.
Jia et al. (2019)	Spotted Hyena Optimization Algorithm With Simulated Annealing for Feature Selection	The combination of two metaheuristic methods enhances their individual performance in selecting features based on the fitness function.	The experiments did not utilize high-dimensional data and the method involves recalculating the fitness function at each iteration on the same samples, which could be optimized.
J. Li et al. (2020)	Elephant Herding Optimization: Variants, Hybrids and Applications	Applicable to various domains, particularly for feature selection using common principles that utilize metaheuristic methods.	There is a need to design optimization operators and apply them to high-dimensional data for feature selection.
Xu et al. (2019)	Applying an Improved Elephant Herding Optimization Algorithm with Spark-based Parallelization to Feature Selection for Intrusion Detection	The improved version utilizes Lévy flight to enhance the original Elephant Herding Algorithm for global optimization.	The study used a relatively small sample (fewer than 50 features) and fewer than 250 samples, suggesting that it can be adjusted to accommodate a larger number of samples and features.
Arora & Agarwal (2024)	Empirical Study of Nature-Inspired Algorithms for Feature Selection in Medical Applications	The survey paper employs various nature-inspired algorithms to identify the most effective one.	One challenge is the number of particles or chromosomes used in the algorithms. Another challenge is the presence of redundant solutions, as many datasets exhibit multimodal solutions. This implies that there can be multiple subsets of features with the same size and error. The nature-inspired algorithms should be capable of detecting all such subsets.
Premalatha et al. (2024)	Comparative evaluation of nature-inspired algorithms for feature selection problems	Various metaheuristic techniques inspired by human behaviour and mammalian traits are compared for their efficiency in feature selection problems.	A challenge remains for researchers to design a universal algorithm that excels across all dimensions, particularly for high-dimensional datasets, as the work utilized relatively small samples.

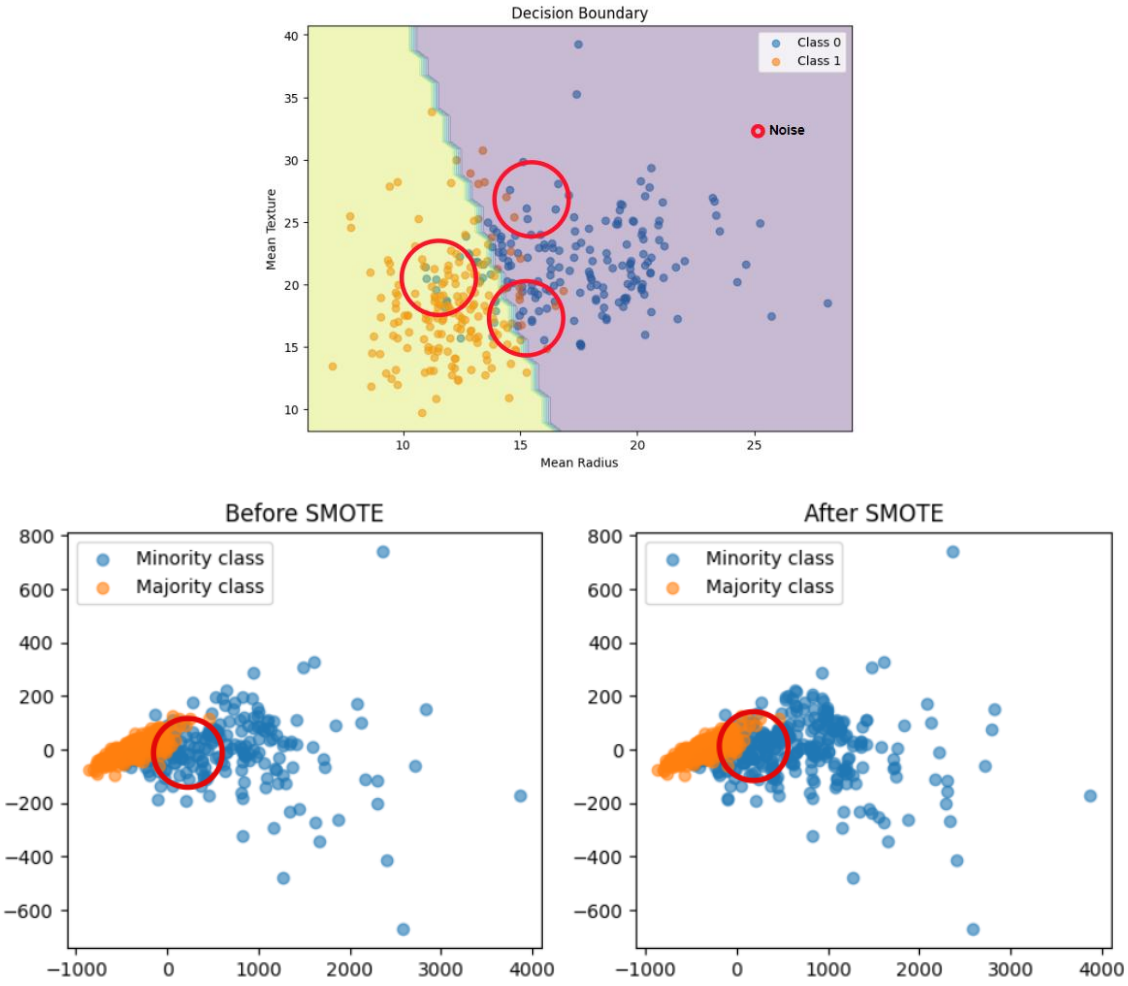


Figure 1. *Illustration of noisy sample.*

As we observed earlier, although we can generate synthetic data using methods such as SMOTE, the generated data can introduce new challenges, including noise. This issue may arise because some features have overlapping information between different object classes. Therefore, it might be beneficial to reduce the features that create overlapping information from different classes. This motivates us to use rough set theory, specifically discernible and incremental discernible scores, to reduce features with overlapping information and to decrease computation time. The feature assessment will not rely on the original data for each assessment and samples will be ordered to help detect optimal samples (including optimized sample partition) and features.

3 PROPOSED METHOD

This research combines a literature review and experimental methods. Initially, a procedural function was developed based on rough set theory using Python. This function was then applied to derive discernible scores. The rough set principle can also be likened to streaming processes, where new data are compared to an existing partitioned dataset (not necessarily the original data). This approach ensures that the learning process does not restart with historical data, thereby reducing learning time. As shown in Figure 2, the original data are divided into k partitions. Some partitions are initially used for sample selection based on discernible scores, while subsequent partitions are incrementally processed, simulating a streaming process with no overlapping samples. The partitioned data are optimized using the long-tail position Grey Wolf Optimizer and the original Grey Wolf Optimizer to assess their effectiveness. Following sample selection, an incremental feature selection process is applied to identify optimal features, resulting in a refined sample set with selected features, which is then used for model training. This process removes unnecessary samples, ensuring that only genuinely new samples and optimal features are included, distinguishing our study from the method proposed by Yang et al. (2022).

