# A Distributed Feature Selection Approach over Hadoop for Accurate Classification based on Grasshopper Algorithm and Rough Sets

Ahmed Hamed

*Department of Computer Science, Faculty of Computers and Information, Damanhour University, Damanhour, 22511, Egypt*

**Abstract**

In the specialized field of data analysis, precise feature selection has become paramount, especially given the extensive and intricate datasets available. Many of these datasets house a plethora of features, of which a substantial number may be redundant, leading to potential inaccuracies and increased computational demands. Although the Rough Set (RS) and Multigranular Rough Set (MGRS) models have demonstrated efficacy in feature selection, their computational complexities can be limiting. To address this, we introduce an innovative solution, integrating the MGRS with the Grasshopper Optimization Algorithm (GOA)—a metaheuristic technique derived from grasshopper foraging behaviors. To manage large-scale data, we employ the Hadoop framework for streamlined distributed processing. By distributing the enhanced GOA tasks within Hadoop, we aspire to efficiently process large-scale datasets. The proposed algorithm's efficacy is assessed using dedicated datasets, benchmarked via classifiers such as Random Forest and K-Nearest Neighbor. Preliminary results highlight the superior performance of our approach compared to prevalent metaheuristic strategies, with the MGRS model enhancing performance notably when employed as an objective function.

*Keywords:* Multigranular Rough Set (MGRS), Grasshopper Optimization Algorithm (GOA), Hadoop framework, Large-scale lung cancer datasets, Feature selection, Lung cancer research

## 1. Introduction

In contemporary data-driven research, the imperative of discerning the most pertinent features within a dataset is underscored, given its pivotal role in a plethora of disciplines such as machine learning, data mining, bioinformatics, signal processing, image analytics, and computer vision [1, 2, 3, 4].

As the dimensionality of data escalates in such disciplines, presenting scenarios where features surpass sample counts, the efficacy of feature selection becomes paramount. While rich information can be harnessed from high-dimensional data, it ushers in computational challenges. Enhanced dimensionality amplifies computational intricacies, elongating analysis timelines, and amplifying resource demands. Additionally, it introduces the potential peril of the "curse of dimensionality," where excessive dimensions might inadvertently compromise prediction model accuracies due to overfitting [5, 6].

Intricacies in the feature selection process arise from the potential interplay of features. For instance, features deemed inconsequential in isolation may bolster classification precision when synergized with complementary counterparts. Inversely, certain standalone robust features could exhibit redundancy in combination with others [7, 8].

Historically, feature selection strategies have been dichotomized as either filter or wrapper methodologies [9]. Wrapper strategies embed a learning algorithm within the feature assessment paradigm, whereas filter strategies decouple the two, evaluating features' classification prowess autonomously. Despite their computational demands, wrapper strategies often outperform, attributing to their consideration of algorithmic interactions. In contrast, filter methods, while resource-efficient, may falter due to their isolation from the learning algorithm [9].

The Rough Set (RS) theory, revered for its prowess in handling data uncertainties, has been instrumental in feature selection, data mining, and pattern recognition [10]. Notwithstanding its merits, conventional RS grapples with computational overheads and real-data challenges.

To ameliorate such shortcomings, advanced models like the multigranular rough set (MGRS) have been proposed [11]. MGRS, epitomizing a refinement over its RS predecessor, offers augmented flexibility and computational efficiency, rendering it apt for today's expansive datasets.

Swarm intelligence approaches, especially metaheuristics like the grasshopper optimization algorithm (GOA), have emerged as potent enhancers for feature selection [12, 13]. Deriving cues from nature, these algorithms have exhibited prowess in deciphering intricate optimization conundrums. GOA, in particular, stands out for its adeptness in leveraging the Hadoop ecosystem, countering conventional MGRS constraints and fortifying computational alacrity [14].

## 2. Related Work

This manuscript offers an extensive critique of data analysis approaches, illuminating methodologies from multiple dimensions, encompassing dataset intricacies, employed techniques, and evaluative metrics. Various modalities have been employed to decipher complex datasets, including Decision Trees (DTs),

Clustering, Probabilistic paradigms, Region-Based tactics, and both linear and non-linear methods [15]. Additionally, machine learning paradigms have also been employed for such analyses [16]. The current discourse also uncovers investigations leveraging Deep Learning models for classificatory and predictive endeavors.

[17], in their seminal work, proposed a detection modality anchored in a Computer-Aided Diagnostic System (CADS). Their method ensured meticulous examination of dataset patterns along multiple axes, segmenting the detection process into three distinct phases, each emphasizing feature map extraction from diverse orientations. Aligning with this, [15] ventured into data classification by employing image segmentation techniques using thresholding tactics and harnessing deep learning for feature extraction. Their employment of the Multi-Layer Perceptron (MLP) classifier yielded a commendable classification success rate of 98.31%.

Further, [18] steered classification tasks relying on image processing techniques, inclusive of morphological and filtering methods. In a complementary vein, [19] demonstrated a classificatory schema utilizing a deep Convolutional Neural Network (CNN) for image datasets, realizing a 71% success quotient. Contrarily, [20] anchored on deep learning frameworks to classify complex data patterns, with the CNN architecture achieving an 84.15% accuracy rate.

In their investigation, [21] harnessed a multi-view convolutional neural network (MV-CNN) to classify patterns in datasets using advanced image processing techniques. Concurrently, [22] adopted a dual-pronged detection strategy for classification tasks, achieving an accuracy of 90%. An innovative algorithm, rooted in CNN, was employed to stage classification categories using fluorodeoxyglucose positron emission tomography (FDG-PET)/CT imagery, achieving a 68% accuracy rate [23].

The domain of data classification has witnessed a myriad of investigative angles. One such research trajectory evaluated the efficacy of low-dose imaging techniques for complex data pattern detection [24]. Another discerning study embarked on a computational histomorphometric classificatory scheme to predict patterns of recurrence in early-stage datasets, achieving an 81% accuracy rate [25]. Furthermore, semi-automated systems were harnessed to analyze volume and dimension attributes in image-based datasets [26], and another inquiry delved into the ramifications of integrating structural dimensions into backpropagation Artificial Neural Networks (ANN) [27].

