Explainable Machine Learning for Data Extraction Across Computational Social System

Hemanta Kumar Bhuyan and Chinmay Chakraborty, *Senior Member, IEEE*

***Abstract*— This article addresses the explainable machine learning for data extraction on diverse datasets. In many cases, individual or specific approaches have been developed for feature selection (FS) on a certain dataset, but collecting the diversity dataset and demonstrating it through different FS methods are challenging. Thus, this article proposed multiapproaches for FS with the classification of diverse datasets. The proposed framework is developed using various methods, such as extendable particle swarm optimization (PSO), global and local searching, feature ranking, feature clustering, computational cost-based FS, and multiobjective optimization. We effectively used these methods in our proposed work in a single-setting framework. We focused on three essential computational items in our framework: classification accuracy, selected features, and computational times. Due to the diverse dataset, few methods have been considered challenging during computational evaluation for classification accuracy with test cost. We tried to manage the classification accuracy based on total cost and high accuracy with less cost. The proposed framework is experimented with the above methods and analyzed through comparative results on diversity datasets. For example, when regular parameter values are in the range of 2^{-13}–2^{-6}, the evaluation result affects all items, i.e., decreasing during this range; other values do not affect results. We used thresholds ranging from 0.6 to 0.9 for highly correlated feature pairs as per the support vector machine (SVM) method for recursive feature elimination.**

***Index Terms*— Classification, computational cost, computational social system, data extraction, explainable machine learning, multiobjective optimization.**

I. Introduction

In THE social system, various social needs have been identified as per public demands, such as healthcare, agriculture, water, and communication issues. Although all issues are impossible to solve immediately, few issues can be solved through data analysis using machine learning approaches. Since extracting features from different social datasets is part of data analysis, we considered various data analyses through the feature selection (FS) process. Different approaches have been used in FS to discover a trivial characteristic from an extensive feature set. Features in classification are selected to reduce data dimension, and the arrangement process can be hasted up.

Manuscript received January 6, 2022; revised February 28, 2022; accepted March 24, 2022. *(Corresponding author: Chinmay Chakraborty.)*

Hemanta Kumar Bhuyan is with the Department of Information Technology, Vignan's Foundation for Science, Technology & Research (Deemed to be University), Guntur 522213, India.

Chinmay Chakraborty is with the Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra, Jharkhand 835215, India (e-mail: cchakrabarty@bitmesra.ac.in).

Digital Object Identifier 10.1109/TCSS.2022.3164993

Finding the best feature set may be accomplished by employing a search technique with testing that assesses the excellent performance of different feature sets. In most cases, FS methods are categorized into filter, wrapper, and embedding methods to evaluate many datasets [1] based on the evaluation measure. In wrapper methods, each feature subset is trained with a classifier as part of a wrapper technique to assess the performance of the chosen features. However, despite being computationally costly, wrapper techniques can generate feature subsets as per specific classification methods with the objective approaches for prediction. They evaluate feature subsets based on information, consistency, and other metrics [1]. Filter techniques are frequently computationally less expensive than wrapper approaches and are more generic in their applicability. Generally, the filter approaches are efficiently searched across the feature domain, while wrappers provide well accuracy in evaluation. It is possible to acquire strong prediction performance using embedded approaches based on training. The most common criteria for evaluating the degree of dependency are correlation measurements between the feature and the class.

Although various approaches are used for FS based on the above approach, they have not considered a generic approach for any dataset with effective computational cost. Thus, we considered multiple approaches with test costs as follows.

1) We considered the most potent global search methods and the least expensive to run compared to the evolutionary computation (EC) algorithms using particle swarm optimization (PSO) [2]–[4]. However, per [5], it reveals that nonstop PSO outperforms binary PSO for FS compared to [6]. For improving classification accuracy, the proposed work is to design a fresh approach of PSO for FS. For this, the global best (gbest) approach is subjected to an iterative local search that mimics a standard backward elimination (BE) procedure using: 1) employing a mutual information (MI) approach; 2) utilizing the position value's additional information; and 3) focusing the testing of tiny subsets instead of the entire set of features. We focused on experiments and compared result analysis of advanced PSO algorithms on different datasets.
2) The large-volume data constitute a significant obstacle to supervised learning applications. We provide a new method called structured feature ranking (SFR) to select supervised features from large, high-dimensional datasets. First, we present a subspace feature clustering (SFC) approach to discover feature-based clusters as

per feature value-related class. The SFC consumes the class labels to extend the subspace weighting coclustering SWCC) as [7]. As per closed feature clusters associated with others, we suggest a structured feature weighting approach to determine the highest-ranking features, which are also informative and diverse. We ran tests on both fictitious and real-world data to assess how well our methods worked.