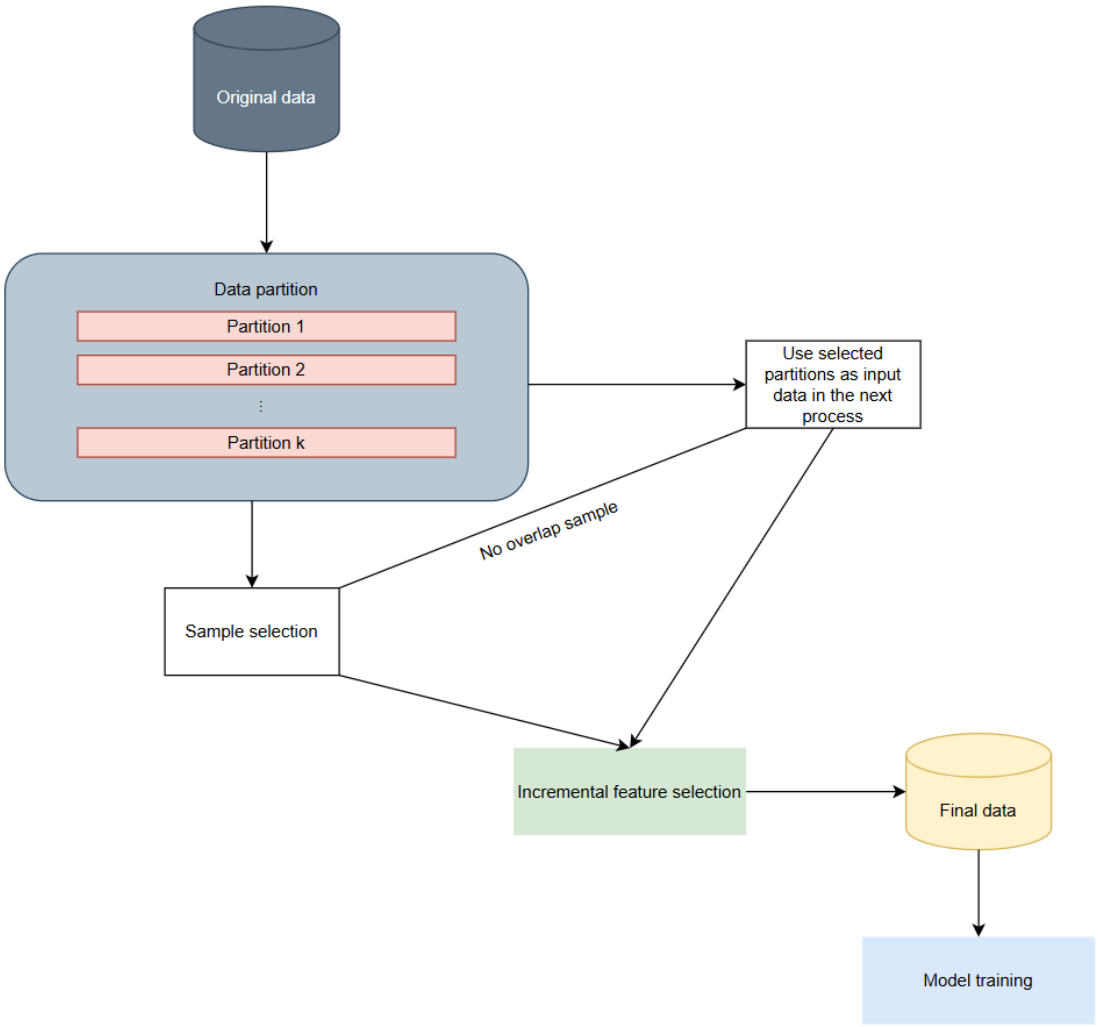


Figure 2. *Overview of proposed method.*

3.1 Rough set theory for sample and feature selection

Pawlak introduced the concept of rough sets in 1982. This theory has been utilized in database mining and knowledge discovery within relational databases. The rough set approach allows the exploration of structural relationships within data that are imprecise or contain noise. In rough set theory, data are organized in a tabular format, where each row represents a fact or object and the entire table is referred to as an information system. Thus, the information table serves as input data, sourced from various domains. Within these tables, multiple objects may share identical features. To streamline the table, it is common practice to retain only one representative object for each set of objects with identical features. These representative objects are referred to as indiscernible objects or tuples. For any subset P of attributes, there exists an association equivalence relation denoted as $IND(P)$:

$$IND(P) = \{(x, y) \in U^2 \mid \forall a \in P, a(x) = a(y)\}. \quad (1)$$

In this section, rough set theory will be directed towards feature selection, particularly as an incremental approach. While the underlying principle is not novel, the primary contribution of this paper lies in advancing the concept of discernible score and introducing the variant GWO to optimize sample partition. The paper introduces a method to assess features without relying on the initial dataset to avoid recalculations. It incorporates adaptive selection of samples and features during new data streaming, or in other words, during incremental processes. The approach prioritizes ordered sampling using a shuffle principle to achieve optimal feature selection more effectively.

Here, a triple (*U, A, D*) is referred to as a decision table, where *U* represents samples, *A* represents attributes and *D* represents decision features. For a given sample $x \in U$, $A(x) = (a_1(x), a_2(x), \dots, a_m(x))$ and *D(x)* is defined. For the sake of clarity and understanding of the concepts employed in this study, several important definitions and theorems related to them are provided below:

Definition 1 (Yang et al., 2022)

Let (*U, A, D*) be a decision table with $U = \{x_1, x_2, \dots, x_n\}$, $B \subseteq A$. The discernible vector $x \in U$ with respect to B is defined as:

$dis_B(x) = (d_1^B, d_2^B, \dots, d_n^B)$, where

$$d_i^B = \begin{cases} 1, B(x) \neq B(x_i), D(x) \neq D(x_i) \\ 0, \quad otherwise. \end{cases} \tag{2}$$

The discernibility score of B is defined as:

$$ds_B(x) = \frac{1}{n}\sum_{i=1}^{n} |dis_B(x_i)|. \tag{3}$$

By **Definition 1**, we have $0 \leq ds_B \leq 1$. The relationship between discernible vectors or observed samples and the discernible score of features selected for samples and subfeatures is elucidated in the following properties:

Theorem 1 (Yang et al., 2022)
Let (*U, A, D*) be a decision table. Then:

1. $ds_B(U) = ds_A(U)$ if and only if $ds_B(x) = ds_A(x), \forall x \in U$. (4)
2. If $ds_p(U) = ds_A(U)$ then $ds_B(U) = ds_A(U), \forall P \subseteq B \subseteq A$. (5)

Theorem 1 states that if we have subfeatures that cover the discernible score from all features, denoted as B, then B is considered the optimal set of features.

Proof:

1. Clear by definition 1.
2. Suppose there exists $P \subseteq B$ with the condition $ds_p(U) \neq ds_A(U)$, by this we have $ds_B(U) < ds_A(U)$ and $ds_P(U) < ds_B(U)$, which means that $ds_P(U) < ds_A(U)$. Contradicts.

Next, some properties and definitions related to feature selection, feature redundancy and sample selection are provided.

Definition 2 (Yang et al., 2022)

Given a decision table (*U, A, D*), $P \subseteq A$ is an optimal feature subset of (*U, A, D*), also called a reduct if it satisfies:

1. $ds_p(U) = ds_A(U)$ (6)
2. $ds_{p-\{a\}}(U) \neq ds_A(U), \forall a \in P$ (7)

By **Definition 2**, the feature $a \in A - P$ is redundant with P, if $ds_{p\cup\{a\}}(U) = ds_P(U)$.

Now, let *U* be the object, divided into *n* parts, denoted as $U'^1, U'^2, \dots, U'^n$. Consider these *n* samples as incoming each other during feature selection process (which we call an incremental process). We have the following theorems:

Theorem 2 (Yang et al., 2022)

Let (*U, A, D*) be an original dataset, U' be an incoming sample set and $x \notin U'$ if x is previously filtered sample, then the same optimal feature subset is obtained from $(U \cup \{x\} \cup U', A, D)$ and $(U \cup U', A, D)$.

Theorem 3 (Yang et al., 2022)

For $B \subseteq A$ and $x_i \in U$, let $I_B^{U_o}(x_i) = (d_1^B, d_2^B, \dots, d_n^B)$ where for $y_i \epsilon U_o$,

$$d_{ij}^B = \begin{cases} 1, B(x) \neq B(y_i), D(x) \neq D(y_i) \\ 0, \quad otherwise. \end{cases} \tag{8}$$

Thus,

$$ds_B(U \cup U_o) = \frac{1}{(n+n_o)^2}(n^2 ds_B(U) + n_o^2 ds_B(U_o) + 2\sum_{i=1}^{n}|I_B^U(x_i)|). \quad (9)$$

A feature will not undergo further examination during the process of determining features B, if $ds_{P\cup B\cup\{a\}}(U \cup U_o) = ds_{P\cup B}(U \cup U_o)$ holds for $a \in A - P - B$. Furthermore, the feature $a \in P$ will be deleted from P if $ds_{(P-\{a\})\cup B}(U \cup U_o) = ds_A(U \cup U_o)$.

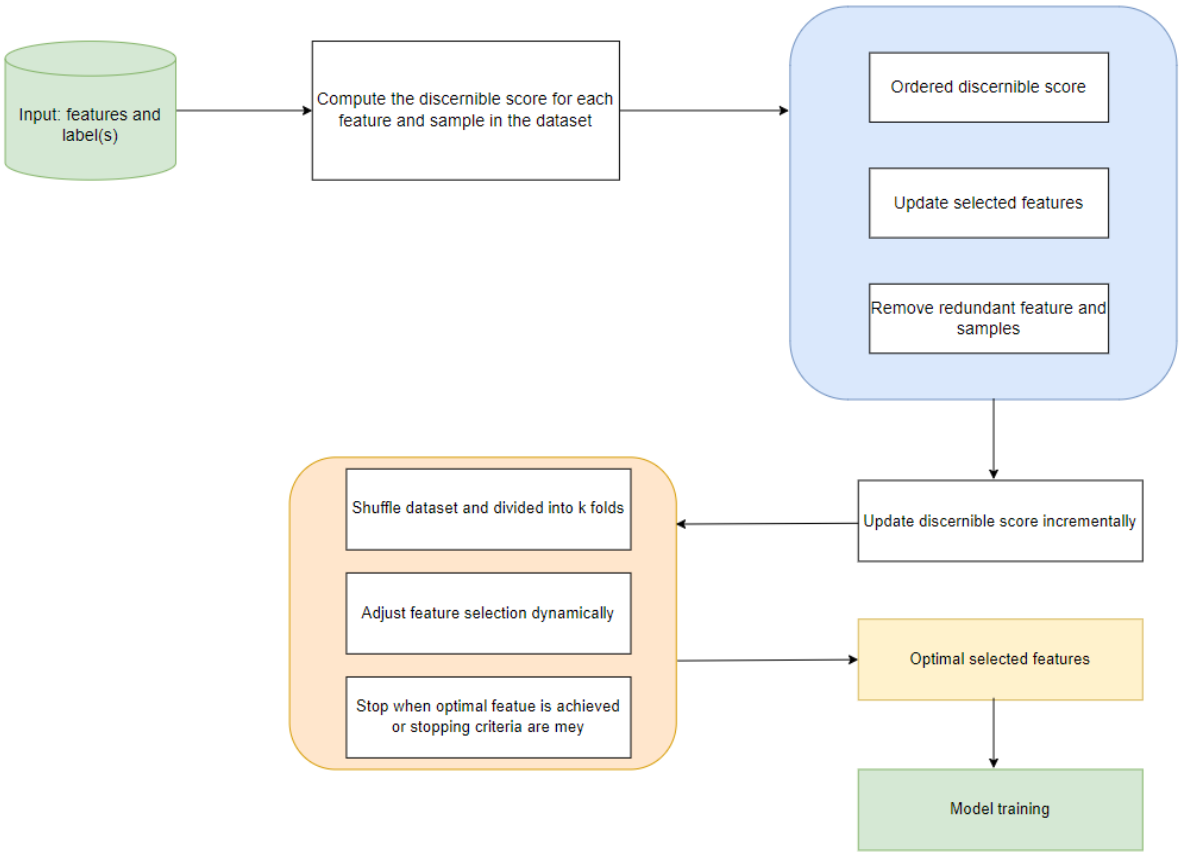


Figure 3. *Process flow of proposed method.*

3.2 Long-tail Position Grey Wolf Optimizer

Grey wolf optimization (GWO) is the swarm intelligence optimization technique which was first introduced in (Mirjalili et al., 2014). It is inspired by the leadership hierarchy and hunting process of the grey wolf in nature. The relatively simple mechanism of the GWO makes it easy to implement over other Nature-Inspired Algorithms (NIAs). Also, it has fewer decision variables, less storage required and does not possess any rigorous mathematical equations of the optimization problem. Mirjalili et al. (2014) explained the hunting behaviour of wolves as follows:

Starting with encircling followed by attacking, where the encircling process is described by the following mathematical equation:

$$\boldsymbol{D} = \boldsymbol{C}.\boldsymbol{X}_{\boldsymbol{p,t}} - \boldsymbol{X}_{\boldsymbol{t}}, \quad (10)$$

$$\boldsymbol{X}_{\boldsymbol{t+1}} = \boldsymbol{X}_{\boldsymbol{p,t}} - \boldsymbol{A}.\boldsymbol{D} \quad (11)$$

where $\boldsymbol{A}$ and $\boldsymbol{C}$ are coefficient vectors, $\boldsymbol{X}_{p,t}$ denotes the position vector of the prey at the current iteration $\boldsymbol{t}$ and $\boldsymbol{X}_{t+1}$ denotes the position vector of a grey wolf at the next iteration. These vectors are determined as:

$$\boldsymbol{A} = 2\boldsymbol{a}.\boldsymbol{r}_1 - \boldsymbol{a}, \quad (12)$$

$$\boldsymbol{C} = 2\boldsymbol{r}_2 \quad (13)$$

where the condition vector $\boldsymbol{a}$ *is a linearly decreasing parameter from 2 to 0 and* $\boldsymbol{r_1}$ *and* $\boldsymbol{r_2}$ are random vectors in [0,1].

To encircle the position of the prey, the wolf position is approximated by the average of the position guided by alpha ($\boldsymbol{X_1}$), beta ($\boldsymbol{X_2}$) and gamma ($\boldsymbol{X_3}$), which can be calculated as:

$$\boldsymbol{D_i} = \left|\boldsymbol{C_i}.\boldsymbol{X_i} - \boldsymbol{X_p}\right|, \tag{14}$$

where i means alpha, beta and delta and $\boldsymbol{p}$ means the position of the prey. Moreover, for each wolf we can estimate their position by:

$$\boldsymbol{X_i} = \boldsymbol{X_i} - \boldsymbol{A_i}.\boldsymbol{D_i}, \tag{15}$$

respectively for alpha, beta and delta. Consequently, the position of the prey in the next iteration can be estimated as:

$$\boldsymbol{X_{t+1}} = \frac{\boldsymbol{X_1}+\boldsymbol{X_2}+\boldsymbol{X_3}}{3}. \tag{16}$$

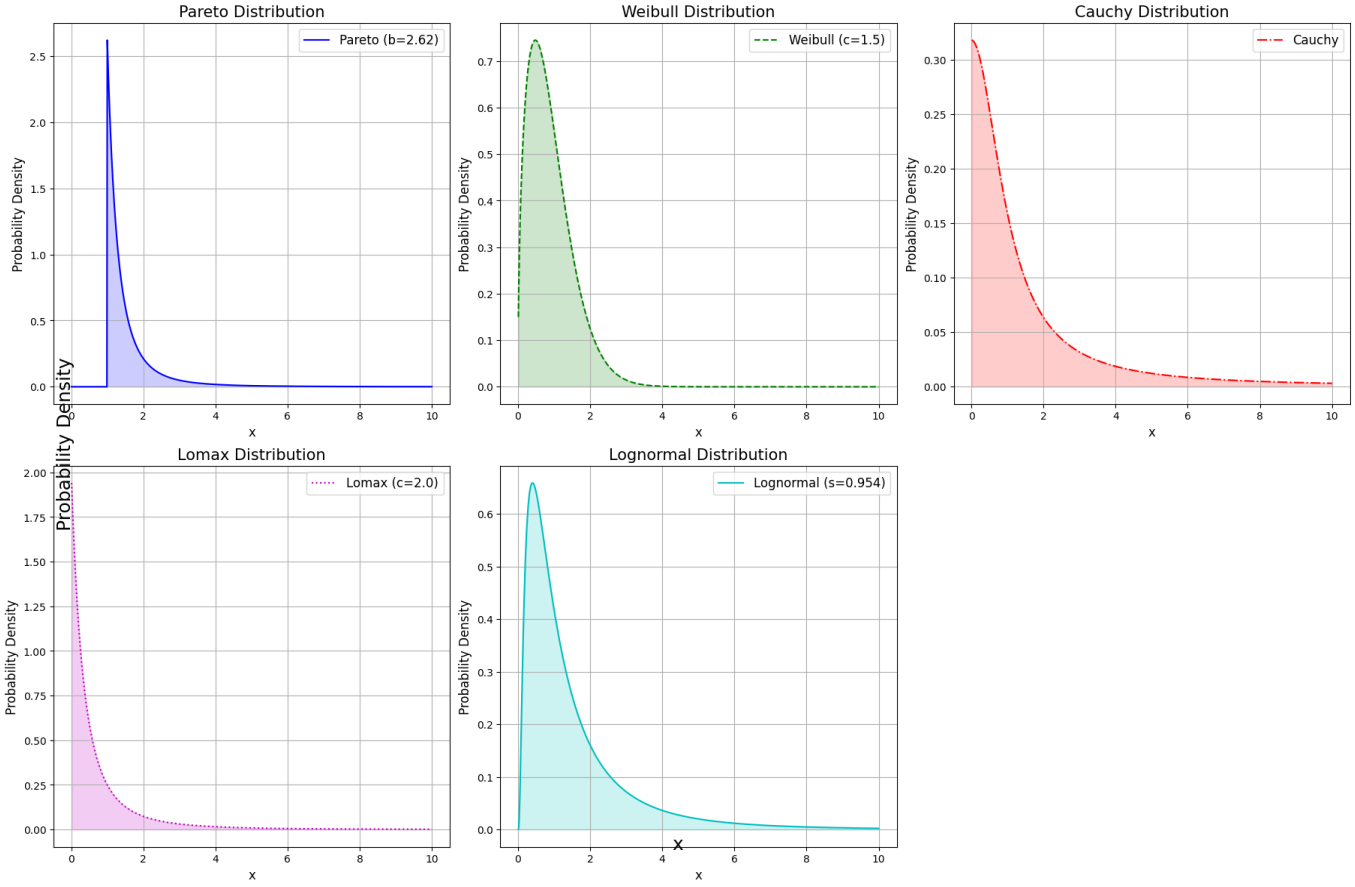


Figure 4. *Illustration of family long-tail distribution.*

As we can see in Equation (16), that final position is determined by the average of all the wolves, implying that each wolf has the same weight and role. This approach does not align well with the natural hierarchy of a wolf pack, where the alpha wolf leads. This uniform weighting can cause premature convergence in the grey wolf optimizer (GWO). To address this issue, we propose assigning a leader role to the alpha wolf, followed by other wolves with weights derived from a long-tail distribution. This distribution is chosen because it allows some regions, less frequently covered by the wolves, to be explored more effectively compared to using a normal distribution. Thus, we have:

$$\boldsymbol{X_{t+1}} = w_\alpha \boldsymbol{X_\alpha} + w_\beta \boldsymbol{X_\beta} + w_\delta \boldsymbol{X_\delta} \tag{17}$$

where $w_\alpha > w_\beta > w_\delta$, $w_i \in \boldsymbol{W}$~long-tail distribution, and we can use the Pareto distribution as the element of the tail distribution. Many researchers have attempted to enhance the original GWO and have applied it in a wide range

of areas (see, Al Afghani Edsa & Sunat, 2023; Almotairi, 2023; Altay & Varol Altay, 2023; Dhargupta et al., 2020; Hashem et al., 2023; Jain et al., 2023; K. Li et al., 2022; Y. Wang et al., 2020; Zhang et al., 2023).

3.3 Proposed method

From (Yang et al., 2022) and the discussion above, it can be understood that the data partition is created randomly without considering the variance of the resulting partition. The assessment of selected features still needs to be compared to the initial sample U. This motivates us to create data partitions with an optimized version to minimize variance by using the long-tail position GWO. We shuffle U before selecting samples and avoid reassessing U during sample selection. We consider U'^{i} and U'^{j} as different partitions and take different values for each $x \in A$, described by the discernible score.

Let (U, A, D) be the decision table, with N samples and M features as cardinality of A.

We have the following procedure:

A. Initialization

Divide U into n partitions, by considering minimum variance for each partition by using long-tail GWO, we will get $U'^{1\_opt}, U'^{2\_opt}, \dots, U'^{n\_opt}$. Compute the initial discernible score for the dataset U by using equations (2) and (3), where U'^{i_opt} represents the i-th optimized partition of the dataset.

B. Feature selection iteration

By applying equations (8) and (9), we obtain the selected features and samples. Meanwhile, the remaining features (*fea_left*) and samples (*sam_left*) are left after initialization. The iteration proceeds as follows:

1. Shuffle the dataset

   Shuffle the dataset U to ensure randomness in sample selection:

   $$U_{shuffled} = shuffled(U'^{1\_opt}, U'^{2\_opt}, \dots, U'^{n\_opt}.) \quad (18)$$

   where $U_{shuffled}$ is the result after shuffling to ensure random sample ordering.