## 3. Preliminaries

### 3.1. Rough Set Theory

Rough set theory, proposed by Zdzislaw Pawlak in the 1980s, is a mathematical approach that deals with the vagueness and uncertainty inherent in many types of information systems. It has been widely applied in various fields, including data analysis, decision-making, and machine learning.

An information system $S$ can be represented as $S = (U, A)$, where $U$ is a non-empty finite set of objects, and $A$ is a non-empty finite set of attributes. For any subset $B \subseteq A$, an indiscernibility relation $IND(B)$ is defined as follows:

$$IND(B) = \{(x, y) \in U \times U | \forall a \in B, a(x) = a(y)\} \quad (1)$$

This means that for any two objects $x$ and $y$ in $U$, if they share the same values for all attributes in $B$, then they are indiscernible in terms of $B$.

In rough set theory, any subset $X$ of $U$ can be approximated using lower and upper approximations, based on the indiscernibility relation.

The lower approximation of $X$ with respect to $B$, denoted $B_*(X)$, is the set of all objects in $U$ that can be certainly classified as belonging to $X$ based on the attributes in $B$:

$$B_*(X) = \{x \in U | [x]_B \subseteq X\} \quad (2)$$

The upper approximation of $X$ with respect to $B$, denoted $B^*(X)$, is the set of all objects in $U$ that can possibly belong to $X$ based on the attributes in $B$:

$$B^*(X) = \{x \in U | [x]_B \cap X \neq \emptyset\} \quad (3)$$

The boundary region of $X$ with respect to $B$, denoted $BN_B(X)$, is the set of all objects that cannot be certainly classified as either belonging or not belonging to $X$ based on $B$:

$$BN_B(X) = B^*(X) \setminus B_*(X) \quad (4)$$

Rough set theory provides a formal framework for dealing with the inherent uncertainty and vagueness in information systems. It is based on the principle of indiscernibility: objects that cannot be distinguished from each other based on the available attributes are considered equivalent.

### 3.2. Multi-Granular Rough Sets (MGRS)

The Multi-Granular Rough Set (MGRS) model is an extension of the classical RST, which allows different levels of granulation. This concept is valuable in many contexts where information can be granulated at different levels, and different granular levels may provide varying degrees of knowledge or insights.

Consider an information system $IS = (U, Q)$ where $U$ is a non-empty finite set of objects and $Q$ is a non-empty finite set of attributes. For $x, y \in U$ and $P \subseteq Q$, if for any $a \in P$, $a(x) = a(y)$, then we denote this by $x[y]_P$.

A multi-granulation rough set model is defined on the basis of several binary relations. Suppose $U$ is a universe and $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$ is a set of equivalence relations on $U$. Each equivalence relation $R_i$ induces a partition of $U$ into some granules. We define the following two types of approximations based on $\mathcal{R}$:

1. **Consistent Multi-granulation Rough Set Model**: The lower and upper approximation of a set $X$ in the consistent MGRS model are respectively defined as:

$$\underline{R}(X) = \bigcap_{i=1}^{m} R_i[X], \quad (5)$$

$$\overline{R}(X) = \bigcup_{i=1}^{m} R_i[X]. \quad (6)$$

2. **Tolerant Multi-granulation Rough Set Model**: The lower and upper approximation of a set $X$ in the tolerant MGRS model are respectively defined as:

$$\underline{R_T}(X) = \bigcup_{i=1}^{m} R_i[X], \quad (7)$$

$$\overline{R_T}(X) = \bigcap_{i=1}^{m} R_i[X]. \quad (8)$$

In both models, the boundary region of a set $X$ is defined as $BN_R(X) = \overline{R}(X) - \underline{R}(X)$.

MGRS provides a more flexible and adaptable approach in situations where knowledge can be understood and modeled at different granular levels. It has found applications in many fields including data mining, machine learning, and decision-making processes.

### *3.3. Grasshopper optimization algorithm*

The Grasshopper Optimization Algorithm (GOA) models the inherent collective behaviour of grasshoppers within their ecological milieu. This algorithm's mathematical constructs are articulated through the subsequent formulas and equations [40, 41]. In the GOA framework, the spatial location of each grasshopper within the swarm signifies a feasible solution to a prescribed optimization challenge. The locus of the $i$-th grasshopper is symbolized as $X_i$ and is formulated as shown in equation 3.1.

$$X_i = S_i + G_i + A_i \quad (9)$$

In this equation, $S_i$ signifies the social interaction, $G_i$ represents the gravitational pull on the i-th grasshopper, and $A_i$ characterizes the wind advection.

The mathematical model embodies three primary constituents: social interaction, gravitational pull, and wind advection, mirroring the movements of grasshoppers in their environment. The dominant component emanating from the grasshoppers themselves is social interaction, depicted in equation 3.2.

$$S_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s(d_{ij})\widehat{d_{ij}}. \quad (10)$$

Here, $d_{ij}$ denotes the distance between the i-th and j-th grasshopper, defined as $d_{ij} = |x_j - x_i|$. The strength of social forces is characterized by the function $s$ as shown in equation 3.3. The vector $\widehat{d_{ij}} = \frac{x_j - x_i}{d_{ij}}$ is a unit vector pointing from the i-th grasshopper towards the j-th grasshopper.

The function $s$, which outlines the social forces, is formulated as follows:

$$s(r) = fe^{\frac{-r}{l}} - e^{-r}. \quad (11)$$

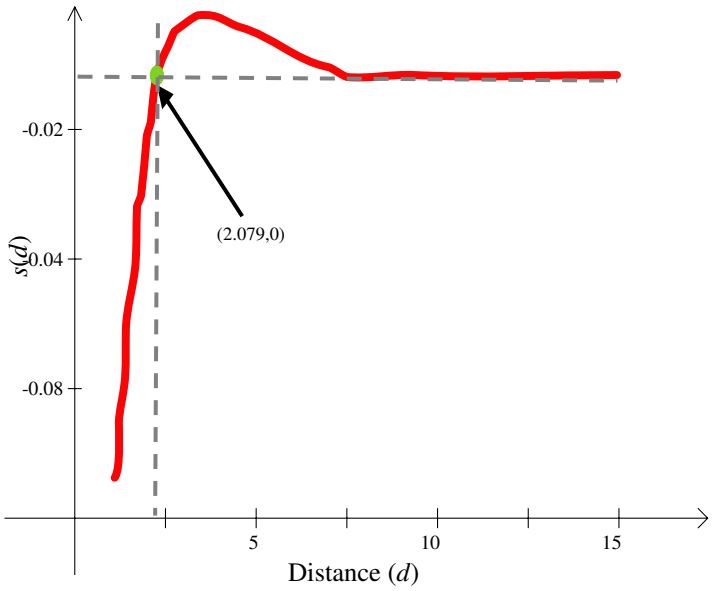


Figure 1: The value of $s(d)$ when $l = 1.15$ and $f = 0.5$

In this equation, $f$ symbolizes the intensity of attraction, while $l$ signifies the attractive length scale. The role of function $s$ in modulating the social interaction of grasshoppers is depicted in Fig. 1.

