Research on the Application of Random Forest-based Feature Selection Algorithm in Data Mining Experiments

Huan Wang*
College of Big Data and Intelligence Engineering, Southwest Forestry University, Kunming, Yunnan, 650224, China

***Abstract*—Handling high-dimensional big data presents substantial challenges for Machine Learning (ML) algorithms, mainly due to the curse of dimensionality that leads to computational inefficiencies and increased risk of overfitting. Various dimensionality reduction and Feature Selection (FS) techniques have been developed to alleviate these challenges. Random Forest (RF), a widely-used Ensemble Learning Method (ELM), is recognized for its high accuracy and robustness, including its lesser-known capability for effective FS. While specialized RF models are designed for FS, they often struggle with computational efficiency on large datasets. Addressing these challenges, this study proposes a novel Feature Selection Model (FSM) integrated with data reduction techniques, termed Dynamic Correlated Regularized Random Forest (DCRRF). The architecture operates in four phases: Preprocessing, Feature Reduction (FR) using Best-First Search with Rough Set Theory (BFS-RST), FS through DCRRF, and feature efficacy assessment using a Support Vector Machine (SVM) classifier. Benchmarked against four gene expression datasets, the proposed model outperforms existing RF-based methods in computational efficiency and classification accuracy. This study introduces a robust and efficient approach to feature selection in high-dimensional big-data scenarios.**

***Keywords*—Random forest; SVM; machine learning; big data; feature selection; best-first search; rough set theory**

I. Introduction

High-dimensional big data poses significant challenges for Machine Learning (ML) algorithms due to the "curse of dimensionality," a phenomenon where the computational complexity and resource requirements increase exponentially as the number of dimensions (features) grows [1]. Traditional algorithms can struggle to make accurate predictions as they become lost in the vastness of the feature space, leading to issues such as overfitting, where the model captures noise instead of the underlying data structure. To mitigate these problems, various dimensionality reduction techniques like Principal Component Analysis (PCA), t-distributed Stochastic Neighbor Embedding (t-SNE), and autoencoders have been developed to compress the feature space while retaining as much of the meaningful information as possible [2]. Additionally, feature selection methods like the Least Absolute Shrinkage and Selection Operator (LASSO), Mutual Information (MI), and Chi-Square Test (CST) are used to identify the most informative features. More sophisticated ML models, such as Deep Learning (DL) models, are also designed to automatically capture hierarchical representations of the data, thereby mitigating some of the challenges posed by high dimensionality.

The Random Forest (RF) technique is a collective learning approach that integrates numerous decision trees to build a more robust and precise forecasting model. Functionally, each tree in the ensemble is built from a bootstrapped sample of the data, and during the tree-building process, a random subset of features is chosen at each node split [3]. This randomization not only decorates the trees but also makes the ensemble less prone to overfitting, enabling it to perform well on unseen data. As a predictor, RF is renowned for its exceptional accuracy, capability to process vast datasets with extensive dimensionality, and capability to handle missing values. One of the lesser-known but advantageous features of an RF model is its innate capability for Feature Selection (FS) [4]. During training, it computes a score for each feature that indicates its importance in making predictions. This feature importance score is often derived from the average reduction in impurity that each feature brings across all trees in the forest. By ranking features based on this score, RF provides a practical and intuitive way for FS, helping to improve the performance of not only itself but also other ML models that may be sensitive to irrelevant or redundant features [5].

Many RF models are specialized for FS, each offering unique advantages and disadvantages. Variants like Boruta [6] focus on systematically identifying important attributes by comparing them to randomly shuffled versions of themselves, while Conditional Inference Forest (CIF) [7] aims for unbiased FS through statistical hypothesis tests. Regularized RF [8] applies a regularization term to prioritize a sparse set of features, and Extremely Randomized Trees (RT) [9] adds an extra layer of randomness for potentially more robust selections. However, a common drawback in most of this RF-based Feature Selection Model (FSM) is their lack of focus on handling large datasets. These methods often strive for computational efficiency, facing challenges related to memory space and computing time. To mitigate these issues, high-performance computing environments and parallel architectures are often necessary for effective FS on big datasets. Failing to use such computational resources can significantly ramp up hardware and software costs. For example, scalable software frameworks like Hadoop MapReduce are often required for the learning and analysis stages to manage large datasets efficiently. Therefore, while RF-based methods offer numerous avenues for FS, their

applicability to big data scenarios often necessitates additional computational resources to overcome inherent limitations.

Considering the challenges associated with computational time complexity and classification accuracy in high-dimensional datasets, a novel FSM integrated with data reduction techniques is proposed. The architecture operates in four distinct phases. Initially, high-dimensional data undergo preprocessing to standardize and clean the dataset. Subsequently, the preprocessed data are processed through a Best-first Search with Rough Set Theory (BFS-RST) Feature Reduction Model (FRM) in the second phase. This specialized model aims to reduce the feature size effectively. In the third phase, a novel proposed variant of RF termed Dynamic Correlated Regularized Random Forest (DCRRF) is employed. This DCRRF model incorporates correlated FSM to identify an optimal set of features from the already-reduced set. The final phase involves a rigorous assessment of the quality and efficacy of the FS using a Support Vector Machine (SVM) classifier. Performance benchmarks indicate that this proposed model outperforms existing RF-based FSM when tested on four gene expression datasets. The architecture aims to mitigate computational inefficiency and enhance classification accuracy, offering a more robust approach to FS in complex data scenarios.

The paper is organized as follows: Section II presents the literature review, Section III presents the methodologies used in the work, Section IV presents the proposed model, Section V analyses the work using different experiments, and Section VI concludes the work.

II. Literature Review

This section delves into various works that have employed RF-based models for FS, offering insights into their efficacy and limitations.

The research in [10-12] presents a two-step RF-based FSM. The first step selects features based on variable importance scores and then employs the search process in the second step to finalize a feature subset. The approach was tested on the KDD'99 intrusion detection dataset, derived from the DARPA 98 dataset. Notably, the KDD'99 dataset was modified to remove redundant records, resulting in a refined dataset called RRE-KDD for training and testing. Experimental results indicated that this approach reduced the feature set and computational time and enhanced classification accuracy. The study in [13-15] explores the use of the RF classifier for FS in prostate cancer detection. Utilizing an ensemble of Decision Trees (DT) for classification, the study notes that the accuracy improves with adding more trees. The classifier is adept at handling incomplete data attributes and is scalable for large datasets. Emphasizing the pivotal role of FS, the research finds that their method boosts detection accuracy by roughly 87%, underscoring the importance of effective FS in enhancing prostate cancer detection. The research in [16-18] study introduces a recursive FSM using RF to enhance protein structural class prediction. The method underwent evaluation through four experiments and was compared to existing prediction techniques. Findings suggest that this feature selection approach significantly bolsters the efficiency of predicting protein structural classes. Remarkably, the method uses fewer than 5% of the features yet boosts prediction accuracy by 4.6-13.3%. Further analysis revealed that features related to predicted secondary structures yielded the best performance, providing insights that could inform the development of even more effective prediction methods for protein structural classes.

The study in [19-23] explores how the number of trees and class separability influence the consistency of variable importance rankings in RF algorithms. The research concludes that achieving stable importance values is possible either by incorporating a large number of trees in a single execution of the model or by taking the average values from multiple runs with fewer trees. While the second approach is more economical regarding computational cost, both methods produce comparable rankings for the variables. The research additionally points out that the ideal number of model iterations fluctuates depending on class separability and offers recommendations for ascertaining the appropriate number of runs or trees to achieve stable rankings of variable importance. In study [24-28], introduce an explainable Artificial Intelligence (AI) model for blood test sample-based COVID diagnosis. Despite the advancements in AI-based diagnostic models, few effectively integrate human-centered and machine-centered approaches. This research employs human-computer interaction design principles to address this gap. Employing graph analysis for the visualization and optimization of features, the model integrates an interpretable decision forest classifier to categorize COVID-19 cases using existing blood test information. This enables clinicians to leverage DT structures and feature visualizations for better model interpretability. They proved that their model had not just better diagnostic accuracy but also reduced computation time.

The research in [29-34] examines the efficacy of ML algorithms like RF and its variations in selecting Single Nucleotide Polymorphisms (SNPs) for fine-scale genetic population assignment in wildlife conservation. The study, which uses unpublished data for Atlantic salmon and published data for Alaskan Chinook Salmon (ACS), found that ML methods outperformed traditional Fixation Index (FST) rankings in identifying informative genetic markers. Specifically, RF-based methods led to an accuracy improvement of up to 7.8% and 11.2% for ACS, respectively. The findings underscore the potential of ML algorithms in enhancing genetic marker selection for conservation efforts. The research in [35-40] addresses the challenges in intrusion detection systems, such as the scarcity of labeled datasets, computational overhead, and suboptimal accuracy. The research introduces an Auto-Encoder Intrusion Detection System (AE-IDS) that leverages the RF algorithm for improved performance. The approach focuses on creating a robust training set through FS and grouping in [41-45]. Post-training, the model employs an auto-encoder for prediction, significantly reducing detection time and enhancing accuracy. Experimental findings suggest that AE-IDS outperform conventional ML-based IDS, offering more accessible training, better adaptability, and higher detection accuracy in [46-50]. The research in [51-55] employs an RF algorithm for county-scale cotton mapping, using spectral, vegetation, and

texture features. The study found that texture features, particularly the Gray Level Co-occurrence Matrix (GLCM), significantly improve classification accuracy. Compared to other classifiers like SVM and ANN, RF exhibited better stability and higher accuracy. The method that combined multiple features achieved an average accuracy of 93.36%, showing the effectiveness of using RF and multiple features for precise cotton mapping [56-58].