3) Classification is a significant part of data mining, which deals with the relationship between feature sets and class labels. Complications associated with the feature acquisition process are called computational expense in fields such as image analysis [8]. When it comes to classification, our goals are twofold: a sensitive computational FS (SCFS) framework makes it possible to maximize classification accuracy and minimize total test cost. Generally, the existing FS approaches try to select the best feature subset to recover classifier performance; however, SCFS attempts to strike a stability between classification performance and the computational cost of the tests as per performance.

A considerable amount of data is generated due to the integration of numerous methods. It is impossible to overstate the magnitude of this data, which provides unparalleled chances to investigate complex social behaviors ranging from the spread of infectious diseases to socioeconomic disparities. Big data (both structured and unstructured) are a key concern in this area because there is an excess of it and a scarcity of management approaches to deal with it. Despite extensive study into big data technologies, accessing data from computational social systems continues to be a technical hurdle to overcome. The process of discovering underlying patterns in social data beyond object-based generalization to some external knowledge is called pattern learning in social data. When it comes to computational social systems, it is precisely at this point that knowledge-infused learning makes a difference. It provides an efficient method of integrating information taken from various data sources. Several essential technologies, including machine learning, PSO, the BE approach, MI, and others, are used to attain these goals.

The primary objective of the proposed work is given as follows: 1) our proposed framework is to examine and progress the result of nonstop PSO for classification; 2) the SFR select supervised features from high-dimensional datasets; and 3) classification and total cost are reciprocal to each other with good performance, i.e., increasing accuracy with decreasing computational cost. As per the proposed approach, we use an optimization problem and propose a new general SCFS wrapper framework. Specifically, we introduce a novel term of testing function for the wrapper method to account for the test costs of FS processing.

Thus, the significant contribution of the proposed work is given as follows: 1) we use the PSO algorithm for both smaller and larger sets of features for classification accuracy; 2) we use different PSO-based algorithms and compare their performance; 3) we designed SFR to select features from the dataset using feature-based cluster and SWCC approaches; and 4) we also developed an SCFS model with possible, increasing accuracy and decreasing computational cost.

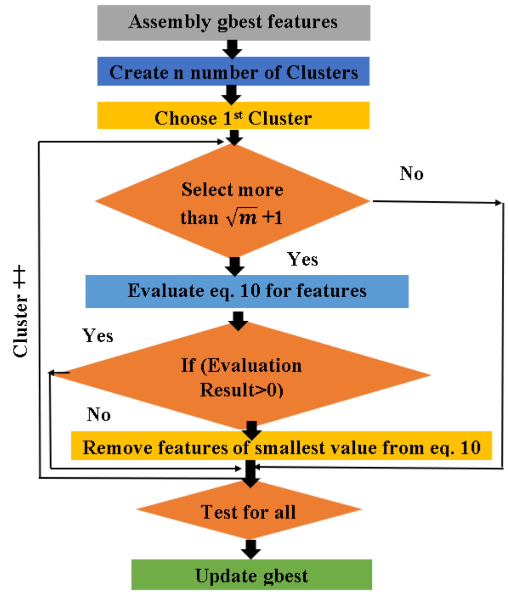

Fig. 1. Process of BE.

The remaining part of this article is arranged in different sections as follows. We considered the relative work of this article in Section II. The proposed methodology related to FS approaches is explained in Section III. This section considered a multiple-approach framework for FS. Section IV has designed a structural feature ranking method. Section V elaborates on cost-based FS. The evaluation performance with analysis is explained in Section VI. The performance analysis on clustered-based FS is given in Section VII. The result analysis on computational cost-based FS is elaborated on in Section VIII. The wrapper method using forward selection and recursive feature elimination (RFE) is explained in Section IX. The overall discussion is mentioned in Section X with future work. This article is concluded in Section XI.

II. Background

We considered the related work to find out the disadvantages and difficulties of earlier models or frameworks from different resources.

A. Conventional Techniques for Selecting Feature Sets

Instead of selecting only the most essential features, a feature ranking method is considered for FS-based classification with specific criteria [1]. It looks for the highest ratings of features to pick the best features. Some selected features may be redundant in a highly correlated dataset, making this approach ineffective. Once updated on any features, it is impossible to reupdate next time, called nesting. Solving this problem, forward and backward time-based FS approaches are considered. Still, identifying the values of this parameter is difficult. Two types of floating selection methods were proposed: 1) sequential backward and 2) forward floating selections. Later, a linear forward selection (LFS) approach was suggested [9]. The gradient descent- and privacy preservation-based FSs are designed in [10] and [11]. By limiting the number of features for most cases, computational costs can be kept low, while the classification accuracy remains high. Few predictive and cryptic techniques are also used in data mining, which assists in selecting features from the dataset [12]–[14].