Analysing Fig. 1, repulsion forces are prevalent within the interval of [0, 2.079]. The "comfort area" is the zone where neither attraction nor repulsion forces exist, which is when the distance equals 2.079. The force of attraction intensifies from a distance of 2.079 units to approximately 4, then it gradually diminishes. Altering the parameters $l$ and $f$ in equation 3.3 results in different social behaviours in artificial grasshoppers, as observed in Fig. 2. Despite the merits of the function $s$, it falls short when dealing with long distances between grasshoppers as it can't exert strong forces. A resolution to this predicament is to map or normalize the distance between grasshoppers within the range [1, 4].

The gravitational component in equation 3.1, $G_i$, is computed as:

$$G_i = -g\hat{e_g} \quad (12)$$

Here, $g$ denotes the gravitational constant, and $\hat{e_g}$ represents a unity vector oriented towards the centre of the earth.

The wind advection component in equation 3.1, $A_i$, is calculated as follows:

$$A_i = u\hat{e_w} \quad (13)$$

In this equation, $u$ signifies a constant drift and $\hat{e_w}$ denotes a unity vector aligned in the wind's direction.

Consequently, equation 3.1 with all its components can be expressed as:

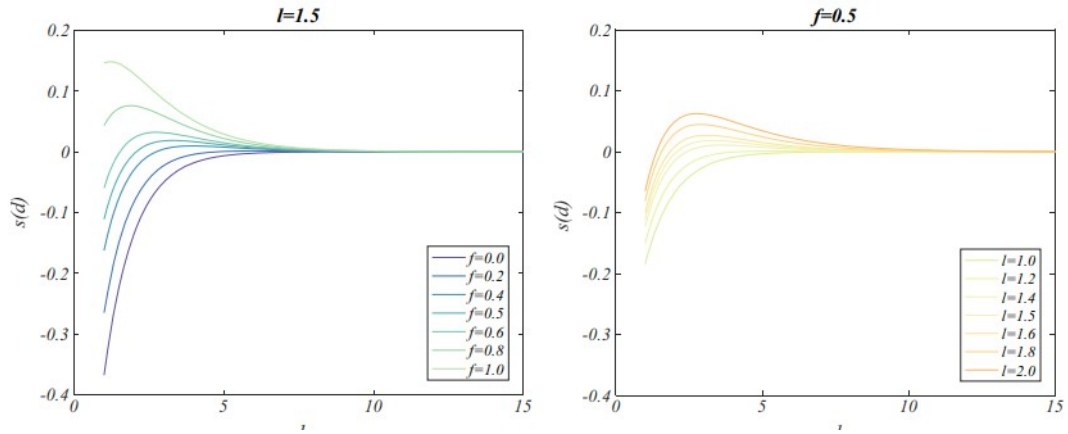


Figure 2: The $s(d)$ value with varying $l$ and $f$.

$$X_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s\left(\left|x_j - x_i\right|\right) \frac{x_j - x_i}{d_{ij}} - g\hat{e_g} + u\hat{e_w} \tag{14}$$

In order to efficiently address optimization issues, a stochastic algorithm needs to perform exploration and exploitation effectively to ascertain an accurate approximation of the global optimum. The preceding mathematical model should be configured with specific parameters to display exploration and exploitation at different stages of optimization. The suggested mathematical model is as follows:

$$X_i^d = c\left(\sum_{\substack{j=1 \\ j\neq i}}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right) \frac{x_j - x_i}{d_{ij}}\right) + \hat{T_d} \tag{15}$$

In this model, $ub_d$ and $lb_d$ represent the upper and lower bounds in the d-th dimension, respectively, $\hat{T_d}$ represents the value of the d-th dimension in the target, and $c$ is a decreasing coefficient to shrink the comfort, repulsion, and attraction zones. Here, we assume the wind direction (the $A$ component) always points towards a target ($\hat{T_d}$), and the gravity ($G$ component) is not considered.

The parameter $c$ plays a dual role in reducing the repulsion/attraction forces between grasshoppers proportional to the iteration count, and in reducing the search coverage around the target as the iteration counter increases.

The parameter $c$ is updated according to the following equation to decrease exploration and increase exploitation in relation to the number of iterations:

$$c = cmax - l\frac{cmax - cmin}{L} \tag{16}$$

Here, $cmax$ and $cmin$ denote the maximum and minimum values respectively, $l$ indicates the current iteration, and $L$ represents the maximum number of iterations. In our study, we use the values 1 and 0.00001 for $cmax$ and $cmin$, respectively.

## 4. The improved grasshopper algorithm

In the subsequent section, advancements in the Grasshopper Algorithm (GA) are delineated, wherein it undergoes a transformation into a binary algorithm, ideally tailored for extracting salient features from high-dimensional datasets. The evolved form, termed as the Improved Grasshopper Algorithm (IGA), incorporates the traditional multi-granular rough set (MGRS) within its objective function for solution assessment. However, it is noteworthy that MGRS encounters certain constraints when deployed for real-world applications. Addressing this, a third notable contribution of this paper introduces a refined multi-granular variant of MGRS, showcasing enhanced compatibility with real-world scenarios.