2. Partitioning for feature assessment

   Partition the shuffled dataset into k equivalent, $partition\_fold(x)$, parts for independent assessment:

   $$parts = partition\_fold(U_{shuffled}, k) \quad (19)$$

   to get new $U'^{1\_opt}, U'^{2\_opt}, \dots, U'^{n\_opt}$.

3. Calculate incremental discernible score

   For each remaining feature $f_j \in fea\_left$, compute its incremental discernible score using the partitioned dataset:

   $$temp\_in\_ds[j] = \max\left(ds(U'^{i\_opt}[:, fea_{slt} \cup \{f_i\}], sam_{left}, U'^{j\_opt})\right) \quad (20)$$

   This step ensures no assessment of the original dataset; in fact, we assess in a new sample partition or a new data stream.

   Where:

   - f_j is a feature in the set of remaining features;
   - fea_{slt} represents the set of already selected features;
   - ds is the discernible score function form (8) and (9) applied to the partitioned dataset; and
   - $temp\_in\_ds[j]$ is the temporary incremental discernible score for the feature f_j.

4. Adaptive threshold calculation

   To select the optimal sample as **Theorem 2** stated, to make a decision optimal sample we calculate the discernible score and compare it with a threshold to determine the new feature; to calculate this, we need an adaptive threshold using a specified percentile:

   $$\text{adaptive\_threshold} = \text{percentile(selected data sample)} \quad (21)$$

5. Select the feature with the maximum incremental discernible score

Select a new feature f_{new} based on the maximum incremental discernible score that exceeds the adaptive threshold:

$$f_{new} = \underset{j}{\text{argmax}}\{temp\_in\_ds[j] | temp\_in\_ds[j] > adaptive\_threshold\} \quad (22)$$

6. Update selected features and remaining features
   Update the selected feature set:
   $$fea\_slt = fea\_slt \cup \{f_{new}\} \quad (23)$$
   Remove f_{new} from fea_left:
   $$fea\_slt = fea\_slt \setminus \{f_{new}\} \quad (24)$$
   where, fea_slt is the updated set of selected features.

7. Update remaining samples and assess feature redundancy
   1. Update the remaining samples *sam_left* based on the discernible score differences.
   2. Assess feature redundancy and adaptively remove redundant features.

For partitioning the dataset, by using the long-tail GWO we need to determine the objective function, consider:

Given dataset X with n samples and m features and k desired number of partitions, we have:

$$f(\boldsymbol{I}) = \frac{1}{k}\sum_{i=1}^{k}\sum_{d=1}^{m} Var(\boldsymbol{X}_{\boldsymbol{I}_i,d}) \quad (25)$$

where:

- $\boldsymbol{I}_i = \{\boldsymbol{I}[j] \mid j \equiv i \ (mod\ k)\}$ represents the indices assigned to the i-th partition.
- $Var(\boldsymbol{X}_{\boldsymbol{I}_i,d})$ is the variance of the d-th feature in the i-th partition.

The proposed method aims to eliminate features that may introduce noise or overlap in samples. By avoiding reliance on the original data for feature assessment, the method significantly reduces computation time. The approach incorporates random shuffling and adapts incrementally to ensure optimal feature selection. This results in faster and potentially superior feature optimization compared to existing methods.

Table 2. *Comparison between method proposed by Yang et al. (2022) and our study.*

Method	Advantages	Disadvantages
Incremental feature selection (Yang et al., 2022)	• Effective feature selection methods that avoid redundant assessments. • Effective for high-dimensional datasets and extreme class ratios in imbalanced classification	• Relies on the original dataset for assessment during incremental feature selection. • The incremental threshold for the selection process is rule-based (determined by the user).
Our study	• Effective feature selection methods that avoid redundant assessments. • Effective for high-dimensional datasets and extreme class ratios in imbalanced classification • The incremental threshold is dynamically changed based on the existing samples that were previously filtered. • In the assessment for feature selection, the incremental process does not rely on the original data; it continues processing the incoming data without referring to the original dataset, using the concept from **Theorem 3**.	• Since we use a threshold for selection based on the percentile of existing data, this may cause issues with small sample sizes, as the samples used for percentile calculations may contain anomalies. This can be improved by applying the neighborhood principle to enhance the threshold method.

Furthermore, the method is versatile and not tied to a specific machine learning model; instead, it focuses on identifying optimal features that can be applied across various models. Its flexibility allows it to adapt to different data problems, making it straightforward to implement and use.

4 RESULTS AND DISCUSSION

This experiment was conducted on a computer with an 11th Gen Intel® Core™ i7-11700 processor, running at 2.50 GHz with a base frequency of 2496 MHz, featuring 8 cores and 16 logical processors. Additionally, we used seven widely adopted datasets from the literature to assess the proposed method. These datasets represent diverse characteristics, including imbalanced class ratios, high-dimensional features with small sample sizes and vice versa. Some of these datasets are available at https://jundongl.github.io/scikit-feature/datasets.html. For the verification of the long-tail position GWO (LTP-GWO) and the GWO, we used the CEC 2019 benchmark with the Opfunu 1.0.4 Python library, while other functions and modules were implemented in Python from scratch.

4.1 Incremental feature selection

In the realm of feature selection, over the past five years, many researchers have endeavoured to develop algorithms based on swarm intelligence. However, challenges arise particularly when dealing with a large number of features, as swarm intelligence algorithms typically require recalculations to obtain optimal features, resulting in increased computation time. Thus, our proposed method aims to address these challenges effectively.

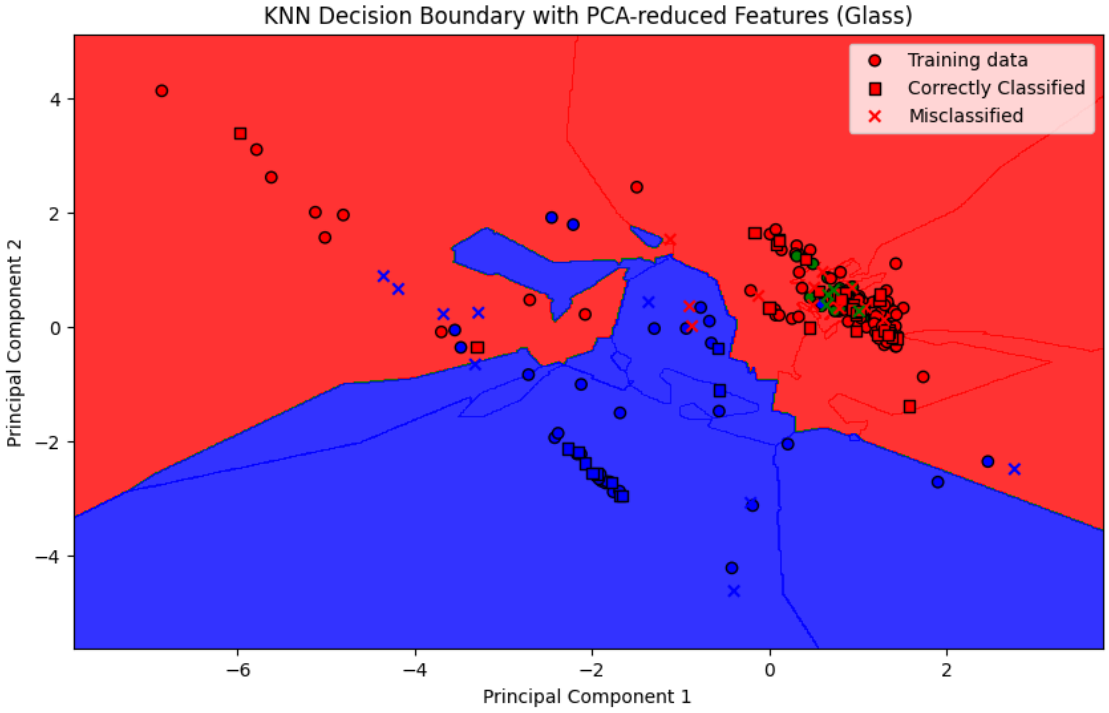


Figure 5. *Illustration of boundary related to misclassification.*

To evaluate the effectiveness of the procedure used, the following performance metrics will be utilized:

1. Accuracy: The proportion of correctly classified instances to the total instances.
$$Accuracy = \frac{(TP+TN)}{(TP+TN+FN+FP)}. \quad (26)$$
where TP = True Positive, FN = False Negative, FP = False Positive and TN = True Negative.
2. Precision: The proportion of correctly predicted positive instances to the total predicted positive instances.
$$Precision = \frac{TP}{(TP+FP)}. \quad (27)$$
3. Recall: The proportion of correctly predicted positive instances to the total actual positive instances.
$$Recall = \frac{TP}{(TP+FN)}. \quad (28)$$

4. F1 Score: The harmonic mean of precision and recall, offering a balance between the two metrics.

$$F1\ Score = 2 \times \frac{(Recall \times Precision)}{(Recall + Precision)}. \quad (29)$$

For dataset information, we utilized open data characterized by high-dimensional features, some of which are accessible at https://jundongl.github.io/scikit-feature/datasets.html. Details regarding the experimental data used in this study are provided in Table 2.