B. Computational Approaches for Feature Selection

Genetic algorithms (GAs) and genetic programming (GP) were utilized to obtain various solutions for FS problems [15]. The results showed that classification accuracy was improved, while the number of features was reduced substantially. According to Zhu *et al.* [15], this approach incorporates GA, and a local search has been proposed for FS. Filter measures and classification performance are used to occupy all particle locations and make a collaborative algorithm for FS [6]. This method generated good performance than PSO according to the results of the experiments. This approach is compared to any wrapper algorithm, which is usually better at classifying than a filter algorithm. The multiswarm PSO method solves FS issues. As a result of its use of multiple subswarms with many particles, the proposed algorithm is computationally intensive. Several methods have been developed for new gbest changing approaches to improve PSO performance. Lin *et al.* [16] suggested a PSO with the support vector machine (SVM) approach for selecting wrapper features. This algorithm aims to optimize the SVM parameters and look for the best feature set simultaneously.

C. Structured Feature Weighting Method for Supervised Feature Selection

There have been many approaches for selecting features from the feature set, eliminating unnecessary features, and improving performance. Filter, wrapper, and embedded approaches are the three main categories of FS methods. The filter methods have no learning algorithm, which chooses feature subsets based on the data's essential appearances. Relieve-F in [17] is one of the most common supervised filtering methods [18]. Such methods can be time-consuming because they treat forecasting the performance with the help of different objective components [19]. The training process for embedded methods includes FS. It has become increasingly popular to use embedded methods better than others [5]. Text mining, bioinformatics, and recommendation systems are just a few areas where it has recently been used. It has been suggested that hierarchical and several clusterings, such as spectral and partitional coclusterings [20], are the best models for coclustering. To deal with high-dimensional data, FS has become increasingly significant [21]–[23].

D. Cost Testing for Feature Selection

The different statistical methods are used to evaluate and remove unimportant features from a dataset. It is common for wrappers to use the expected feature subset based on its performance with various features. These algorithms involve different searching methods, including greedy search and random [24] searches to find the best feature subset. Even though wrappers are more precise, the computational burden that they impose is prohibitive. During the computational process, embedded methods use FS, which is only applicable to a particular machine. RFE achieves FS and extracts feature subset using the SVM method. The RFE concept was used by Archibald and Fann [25] to create a new procedure for picking features from image data. The computational cost for FS is designed using the wrapper framework in [26].

Feature acquisition time and space complexity are used for computational expense [8]. Instead of focusing solely on the accuracy, the computational cost evaluation approach was used to balance accuracy efficiency and measuring cost. For the best decisions, both the computational cost and the accuracy of the classification must be taken into consideration. Despite the abundance of prior work on FS, less approach was taken with a multiobjective optimization problem. In this case, the weight of the test cost could be controlled by entering the parameter manually. The wrapper model-based test cost FS has been designed by Jiang *et al.* [26].

E. Mechanism for Computational Social System

We explained few approaches related to the social system with the proposed mechanism as follows. Different mobile apps are used through Social Internet of Things (SIoT) systems, which assists to distribute various message records and regular events, such as health, incidents, and sports [27]. Soil and environmental characteristics are used in crop cultivation, which is analyzed by machine learning techniques, such as the RFE technique with several classifiers [28]. Different types of smart devices, such as smart mobiles and smart wearable devices, are used in the human-in-the-loop (HITL) system [29] as users' requirements. Different user connectivity and collaborative computation offloadings for developing smart cities endorse citizen life [30]. Various depression intensity problems of any person have been issued in present society. Few authors find out its reason and analyze it with a deep learning approach [31].

III. Multiple-Approach Framework for Feature Selection

We have tried to find out the generic approach for many applications on different datasets. Although different approaches have been developed for FS and classification, they used specific methods for certain datasets. Thus, we considered the following approaches to make a common elementary approach for the diversity of research work.