III. Proposed Methodologies

A. Random Forest

RF is an Ensemble Learning (EL) algorithm that builds a forest of DT, usually trained with the "bagging" method. The general idea of the Ensemble Learning Method (ELM) is to combine weak learners to create a robust model. In RF, each DT, T_m is trained on a different bootstrap sample D_m Drawn from the original dataset. The algorithm performs this operation M times based on the parameter n_{est}, effectively creating M different trees. A unique aspect of RF is that it considers only a subset of features when making each split, a number specified by the parameter max_f. This random subset of features introduces diversity among the trees, leading to a more robust model.

For regression problems, the output of a RF model is the mean prediction of all the trees, mathematically expressed as Eq. (1).

$$\hat{Y}(X) = \frac{1}{M}\sum_{m=1}^{M} T_m(X) \quad (1)$$

In classification tasks, the model employs a Majority Voting Scheme (MVS), choosing the mode of the classes predicted by individual trees, given as Eq. (2).

$$\hat{Y}(X) = \text{mode}(T_1(X), T_2(X), \dots, T_M(X)). \quad (2)$$

This EML provides a way to reduce the variance that might be present in a single DT, improving generalization to unseen data. One of the essential aspects of RF is the criteria used for node splitting, often specified by the Gini Impurity (GI) as shown in Eq. (3).

$$G(S) = 1 - \sum_{i=1}^{k} p_i^2, \quad (3)$$

where p_i is the proportion of samples of class i at a node, GI quantifies the "messiness" of the data. The algorithm aims to minimize the weighted sum of the GI of child nodes when making each split. This weighted sum can be calculated as in Eq. (4).

$$\Delta G = \sum_{j=1}^{2} \frac{|S_j|}{|S|} G(S_j). \quad (4)$$

In addition to GI, entropy is another criterion which is sometimes used for splitting nodes, defined as $H(S) = -\sum_{i=1}^{k} p_i \log_2 p_i$. The algorithm then selects the split that maximizes the information gain, calculated as in Eq. (5).

$$IG(S, f) = H(S) - \sum_{j=1}^{2} \frac{|S_j|}{|S|} H(S_j). \quad (5)$$

One lesser-known but critical aspect of RF is the Out-of-Bag (OOB) error. This internal error estimate eliminates the need for a separate validation set. Each tree in the forest leaves out some samples during its bootstrap training, called OOB samples. The OOB error for each tree T_m is calculated using its corresponding OOB_m samples as in Eq. (6).

$$OOB_{error\,T_m} = \frac{1}{|OOB_m|}\sum_{(x_i,y_i)\in OOB_m} L(y_i, T_m(x_i)) \quad (6)$$

The overall OOB error for the RF is the average of these individual tree OOB errors as shown in Eq. (7).

$$\text{Overall } OOB_{error} = \frac{1}{M}\sum_{m=1}^{M} OOB_{error\ T_m} \quad (7)$$

where, $L(y, \hat{y})$ is a loss function measuring the difference between the true label y and the predicted label $\hat{y}$.

Algorithm 1 for RF Algorithm

Initialize Parameters:

(i) tree count in the forest (M): "n_{est}"

(ii) features needed for each split: "max_f"

(iii) each tree's maximum depth: "max_d"

(iv) the sample count needed to split a node: "$minsam_{split}$"

(v) The sample count needed to be a leaf node: "$minsam_{leaf}$".

For $m = 1$ to:

- *Bootstrap Sampling*
- *Draw a bootstrap sample D_m of size N from the training dataset.*
- *Identify Out-of-Bag samples $OOB_m = D - D_m$*
- *Build Tree T_m*
- *Initialize the root node with D_m*

For Each node:

- *Check Terminal Conditions*

If depth equals max_d or $|D_m| < minsam_{split}$ or $|D_m| < minsam_{leaf}$, make the node a leaf and stop.

Feature Selection

- *Randomly select max_f Features without replacement.*

Find Best Split

For Each

- *FS compute the best split based on either GI or Entropy.*
- *The best split minimizes the weighted GI or maximizes Information Gain.*

Split the Node

- *Divide D_m into two subsets $D_{m,left}$ and $D_{m,right}$ based on the best split.*

Recursive Split

- *Repeat steps 1.2.1 to 1.2.4 for the child nodes $D_{m,left}$ and $D_{m,right}$.*

Calculate OOB Error for T_m

- $OOB_{error\ T_m} = \frac{1}{|OOB_m|}\sum_{(x_i,y_i)\in OOB_m} L\big(y_i, T_m(x_i)\big)$

Calculate Overall OOB Error

- *Overall* $OOB_{error} = \frac{1}{M}\sum_{m=1}^{M} OOB_{error\ T_m}$

End If

End

1) Variable importance scores from RF: RF is not only known for its robust predictive power but also for its built-in FS capabilities. One of the metrics that the algorithm provides for understanding the dataset is the variable importance score for each feature. Understanding variable importance is crucial for improving and interpreting the model's decisions.The variable importance score in an RF algorithm is computed based on two principal factors:

a) Mean Decrease in Impurity (MDI): This method calculates the average reduction in impurity-Gini impurity or entropy, for example, for each feature brought about when used for node splitting. Mathematically, the Mean Decrease in Impurity for feature f is computed as shown in Eq. (8).

$$\text{MDI}(f) = \sum_{\text{all nodes using } f} (\text{Impurity of Parent Node - Weighted Impurity of Child Nodes}) \quad (8)$$

b) Mean Decrease in Accuracy (MDA): Another method, which usually involves using the Out-of-Bag (OOB) error, calculates the decrease in model accuracy when a particular feature is permuted. The idea is to assess how much worse the model performs without each feature. The formula for MDA can be generalized as Eq. (9).

$$\text{MDA}(f) = \frac{1}{M}\sum_{m=1}^{M}\left(\text{OOB\_Error}_{\text{ with } f} - \text{OOB\_Error}_{\text{ without } f}\right) \quad (9)$$

Calculation Steps:

Step 1. Run the RF Algorithm: First, generate the RF model using all variables and calculate the OOB error rate.

Step 2. Permute Each Variable: For Each feature f in the dataset, randomly permute the values of f in the OOB samples and record the new OOB error.

Step 3. Compute Importance: For Each feature$f$, compute the Mean Decrease in Accuracy or Mean Decrease in Impurity, depending on which method you're using.

Step 4. Normalize Scores: The raw importance scores can be normalized to sum to one, making them easier to interpret and compare.

2) Regularized random forest (RRF): RRF is an advanced extension of the traditional RF algorithm. While RF is already effective in ensemble learning, RRF takes a step further by incorporating regularization techniques aimed at reducing overfitting and improving feature selection. In standard RF models, each DT T_m is trained independently on a bootstrap sample D_m , with no explicit mechanism for feature regularization. RRF, however, adds a regularization term to the ELM, effectively penalizing the complexity of individual trees.

The objective function for each tree in RRF can be mathematically represented as in Eq. (10).

$$\text{Objective}(T_m) = \text{Impurity}(T_m) + \lambda\text{Complexity}(T_m) \quad (10)$$

Here, Impurity (T_m) refers to the impurity measure, which can be either GI or entropy. Complexity (T_m) is a function quantifying the complexity of the tree, such as the depth or the number of leaves. λ is the regularization parameter controlling the trade-off between impurity and complexity. This parameter is usually determined through cross-validation. In the RRF model, the standard information gain is replaced by a regularized form, $\text{Gain}_R(X_i, v)$, which integrates the regularization term:

$$\text{Gain}_R(X_i, v) = \begin{cases} \lambda \times \text{Gain}(X_i, v) & \text{if } i \notin F \\ \text{Gain}(X_i, v) & \text{if } i \in F \end{cases} \quad (11)$$

Here, F is the set of feature indices already used for splitting in previous nodes. The term λ serves as the penalty coefficient. Regularization in RRF can be applied at different stages:

- During Feature Selection: The regularization term is incorporated into the evaluation metric used for selecting the features for node splitting.
- During Tree Pruning: After constructing the trees, they can be pruned to minimize the regularized objective function.

By introducing the regularization term, RRF balances model complexity and fit quality, ensuring a more interpretable and robust ensemble model. This is particularly useful in cases where the dataset contains many irrelevant features or when overfitting is a concern. Therefore, RRF benefits from the inherent advantages of Random Forests while simultaneously mitigating some of their limitations.