Table 3. *Summary of datasets.*

Dataset	N sample	N feature	Class ratio	Class type
Glass	214	9	0.35/0.33/0.14/0.07/0.06/0.04	Multi class
Libras movement	360	90	0.52/0.47/0.01	Multi class
QSAR biodegradation	1055	41	0.48/0.45/0.07	Multi class
PCMAC	1943	3288	0.98/0.01/0.003/0.002/0.0006/0.0003	Multi class
Pima	768	7	0.65/0.35	Binary class
Sick	2800	29	0.61/0.17/0.17/0.07	Multi class
Brain1	90	5920	0.67/0.11/0.11/0.07/0.04	Multi class

In this experiment, the focus is on model performance (we use KNN as the machine learning model because it is relatively straightforward to implement with the selected features obtained from the proposed procedure), the number of selected features and runtime. By extending the method introduced by Yang et al. (2022) and making it more adaptive based on the available sample rather than reverting to the original data, ordered shuffle sampling and optimized data partitioning can save computation time and achieve better optimal features. Additionally, the proposed method is flexible and can be used with any machine learning model, not being tied to a specific one. This approach can be particularly useful for feature selection in cases of imbalanced or high-dimensional datasets. To ensure that our results are reliable and not due to chance, we repeated the procedure 30 times to obtain statistical values.

Table 4. *Experimental results – average values (long-tail position GWO based).*

Dataset	Value	Accuracy	F1 score	Recall	Precision	N selected features
Glass	Mean	0.9288	0.6550	0.6684	0.6466	1
	Std	0.0691	0.2544	0.2410	0.2701	0
Libras movement	Mean	0.8083	0.6347	0.6547	0.6553	10.3667
	Std	0.0696	0.1302	0.1254	0.1411	1.4019
QSAR biodegradation	Mean	0.8679	0.7864	0.7790	0.8138	12.3000
	Std	0.0312	0.0753	0.0855	0.0634	2.8500
PCMAC	Mean	0.9980	0.8496	0.8499	0.8493	15.2000
	Std	0.0033	0.2529	0.2523	0.2533	14.7725
Pima	Mean	0.75	0.7236	0.7244	0.7375	1
	Std	0.0502	0.0541	0.0534	0.0547	0
Sick	Mean	0.9532	0.5431	0.5319	0.6501	12.4000
	Std	0.0144	0.0523	0.0325	0.1806	5.0173
Brain1	Mean	0.9037	0.7305	0.7511	0.7240	1
	Std	0.0983	0.2632	0.2526	0.2753	0

The role of adaptive thresholds in equations (21) and (22) is to ensure that the thresholds are data-driven rather than rule-based, allowing them to adapt to the specific conditions of the data. By evaluating the discernible scores of the

selected samples, the percentile that will serve as the threshold can be determined. Additionally, careful consideration must be given to the selection of samples for assessment. This is crucial because, based on the principles of rough set theory and equations (2) and (3), sorting the samples can help identify the optimal features more effectively.

Table 5. *Comparison of test results (best scores in bold) (Friedmann rank test, p-value = 0.00049).*

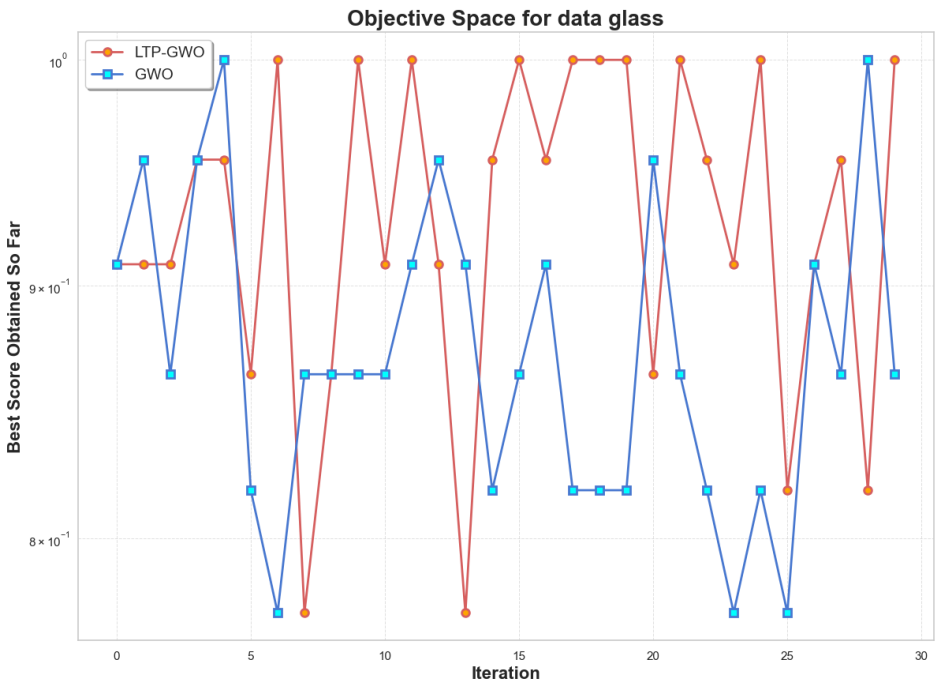
Dataset	Methods	Accuracy	N selected features	Runtime (s)	Rank
Glass	SMOTE-LOF (Asniar et al., 2022)	0.6484	*NA*	*NA*	2
	This study (LTP-GWO)	**0.7272**	**1**	**0.0156**	1
	This study (GWO)	**0.7272**	**1**	0.0330	1
Libras movement	DFFS (Yang et al., 2022)	0.6250	11	25.3700	3
	IFS-SSFA (Yang et al., 2022)	0.5722	25	5.5500	4
	This study (LTP-GWO)	0.7222	13	**3**	2
	This study (GWO)	**0.8333**	**9**	3.7781	1
QSAR biodegradation	DFFS (Yang et al., 2022)	0.7990	35	54.3800	2
	IFS-SSFA (Yang et al., 2022)	0.7603	**9**	6.7400	3
	This study (LTP-GWO)	**0.8679**	13	**4**	1
	This study (GWO)	0.7358	15	9	4
PCMAC	DFFS (Yang et al., 2022)	0.7452	39	36,070.6300	3
	IFS-SSFA (Yang et al., 2022)	0.6562	9	1,292.0400	4
	This study (LTP-GWO)	**1**	**9**	**346**	2
	This study (GWO)	**1**	10	770.2779	2
Pima	SMOTE-LOF (Asniar et al., 2022)	0.7396	*NA*	*NA*	2
	This study (LTP-GWO)	**0.7662**	**1**	**0.05**	1
	This study (GWO)	0.6883	**1**	0.0730	3
Sick	DFFS (Yang et al., 2022)	0.9182	24	58.7600	4
	IFS-SSFA (Yang et al., 2022)	0.9368	10	3.4600	3
	This study (LTP-GWO)	**0.9532**	10	5	1
	This study (GWO)	0.9464	**9**	**4.3145**	2
Brain1	Relief (Pan et al., 2023)	0.6723	50	712	4
	MGWO (Pan et al., 2023)	0.9000	18	2,574	2
	This study (LTP-GWO)	**0.9037**	**1**	**3**	1
	This study (GWO)	0.7778	**1**	7	3

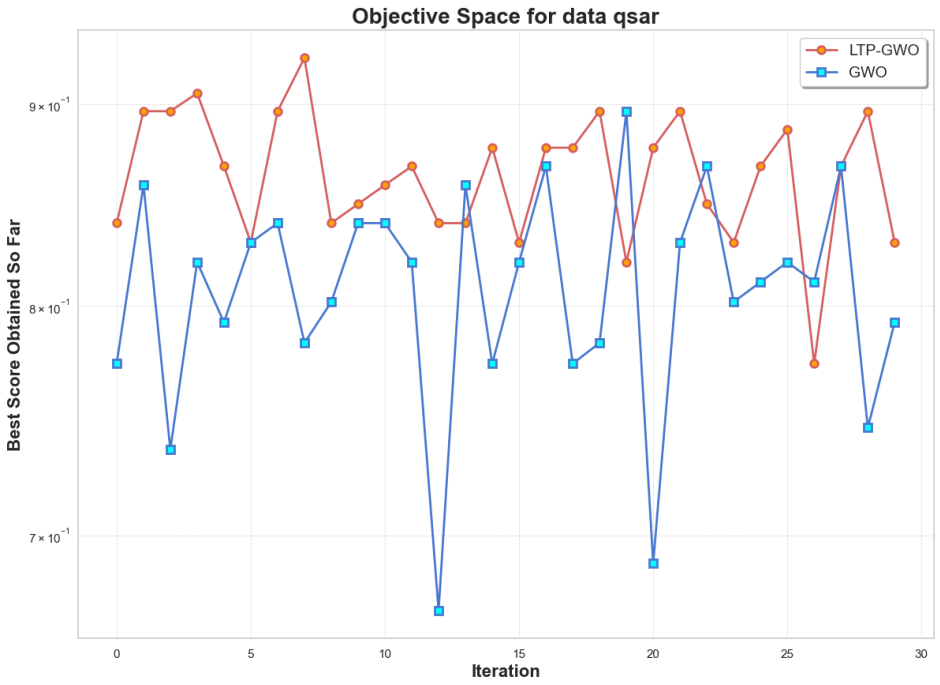
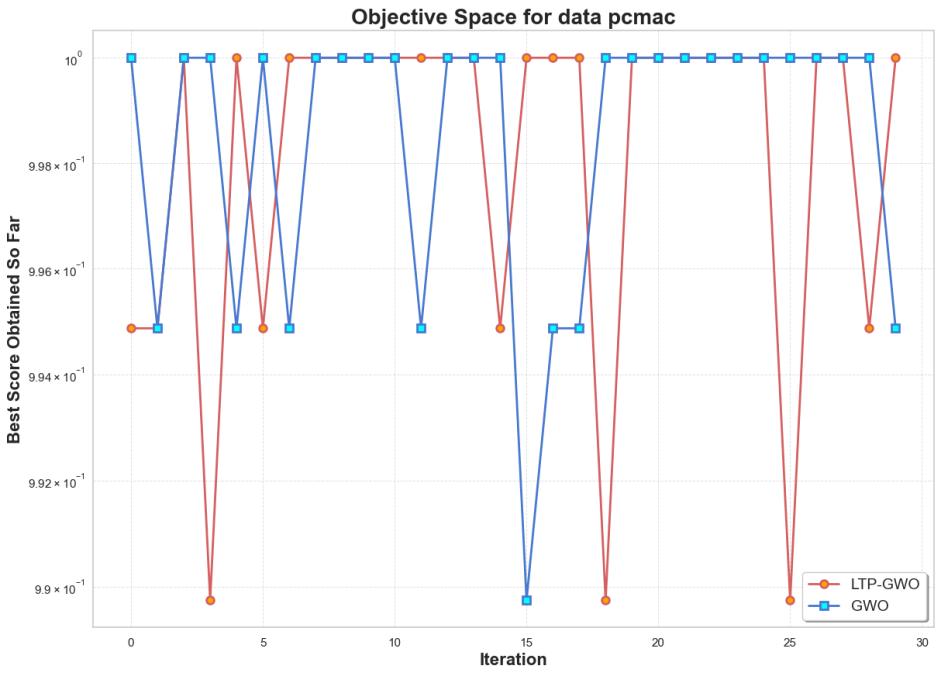
The proposed method was then applied to an experimental dataset and its results were compared with another method using the same dataset. The findings indicate that the proposed method is capable of providing a relatively shorter processing runtime and achieving better performance and feature count compared to the other method. This demonstrates that the extension of the existing method served as our motivation for improvement.