A. Particle Swarm Optimization Method

Though the PSO approach has already developed, its comprehensive approach is considered for our framework. As per the fundamentals of swarm intelligence, PSO is motivated by collective attributes of a community, such as fish grouping and birds grouping [2], [3]. A particle is used to represent a potential solution in PSO. A group of particles works together to get the appropriate solutions in the available space. Thus, the movement of a particle is designed as follows. When a particle (i) moves, its position and velocity are defined by x_i and v_i in the searching space D as per dimension, respectively. We considered the two best solutions for the movement of a particle. It is possible for each particle to remember its personal best (pbest) and the prior visited by all swarms considered as gbest. The PSO approach iteratively changes x_i

and υ_i of each particle to find the best solution by formulating the following equations:

$$\upsilon_{\text{id}}^{t+1} = w * \upsilon_{\text{id}}^{t} + c_1 * r_{i1} * \left(p_{\text{id}} - x_{\text{id}}^{t}\right) + c_2 * r_{i2} * \left(p_{\text{gd}} - x_{\text{id}}^{t}\right) \tag{1}$$

$$x_{\text{id}}^{t+1} = x_{\text{id}}^{t} + \upsilon_{\text{id}}^{t+1} \tag{2}$$

where particle i's velocity in the $(t+1)$th iteration is $\upsilon_{\text{id}}^{t+1}$, where υ is considered as the velocity of particles in the $(t+1)$th iteration in the d dimension. Position x_{id}^{t+1} and weight w recognize the influence of the previous velocity on the current position of particle i. There are two acceleration constants $c1$ and $c2$ in this equation. The rand() function returns random values that are evenly distributed between 0 and 1. There are two ways to look at this data: p_{id} and p_{gd} are used in the dth dimension for two solutions: local best position (pbest) and gbest position.

Since the PSO model has the limitations of the existing approaches, we try to extend the PSO-based model approaches to avoid the limitation of this model. Thus, we have made the proposed model to find different solutions based on multiobjective approaches. Initially, we considered the particle swarm optimization backward elimination (PSOBE) algorithm for the proposed model as Algorithm 1 that depicts its overall structure. We use two kinds of particles for PSO position. The particle occupies its pbest and the gbest. The computational complexity of Algorithm 1 is $O(n^2)$. The BE procedure on gbest is omitted in PSOBE. As a result of PSOBE, the information of each particle is encrypted as an array of numbers, with each element's value denoted by the $x_i = (x_{i1}, x_{i2}, \ldots, x_{\text{id}}, \ldots, x_{i\text{D}})$,. The dth feature's probability will be selected as $0 \leq x_{\text{id}} \leq 1$. This feature is selected or deselected based on a threshold value θ. If $\theta \leq x_{\text{id}}$, the dth feature will be enabled for selection; otherwise, it would not use the dth feature. For lessening the evaluation error for classification, the fitness function in PSOBE is defined as (3) based on a wrapper measure

$$\text{Error Rate} = \frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{3}$$

where FP, FN, TP, and TN stand for false positives, false negatives, true positives, and true negatives. To mimic the distinctive BE algorithm, a "BE" process is designed [33]. For gbest, the BE seeks the best solution while improving search capability and preventing algorithm inactivity. Well evaluation and good performance are two essential components in developing this BE. The position's value shows how the corresponding features will be selected. Thus, BE seeks to utilize position value information even further. The following consecutive sections described, in detail, the BE with its performance.

Algorithm 1 PSOBE

```
Input: Data set, parameters in PSO;
Output: gbest; Overall classification accuracies.
Begin
  Start with initial value of the position and velocity of each particle;
  If (iteration ≠ Max) then
     compute fitness for each particle as eq. 3
   Based on PopulationSize
     modify the pbest of particle i; and
     modify the gbest
   Based on Dimensionality, do
     modify v_id as Eq. 1;
     modify x_id as Eq. 2;
     compute "BE" on gbest;
End
   Evaluate the classification accuracy on the test set;
  return the position of gbest, the training and test classification
   accuracies;
End
```

B. Method of Backward Elimination

The BE method must be simple computation for needed features because wrapper-based FS methods have no low evaluation performance. Thus, we considered the filter approach using MI in this article. In addition, PSOBE is expected to benefit from both filter and wrapper methods by implementing this filter measure. Generally, the MI method is used to select features based on the variables such as both features and the class label [12], [33]. We considered the set of features to be F (f_i€$|F|$), and the class label is c. The relevance measurement between F and c is shown in (5) and can show the contribution of f_i to the relevance as (6) where the calculation of $I(f_i; c)$ is defined in (4). For (4), the reader can refer [34]

$$I(f_i; c) = -\sum_{i \in |F|} p(f_i, c) \log_2 \frac{p(f_i, c)}{p(f_i) p(c)} \tag{4}$$

$$\text{Rel}(F) = \sum_{f_i \in F} I(f_i; c) \tag{5}$$

$$\Delta\text{Rel}(f_i) = I(f_i; c). \tag{6}$$

Furthermore, the redundancy in feature set F is determined in (7) as

$$\text{Red}(F) = \sum_{f_i, f_j \in F} I(f_i, f_j). \tag{7}$$

Equation (5) is also brought by f_i

$$\Delta\text{Red}(f_i) = \sum_{f_i, f_j \in F;\quad f_{i \neq} f_j} I(f_i, f_j). \tag{8}$$