B. Best-First Search (BFS)

BFS is a tree-based search algorithm that aims to find the most optimal solution by navigating through the state space of possible solutions. In the context of feature selection, each node N in the search tree represents a subset of features S, and the root node usually represents an empty set or the complete feature set. The primary driving force of the algorithm is an evaluation function $f(N)$, which measures the 'quality' or 'promising nature' of node N. Mathematically, the evaluation function $f(N)$ can be expressed as in Eq. (12).

$$f(N) = g(N) + h(N) \quad (12)$$

where, $g(N)$ is the cost to reach the current node from the root (often equal to the number of features in S when feature reduction is the goal), and $h(N)$ is the heuristic estimate of the cost to reach an optimal solution from N. The algorithm maintains a priority queue Q, initialized with the root node. The nodes are sorted in Q based on their evaluation scores. The algorithm iteratively performs the following steps until a stopping criterion is met:

Algorithm 2 for BFS for FS
Input:

- *S : Complete set of features*
- *$f(N)$: Evaluation function for a node N*
- *Stopping Criteria: C*

Output:

- *Optimal subset of features $S_{optimal}$*
  Initialize:
- *Create an empty priority queue Q*
- *Create a root node N_{root} with no features or all features, add N_{root} to Q*

Steps:

While Q is not empty and stopping criteria C are not met:

- *Pop the node N with the lowest $f(N)$ from Q*
  If N satisfies C :

Return $S_{optimal}$ as the feature subset in N

Exit

Else

- *Generate child nodes of N by adding or removing features from S*
  Evaluate $f(N)$ For Each Child Node
- *Insert the child nodes into Q*
- *Re-sort Q based on $f(N)$*

The mathematical representation of the priority queue after *'k'* iterations can be represented as in Eq. (13).

$$Q_k = \{N_1, N_2, \dots, N_m\} \text{ s.t. } f(N_1) \le f(N_2) \le \cdots \le f(N_m) \quad (13)$$

By focusing on the most promising subsets of features, BFS achieves a balance between exhaustive search and greedy algorithms. However, it can be computationally intensive, especially when the feature space is ample, as the time complexity can go up to $O(b^d)$, where b is the branching factor, and d is the depth of the solution.

C. Rough Set Theory (RST)

FR helps reduce the computational cost, simplifying models and sometimes even improving the performance by eliminating irrelevant or redundant features. RST developed that can be employed for feature reduction. RST provides a formal mathematical framework to deal with vagueness and uncertainty in data. In the context of Feature Reduction (FR), it helps identify the minimal set of features indispensable for preserving the discernibility between objects. In simpler terms, it helps find the most miniature set of features necessary and sufficient for classification tasks.

Let U represent the universe of objects or instances in the dataset, and let A denote the set of attributes or features. A decision table $T = (U, A)$ may be formed, in which U comprises the rows, and A makes up the columns. Additionally, D a subset of A, can be introduced as the decision attribute(s) of interest. Using this foundation, the following aspects of RST are discussed:

a) Indiscernibility Relation: The fundamental concept in RST is the indiscernibility relation. For a given subset of attributes, $B \subseteq A$an indiscernibility relation $IND(B)$ is defined as follows in Eq. (14).

$$IND(B) = \{(x, y) \in U \times U \mid \forall a \in B, a(x) = a(y)\} \quad (14)$$

Here, $a(x)$ is the value of attribute a for object x. The indiscernibility relation $IND(B)$ groups objects that cannot be distinguished by attributes in B

b) Lower and Upper Approximations: Given a target set $X \subseteq U$, the lower and upper approximations are defined as in Eq. (15) and Eq. (16).

Lower Approximation: $\underline{B_X} = \{x \in U \mid [x]_B \subseteq X\}$ (15)

Upper Approximation: $\overline{\overline{B_X}} = \{x \in U \mid [x]_B \cap X \neq \emptyset\}$ (16)

Here, $[x]_B$ represents the equivalence class of x concerning B.

c) Core and Reduct: The core attributes are indispensable for maintaining the exact lower approximation for every subset of U as the entire set A. Mathematically, Eq. (17).

$$\text{Core}(A, D) = \left\{a \in A \mid \underline{A_D} \neq \underline{(A - \{a\})_D}\right\} \quad (17)$$

A reduct is a minimal subset B of A such that $\underline{B_D} = \underline{A_D}$. In other words, B and A give the same lower approximations for each decision class.

D. Feature Reduction using RST

The overarching goal is to identify all possible reducts and then choose the one with the least number of attributes while preserving the classification power of the original dataset. However, finding all reducts can be computationally taxing. For this reason, heuristic approaches are frequently used to approximate a minimal reduct effectively.

1) Initialize with core attributes: Start by calculating the core attributes, denoted as Core (A, D), which are essential for classification. Initialize the reduct set, Reduct, with these core attributes.

2) Iterative refinement: Continue refining the reduct set until it provides the same classification power as the complete

attribute set A. Specifically, iterating while reducing $_D$ is not equal to $\underline{A_D}$.

- Evaluate Significance: For each remaining attribute a in A - Reduct, evaluate its significance in distinguishing between different classes.
- Select Most Significant Attribute: Add the attribute with the highest significance score to the Reduct set.

By the end of this iterative process, Reduct will contain a minimal set of attributes that retains the original dataset's ability to distinguish between different classes.

E. Correlation-based Feature Selection (CFS)

CFS is an FSM designed to improve model performance by FS that are highly correlated with the target variable and minimally correlated with each other. The process typically begins with data preprocessing to standardize or normalize the features then calculating a correlation matrix. Based on this matrix, an initial subset of FS either through predefined correlation thresholds or optimization algorithms. The criterion Cr, often used to maximize the quality of the feature subset, is Eq. (18).

$$Cr = \frac{k \cdot \bar{\tau_{cf}}}{\sqrt{k + k \cdot (k-1) \cdot \bar{r_{ff}}}}, \qquad (18)$$

where, k is the number of features, $\bar{r_{cf}}$ is the average correlation between features and the class label, and $\bar{r_{ff}}$ is the average inter-feature correlation. This subset is then further evaluated using methods like cross-validation.

It is important to note that the CFS employs a heuristic search strategy within its multivariate FS algorithm to pinpoint optimal attributes in a given dataset. The criteria for selection are anchored in the correlation strength and statistical significance between a feature and its associated category. This unique capability has solidified CFS's role as a go-to method for Feature Extraction (FE), especially in large-scale data environments. Moreover, CFS has yielded numerous impactful findings that contribute to elevating the efficacy of Decision-Making System (DMS).

The advantages of CFS are manifold. It tends to produce more superficial and interpretable models by decreasing the feature size, thus mitigating the risk of overfitting. However, the method is not without limitations. For example, Pearson's correlation, which is commonly used, assumes a linear relationship between variables and does not capture feature interactions. Despite this, CFS remains a powerful FSM, aiming to optimize the model's performance and generalization capabilities. When integrated with techniques like Regularized Random Forest (RRF), CFS can further enhance the FSM, leveraging the regularization capabilities of RRF to produce an even more robust and interpretable model.

IV. PROPOSED FSM

In the architecture of the proposed FSM, as shown in Fig. 1, four key steps seamlessly integrate to provide a holistic solution. Initially, the dataset undergoes a preprocessing phase, which includes tasks like data normalization, formatting, and randomization, preparing the data for rigorous analysis. Following preprocessing, the first significant phase employs the innovative BFS-RST Adaptive Algorithm to reduce the feature set effectively. Utilizing this algorithm allow for a focus on a subset of features that are most relevant to the task, thereby enhancing the model's efficiency. This reduced feature set serves as the input to the second crucial phase, which features the Dynamic Correlated Regularized Random Forest (DCRRF) application. DCRRF refines FS dynamically, optimizing performance and interpretability through a combination of Correlation-based Feature Selection (CFS) and Regularized Random Forest (RRF) methodologies. After the optimal feature set has been identified, the final step involves a data analysis phase where the effectiveness of the selected features is rigorously tested using a Support Vector Machine (SVM) classifier. This multi-layered approach enhances the feature selection process and lends itself to detailed performance evaluation, making it a comprehensive solution for complex data analysis scenarios.

A. Data Preprocessing

The first step in the proposed FSM is Data Preprocessing. This phase is crucial because it converts the raw dataset into a more manageable, clean form, making it easier to analyze and feed into subsequent FR and FS stages. A properly preprocessed dataset not only streamlines the FSM but also contributes to the robustness and interpretability of the resulting model. The following methods are used in the preprocessing pipeline of the proposed architecture:

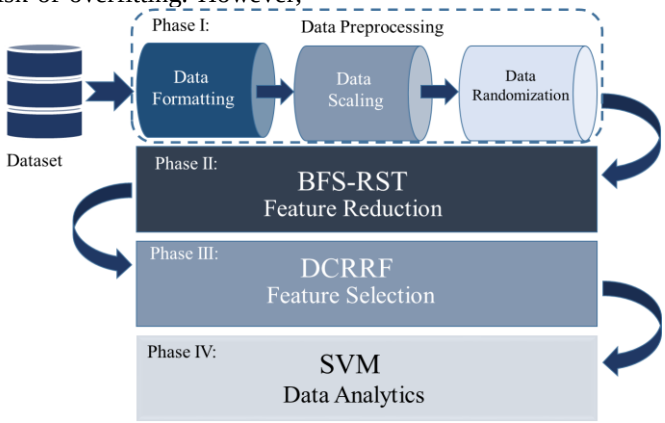

Fig. 1. Proposed FSM.