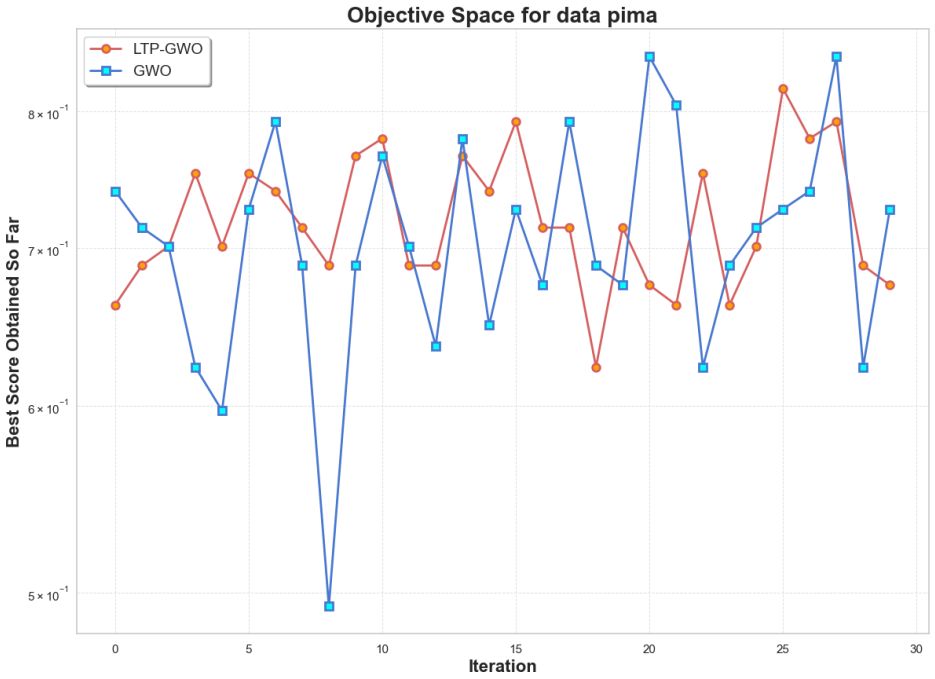
In Table 4, on the PCMAC dataset, our study demonstrates a reduction in processing time and an improvement in model performance compared to IFS-SSFA (Yang et al., 2022) and DFFS (Yang et al., 2022). The processing time decreased from 36,070.6300 and 1,292.0400 to 346 and 770.2779, respectively. Furthermore, PCMAC is a high-dimensional dataset with 3,288 features and an extreme class ratio, making it a challenging multi-class classification task. Another experiment on the Brain1 dataset, which has 5,920 features and a sample size of 90, also showed improvements in both processing time and model performance, despite its extreme class ratio and multi-class classification setup.

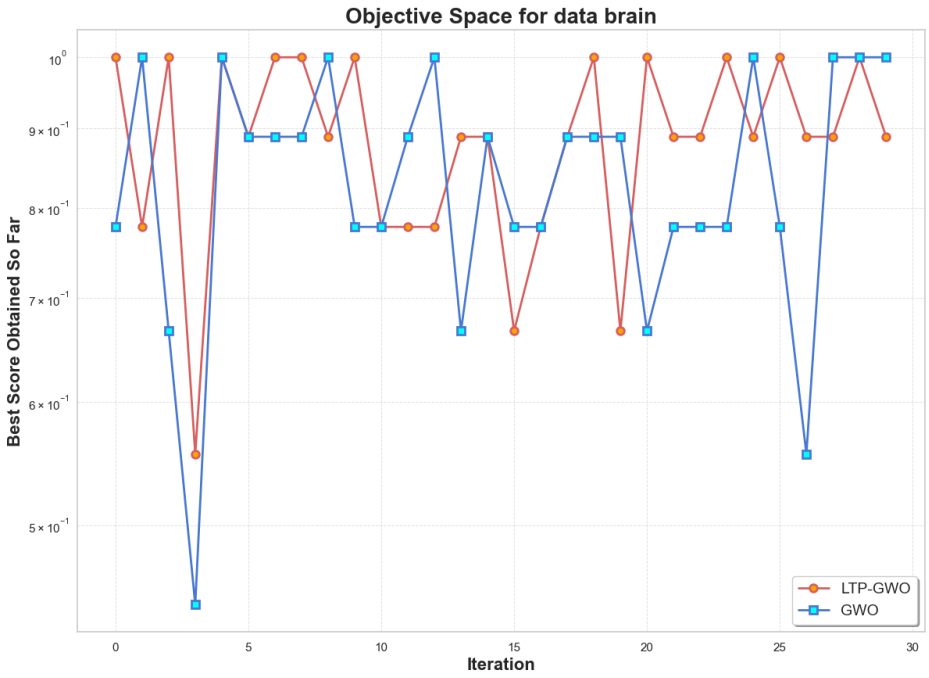
However, the proposed feature selection method has a drawback when it comes to sample selection during data partitioning. Specifically, it uses a percentile-based threshold from the partitioned samples, which is updated in subsequent iterations as the optimized partition changes. As a result, the selected samples may not entirely match the original data split used for incremental learning. The original data are divided into two sets: initial data and incremental data. As the process progresses, feature selection is no longer based on the initial data, but instead relies on the incremental data. This becomes a limitation when the sample size is small, for example, 100 samples. The incremental data might become too small, leading to selection of very few features, as seen in our experiment, where only one feature was selected, although it improved model performance. This issue can be mitigated by adjusting the threshold approach to a neighbourhood principle, ensuring that neighbouring samples are grouped together, smoothing the sample selection and optimizing the partition further.

Table 6. *Convergence curve (Accuracy) of LTP-GWO vs GWO on dataset.*

Dataset	Convergence curve
Glass	
Libras movement	

Dataset	Convergence curve
QSAR biodegredation	
PCMAC	

Dataset	Convergence curve
Pima	
Sick	

Dataset	Convergence curve
Brain1	

We also present the convergence curve of the partitioning method using the grey wolf optimizer and its improvement using a long-tail distribution. The improved version showed better accuracy in most cases across iterations.

From the discussion above, our proposed method—an extension of Yang et al. (2022) with a focus on optimized data partitioning—does not rely on the original dataset when evaluating selected features. We ran the procedure 30 times to assess its reliability, finding that it significantly improves the selection of samples and features, making it suitable for further modelling, such as with the K-nearest neighbours (KNN) model. Notably, the proposed method efficiently handles more than 1,000 features in a relatively short time, which is a distinct advantage for high-dimensional data analysis, where traditional methods often struggle. For instance, widely used techniques such as PCA can become inadequate when dealing with such a large number of features.

To statistically validate our results, we applied the Friedman rank test, a non-parametric test often used to compare multiple algorithms across different datasets. Instead of ranking the average performance, in this study, we specifically ranked the best scores obtained in each iteration for each method. This approach helps to highlight the peak performance of the method, which is especially relevant for optimization tasks. The Friedman rank test yielded a p-value of 0.00049, indicating a statistically significant improvement of our proposed method compared to the other methods evaluated.

The Friedman test was selected because our method makes use of an optimized data partitioning approach, based on an improved version of the grey wolf optimizer (GWO). Since the GWO is a swarm-based algorithm, we ran the procedure 30 times to assess reliability, focusing on the best scores, as detailed in Tables 3 and 4. Additionally, we compared the performance of our method against the original GWO.

Our study also differs from existing literature, where most approaches directly employ bio-inspired algorithms for feature selection, as seen in J. Li et al. (2020) and Pham & Raahemi (2023). Aware of the inefficiency of recalculating the same samples in such approaches, we combined rough set theory with optimized data partitioning, supported by bio-inspired methods—particularly the GWO.