We have generated a good feature set by the difference between $\text{Rel}(F)$ and $\text{Red}(F)$. Thus, we proposed the best feature set as $B(f_i)$ in the following equation:

$$B(f_i) = \Delta\text{Rel}(f_i - \frac{1}{|F| - 1} \Delta\text{Red}(f_i)). \tag{9}$$

For BE, the feature has the largest redundancy and smallest relevance. A higher value of (9) represents a better value. Removing a feature with the lowest $B(f_i)$ value will help BE because it has the most remarkable redundancy and the smallest amount of relevance. f_i should not be removed if $B(f_i) \geq 0$ because its relevance is more significant or at least equal to its redundancy when $B(f_i) \geq 0$; the relevance of feature f_i is larger. Thus, feature f_i is removed only when $B(f_i) < 0$, and $B(f_i)$ is the smallest value in F, i.e., all other features in F are reduced to their minimum values, which is the case here because $B(f_i) \leq B(f_j)$, where $j \neq i$.

C. Based on Position Value

The BE considers the position value to optimize the data value, which contains probability information. Equation (9) is changed to form a new equation by joining the position value of f_i (x_i) as (10). In this case, the value is 1.0 because the relevant feature has been selected. Equation (10) adds x_i to ensure that, if the same value has happened for two features as (9), it neglects the lesser position value (i.e., more negligible value) from F. Because of this, f_i can only be eliminated if $B'(f_i) < 0$ and $B'(f_i) \leq B'(f_j)$, where $j = 1, 2, \ldots, |F|$ and $j \neq i$

$$B'(f_i) = \frac{1}{x_i} * (\Delta\text{Rel}(f_i) - \frac{1}{|F|-1}\Delta\text{Red}(f_i)). \quad (10)$$

D. Measurement of Be

As per the BE method, the number of features will need to remove during the FS process. We considered the dynamic values that are determined by gbest's preferences for BE. Using a clustering approach, similar features are first grouped into several clusters and then into clusters within the same cluster to solve the problem. Instead of using gbest's features, the BE is applied to each cluster individually. According to expected performance, the elimination of a single feature does not significantly impact the classification performance if many features are chosen from a single cluster. Thus, the BE is performed on a subset of the gbest-selected features for each cluster (i.e., F used above).

If features are selected greater than $\sqrt{m} + 1$ by gbest (i.e., $|F| > \sqrt{m}$). This ensures that, even if gbest only uses a small portion of the features available, the essential data will be preserved. When many features are chosen, removing one would not affect classification accuracy. However, (10) will reduce one feature as per the performance of BE in the same cluster. Equation (10) on smaller feature clusters will decrease the computation costs on the extensive feature set.

Fig. 1 depicts the BE process flowchart in the diagram form. While there are many steps in this BE, its evaluation is quicker than a classifier assessment using the wrapper approach. The algorithm will be guided to look for small feature subsets if the quantity of features in gbest is reduced, and this will take less time as per the reduction of features. Thus, as per the BE approach, the reduction of features creates good classification performance and reduces the computational cost on gbest accordingly.

IV. Structural Feature Ranking Method

We have considered the structural feature ranking approaches for choosing features. We developed a cluster with high correlated features and chose features from the feature set with the help of a cluster to minimize redundancy. Chen *et al.* [7] had made earlier clusters based on highly correlated features in subspace.

The structured ranking approach is proposed for FS in this article. The new method begins by grouping the features into a set of feature clusters, which will be considered for feature ranking as per the method. A structured weighting ranking method makes a list of features as their ranking as per SFR and feature clusters, which is proposed to obtain a concluding list of features as per ranked from various feature clusters.