- Data Formatting: Including Data Formatting as the initial step in the preprocessing pipeline is fundamental for making the dataset readable and interpretable by the following algorithms. Addressing issues like inconsistent data types and missing values eliminates the potential for errors or biases that could significantly affect the performance of the BFS-RST Adaptive Algorithm and DCRRF. Proper formatting is a stable foundation for the stages that follow in the feature selection model.
- Data Scaling: Data Scaling finds its place in the preprocessing phase due to its significant impact on any ML algorithm's performance. The scales of numerical features can vary widely, and a feature with a more extensive range could overshadow those with smaller ranges, leading to suboptimal FS or model training. Scaling standardizes this, ensuring that each feature is given equal importance, thereby enhancing the overall reliability and effectiveness of the BFS-RST and DCRRF algorithms.
- Data Randomization: Data Randomization is incorporated to mitigate any sequence-based biases in the dataset. Data points can come in sequences that may reflect various forms of underlying structure or bias, such as time-based or class-based ordering. Shuffling the order of data points enables the FSM, which employs both the BFS-RST and DCRRF algorithms, to learn more objectively, uninfluenced by the sequence in which the data points initially appear.

B. Adaptive Feature Reduction (AFR) using BFS and RST

FR is a vital process in ML pipelines, as it aims to cut down on the data dimensions without significantly affecting the model's performance. While several algorithms aim to do this, each has advantages and disadvantages. BFS is known for its ability to traverse the feature space optimally but can be computationally expensive. On the other hand, RST provides a formal framework to identify indispensable features but can be heuristic and computationally intensive for calculating reducts.

An adaptive approach that combines the strengths of both algorithms is thus conceived to achieve effective FR. The rationale is to employ RST's capability to identify core attributes indispensable for the DMS and then use BFS to navigate the feature space efficiently. In the given dataset $\mathcal{D}$, each feature subset $S \subseteq \mathcal{A}$ is a potential candidate for FR. These subsets are represented as nodes in the search space that BFS navigates. An evaluation function $f(S)$ is used to assess the "quality" of each subset S, analogous to how each node in a traditional BFS comes with an associated cost or value.

1) Initialization using core attributes from RST: Rough Set Theory first identifies a set of core attributes $\mathcal{R}_{\text{core}}$ from $\mathcal{A}$. These are the features that are indispensable for maintaining discernibility among the classes in $\mathcal{D}$ as shown in Eq. (19)

$$\mathcal{R}_{\text{core}} = \{a \in \mathcal{A} \mid a \text{ is indispensable for discernibility }\} \quad (19)$$

2) Evaluation function in BFS: The evaluation function $f(S)$ used in BFS combines a cost function $g(S)$ and a heuristic $h(S)$ to guide the search. $g(S)$ could represent how well S performs in terms of model accuracy or any other metric, and $h(S)$ is a heuristic estimate of the "distance" to the optimal feature subset, see Eq. (20).

$$f(S) = \omega_1 \cdot g(S) + \omega_2 \cdot h(S) \quad (20)$$

Here, ω_1 and ω_2 are weight parameters.

3) Priority assignment using RST: During the BFS traversal, RST is used to identify if a subset S is a reduct minimal set of features with discernibility power comparable to $\mathcal{A}$. Such subsets are flagged for higher priority in the BFS queue.

$$\text{Priority}(S_i) = \begin{cases} \alpha \cdot f(S_i), & \text{If } S_i \text{ is a reduct} \\ f(S_i), & \text{Otherwise} \end{cases} \quad (21)$$

In Eq. (21), $\alpha < 1$ is a factor that lowers the evaluation function $f(S_i)$ for reducts, they are effectively giving them a higher priority in the queue. By methodically integrating RST for initial setup and ongoing evaluation with the traversal capabilities of BFS, the algorithm aims to find an optimal and minimal feature subset from $\mathcal{A}$ for dataset $\mathcal{D}$. In the proposed algorithm, the focus is on reducing features by generating child nodes with fewer attributes, followed by an evaluation of their effectiveness using both BSF and RST techniques. Feature sets that neither improve nor degrade the quality of the model will be pruned. With this understanding now established, Algorithm 3, shown below, illustrates the steps involved:

Algorithm 3 for BFS-RST based on Adaptive Feature Reduction

Input:

- $\mathcal{F}$*: Complete set of features*
- $f(S)$*: Evaluation function for a feature subset S*
- $\mathcal{C}$ *: Stopping Criteria (e.g., a set limit on the number of features)*

Output:

- $\mathcal{S}_{opt}$ *: Optimal reduced set of features*

  Initialize:

  - *Compute the core attributes* $\mathcal{R}_{core}$ *using RST. These are the features that are indispensable for the discernibility of classes.*
  - *Initialize priority queue* Q *with* $\mathcal{R}_{core}$ *as the root node, evaluated by* $f(\mathcal{R}_{core})$.
  - *Algorithm Steps:*
    - *While* Q *is not empty and* $\mathcal{C}$ *is not met, Do*
      - *Dequeue: Pop the node S with the lowest* $f(S)$ *from* Q.

- *Check Stopping Criteria:*
- *If S satisfies $\mathcal{C}$, Then*
- *Return $\mathcal{S}_{opt} = S$*
- *Exit*
- *FR and Child Node Generation:*
  - *Remove one feature at a time from S to create smaller subsets $S_1, S_2, \dots, S_n$*
  - *For Each S_i , if S_i is a reduct according to RST, flag it as a high-priority node.*

Evaluate and Enqueue:

- ***For Each** Child Node S_i, Compute $f(S_i)$*
- ***If** S_i is flagged as high-priority, Adjust $f(S_i)$ to reflect its importance.*
- *Insert S_i into Q*
- *Re-sort Priority Queue: Sort Q based on $f(S)$.*

In Step 3.3.1, each child node S_i has one less feature than its parent S. This is where FR is explicitly done. Here, Rough Set Theory is used for two purposes:

(i) It provides a robust starting point $\mathcal{R}_{core}$ that contains indispensable features, ensuring that the essential features are not eliminated in the initial stages.

(ii) It helps to flag high-priority nodes (reducts) during the FR process, guiding the algorithm toward a more meaningful feature subset.

The BFS evaluates these smaller feature sets and prioritizes them in the queue. If a reduced feature set satisfies the stopping criteria, it is output as the optimal set of features $\mathcal{S}_{\text{opt}}$. In essence, this algorithm combines the strengths of both RST and BFS to perform feature reduction in a more effective and informed manner.

C. Dynamic Correlated Regularized Random (DCRRF)

DCRRF is a novel hybrid model that aims to combine the strengths of Correlation-based Feature Selection (CFS) and Regularized Random Forest (RRF) to optimize FS and improve model performance dynamically. By incorporating CFS into the training of each tree within the RRF ensemble, DCRRF aims to maximize model robustness and interpretability. The model takes a reduced feature set D' as input from the BFS-RST Adaptive algorithm. This feature set is standardized or normalized to make feature values comparable using the following steps:

1) Standardization: In the standardization process, every attribute is adjusted to have a zero mean and a unit standard deviation. This becomes particularly crucial when dealing with features in disparate units or varying in scale. To standardize a given feature x, a commonly used mathematical EQU (22) is typically employed.

$$\text{Standardized}(x) = \frac{x - \text{mean}(x)}{\text{std}(x)} \quad (22)$$

2) Normalization: In normalization, the features are typically scaled to lie in a given range $[0,1]$. This is often beneficial when the algorithm involves distance metrics or when the feature has a skewed distribution. Normalization of a feature x is generally achieved by Eq. (23):

$$\text{Normalized}\ (x) = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (23)$$

The normalized feature set is the foundation for the subsequent feature selection process in the DCRRF model. For each tree T_m in the ensemble, a distinct bootstrap sample D'_m is chosen from this processed feature set. A correlation matrix $\text{Corr}(D'_m)$ is then computed for each of these samples, expressed as in Eq. (24).

$$\text{Corr}(D_m) = \begin{pmatrix} \text{corr}(X_1, X_1) & \text{corr}(X_1, X_2) & \cdots & \text{corr}(X_1, X_n) \\ \text{corr}(X_2, X_1) & \text{corr}(X_2, X_2) & \cdots & \text{corr}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \text{corr}(X_n, X_1) & \text{corr}(X_n, X_2) & \cdots & \text{corr}(X_n, X_n) \end{pmatrix} \quad (24)$$