Based on our study, the proposed method can be applied across a range of domains, including healthcare, social network analysis, text mining for aspect-based sentiment analysis and other use cases involving high-dimensional data. Its ability to handle large feature sets efficiently makes it versatile and suitable for any domain requiring robust feature selection under high-dimensional conditions.

Despite its strengths, our method does have some limitations. The selection process for optimal samples is dynamic and based on a percentile approach, which, while effective, could be further improved. A potential enhancement would involve employing a neighbourhood-based approach, similar to that of Y. Li et al. (2023), which considers surrounding samples to refine the selection. Moreover, our method could be extended to handle pixel data for image processing tasks, which is another area for future research.

4.2 Verification of Long-tail Position GWO

As previously mentioned, the long-tail position grey wolf optimizer (GWO) is more effective at covering areas that the original GWO may not reach. To evaluate our optimizer, we will compare the proposed Long-Tail Position GWO with the original GWO using benchmark functions with a dimension of 100, including the CEC 2019 test suite. The number of evaluations is set to 10,000, with the same number of agents (50) for both optimizers.

Table 7. *Conventional benchmark function (Dimension = 100).*

Function	Formula	Category	Search space	Optimal value
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	Unimodal	[-100,100]	0
F2	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	Unimodal	[-10,10]	0
F3	$f(x) = \sum_{i=1}^{n} \left(\left(\sum_{j=1}^{i} x_j \right)^2 \right)$	Unimodal	[-100,100]	0
F4	$f(x) = max\{\lvert x_i \rvert, i \le i \le n\}$	Unimodal	[-100,100]	0
F5	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + ((x_i - 1)^2)]$	Unimodal	[-30,30]	0
F6	$f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	Unimodal	[-100,100]	0
F7	$f(x) = \sum_{i=1}^{n} i x_i^4 + random[0,1)$	Unimodal	[-1.28,1.28]	0
F8	$f(x) = \sum_{i=1}^{n} -x_i sin(\sqrt{\lvert x_i \rvert})$	Multimodal	[-500,500]	-418.9 *Dimension
F9	$f(x) = \sum_{i=1}^{n} [x_i^2 - 10\, cos(2\pi x_i) + 10]$	Multimodal	[-5.12,5.12]	0
F10	$f(x) = -20exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2} \right) - exp(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)) + 20 + e$	Multimodal	[-32,32]	8.88E-16
F11	$f(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	Multimodal	[-600,600]	0
F12	$f(x) = \frac{\pi}{n} \left\{ 10 sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	Multimodal	[-50,50]	0

F13	$f(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	Multimodal	[-50,50]	0

Table 8. *Classical benchmark function comparison.*

Function	Value	LTP - GWO	GWO
F1	Avg	0	0
	Std	0	0
F2	Avg	0	0
	Std	0	0
F3	Avg	0	0
	Std	0	0
F4	Avg	0	0
	Std	0	0
F5	Avg	98.2534	98.1318
	Std	0	0.07541
F6	Avg	0	0
	Std	0	0
F7	Avg	1.6630e-06	2.3369e-06
	Std	0	2.1520e-06
	Rank		
F8	Avg	-41898.2887	-41803.0821
	Std	0	343.2972
F9	Avg	0	0
	Std	0	0
F10	Avg	4.4408e-16	4.4408e-16
	Std	0	0
F11	Avg	0	0
	Std	0	0
F12	Avg	0.0966	0.1362
	Std	0	0.0648
F13	Avg	2.4908	2.7199
	Std	0	0.8318

Table 9. *CEC 2019 benchmark function.*

No	Function	Dimension	Search Space	Best
F1	Storn's Chebyshev Polynomial Fitting Problem	9	[-8192,8192]	1
F2	Inverse Hilbert Matrix Problem	16	[-16,384,16,384]	1
F3	Lennard-Jones Minimum Energy Cluster	18	[-4,4]	1
F4	Rastrigin's Function	10	[-100,100]	1
F5	Griewangk's Function	10	[-100,100]	1
F6	Weierstrass Function	10	[-100,100]	1

No	Function	Dimension	Search Space	Best
F7	Modified Schewefel's Function	10	[-100,100]	1
F8	Expanded Schaffer's F6 Function	10	[-100,100]	1
F9	Happy Cat Function	10	[-100,100]	1
F10	Ackley Function	10	[-100,100]	1

The rationale for using 100 dimensions in the conventional benchmark functions and CEC 2019 is to evaluate the effectiveness of the long-tail position GWO in solving complex problems, particularly in multimodal scenarios.

Table 10. *LTP-GWO vs GWO (on CEC2019).*

Function	Value	LTP-GWO	GWO
F1	Avg	1	1
	Std	1.3701-09	2.0613-09
	Rank	1	1
F2	Avg	4.9941	4.9991
	Std	0.0196	0.0050
	Rank	1	2
F3	Avg	5.1374	5.4914
	Std	1.1935	1.0061
	Rank	1	2
F4	Avg	6346.8212	5875.1287
	Std	3041.9860	4082.4830
	Rank	2	1
F5	Avg	2.3173	2.5575
	Std	0.4314	0.555
	Rank	1	2
F6	Avg	9.5171	9.9869
	Std	0.8832	0.6880
	Rank	1	2
F7	Avg	699.0675	785.3209
	Std	372.4067	379.0702
	Rank	1	2
F8	Avg	1.1649	1.2229
	Std	0.0943	0.0953
	Rank	1	2
F9	Avg	107.7746	114.8789
	Std	68.0704	54.8689
	Rank	1	2
F10	Avg	21.2793	21.32
	Std	0.0976	0.1066
	Rank	1	2

Based on the above table, the Long-Tail Position GWO enhances the search capability of the original GWO, enabling it to find optimal or near-optimal solutions more effectively across various benchmark functions. By integrating both the Long-Tail Position GWO and original GWO with real-world scenarios, as demonstrated in our proposed method for optimizing data partitioning and feature selection in an incremental setting using Rough Set Theory, we achieved

promising results, confirmed through experimentation in this study. We can observe that the use of the optimizer significantly aids in determining the best partition for incremental feature selection.

5 CONCLUSION

In this paper, we introduce an enhanced version of incremental feature selection supported by rough set theory and optimized partitioning, utilizing the long-tail position grey wolf optimizer (LTP-GWO). The primary objective was to reduce computation time, minimize rule-based processes in incremental feature evaluation and improve the performance of machine learning models while ensuring that feature evaluation remains independent of the original data. Our approach achieves this by increasing adaptability during incremental updates through percentile thresholds derived from existing data. Additionally, we use shuffled ordering to accelerate the detection and computation of discernibility scores, eliminating the need to reference the original dataset.

The optimized partitioning method reduces variance within each partition, which is exclusively used for the initial determination of optimal features. Experimental results, comparing our proposed method with state-of-the-art feature selection techniques across various scenarios, including imbalanced class distributions, demonstrate its superiority in most cases. Consequently, our method proves effective in both practical applications and academic settings.

Furthermore, the Grey Wolf Optimizer becomes adaptive by incorporating elements from long-tail distributions. For instance, if no improvement is observed, the method can switch to another long-tail distribution. This adaptability, combined with our feature selection approach, can also serve as a preprocessing step for clustering tasks.

For future work, the dynamic sample partitioning could be refined by adopting a neighbourhood-based approach. In this study, we employed a percentile-based approach, which may be less effective if anomalies exist in the original partitions. Using a neighbourhood-based approach could better handle such anomalies, leading to more optimal sample partitioning and improved feature selection.

ADDITIONAL INFORMATION AND DECLARATIONS

Acknowledgments: The authors would like to thank the College of Computing and Khon Kaen University through the KKU Scholarship for ASEAN and GMS Countries' Personal Academic Year 2023 program.

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: S.A.A.E.: Conceptualization, Methodology, Writing – Original draft preparation, Writing – Reviewing and Editing. K.S.: Conceptualization, Methodology, Data curation, Supervision.

Statement on the Use of Artificial Intelligence Tools: The authors declare that they didn't use artificial intelligence tools for text or other media generation in this article.

Data Availability: The data that support the findings of this study are available from the corresponding author.