We proposed the SFR method's steps for analyzing selected features, as shown in Fig. 2. To find the disjoint feature clusters, we first group the labeled dataset X with n number of

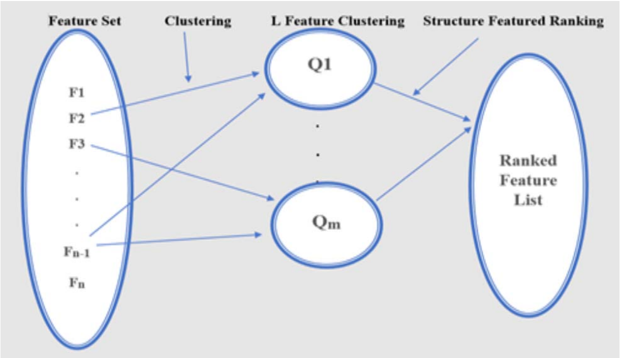

Fig. 2. Process of the structured featured ranking method.

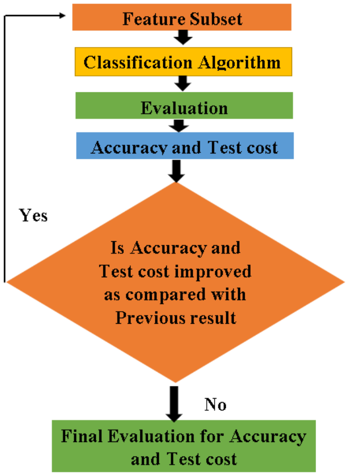

Fig. 3. Wrapper framework for SCFS.

features as $F = \{f_1, \ldots, f_n\}$ into a number of disjoint feature clusters as $\{Q_1, \ldots, Q_m\}$, where $Q_j \cap Q_i = \Phi(\forall i \neq j)$ and $\bigcup_{j=1}^{m} Q_j = F$ Finally, we use a structured weighting feature ranking method to determine how the n features should be ranked. We will go over feature clustering and structured weighting in these sections as follows.

A. Optimized Features

Let the labeled data matrix $X \in R^{\text{nxm}}$ contain n instances and m features. We follow a subspace weight matrix (SWM) $C \in R^{\text{kxl}}$ as [40], in which c_{gj} is considered as the weight of the cluster with the jth column and the gth row. We made cluster X with k row and l column for the labeled data matrix. We also developed the objective function based on SWCC as [40] in the following equation:

$$\min_{U,V,Z,C} \frac{1}{mn} \sum_{g=1}^{k} \sum_{h=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ig} v_{jh} c_{gj} (x_{i,j} - z_{g,h})^2 + \frac{\eta}{m} \sum_{g=1}^{k} \sum_{j=1}^{m} c_{gi} \log c_{gi}$$

$$\text{s.t.} \sum_{g=1}^{k} u_{ig} = 1, \quad u_{ig} \in \{0, 1\}$$

$$\sum_{h=1}^{l} v_{jh} = 1, \quad v_{jh} \in \{0, 1\}$$

$$\sum_{j=1}^{m} c_{gj} = 1, \quad c_{gj} \in \{0, 1\}. \quad (11)$$

Based on the class label in X, it is possible to construct $U \in R^{\text{nxk}}$ with known class labels by setting $u_{ig} = 1$ when $x_i \in g$th class and 0 otherwise. This is called supervised FS. The feature clustering process's goal is to group n features in cluster X and create l number of feature clusters. Getting this goal, we considered SFC to generate the objective function as (12) from (11)

$$\min_{V,Z,C} \frac{1}{mn} \sum_{g=1}^{k} \sum_{h=1}^{l} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ig} v_{jh} c_{gj} (x_{i,j} - z_{g,h})^2 + \frac{\eta}{m} \sum_{g=1}^{k} \sum_{j=1}^{m} c_{gi} \log c_{gi}$$
$$\text{s.t.} \sum_{h=1}^{l} v_{jh} = 1, \quad v_{jh} \in \{0, 1\}$$
$$\sum_{j=1}^{m} c_{gj} = 1, \quad c_{gj} \in \{0, 1\}. \tag{12}$$

Equation (12) has the approximately same solution as (11) of V, Z, and C. As per [23], we can find the solution of V, Z, and C from (12), which are explained as in the following. When Z and C are static, V can be solved as follows:

$$\begin{cases} v_{jh} = 1, \quad \text{if } P_{(h)} \leq P_{(t)} \text{ for } 1 \leq t \leq L \text{ where} \\ P_{(t)} = \sum_{g=1}^{k} \sum_{i=1}^{n} u_{ig} c_{gj} (x_{ij} - z_{gt})^2 \\ v_{jt} = 0, \quad \text{for } t \neq h. \end{cases} \tag{13}$$

When both V and C are stable, Z can be defined as follows:

$$z_{gh} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} u_{ig} v_{jh} c_{gj} x_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{m} u_{ig} v_{jh} c_{gj}}. \tag{14}$$

If both Z and V are stable, the optimal solution for C is

$$c_{gj} = \frac{\exp\left\{-\frac{E_{gj}}{\eta}\right\}}{\sum_{j'=1}^{m} \exp\left\{-\frac{E_{gj'}}{\eta}\right\}} \tag{15}$$

where

$$E_{gi} = \frac{1}{n} \sum_{h=1}^{l} \sum_{i=1}^{n} u_{ig} v_{jh} (x_{ij} - z_{gt})^2. \tag{16}$$

Based on the above equations, we developed Algorithm 2 for the workflow of feature clustering ranking based on the subspace weighted matrix.