Using the CFS criterion Cr, a tailored feature subset F_m is dynamically selected for each tree T_m. The Cr_m The criterion is calculated as Eq. (25).

$$Cr_m \frac{k_m \cdot \bar{\tau_{cf_m}}}{\sqrt{k_m + k_m \cdot (k_m - 1) \cdot \bar{r_{ff_m}}}}, \quad (25)$$

where, k_m is the number of features in F_m, $\bar{\tau_{cf_m}}$ is the average correlation between features and the class label for D'_m, and $\bar{r_{ff_m}}$ is the average inter-feature correlation for D'_m. This criterion Cr_m is used to select an optimal subset of features F_m for each T_m. After this dynamic FS, each tree T_m is trained using its respective selected feature subset F_m. The training process adopts the regularized objective function which is shown in Eq. (26).

$$\left(T_{m,F_m}\right) = \text{Impurity}\left(T_{m,F_m}\right) + \lambda \text{Complexity}\left(T_{m,F_m}\right) \quad (26)$$

Importantly, this objective function is indexed with F_m to signify the dynamically chosen features for that specific tree. The optimal feature set F^* is then determined by an intersection operation over all the dynamically selected feature subsets F_m. This can be formally expressed as in Eq. (27).

$$F^* = F_1 \cap F_2 \cap \dots \cap F_M \quad (27)$$

The dynamic FS introduces diversity among the individual trees, making the ensemble model more resilient and adaptable. It also enables optimized FS, thereby potentially improving both performance and interpretability. The following Algorithm 4 presents the steps involved in the proposed FSM.

D. Algorithm 4 for DCRRF for Feature selection

Input:

- *Reduced Feature set D'*
- *Number of trees M*
- *Regularization parameter λ*

Output

- *Optimal set of features F^**
    - *Data Preprocessing*
    - *Normalize or standardize the Feature set D'.*
- *Initialize Feature Set*
- *Initialize feature set $F^* = \emptyset$.*

Ensemble Training and FS

- *For $m = 1$ to M do the following:*
- *Bootstrap Sample*
- *Draw a bootstrap sample D'_m from D'.*
- *Calculate Correlation Matrix*
- *Compute $Corr(D'_m)$ for the bootstrap sample D'_m.*
- *FS with CFS*
- *Compute the CFS criterion Cr_m using $Corr(D'_m)$.*
- *Select the optimal feature subset F_m based on Cr_m.*
- *Train Regularized RF Tree*
- *Use F_m and D'_m to train a tree T_m with the objective function:*
  $$Objective(T_{m,F_m}) = Impurity(T_{m,F_m}) + \lambda\, Complexity(T_{m,F_m})$$
- *Update Feature Aggregation*
- *Update F^* based on F_m using an intersection operation: $F^* = F^* \cap F_m$.*
- *Determine the Optimal Feature Set*
- *Return F^* as the optimal feature set that captures the most important features across the ensemble.*

D. Data Analysis

The data analysis phase serves as the final phase of the FSM's pipeline. This phase is significant because it provides the final validation of the feature sets that have been carefully reduced and selected through the preceding stages. The focus here is on evaluating these feature sets within the specific context of the problem, be it classification, clustering, or some other form of ML task. For the purpose of this paper, the efficacy of the proposed FR and FS model is examined using a SVM classifier. The reason for choosing SVM for analysis is twofold. First, SVMs are known for their effectiveness in high-dimensional spaces, making them a suitable choice for testing the quality of the FS. Second, SVMs are robust to overfitting, especially in cases where the number of dimensions is greater than the number of samples, further validating the quality of the FS. The features that have passed through the BFS-RST Adaptive Algorithm and the DCRRF are fed into the SVM model. Performance metrics such as accuracy, precision, recall, and F1-score are computed to evaluate the classifier's performance on the selected feature sets.

V. Experimental Analysis

A. Dataset and Implementation

In the current research, experiments were conducted on four gene expression datasets analyzed by [59], namely: i) Prostate [60], ii) Brain [61], iii) NCI60 [62] and iv) Adenocarcinoma [63]. The specifics regarding the number of instances and attributes for each dataset are detailed in Table. I. All methods and experimental procedures were executed in a Jupyter Notebook environment, utilizing the Python 3.6 language. Computations and tests were carried out on a system equipped with a Windows 10 operating system, powered by a 2.8GHz AMD Ryzen 5 processor, and supplemented by 8GB RAM. Various stages of data processing, feature selection, and machine learning implementations leveraged pre-existing software libraries.

The datasets mentioned above are partitioned into an 80:20 ratio for the purposes of training and evaluation. The SVM model is calibrated using specific hyperparameter settings, as shown in Table II, for optimal performance.

The regularization parameter C is set to 1 to maintain a compromise between maximizing the margin and minimizing the classification error. The Radial Basis Function (RBF) kernel is chosen for its ability to handle both linear and nonlinear patterns in the data [64-73]. The model undergoes 100 iterations during training to ensure convergence and optimal performance. The performance of the proposed feature selection model is compared with RF-based baseline models such as i) Boruta, ii) RRF, iii) VSURF and iv) GRRF. The effectiveness of the SVM, when employing each feature mentioned above, FSM, is assessed through metrics such as accuracy, sensitivity, specificity, precision, and F-score. The results achieved by all the models for the listed performance metrics are shown in Table III.

TABLE I. Dataset Description

Dataset	Instance	Attribute	Class
Prostate	102	6033	2
Brain	42	5597	5
nci60	61	5244	8
Adenocarcinoma	76	9868	2

TABLE II. SVM Learning Parameters

Hyperparameter	Specific Value
Regularization C	1
Kernel Type	RBF
Number of Iterations	100

TABLE III. PERFORMANCE COMPARISON FOR DIFFERENT BASELINES AGAINST FOUR DATASETS

Models	FS	Accuracy	Sensitivity	Specificity	Precision	F1-score
Prostate Dataset (No. of Features: 6033)						
Boruta	96	0.9306	0.9242	0.9558	0.9579	0.9407
RRF	88	0.9479	0.9413	0.9558	0.9581	0.9496
VSURF	78	0.9385	0.9356	0.9421	0.9466	0.9411
GRRF	18	0.9488	0.9518	0.9557	0.9572	0.9497
DCRRF	29	0.9514	0.9582	0.9559	0.9582	0.9534
BFSRST+ DCRRF	12	0.9544	0.9589	0.9593	0.9625	0.9562
Brain Dataset (No. of Features: 5597)						
Boruta	83	0.8761	0.8464	0.8990	0.8464	0.8464
RRF	97	0.8861	0.8236	0.9342	0.8797	0.8507
VSURF	56	0.9060	0.8693	0.9342	0.8882	0.8787
GRRF	22	0.8960	0.8693	0.9166	0.8693	0.8693
DCRRF	28	0.9073	0.8802	0.9292	0.8986	0.8894
BFSRST+ DCRRF	19	0.9126	0.9096	0.9114	0.8910	0.9003
NCI60 Dataset (No. of Features: 5244)						
Boruta	93	0.8794	0.9004	0.9257	0.9660	0.8408
RRF	197	0.9157	0.8823	0.9414	0.8823	0.8823
VSURF	83	0.8878	0.7351	0.9549	0.9485	0.8283
GRRF	63	0.9233	0.8598	0.9680	0.9075	0.8830
DCRRF	58	0.9283	0.8765	0.9405	0.9008	0.8885
BFSRST+ DCRRF	53	0.9317	0.8960	0.9317	0.8960	0.8960
Adenocarcinoma Dataset (No. of Features: 9868)						
Boruta	143	0.8757	0.8027	0.9532	0.9485	0.8695
RRF	86	0.9098	0.8933	0.9274	0.9290	0.9108
VSURF	106	0.8784	0.7844	0.9781	0.9712	0.8679
GRRF	20	0.9057	0.9268	0.9785	0.9768	0.9030
DCRRF	36	0.9101	0.8977	0.9367	0.9274	0.8992
BFSRST+ DCRRF	14	0.9249	0.9130	0.9387	0.9405	0.9179

In both the Prostate and Brain datasets, as shown in Fig. 2 and Fig. 3, DCRRF demonstrates superior performance across multiple metrics. For the Prostate dataset, DCRRF achieves an accuracy of 0.9514, edging out the second-best model, RRF, by 0.37%. It also excels in sensitivity with a score of 0.9582, which is notably higher than RRF's 0.9413. Regarding specificity and precision, DCRRF performs on par with RRF and GRRF, highlighting its balanced efficiency in identifying True Negatives (TN) and minimizing False Positives (FP). The F1-score for DCRRF is the highest at 0.9534, and it further improves to 0.9562 when augmented with BFS-RST, all while requiring only 12 selected features. For the Brain dataset, DCRRF again leads in accuracy and sensitivity, with scores of 0.9073 and 0.8802, respectively. While its specificity score of 0.9292 is not the highest, it still indicates a balanced performance compared to RRF's higher specificity but lower sensitivity. In the precision metric, DCRRF is slightly edged out by VSURF but still performs strongly with a score of 0.8986. Its F1-score stands at 0.8894, and when combined with BFS-RST, it further improves to 0.9003, again achieving this with fewer features. These metrics collectively indicate that DCRRF, particularly when enhanced with BFS-RST, offers balanced, efficient, and robust performance across both datasets in FS and classification tasks.