The initiation of IGA is characterized by the creation of a random population, analogous to its metaheuristic counterparts, symbolizing a gamut of solutions. Each solution undergoes a transformation into a Boolean vector, a pivotal step for feature extraction endeavors. Within this vector, entities marked by '1' signify pertinent features, whereas those demarcated by '0' point to non-essential features recommended for omission. Following this, the objective function's computation evaluates the merit of the selected features. For the purposes of this research, the objective function amalgamates two components: (i) the MGRS, and (ii) the proportion of selected features. In the ensuing phase, IGA is employed to pinpoint the solution boasting the apex value of the objective function. Thereafter, the solution ensemble is rejuvenated, harnessing the operations of the traditional GA algorithm, as elucidated in Section 2.3. The aforementioned procedures are reiterated until predefined termination criteria are satisfied. A comprehensive elucidation of the proposed algorithmic steps is provided in the ensuing sections.

### 4.1. Initial population

The generation of a random mother plant population of size $N$ is the first step in the proposed algorithm (akin to any swarm algorithm); thereafter, the conversion of each mother plant (solution) $s_i$ at the current iteration $t$ to a Boolean vector is given

as:

$$s_{i,j}^{t} = \begin{cases} 1 & \text{if } x_{i,j}^{t} > \theta \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where $\theta$ is a random value that signifies the threshold value, and it falls within the interval $[0, 1]$. Equation (18) is utilized to pinpoint the features that must be chosen (corresponding to 1's) and those that must be removed (corresponding to 0's).

### 4.2. Objective function

To identify which subset of features (solution) is useful, the objective function is calculated for it, where the degree of dependency $D$ defined in Eq. (7) is used. Since the $D$ assesses only the quality of features, we add another term that represents the number of chosen features; therefore, the objective function is defined as follows.

$$f_i^t = D\left(s_i^t\right) + \lambda\left(\frac{c_i^t}{n}\right) \quad (18)$$

where $\lambda$ represents the parameter that maintains a balance between the size of the chosen features and their quality [37]. $c_i^t$ and $n$ represent the number of chosen features using the $i$th solution and the total number of features in the dataset, respectively.

The optimal mother plant is selected based on the apex value of the objective function. Subsequently, for each mother plant $s_i$, a corresponding daughter plant $s_{i,j}$ is generated as per Eq. (11), followed by the computation of the objective function $f_{i,j}$ for each daughter plant. Amongst these, the most prominent daughter plant is designated as $s_{i,j}^{*}$. Should the condition detailed in Eq. (12) prevail, a global search is pursued; in its absence, a local search is initiated, utilizing the function $f_{i,j}^{*}$ as delineated in Eq. (13).

Elaborating further on the local search, as previously articulated, it transpires on two distinct scales: random expansive strides and random diminutive strides. Within the ambit of an expansive stride, each $s_{i,j}$ is juxtaposed against $s_{i,j}^{*}$, aiming to discern the one demonstrating superior prowess, as gauged by the objective function value. Analogous procedures are adopted for the local search pertaining to diminutive strides.

Progressing to the subsequent phase, mother plants are synthesized from the daughter plants, employing a fusion of elite and roulette methods as specified in Eqs. (15)–(17), priming them for the forthcoming iteration. The presence of a stall condition is subsequently scrutinized; contingent upon its affirmation, $s_{i,j}$ is selected, else $s_{i,j}^{*}$. These procedures are cyclically executed until the cessation criteria materialize, which may manifest as $s_{i,j}^{*}$ attaining its zenith, denoted as $s_{i,j}^{t_{\max}}$, or the culmination of the stipulated iteration count. A meticulous representation of the proposed IGAMGRS is encapsulated in Algorithm 1.

**Algorithm 1** Improved Grasshopper Algorithm based on Multi-Granular Rough Set (IGAMRS)

**Result:** The best solution $x_{\text{mbest}}$ that represents the optimal subset of features.

**Input:** The dataset with number of features $E$.
**Initialize:** Define the maximum number of iterations $t_{\max}$, the size of population $N_{\text{pop}}$, the maximum stall $stall_{\max}$, and $tol$.
**Procedure:**
Initialize the best function $fit_{\text{best}}$ Build a random population of mother grasshoppers, $x_m$ with size $N_{\text{pop}}$ $stall_{\text{count}} = 0, t = 1$
**while** *termination condition not satisfied* **do**
  **for** $k = 1$ **to** $N_{pop}$ **do**
    Generate daughter grasshopper $x_{dk}(t)$ using Eq. (11)
  **end**
  Compute the $f(x_d(t))$ for each $x_{dk}$ using Eq. (19) Determine the best daughter $x_{\text{dbest}}(t)$ as $x_{\text{dbest}}(t) = argmin_x = x_{dk} f(x)$ **if** *Condition in Eq. (12) is not true and* $t > 1$ **then**
    Construct the $x_{\text{perk}}$ for each $x_d$ using the local search with the large steps as in Eq. (13) Compute the objective function for $x_{\text{perk}}$ Update the best daughter $x_{\text{dbest}}(t)$ by comparing $f(x_{\text{dbest}}(t))$ Construct the $x_{\text{perk}}$ for each $x_{dk}(t)$ using the local search with the small steps using Eq. (14) Compute the objective function for $x_{\text{perk}}$ Update the best daughter $x_{\text{dbest}}(t)$ according to the local search with the small steps
  **end**
  Generate mother plants from the daughter plants using Eqs. (15)-(17) **if** $\left|\frac{f(x_{dbest}(t)) - f(x_{dbest}(t-1))}{f(x_{dbest}(t-1))}\right| \geq tol$ **then**
    $stall_{\text{count}} = 0$
  **end**
  **else**
    $stall_{\text{count}} = stall_{\text{count}} + 1$
  **end**
**end**

### 4.3. Distributed Computation of IGAMRS using Hadoop MapReduce

The Hadoop MapReduce paradigm can be used to distribute the computations involved in the Improved Grasshopper Algorithm based on Multi-Granular Rough Set (IGAMRS), thereby enhancing the efficiency of the algorithm when dealing with large datasets.

The MapReduce paradigm comprises two main stages: the Map stage and the Reduce stage. In the context of the IGAMRS algorithm, these stages can be defined in the following subsection. The description of the distributed version is shown in Fig. 3.

#### 4.3.1. Map Algorithm

In the Map stage, the population of mother grasshoppers, denoted as $x_m$, is divided into subsets. Each Map task works on one of these subsets. Each Map task then proceeds with the IGAMRS computation for its respective subset of $x_m$, involving the generation of daughter grasshoppers $x_{dk}(t)$, determination of the best daughter $x_{\text{dbest}}(t)$, and generation of mother plants. The

output of each Map task is a key-value pair, where the key represents a unique identifier of the task, and the value represents the best daughter grasshopper $x_{\text{dbest}}(t)$ and its corresponding objective function value. The map function pseudocode is given in Algorithm 2.