REFERENCES

Abdel-Basset, M., El-Shahat, D., El-henawy, I., de Albuquerque, V. H. C., & Mirjalili, S. (2020). A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection. *Expert Systems with Applications*, 139, 112824. https://doi.org/10.1016/j.eswa.2019.112824

Al Afghani Edsa, S., & Sunat, K. (2023). Hybridization of Modified Grey Wolf Optimizer and Dragonfly for Feature Selection. In *Data Science and Artificial Intelligence, DSAI 2023*, (pp. 35–42). Springer. https://doi.org/10.1007/978-981-99-7969-1_3

Almotairi, K. H. (2023). MiRNA subset selection for microarray data classification using grey wolf optimizer and evolutionary population dynamics. *Neural Computing and Applications*, 35(25), 18737–18761. https://doi.org/10.1007/s00521-023-08701-y

Altay, O., & Varol Altay, E. (2023). A novel hybrid multilayer perceptron neural network with improved grey wolf optimizer. *Neural Computing and Applications*, 35(1), 529–556. https://doi.org/10.1007/s00521-022-07775-4

Arora, V., & Agarwal, P. (2024). An Empirical Study of Nature-Inspired Algorithms for Feature Selection in Medical Applications. *Annals of Data Science*, (in press). https://doi.org/10.1007/s40745-024-00571-y

Asniar, Maulidevi, N. U., & Surendro, K. (2022). SMOTE-LOF for noise identification in imbalanced data classification. *Journal of King Saud University - Computer and Information Sciences*, 34(6), 3413–3423. https://doi.org/10.1016/j.jksuci.2021.01.014

Dash, M., & Liu, H. (2000). Feature Selection for Clustering. In T. Terano, H. Liu, & A. L. P. Chen (Eds.), *Knowledge Discovery and Data Mining. Current Issues and New Applications* (pp. 110–121). Springer. https://doi.org/10.1007/3-540-45571-X_13

Dehghan, Z., & Mansoori, E. G. (2018). A new feature subset selection using bottom-up clustering. *Pattern Analysis and Applications*, 21(1), 57–66. https://doi.org/10.1007/s10044-016-0565-8

Dhargupta, S., Ghosh, M., Mirjalili, S., & Sarkar, R. (2020). Selective Opposition based Grey Wolf Optimization. *Expert Systems with Applications*, 151, 113389. https://doi.org/10.1016/j.eswa.2020.113389

Gilal, A. R., Abro, A., Hassan, G., Jaafar, J., & Rehman, F. (2019). A Rough-Fuzzy Model for Early Breast Cancer Detection. *Journal of Medical Imaging and Health Informatics*, 9(4), 688–696. https://doi.org/10.1166/jmihi.2019.2664

Gu, S., Cheng, R., & Jin, Y. (2018). Feature selection for high-dimensional classification using a competitive swarm optimizer. *Soft Computing*, 22(3), 811–822. https://doi.org/10.1007/s00500-016-2385-6

Hancer, E., Xue, B., & Zhang, M. (2020). A survey on feature selection approaches for clustering. *Artificial Intelligence Review*, 53(6), 4519–4545. https://doi.org/10.1007/s10462-019-09800-w

Hashem, M. H., Abdullah, H. S., & Ghathwan, K. I. (2023). Grey Wolf Optimization Algorithm: A Survey. *Iraqi Journal of Science*, 64(11), 5964–5984. https://doi.org/10.24996/ijs.2023.64.11.40

Jain, A., Nagar, S., Singh, P. K., & Dhar, J. (2023). A hybrid learning-based genetic and grey-wolf optimizer for global optimization. *Soft Computing*, 27(8), 4713–4759. https://doi.org/10.1007/s00500-022-07604-9

Jia, H., Li, J., Song, W., Peng, X., Lang, C., & Li, Y. (2019). Spotted Hyena Optimization Algorithm with Simulated Annealing for Feature Selection. *IEEE Access*, 7, 71943–71962. https://doi.org/10.1109/ACCESS.2019.2919991

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., & Togneri, R. (2017). Cost-Sensitive learning of deep feature representations from imbalanced data. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8), 3573–3587. https://doi.org/10.1109/tnnls.2017.2732482

Kwakye, B. D., Li, Y., Mohamed, H. H., Baidoo, E., & Asenso, T. Q. (2024). Particle guided metaheuristic algorithm for global optimization and feature selection problems. *Expert Systems with Applications*, 248, 123362. https://doi.org/10.1016/j.eswa.2024.123362

Li, F., Zhang, Z., & Jin, C. (2016). Feature selection with partition differentiation entropy for large-scale data sets. *Information Sciences*, 329, 690–700. https://doi.org/10.1016/j.ins.2015.10.002

Li, J., Lei, H., Alavi, A. H., & Wang, G. G. (2020). Elephant herding optimization: Variants, hybrids, and applications. *Mathematics*, 8(9), 1415. https://doi.org/10.3390/MATH8091415

Li, K., Li, S., Huang, Z., Zhang, M., & Xu, Z. (2022). Grey Wolf Optimization algorithm based on Cauchy-Gaussian mutation and improved search strategy. *Scientific Reports*, 12(1), 18961. https://doi.org/10.1038/s41598-022-23713-9

Li, Y., Wu, X., & Wang, X. (2023). Incremental reduction methods based on granular ball neighborhood rough sets and attribute grouping. *International Journal of Approximate Reasoning*, 160, 108974. https://doi.org/10.1016/j.ijar.2023.108974

Ma, T., Lu, S., & Jiang, C. (2024). A membership-based resampling and cleaning algorithm for multi-class imbalanced overlapping data. *Expert Systems with Applications*, 240, 122565. https://doi.org/10.1016/j.eswa.2023.122565

Meng, Z., & Shi, Z. (2016). On quick attribute reduction in decision-theoretic rough set models. *Information Sciences*, 330, 226–244. https://doi.org/10.1016/j.ins.2015.09.057

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Pan, H., Chen, S., & Xiong, H. (2023). A high-dimensional feature selection method based on modified Gray Wolf Optimization. *Applied Soft Computing*, 135. https://doi.org/10.1016/j.asoc.2023.110031

Pham, T. H., & Raahemi, B. (2023). Bio-Inspired Feature Selection Algorithms With Their Applications: A Systematic Literature Review. *IEEE Access*, 11, 43733–43758. https://doi.org/10.1109/ACCESS.2023.3272556

Pichai, S., Sunat, K., & Chiewchanwattana, S. (2020). An asymmetric chaotic competitive swarm optimization algorithm for feature selection in high-dimensional data. *Symmetry*, 12(11), 1–13. https://doi.org/10.3390/sym12111782

Premalatha, M., Jayasudha, M., Čep, R., Priyadarshini, J., Kalita, K., & Chatterjee, P. (2024). A comparative evaluation of nature-inspired algorithms for feature selection problems. *Heliyon*, 10(1), e23571. https://doi.org/10.1016/j.heliyon.2023.e23571

Raza, M. S., & Qamar, U. (2018). Feature selection using rough set-based direct dependency calculation by avoiding the positive region. *International Journal of Approximate Reasoning*, 92, 175–197. https://doi.org/10.1016/j.ijar.2017.10.012

Roth, V., & Lange, T. (2003). Feature Selection in Clustering Problems. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems 2003,* (pp. 473–480). NeurIPS. https://proceedings.neurips.cc/paper_files/paper/2003/file/bb03e43ffe34eeb242a2ee4a4f125e56-Paper.pdf

Sharma, M., & Kaur, P. (2021). A Comprehensive Analysis of Nature-Inspired Meta-Heuristic Techniques for Feature Selection Problem. *Archives of Computational Methods in Engineering*, 28(3), 1103–1127. https://doi.org/10.1007/s11831-020-09412-6

Shikoun, N. H., Al-Eraqi, A. S., & Fathi, I. S. (2024). BINCOA: An efficient binary crayfish optimization algorithm for feature selection. *IEEE Access*, 12, 28621–28635. https://doi.org/10.1109/access.2024.3366495

Tran, B., Xue, B., & Zhang, M. (2019). Variable-Length Particle Swarm Optimization for Feature Selection on High-Dimensional Classification. *IEEE Transactions on Evolutionary Computation*, 23(3), 473–487. https://doi.org/10.1109/TEVC.2018.2869405

Wang, C., Huang, Y., Ding, W., & Cao, Z. (2021). Attribute reduction with fuzzy rough self-information measures. *Information Sciences*, 549, 68–86. https://doi.org/10.1016/j.ins.2020.11.021

Wang, Y., Wang, T., Dong, S., & Yao, C. (2020). An Improved Grey-Wolf Optimization Algorithm Based on Circle Map. *Journal of Physics Conference Series*, 1682(1), 01202. https://doi.org/10.1088/1742-6596/1682/1/012020

Xu, H., Cao, Q., Fu, H., & Chen, H. (2019). Applying an Improved Elephant Herding Optimization Algorithm with Spark-based Parallelization to Feature Selection for Intrusion Detection. *International Journal of Performability Engineering*, 15(6), 1600–1610. https://doi.org/10.23940/ijpe.19.06.p11.16001610

Yang, Y., Chen, D., Wang, H., & Wang, X. (2018). Incremental Perspective for Feature Selection Based on Fuzzy Rough Sets. *IEEE Transactions on Fuzzy Systems*, 26(3), 1257–1273. https://doi.org/10.1109/TFUZZ.2017.2718492

Yang, Y., Chen, D., Zhang, X., Ji, Z., & Zhang, Y. (2022). Incremental feature selection by sample selection and feature-based accelerator. *Applied Soft Computing*, 121, 108800. https://doi.org/10.1016/j.asoc.2022.108800

Yang, Y., Song, S., Chen, D., & Zhang, X. (2020). Discernible neighborhood counting based incremental feature selection for heterogeneous data. *International Journal of Machine Learning and Cybernetics*, 11(5), 1115–1127. https://doi.org/10.1007/s13042-019-00997-4

Zhang, H., Chen, J., Zhang, Q., Chen, Z., Ding, X., & Yao, J. (2023). Grey Wolf Optimization Algorithm Based on Follow-Controlled Learning Strategy. *IEEE Access*, 11, 101852–101872. https://doi.org/10.1109/ACCESS.2023.3314514

Zhang, H., Chen, Q., Xue, B., Banzhaf, W., & Zhang, M. (2024). A geometric semantic macro-crossover operator for evolutionary feature construction in regression. *Genetic Programming and Evolvable Machines*, 25(1), Article no. 2. https://doi.org/10.1007/s10710-023-09465-z

Zhao, P., Zhang, Y., Ma, Y., Zhao, X., & Fan, X. (2023). Discriminatively embedded fuzzy K-Means clustering with feature selection strategy. *Applied Intelligence*, 53(16), 18959–18970. https://doi.org/10.1007/s10489-022-04376-5

Acta Informatica Pragensia is published by the Prague University of Economics and Business, Czech Republic | eISSN: 1805-4951