In the NCI60 dataset, as shown in Fig. 4, DCRRF stands out with an accuracy of 0.9283, outperforming the next-best model, RRF, which scores 0.9157. While its sensitivity score of 0.8765 isn't the highest, it's balanced by a strong specificity of 0.9405. The model's F1-score is 0.8885, which is superior to both Boruta's 0.8408 and RRF's 0.8823. Its precision score

of 0.9008 is commendable, though it is slightly eclipsed by Boruta's 0.9660. Notably, when integrated with BFS-RST, the model's F1-score rises to 0.8960 with a reduced feature count of 53. In the Adenocarcinoma dataset, as shown in Fig. 5, DCRRF maintains its strong performance with an accuracy of 0.9101, closely followed by RRF at 0.9098. DCRRF shines in sensitivity with a score of 0.8977, substantially better than Boruta's 0.8027 and slightly edging out RRF's 0.8933. With well-rounded scores in specificity (0.9367) and precision (0.9274), it also maintains a balanced F1-score of 0.8992. When enhanced by BFS-RST, the F-score improves to 0.9179 with just 14 FS, demonstrating the model's efficiency and efficacy in FS and classification.

In a comprehensive review of the results for OOB error, time consumption and AUC efficiently, as shown in Fig. 6 to Fig. 8, BFSRST+DCRRF consistently delivers outstanding performance across all datasets, excelling in AUC and minimizing OOB errors. For instance, in the Prostate dataset, this model achieves the highest AUC of 0.893, using the fewest features (12) and an OOB error of just 0.11. The computational time, although slightly higher than its DCRRF counterpart, remains modest at 0.06 minutes. Similarly, in the Brain and NCI60 datasets, BFSRST+DCRRF again tops the chart in AUC, recording 0.911 and 0.914, respectively, while maintaining low OOB errors and computational times. On the Adenocarcinoma dataset, it achieves an AUC of 0.902, leading the pack. VSURF performs well but is computationally expensive, particularly noticeable in the Prostate and Adenocarcinoma datasets, where the computational times are 0.08 and 0.1 minutes, respectively. DCRRF alone also shows promise, particularly in the NCI60 and Adenocarcinoma datasets, where it nearly matches the performance of its BFSRST-enhanced version but with more features. Boruta and RRF, although competent, generally lag in AUC and OOB error metrics. Notably, GRRF consistently demands fewer features but doesn't offer a compelling trade-off regarding AUC or OOB error. The BFSRST+DCRRF model demonstrates superior, balanced performance across all four datasets.

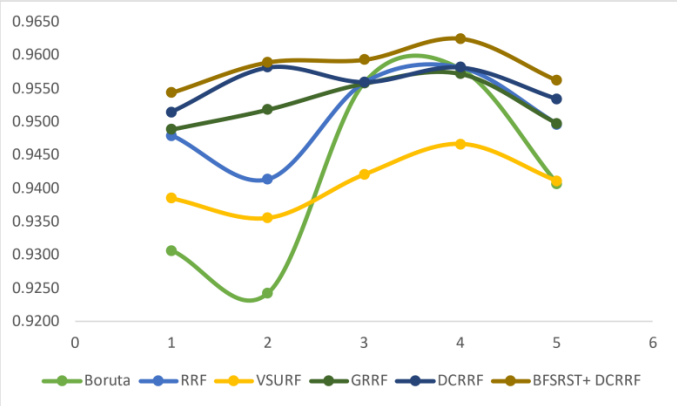

Fig. 2. Performance comparison for prostate dataset.

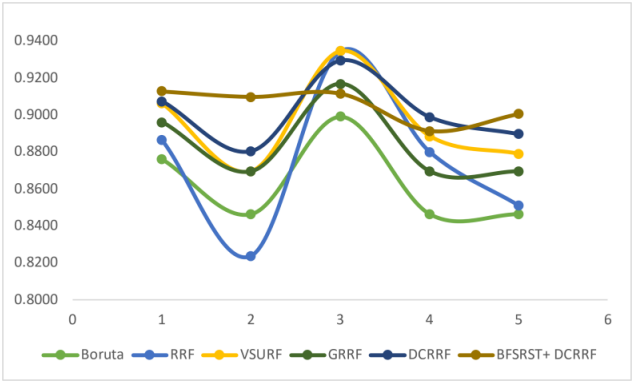

Fig. 3. Performance comparison for brain dataset.

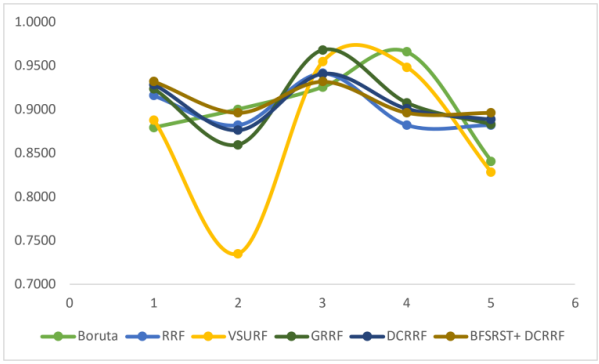

Fig. 4. Performance comparison for NCI60 dataset.

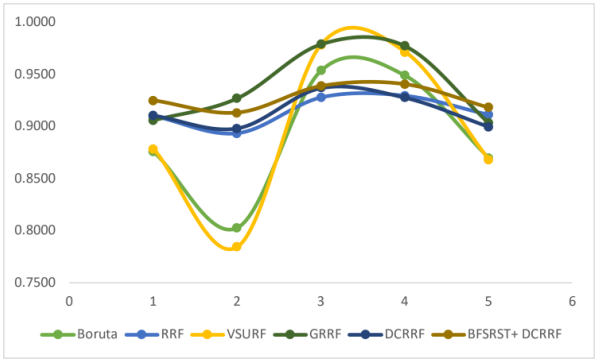

Fig. 5. Performance comparison for adenocarcinoma dataset.

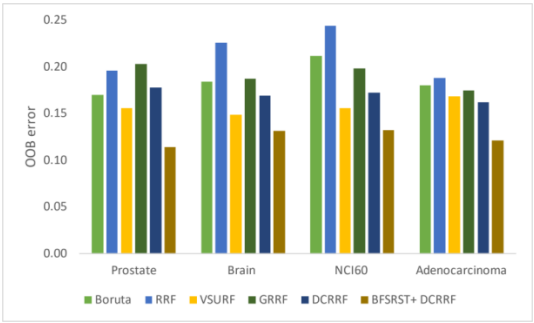

Fig. 6. OOB error comparison.

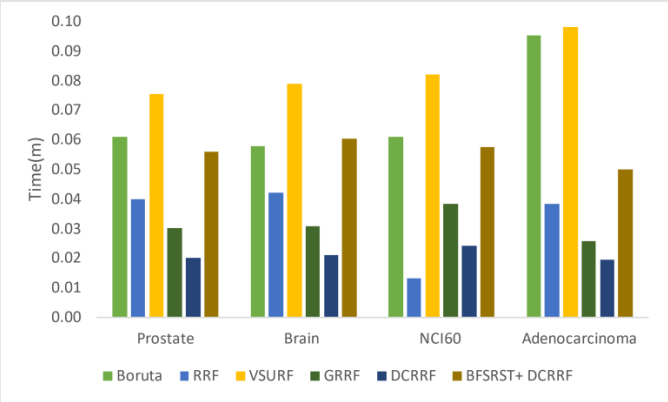

Fig. 7. FS-time comparison.

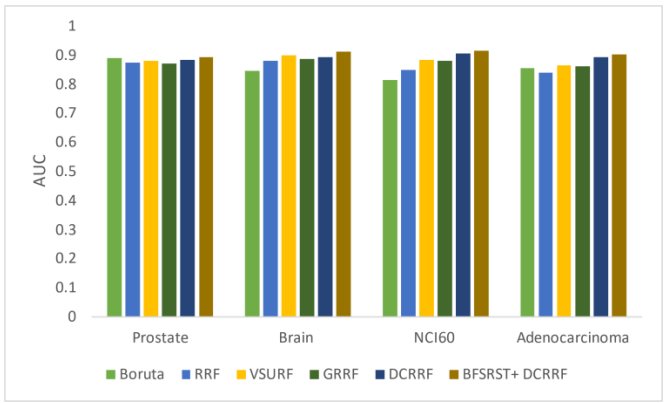

Fig. 8. AUC comparison.