**Algorithm 2** Map Function for Distributed IGAMRS

**Result:** Key-value pairs of the best local solutions.
**Input:** The population subsets of mother grasshoppers $x_m$.
**Procedure:**
**foreach** *subset in* $x_m$ **do**
  Initialize the best function $fit_{\text{best}}$ Initialize the population of mother grasshoppers $x_m$ for the subset Initialize $stall_{\text{count}} = 0$, $t = 1$ **while** *termination condition not satisfied* **do**
    **for** $k = 1$ **to** $N_{pop}$ **do**
      Generate daughter grasshopper $x_{dk}(t)$ using Eq. (11)
    **end**
    Compute the $f(x_d(t))$ for each $x_{dk}$ using Eq. (19) Determine the best daughter $x_{\text{dbest}}(t)$ as $x_{\text{dbest}}(t) = argmin_x = x_{dk} f(x)$ Generate mother plants from the daughter plants using Eqs. (15)-(17)
  **end**
  Emit (subset identifier, $x_{\text{dbest}}(t)$)
**end**

*4.3.2. Reduce Algorithm*

In the Reduce stage, all key-value pairs from the Map tasks are aggregated. The Reduce task identifies the global best daughter grasshopper $x_{\text{dbest}}(t)$ from the collected values. This involves a comparison of the objective function values of the best daughter grasshoppers obtained from all Map tasks. The grasshopper with the lowest objective function value is then selected as the global best.

Using this MapReduce implementation of the IGAMRS algorithm, the time complexity can be significantly reduced, making the algorithm more suitable for applications involving large-scale feature selection problems. The reduce function pseudocode is given in Algorithm 3.

**Algorithm 3** Reduce Function for Distributed IGAMRS

**Result:** The global best solution.
**Input:** Key-value pairs from Map function.
**Procedure:**
Initialize the global best solution $x_{\text{gbest}}$ **foreach** *key-value pair* **do**
  Extract the best local solution $x_{\text{dbest}}(t)$ **if** $f(x_{dbest}(t))$ *is better than* $f(x_{gbest})$ **then**
    $x_{\text{gbest}} = x_{\text{dbest}}(t)$
  **end**
**end**

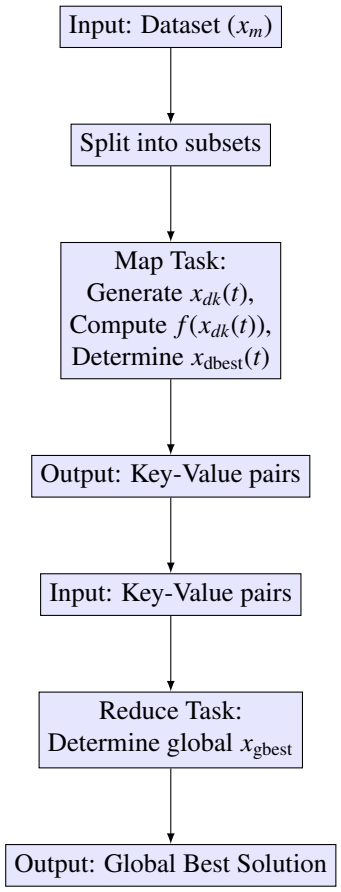


Figure 3: IGAMGRS in its distributed form using MapReduce.

## 5. Experimental work

The proposed IGAMGRS's performance was rigorously evaluated via a series of experimental tests. Initially, a diversified array of datasets from the UCI machine learning repository, each manifesting distinct attributes, were employed. The primary intent of such an examination was to identify a concise feature subset that preserves the dataset's intrinsic classificatory essence across generic data instances. As a secondary validation, a collection of lung cancer datasets served to ascertain the applicability and robustness of the IGAMGRS within the biomedical domain.

Throughout the experimental framework, classification efficacy was gauged using two renowned classifiers: Random Forest (RF) and $K$-Nearest Neighbor ($K$NN). These benchmarks were reached by executing diverse algorithms 50 times over the designated datasets and subsequently deriving the average accuracy. The hyperparameters for $K$NN, as well as the forest size for the RF classifier, were meticulously determined through iterative experimentation. Such iterations culminated in optimal configurations, settling on $K = 6$ for the $K$NN classifier and an ensemble size of 95 trees for the RF classifier. All computational experiments were orchestrated using Python, executed on a 64-bit Windows 11 operating system infrastructure.

Table 1: Description of datasets used in experiment I.

| # | Dataset | Samples | Features |
|---|---|---|---|
| 1 | Zoo | 101 | 17 |
| 2 | BreastCW | 699 | 10 |
| 3 | Lung Cancer | 32 | 56 |
| 4 | Heart | 270 | 13 |
| 5 | Congress | 435 | 16 |
| 6 | Ionosphere | 351 | 34 |
| 7 | WaveformEW | 5000 | 40 |
| 8 | Exactly | 1000 | 13 |
| 9 | Exactly2 | 1000 | 13 |
| 10 | M-of-n | 1000 | 12 |
| 11 | PenglungEW | 73 | 325 |
| 12 | Hayesroth | 160 | 5 |
| 13 | Madelon | 2600 | 500 |
| 14 | Isolate5 | 1559 | 617 |

### *5.1. Experiment I: Comparison using general benchmark dataset*

An exhaustive assessment of the proposed algorithm's performance was conducted employing a selection of fourteen distinct datasets sourced from the UCI repository. An elaborate description of these datasets is tabulated in Table 1. To prime the datasets for feature extraction by the proposed algorithm, a discretization step was deemed essential. For this procedure, the Boolean Reasoning method was chosen, primarily due to its intuitive nature and straightforwardness. Notwithstanding, it is pertinent to highlight that alternative discretization methodologies can be equivalently integrated within this framework, contingent upon the specific use case. Leveraging the Boolean Reasoning method facilitates the transmutation of the original datasets, yielding a format imbued with definitive values.