Algorithm 2 predicts the comprehensive information to (12) and also updates V, Z, and C cyclically until convergence is achieved. Because we are getting closer and closer to the local minima of (12), the optimization process is strictly decreasing to local minima. SFC's computational complexity is O (rnmkl), when the algorithm converges after r iterations. Using k-means, we can see that clustering extensive high-dimensional data can be efficient because the computational cost of SFC is proportional to the number of features and records. The SFC algorithm searches dissimilar starting clusters with the center to provide several feature clusters. Then, we run SFC with different initial cluster centers for each of l and η to provide a coclustering output $\hat{H}$, run with $H \in \hat{H}$, and choose $H^* \in \hat{H}$ as well clustering performance. Each object's label is predicted by applying the learned V^*, Z^*, and C^* to determine the class to which it belongs. This is done by placing the object in the class with the least weighted distance from the others

$$\text{label}(x_i) = \arg\min_{g} \left[\sum_{h=1}^{l} \sum_{j=1}^{m} v_{jh}^* c_{gj}^* \left(x_{ij} - z_{gh}^*\right)^2 \right]. \tag{17}$$

After that, various evaluation indices, such as accuracy, recall, and others, can be used to assess the accuracy of the classification output getting from H. The user typically specifies the number of feature clusters l to use. The best coclustering result can be chosen from a set of multiple coclustering results by selecting multiple l.

Algorithm 2 SFC

1: Input: Dataset X, l- feature clusters and regularized parameter η.
2: Output: SFC outcome V and the SWM C.
3: Make a binary matrices $U \in R^{n \times k}$ from class labels, where $u_{ig} = 1$ for the i^{th} feature with the g^{th} class.
4: i initialize 0 with arbitrary Z
5: Assume $c_{gj} = 1/m$ for $\forall$ g and j.
6: iterate
7: Compute V^{i+1} using (13).
8: Compute Z^{i+1} using (14).
9: Compute C^{i+1} using (15) and (16).
10: i++
11: upto(12) gets local least value

B. Structured Weighting Feature Ranking

Since each feature's contribution to each class is identified by the learned weight matrix C in H^*, ranking the features to C is logical. As a result, a projection matrix $W \in R^{\text{mxk}}$ is learned, and the importance of the features can be estimated as $\{||w^1||_2, \ldots, ||w^m||_2\}$ using the least-squares regression method. Due to the non-negative nature of C in SFC, we can assess the relative importance of various features using the formulas $\{||c_1||_1, \ldots, ||c_m||_1\}$. The nominated features are involved in a few feature clusters with well correlation if it chooses r high-rank features as C, which can select r import features. We proposed a ranked weighting method for ranking features to choose features with well performance. The new method sorts feature in feature clusters according to $\{||c_1||_1, \ldots, ||c_m||_1\}$; then, it uses in ascending order to sort the remaining features in the clusters in reverse order. Considering that the jth feature index as per feature cluster is l_j, we execute a weighted feature ranking vector $\theta \in R^m$ as per the jth feature index with cluster l_j as follows:

$$\theta_j = ||c_j||_1 \lambda^{lj} \tag{18}$$

where the user specifies the weighting parameter $\lambda \in (0, 1]$. In this case, the weights in a feature cluster are geometrically decreased using λ^{lj}. To degenerate into the traditional ranking approach, set $\lambda = 1$, and use θ_j. If $\lambda < 1$, it geometrically decreased weights as per the number of features in a cluster, deemphasizing features with the lower order. As a result, selecting various features from a feature cluster will be avoided. Consequently, using the criteria, we can choose features from the cluster as per θ.

C. High-Performance Ranking

Algorithm 3 summarizes the detailed procedure above. First, we use SFC to create l disjoint feature clusters from m features in X using the new method. Finally, we use a structured weighting feature ranking method to determine how the m features should be ranked. The computational cost in the algorithm is $O(n^2)$.