VI. Conclusion

Handling big data with high dimensions presents unique challenges, particularly regarding computational resources and predictive accuracy. To address these issues, an all-encompassing Feature Selection Model (FSM) has been developed. This system incorporates initial data cleaning and feature reduction through Best-first Search and Rough Set Theory (BFS-RST). It culminates in deploying a specialized Random Forest (RF) algorithm called Dynamic Correlated Regularized Random Forest (DCRRF). Each stage of this four-tiered architecture serves a specific function, from initial data refinement to advanced FSM. The final assessment phase employs a Support Vector Machine (SVM) classifier to evaluate the quality and utility of the selected features rigorously. When tested against existing RF-based FSM on four gene expression datasets, this innovative approach improved computational efficiency and classification accuracy. The system's enhanced performance indicates its potential as a scalable solution for tackling the unique challenges presented by high-dimensional big data across various applications.

Funding Statement

This work was supported by Yunnan Provincial Education Department Scientific Research Fund Project. (2022J0494).

References

[1] E. Debie and K. Shafi, Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses. Pattern Analysis and Applications, vol. 22, pp. 519-536, 2019.

[2] S. Shi, Y. Xu, X. Xu, X. Mo and J. Ding, A Preprocessing Manifold Learning Strategy Based on t-Distributed Stochastic Neighbor Embedding. Entropy, vol. 25, no. 7, pp:1065, 2023.

[3] C. Kern, T. Klausch and F. Kreuter, Tree-based Machine Learning Methods for Survey Research. Surv Res Methods, vol. 13, no. 1, pp. 73-93, 2019.

[4] F. Tang and H. Ishwaran, Random forest missing data algorithms. Statistical Analysis and Data Mining: The ASA Data Science Journal, vol. 10, no. 6, pp. 363-377, 2017.

[5] S. Georganos, T. Grippa, S. Vanhuysse, M. Lennert, M. Shimoni, S. Kalogirou and E. Wolff, Less is more: Optimizing classification performance through feature selection in a very-high-resolution remote sensing object-based urban application. GIScience & remote sensing, vol. 55, no. 2, pp. 221-242, 2018.

[6] M. B. Kursa and W. R. Rudnicki, Feature selection with the Boruta package. Journal of statistical software, vol. 36, pp. 1-13, 2010.

[7] T. Hothorn, K. Hornik and A. Zeileis, Unbiased recursive partitioning: A conditional inference framework. Journal of Computational and Graphical Statistics, vol. 15, no. 3, pp. 651-674, 2006.

[8] H. Deng and G. Runger, Feature selection via regularized trees. International Joint Conference on Neural Networks (IJCNN), pp. 1-8, 2012.

[9] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees. Machine learning, vol. 63, pp. 3-42, 2006.

[10] M. A. M. Hasan, M. Nasser, S. Ahmad and K. I. Molla, Feature selection for intrusion detection using random forest. Journal of information security, vol. 7, no. 3, pp. 129-140, 2016.

[11] M. Huljanah, Z. Rustam, S. Utama and T. Siswantining, Feature selection using random forest classifier for predicting prostate cancer. In IOP Conference Series: Materials Science and Engineering, vol. 546, no. 5, p. 052031, IOP Publishing, 2019.

[12] Y. Wang, Y. Xu, Z. Yang, X. Liu and Q. Dai, Using recursive feature selection with random forest to improve protein structural class prediction for low-similarity sequences. Computational and Mathematical Methods in Medicine, 2021.

[13] A. Behnamian, K. Millard, S. N. Banks, L. White, M. Richardson and J. Pasher, A systematic approach for variable selection with random forests: achieving stable variable importance values. IEEE Geoscience and Remote Sensing Letters, vol. 14, no. 11, pp. 1988-1992, 2017.

[14] M. Rostami and M. Oussalah, A novel explainable COVID-19 diagnosis method by integration of feature selection with random forest. Informatics in Medicine Unlocked, vol. 30, no. 100941, 2022.

[15] E. V. Sylvester, P. Bentzen, I. R. Bradbury, M. Clément, J. Pearce, J. Horne and R. G. Beiko, Applications of random forest feature selection for fine-scale genetic population assignment. Evolutionary Applications, vol. 11, no. 2, pp. 153-165, 2018.

[16] X. Li, W. Chen, Q. Zhang and L. Wu, Building auto-encoder intrusion detection system based on random forest feature selection. Computers & Security, vol. 95, no. 101851, 2020.

[17] H. Fei, Z. Fan, C. Wang, N. Zhang, T. Wang, R. Chen and T. Bai, Cotton classification method at the county scale based on multi-features and random forest feature selection algorithm and classifier. Remote Sensing, vol. 14, no. 4, pp. 829, 2022.

[18] Z. Pawlak, Rough set theory and its applications to data analysis. *Cybernetics & Systems*, vol. *29*, *no.* 7, pp. 661-688, 1998.

[19] T. Kavitha *et al.*, 'Deep Learning Based Capsule Neural Network Model for Breast Cancer Diagnosis Using Mammogram Images', *Interdisciplinary Sciences – Computational Life Sciences*, vol. 14, no. 1, pp. 113–129, 2022.

[20] S. Sengan, O. I. Khalaf, P. Vidya Sagar, D. K. Sharma, L. Arokia Jesu Prabhu, and A. A. Hamad, 'Secured and privacy-based IDS for healthcare systems on e-medical data using machine learning approach', *International Journal of Reliable and Quality E-Healthcare*, vol. 11, no. 3, 2022.

[21] S. Namasudra, R. Chakraborty, A. Majumder, and N. R. Moparthi, 'Securing Multimedia by Using DNA-Based Encryption in the Cloud Computing Environment', *ACM Transactions on Multimedia Computing, Communications and Applications*, vol. 16, no. 3s, 2021.

[22] K. K. D. Ramesh, G. Kiran Kumar, K. Swapna, D. Datta, and S. Suman Rajesh, 'A review of medical image segmentation algorithms', *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 7, no. 27, 2021.

[23] A. Naik, S. C. Satapathy, and A. Abraham, 'Modified Social Group Optimization—a meta-heuristic algorithm to solve short-term hydrothermal scheduling', *Applied Soft Computing Journal*, vol. 95, 2020.

[24] N. Satheesh *et al.*, 'Flow-based anomaly intrusion detection using machine learning model with software-defined networking for OpenFlow network', *Microprocessors and Microsystems*, vol. 79, 2020.

[25] K. N. Dattatraya and K. R. Rao, 'Hybrid based cluster head selection for maximizing network lifetime and energy efficiency in WSN', *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 3, pp. 716–726, 2022.

[26] N. Yuvaraj, T. Karthikeyan, and K. Praghash, 'An Improved Task Allocation Scheme in Serverless Computing Using Gray Wolf Optimization (GWO) Based Reinforcement Learning (RIL) Approach', *Wireless Personal Communications*, vol. 117, no. 3, pp. 2403–2421, 2021.

[27] C. Sridhar, P. K. Pareek, R. Kalidoss, S. S. Jamal, P. K. Shukla, and S. J. Nuagah, 'Optimal Medical Image Size Reduction Model Creation Using Recurrent Neural Network and GenPSOWVQ', *Journal of Healthcare Engineering*, vol. 2022, 2022.

[28] S. Deshmukh, K. Thirupathi Rao, and M. Shabaz, 'Collaborative Learning Based Straggler Prevention in Large-Scale Distributed Computing Framework', *Security and Communication Networks*, vol. 2021, 2021.

[29] P. K. Pareek *et al.*, 'IntOPMICM: Intelligent Medical Image Size Reduction Model', *Journal of Healthcare Engineering*, vol. 2022, 2022.

[30] S. Mishra, L. Jena, H. K. Tripathy, and T. Gaber, 'Prioritized and predictive intelligence of things enabled waste management model in smart and sustainable environment', *PLoS ONE*, vol. 17, no. 8, 2022.

[31] S. Stalin *et al.*, 'A Machine Learning-Based Big EEG Data Artifact Detection and Wavelet-Based Removal: An Empirical Approach', *Mathematical Problems in Engineering*, vol. 2021, 2021.

[32] C. Banchhor and N. Srinivasu, 'Integrating Cuckoo search-Grey wolf optimization and Correlative Naive Bayes classifier with Map Reduce model for big data classification', *Data and Knowledge Engineering*, vol. 127, 2020.

[33] K. Saikumar, V. Rajesh, and B. S. Babu, 'Heart Disease Detection Based on Feature Fusion Technique with Augmented Classification Using Deep Learning Technology', *Treatment du Signal*, vol. 39, no. 1, pp. 31–42, 2022.

[34] S. D. M. Achanta, T. Karthikeyan, and R. Vinoth Kanna, 'A wireless IoT system towards gait detection technique using FSR sensor and wearable IoT devices', *International Journal of Intelligent Unmanned Systems*, vol. 8, no. 1, pp. 43–54, 2020.

[35] S. Sengan, G. R. K. Rao, O. I. Khalaf, and M. R. Babu, 'Markov mathematical analysis for comprehensive real-time data-driven in healthcare', *Mathematics in Engineering, Science and Aerospace*, vol. 12, no. 1, pp. 77–94, 2021.

[36] V. Kumar *et al.*, 'Addressing Binary Classification over Class Imbalanced Clinical Datasets Using Computationally Intelligent Techniques', *Healthcare (Switzerland)*, vol. 10, no. 7, 2022.