However, it is crucial to elucidate that the computational overhead associated with the Boolean Reasoning method is not incorporated into the cumulative CPU time, denoted in seconds, attributed to the feature extraction process. This demarcation stems from the intention to spotlight the intrinsic efficiency of the algorithm when engaged in the task of feature delineation.

The performance indicators utilized in this study include the overall accuracy, the Fisher score (F-score), the proportion of selected features, along with the average, standard deviation, the best and the worst of the objective function.

1. The *Overall Accuracy* is used to evaluate the classification model. It is defined by the number of accurate predictions the algorithm makes. Mathematically, accuracy can be denoted by Eq. (20).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{20}$$

Here, TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively.

2. The *Average Fisher Score (F-score)* is utilized to assess the efficacy of the selected features. It is represented by Eq. (21) [**? ? ?** ] as:

$$FS_j = \frac{\sum_{k=1}^{c} NC_k(\mu_{kj} - \mu_j)^2}{\sigma_j^2} \tag{21}$$

In this equation, $FS_j$ is the Fisher index of the $j$-th feature, $\sigma_j^2$ is the standard deviation, $\mu_{kj}$ and $\mu_j$ represent the mean of the $k$-th class and the mean of all datasets, respectively, and $NC_k$ is the size of the $k$-th class.

3. *The average ratio of the selected features* is another measure used to estimate the proportion of the selected features to the total features over $M$ iterations. The metric is expressed in Eq. (24) as:

$$SelR = \frac{1}{M}\sum_{i=1}^{M}\frac{N_{i_{Sel}}}{D} \tag{24}$$

Here, $N_{i_{Sel}}$ is the number of features selected in the $i$-th iteration, and $D$ is the total number of features in the dataset.

4. The *Average of the Objective Function, Averagef*, which is presented in Eq. (25):

$$Averagef = \frac{1}{M}\sum_{i=1}^{M} f_i \tag{25}$$

5. The *Standard Deviation, STDf* of the objective function values is a measure of the dispersion from the central point (Averagef) over $M$ iterations. It is computed as in Eq. (26):

$$STDf = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(f_i - Averagef)^2} \tag{26}$$

The algorithm is deemed stable and robust if the value of STDf is small; otherwise, if STDf is large, the algorithm results can be regarded as inconsistent and non-convergent.

The *best objective function, fBest*, is the minimum objective function value obtained over $M$ iterations, defined by Eq. (27):

$$fBest = \min_{i=1}^{M} f_i^* \tag{27}$$

where $f_i^*$ is the best fitness value achieved in the $i$-th iteration.

The *worst objective function, fWorst*, is the maximum objective function value achieved over $M$ iterations, and is defined by Eq. (28):

$$fWorst = \min_{i=1}^{M} f_i^* \tag{28}$$

where, similar to above, $f_i^*$ is the best objective value obtained at the $i$-th run.

In Table **??**, we provide a detailed account of the results pertaining to feature selection and the concomitant computational times. It is salient to observe that the proffered method, in a majority of datasets, predominantly identifies the most concise feature subsets while incurring the least computational expenditure. Nevertheless, for the Heart dataset, SSORS discerns the most minimal feature subsets, while CSRS does the same for both the Lung Cancer and Ionosphere datasets. When ap-

Table 3: The results of the standard deviation values of the objective function based on RS.

| | GARS | PSORS | ABCRS | FARS | SSORS | CSRS | HSRS | IGAMGRS |
|---|---|---|---|---|---|---|---|---|
| Zoo | 0.0541 | 0.0759 | 0.0645 | 0.0438 | 0.0531 | 0.0454 | 0.0334 | 0.0348 |
| BreastCW | 0.0147 | 0.0158 | 0.0144 | 0.0164 | 0.0072 | 0.0109 | 0.0157 | 0.0115 |
| Lung Cancer | 0.193 | 0.171 | 0.21 | 0.183 | 0.159 | 0.0103 | 0.174 | 0.052 |
| Heart | 0.061 | 0.032 | 0.045 | 0.052 | 0.054 | 0.059 | 0.023 | 0.021 |
| Congress | 0.019 | 0.017 | 0.007 | 0.015 | 0.012 | 0.014 | 0.007 | 0.007 |
| Ionosphere | 0.022 | 0.019 | 0.017 | 0.022 | 0.06 | 0.059 | 0.015 | 0.018 |
| WaveformEW | 0.012 | 0.007 | 0.009 | 0.009 | 0.0131 | 0.008 | 0.007 | 0.005 |
| Exactly | 0.024 | 0.091 | 0.092 | 0.087 | 0.024 | 0.018 | 0.071 | 0.022 |
| Exactly2 | 0.015 | 0.013 | 0.01 | 0.018 | 0.015 | 0.015 | 0.012 | 0.01 |
| M-of-n | 0.051 | 0.061 | 0.032 | 0.022 | 0.026 | 0.016 | 0.026 | 0.014 |
| PenglungEW | 0.0049 | 0.0086 | 0.0059 | 0.0078 | 0.0088 | 0.0045 | 0.007 | 0.0041 |
| Hayesroth | 0.0193 | 0.0312 | 0.0226 | 0.0431 | 0.0108 | 0.0267 | 0.0498 | 0.0316 |
| Madelon | 0.154 | 0.1109 | 0.0533 | 0.0405 | 0.1672 | 0.1581 | 0.1094 | 0.0756 |
| Isolet5 | 0.1329 | 0.069 | 0.0701 | 0.0759 | 0.0575 | 0.1234 | 0.0867 | 0.0864 |

praising computational efficiency, SSORS stands out for the BreastCW dataset, whereas CSRS emerges preeminent for the WaveformEW and Ionosphere datasets.

For datasets such as Hayesroth and Exactly2, ABCRS emerges as the pinnacle of time efficiency in earmarking salient features. Conversely, GARS and PSORS outshine others in the Exactly and PenglungEW datasets, respectively. The method under discussion manifests marked enhancements in performance for the following quintet of datasets: Heart, Lung Cancer, Zoo, Congress, and M-of-n.

When navigating datasets endowed with pronounced dimensionality, like Madelon and Isolet5, the advanced IRRA method evidences an accelerated efficiency compared to its counterparts. That being said, ABCRS and HSRA confer enhanced SelR for the datasets Madelon and Isolet5, in that order. In a consistent manner, the IGAMGRS algorithm optimally performs, producing both the premier average and the nadir objective function values across the entirety of the datasets.

Further, SSORS claims the runner-up position in terms of the superlative and least objective values. Following the IGAMGRS, ABCRS stands next in the hierarchy concerning the mean objective value. It is pivotal to annotate that GARS, unfortunately, registers as the most underperforming algorithm in this evaluation.

Furthermore, to assess the robustness of the proposed approach, we computed the standard deviation of the fitness function, as depicted in Table 3. It is noteworthy that the proposed algorithm, IGAMGRS, demonstrates the lowest standard deviation value across all datasets. This observation points towards its superior stability when compared with the other competing algorithms. Moreover, the CSRS algorithm exhibits a commendable standard deviation value, outperforming the other algorithms, while the standard deviation values of SSORS and HSRS are observed to be comparable.

Table 4 provides a depiction of the average classification accuracy of Random Forest (RF) and K-Nearest Neighbours (KNN) classifiers based on the selected features using various algorithms for each dataset. A generalized observation reveals that IGAMGRS delivers the highest accuracy across all datasets with both classifiers, followed closely by SSORS. Nevertheless, ABCRS maintains a solid third position, succeeded by CSRS and HSRS that showcase nearly equivalent accuracies. The conventional GARS algorithm, however, exhibits the least accuracy. In addition, the accuracy performance of PSORS supersedes that of the FARS algorithm.

Further dissecting the performance of the proposed method in each dataset, it can be noted from Table 4 that the SSORS algorithm delivers superior results for both BreastCW and Congress datasets. For the Exactly dataset, the highest accuracy is attained by employing ABCRS. On the other hand, IGAMGRS outperforms other algorithms in the remaining datasets.

## 6. Conclusions

In this investigation, we introduced a synergistic model that incorporated six distinct machine learning classifiers, three Convolutional Neural Networks (CNN) models, and the minimum-Redundancy Maximum-Relevance (mRMR) feature selection technique for data classification tasks. The model was trained and evaluated on a publicly accessible dataset comprising 100 samples. We utilized a 10-fold cross-validation approach to ensure the generalizability of the results.

The research was structured into five experiments. The primary goal of the first two experiments was to gauge the effectiveness of the CNNs and machine learning classifiers in the absence of data augmentation techniques. Given the limited size of the original dataset, the necessity for augmentation techniques became apparent. The third and fourth experiments followed the same protocol as the initial two, but we introduced augmentation techniques to assess their potential impact on model performance. As a consequence, we achieved a classification success rate of 98.74%, indicating that augmentation methods significantly enhanced performance.

The final experiment was designed to further improve the success rate achieved in the fourth experiment by using a more efficient subset of features. This experiment differed from its

Table 4: Comparison of the accuracy values of the proposed algorithm with other existing algorithms based on RS using RF and KNN classifiers.

| | GARS | PSORS | ABCRS | FARS | SSORS | CSRS | HSRS | IGAMGRS |
|---|---|---|---|---|---|---|---|---|
| Zoo (RF) | 82.79 | 84.61 | 88.48 | 84.06 | 86.00 | 86.47 | 86.67 | 89.67 |
| Zoo (KNN) | 78.79 | 80.61 | 84.85 | 81.88 | 83.50 | 84.12 | 83.88 | 86.88 |
| BreastCW (RF) | 88.07 | 92.54 | 96.40 | 93.26 | 96.83 | 95.67 | 95.92 | 96.69 |
| BreastCW (KNN) | 87.28 | 91.05 | 96.49 | 92.73 | 95.39 | 93.42 | 97.37 | 97.99 |
| Lung Cancer (RF) | 70.33 | 67.81 | 76.36 | 69.00 | 63.64 | 76.36 | 73.00 | 78.07 |
| Lung Cancer (KNN) | 60.00 | 65.36 | 69.64 | 64.60 | 69.64 | 70.64 | 66.00 | 75.98 |
| Heart (RF) | 67.35 | 64.85 | 71.64 | 66.47 | 81.64 | 64.18 | 77.61 | 84.91 |
| Heart (KNN) | 65.06 | 66.41 | 70.15 | 69.41 | 71.57 | 70.15 | 73.13 | 75.18 |
| Congress (RF) | 91.66 | 91.24 | 94.48 | 92.79 | 96.86 | 95.17 | 94.48 | 96.58 |
| Congress (KNN) | 89.66 | 90.41 | 91.38 | 91.38 | 95.86 | 93.79 | 92.79 | 95.17 |
| Ionosphere (RF) | 82.60 | 83.41 | 85.10 | 82.44 | 86.10 | 86.20 | 86.40 | 87.01 |
| Ionosphere (KNN) | 79.85 | 80.40 | 84.90 | 80.80 | 84.70 | 85.40 | 84.00 | 86.07 |
| WaveformEW (RF) | 79.01 | 78.12 | 82.90 | 79.89 | 83.10 | 82.40 | 83.20 | 85.09 |
| WaveformEW (KNN) | 74.82 | 76.82 | 80.60 | 77.90 | 81.90 | 80.60 | 80.20 | 83.25 |
| Exactly (RF) | 81.30 | 84.93 | 89.10 | 83.89 | 87.00 | 88.30 | 85.40 | 88.17 |
| Exactly (KNN) | 78.98 | 79.50 | 88.40 | 80.70 | 84.70 | 86.00 | 84.60 | 86.40 |
| Exactly 2 (RF) | 72.70 | 72.37 | 74.50 | 72.35 | 77.19 | 75.80 | 76.04 | 77.55 |
| Exactly 2 (KNN) | 70.60 | 71.09 | 73.80 | 71.40 | 74.50 | 73.20 | 73.60 | 75.80 |
| M-of-n (RF) | 92.80 | 93.20 | 96.50 | 92.72 | 96.90 | 97.10 | 96.90 | 98.70 |
| M-of-n (KNN) | 91.10 | 92.39 | 95.40 | 91.37 | 94.80 | 94.30 | 95.70 | 97.09 |
| PenglungEW (RF) | 61.64 | 62.34 | 67.51 | 63.49 | 67.29 | 65.38 | 67.20 | 70.09 |
| PenglungEW (KNN) | 59.64 | 60.14 | 65.20 | 60.44 | 65.30 | 64.30 | 66.52 | 67.90 |
| Hayesroth (RF) | 85.61 | 88.41 | 96.57 | 90.26 | 96.53 | 95.05 | 96.92 | 98.25 |
| Hayesroth (KNN) | 83.68 | 86.78 | 96.48 | 87.33 | 95.61 | 93.74 | 96.04 | 97.37 |
| Madelon (RF) | 74.02 | 76.67 | 74.02 | 71.86 | 77.16 | 75.06 | 71.28 | 79.79 |
| Madelon (KNN) | 70.81 | 74.25 | 75.06 | 68.48 | 76.67 | 73.04 | 67.32 | 78.98 |
| Isolet5 (RF) | 80.77 | 89.33 | 79.17 | 83.33 | 85.43 | 87.50 | 84.62 | 90.59 |
| Isolet5 (KNN) | 71.15 | 86.67 | 75.00 | 79.17 | 71.88 | 75.00 | 74.04 | 88.15 |