[37] S. Kumar, A. Jain, A. Kumar Agarwal, S. Rani, and A. Ghimire, 'Object-Based Image Retrieval Using the U-Net-Based Neural Network', *Computational Intelligence and Neuroscience*, vol. 2021, 2021.

[38] M. S. Mekala and P. Viswanathan, '(t,n): Sensor Stipulation with THAM Index for Smart Agriculture Decision-Making IoT System', *Wireless Personal Communications*, vol. 111, no. 3, pp. 1909–1940, 2020.

[39] P. Sharma, N. R. Moparthi, S. Namasudra, V. Shanmuganathan, and C.-H. Hsu, 'Blockchain-based IoT architecture to secure healthcare system using identity-based encryption', *Expert Systems*, vol. 39, no. 10, 2022.

[40] S. Joshi *et al.*, 'Unified Authentication and Access Control for Future Mobile Communication-Based Lightweight IoT Systems Using Blockchain', *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[41] M. Baskar, J. Ramkumar, C. Karthikeyan, V. Anbarasu, A. Balaji, and T. S. Arulananth, 'Low rate DDoS mitigation using real-time multi-threshold traffic monitoring system', *Journal of Ambient Intelligence and Humanized Computing*, 2021.

[42] A. V. N. Reddy, C. P. Krishna, and P. K. Mallick, 'An image classification framework exploring the capabilities of extreme learning machines and artificial bee colony', *Neural Computing and Applications*, vol. 32, no. 8, pp. 3079–3099, 2020.

[43] C. Banchhor and N. Srinivasu, 'Integrating Cuckoo search-Grey wolf optimization and Correlative Naive Bayes classifier with Map Reduce model for big data classification', *Data and Knowledge Engineering*, vol. 127, 2020.

[44] V. Talasila, K. Madhubabu, M. C. Mahadasyam, N. J. Atchala, and L. S. Kande, 'The prediction of diseases using rough set theory with recurrent neural network in big data analytics', *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 10–18, 2020.

[45] S. P. Praveen, T. B. Murali Krishna, C. H. Anuradha, S. R. Mandalapu, P. Sarala, and S. Sindhura, 'A robust framework for handling health care information based on machine learning and big data engineering techniques', *International Journal of Healthcare Management*, 2022.

[46] C. Banchhor and N. Srinivasu, 'FCNB: Fuzzy Correlative Naive Bayes Classifier with MapReduce Framework for Big Data Classification', *Journal of Intelligent Systems*, vol. 29, no. 1, pp. 994–1006, 2020.

[47] C. Banchhor and N. Srinivasu, 'Analysis of Bayesian optimization algorithms for big data classification based on Map Reduce framework', *Journal of Big Data*, vol. 8, no. 1, 2021.

[48] A. D. Jadhav and V. Pellakuri, 'Highly accurate and efficient two phase-intrusion detection system (TP-IDS) using distributed processing of HADOOP and machine learning techniques', *Journal of Big Data*, vol. 8, no. 1, 2021.

[49] A. V. Brahmane and C. B. Krishna, 'Rider chaotic biography optimization-driven deep stacked auto-encoder for big data classification using spark architecture: Rider chaotic biography optimization', *International Journal of Web Services Research*, vol. 18, no. 3, pp. 42–62, 2021.

[50] K. Jammalamadaka and N. Parveen, 'Testing coverage criteria for optimized deep belief network with search and rescue', *Journal of Big Data*, vol. 8, no. 1, 2021.

[51] S. K. Funde and G. Swain, 'Security aware information classification in health care big data', *International Journal of Electrical and Computer Engineering*, vol. 11, no. 5, pp. 4439–4448, 2021.

[52] S. Roy, B. Patel, D. Bhattacharyya, K. Dhayal, T.-H. Kim, and M. Mittal, 'Demographical gender prediction of Twitter users using big data analytics: An application of decision marketing', *International Journal of Reasoning-based Intelligent Systems*, vol. 13, no. 2, pp. 41–49, 2021.

[53] C. Banchhor and N. Srinivasu, 'Holoentropy based Correlative Naive Bayes classifier and MapReduce model for classifying the big data', *Evolutionary Intelligence*, vol. 15, no. 2, pp. 1037–1050, 2022.

[54] C. Banchhor and N. Srinivasu, 'Grey Wolf Shuffled Shepherd Optimization Algorithm-Based Hybrid Deep Learning Classifier for Big Data Classification', *International Journal of Swarm Intelligence Research*, vol. 13, no. 1, 2022.

[55] S. Funde and G. Swain, 'Big Data Privacy and Security Using Abundant Data Recovery Techniques and Data Obliviousness Methodologies', *IEEE Access*, vol. 10, pp. 105458–105484, 2022.

[56] C. Banchhor and N. Srinivasu, 'A comprehensive study of data intelligence in the context of big data analytics', *Web Intelligence*, vol. 20, no. 1, pp. 53–66, 2022.

[57] A. V. Brahmane and B. C. Krishna, 'Big data classification using deep learning and Apache spark architecture', *Neural Computing and Applications*, vol. 33, no. 22, pp. 15253–15266, 2021.

[58] A. V. Brahmane and B. C. Krishna, 'DSAE-Deep Stack Auto Encoder and RCBO-Rider Chaotic Biogeography Optimization Algorithm for Big Data Classification', *Advances in Parallel Computing*, vol. 39, pp. 213–227, 2021.

[59] R. Díaz-Uriarte and S. Alvarez de Andrés, Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, vol. *7*, pp. 1-13, 2006.

[60] D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd and W. R. Sellers, Gene expression correlates of clinical prostate cancer behaviour. *Cancer Cell*, vol. *1*, no. 2, pp. 203-209, 2022.

[61] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin and T. R. Golub, Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, vol. *415*, *no.* 6870, pp. 436-442, 2002.

[62] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman and P. O. Brown, Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics*, vol. *24*, *no.* 3, pp. 227-235, 2000.

[63] S. Ramaswamy, K. N. Ross, E. S. Lander and T. R. Golub, A molecular signature of metastasis in primary solid tumours. *Nature Genetics*, vol. *33*, no. 1, pp. 49-54, 2003.

[64] P. Sharma, N. R. Moparthi, S. Namasudra, V. Shanmuganathan, and C.-H. Hsu, 'Blockchain-based IoT architecture to secure healthcare system using identity-based encryption', *Expert Systems*, vol. 39, no. 10, 2022.

[65] S. Joshi *et al.*, 'Unified Authentication and Access Control for Future Mobile Communication-Based Lightweight IoT Systems Using Blockchain', *Wireless Communications and Mobile Computing*, vol. 2021, 2021.

[66] P. Chithaluru, F. Al-Turjman, T. Stephan, M. Kumar, and L. Mostarda, 'Energy-efficient blockchain implementation for Cognitive Wireless Communication Networks (CWCNs)', *Energy Reports*, vol. 7, pp. 8277–8286, 2021.

[67] N. Yuvaraj, K. Praghash, R. A. Raja, and T. Karthikeyan, 'An Investigation of Garbage Disposal Electric Vehicles (GDEVs) Integrated with Deep Neural Networking (DNN) and Intelligent Transportation System (ITS) in Smart City Management System (SCMS)', *Wireless Personal Communications*, vol. 123, no. 2, pp. 1733–1752, 2022.

[68] M. Z. U. Rahman, S. Surekha, K. P. Satamraju, S. S. Mirza, and A. Lay-Ekuakille, 'A Collateral Sensor Data Sharing Framework for Decentralized Healthcare Systems', *IEEE Sensors Journal*, vol. 21, no. 24, pp. 27848–27857, 2021.

[69] S. Sekar *et al.*, 'Autonomous Transaction Model for E-Commerce Management Using Blockchain Technology', *International Journal of Information Technology and Web Engineering*, vol. 17, no. 1, 2022.

[70] S. R. Dasari, S. Tondepu, L. R. Vadali, and N. Seelam, 'PEG-400 mediated an efficient eco-friendly synthesis of new isoxazolyl pyrido[2,3-d] pyrimidines and their anti-inflammatory and analgesic activity', *Synthetic Communications*, pp. 2950–2961, 2020.

[71] S. Rajasoundaran *et al.*, 'Secure watchdog selection using intelligent key management in wireless sensor networks', *Materials Today: Proceedings*, 2021.

[72] N. V. Rani and K. Ravindhranath, 'PEG-400 promoted a simple, efficient and eco-friendly synthesis of functionalized novel isoxazolyl pyrido[2,3-d]pyrimidines and their antimicrobial and anti-inflammatory activity', *Synthetic Communications*, vol. 51, no. 8, pp. 1171–1183, 2021.

[73] A. Bhattacharjya, 'A Holistic Study On The Use Of Blockchain Technology In CPS And IoT Architectures Maintaining The Cia Triad In Data Communication', *International Journal of Applied Mathematics and Computer Science*, vol. 32, no. 3, pp. 403–413, 2022.