predecessors by focusing on enhancing time and speed efficiency in the classification process. To achieve this, we opted to reduce the feature dimensionality using the mRMR feature selection method. This strategy led to a more time-efficient approach. We discovered that a combination of IGAMGRS, Neural Network (NN), and the mRMR method yielded the most promising results, with an accuracy of 99.51%, a sensitivity of 99.32%, and a specificity of 99.71%.

## References

[1] Y. Li, A. Ngom, Machine learning applications in genetics and genomics, Genes 11 (6) (2020) 602.
[2] W. Liu, S. Zhang, W. Luo, Deep learning in spectral analysis and signal processing, Journal of Signal Processing Systems 92 (2) (2020) 151–162.
[3] C. Shorten, T. M. Khoshgoftaar, A survey on image data augmentation for deep learning, Journal of Big Data 6 (1) (2019) 1–48.
[4] Y. Guo, Y. Liu, T. Georgiou, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 415 (1) (2020) 296–317.
[5] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, Springer series in statistics New York, 2001.
[6] G. Chandrashekar, F. Sahin, A survey on feature selection methods, Computers & Electrical Engineering 40 (1) (2014) 16–28.
[7] X. Wang, I. Mitra, W. Cheng, Interaction-aware feature selection with application in bioinformatics, BMC Bioinformatics 20 (1) (2019) 1–10.
[8] A. Hamed, M. Torky, S. AlJanah, A. Alshammari, T. Gaber, A blockchain-based fox optimization algorithm for optimizing and securing electrical vehicles charging, IEEE Transactions on Intelligent Vehicles (2024).
[9] A. Bommert, X. Sun, U. Brefeld, Neural networks for feature selection and classification, Journal of Machine Learning Research 21 (110) (2020) 1–32.
[10] N. Gupta, G. Purohit, Rough set theory for feature selection in data mining: a review, Archives of Computational Methods in Engineering 26 (6) (2019) 1519–1533.
[11] B. Zhou, Q. Li, Multi-granulation rough set over two universes and its properties, International Journal of Pattern Recognition and Artificial Intelligence 33 (05) (2019) 1950019.
[12] J. Kennedy, R. Eberhart, Particle swarm optimization, Encyclopedia of Machine Learning and Data Mining (2020) 1–8.
[13] T. Gaber, M. Nicho, E. Ahmed, A. Hamed, Robust thermal face recognition for law enforcement using optimized deep features with new rough sets-based optimizer, Journal of Information Security and Applications 85 (2024) 103838.
[14] S. Saremi, S. M. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, Advances in Swarm Intelligence for Optimizing Problems in Computer Science (2019) 1–37.
[15] S. Potghan, Multi-layer perceptron based lung tumor classification. 2018 second int conf electron commun aerosp technol, 2018, pp. 499–502.
[16] H. Wang, Z. Zhou, Y. Li, Z. Chen, P. Lu, W. Wang, W. Liu, L. Yu, Comparison of machine learning methods for classifying mediastinal lymph node metastasis of non-small cell lung cancer from 18 f-fdg pet/ct images, EJNMMI research 7 (2017) 1–11.
[17] S. K. Bandyopadhyay, Edge detection from ct images of lung (2012) 34–37.
[18] N. Aggarwal, V. Kumar, S. Verma, A novel approach to classify the lung cancer using artificial neural network (2015) 109–112.

[19] A. Teramoto, T. Tsukamoto, Y. Kiriyama, H. Fujita, Automated classification of lung cancer types from cytological images using deep convolutional neural networks, Biomedical Physics & Engineering Express 3 (2017) 025014.
[20] Q. Song, L. Zhao, X. Luo, X. Dou, Deep learning based classification of lung nodules on computed tomography images (2017) 687–695.
[21] S. Liu, Q. Zhang, Q. Wang, P. Chen, W. Gao, H. Yu, Multi-view convolutional neural networks for lung nodule classification (2017).
[22] R. Pandey, N. Pandey, Early detection of lung cancer based on wavelet transform, International Journal of Computer Applications 46 (2012) 11–15.
[23] N. Margarita, Y. Hirano, H. Watanabe, Y. Shintani, S. Funaki, N. Oyama-Manabe, T. Tsujikawa, Y. Kiyono, H. Okazawa, H. Kondo, K. Tomiyoshi, Deep learning approach to classify lung cancer on fdg-pet/ct images (2018).
[24] R. J. Thomas, Y. Liu, S. Park, A. Krishnan, A. Baidoshvili, J. Cebula, K. Murphy, J. Steinkamp, S. Ramalingam, S. Force, I. Pastan, A. A. Gal, Performance of a deep learning based neural network in the selection of lung cancer screening candidates (2018).
[25] Q. Wang, J. Hou, H. Liu, Y. Li, W. Huang, X. Yan, Y. Xia, A computational prognostic model for lung adenocarcinoma based on histomorphometric features and clinical variables (2018).
[26] S. Watson, B. Hall, P. Bannon, J. Ferguson, G. Schembri, The use of semi-automated volumetric diameter and density measurements to assess stability in solid pulmonary nodules (2016).
[27] C. M. Lynch, V. H. van Berkel, H. B. Frieboes, Application of unsupervised analysis techniques to lung cancer patient data, PLoS One 12 (9) (2017) e0184370.