Algorithm 3 SFR

1: Input: Dataset X, l- feature clusters, the parameter η, the structured weighting parameter λ and repetition of clusters rep.
2: Start with clustering output list $\boldsymbol{H}$ =0.
3: For j: 1 to rep do
4: Evaluate SFC(X, l, η) and start cluster centers H with results
5: Combine H into $\boldsymbol{H}$.
6: end for
7: Authenticate the output H $\in \boldsymbol{H}$, and choose H* $\in$ R with the best clustering output
8: Evaluate the l_1-norm of C and arrange a sorting approach for features from the feature cluster as their values.
9: Evaluate $\theta \in R^{m\times 1}$ as eq. (18).
10: Arrange top to bottom as top r ranked features with the help of θ.

V. Cost-Based Feature Selection

We further considered the computational cost of evaluating different FS approaches. Evaluating and selecting the most cost-effective filter method are used as correlation-based FS (CFS) [35]. Using cost-sensitive CFS (CSCFS) as [8], a new evaluation function is proposed to account for the total cost of testing the selected features while keeping the original CFS evaluation function intact. The CS-CFS assesses the value of k features from subset S as follows:

$$\mathrm{MC}_s = \frac{k\bar{r_{ci}}}{\sqrt{k+k(k-1)\bar{r_{ii}}}} - \lambda\frac{\sum_{i=1}^{k} C_i}{k}. \tag{19}$$

To calculate MC_S (total cost of k selected features of subset S), we need to evaluate (19), where k = selected features from subset S, $\bar{r_{ci}}$ is mean correlation between feature class in S, $\bar{r_{ii}}$ is the mean correlation among features in S, C_i is the ith test cost in S, and λ is used as a weight for the cost function. According to the observation, CS-CFS performed better with higher values of λ, but the overall test cost is lower, and the classification accuracy decreased. We used $\lambda = 1$ for managing classification accuracy with the test cost. We use additional terms for test cost based on the FS approach and consider a filter framework within SCFS. When comparing the objective of the total cost, Kohavi and John [36] show better performance on wrapper than the filter method because they optimize a classifier during FS processing.

However, there has been very little progress in developing a wrapper approach for SCFS. Thus, this framework aims to make an enhanced SCFS wrapper framework in a general form to make sure that our wrapper framework will work with good validation.

Fig. 3 depicts the wrapper framework for SCFS as per CS-CFS and framework [36]. SCFS has two primary goals: 1) classification accuracy must be improved and 2) test costs must be reduced. Thus, we consider that the SCFS issue is a multiobjective optimization issue. Multiobjective optimization problems can be solved by turning them into single-objective problems. The CS-CFS incorporates the total test cost as a new term when evaluating filter FS methods. To counter this, we came up with the idea of creating a modified wrapper approach for SCFS that incorporates the computational cost associated with computing features. We developed our framework with a computing function for determining classification accuracy using specific classifiers and measuring total test cost for the selected features (S)

$$T_c(\text{Model}, S) = \text{Model}_{\text{accuracy}} - \sum_{i=1}^{k} C_i. \tag{20}$$

T_c(Model, S) is evaluating the overall cost based on classification accuracy and selected feature subset S. $\text{Model}_{\text{accuracy}}$ is the built model's accuracy in classifying test instances (the percentage of correctly classified test instances). The test cost is defined from the ith to the kth selected features from S as C_i ($i = 1, 2, \ldots, k$). We tried to make good models for advanced classification accuracy with lower total test costs according to the hypothesis of the proposed evaluation function. In (20), a better-performing built model has well accuracy with less test cost. This article chooses relevant features with low test costs by balancing a wrapper method against the total test cost. Here, we considered two distributions: beta and uniform distributions on [0, 1]. However, before applying the algorithm, the test cost needs to be 1. This can be accomplished by utilizing a large number of normalization strategies. Because normalizing real-world test costs is not the focus of this article, we used the most basic normalization approach as

$$C_i^* = \frac{C_i}{C_{\max} + 1.0}. \tag{21}$$

For all features, C_i contains the original test cost, and C_i^* represents the normalized test cost. Finally, $C_{\max}$ contains the highest value of all original test costs. We aim to obtain a subset of the original feature space using different searching approaches. The framework that we are presenting here is generic enough to use any search strategy. To test and validate our framework, it considers the forward best-first search that creates all possible single feature expansions by starting with null features.

Selecting the item with the highest overall rating leads to a new round of searching that includes additional features that were not previously considered. If adding more features to a feature subset does not improve performance, the search goes back to the previous best-unexpanded feature subset and keeps searching until adding more features does not help. Our proposed SCFS wrapper framework's detailed process is shown in Algorithm 4. The greedy hill-climbing is combined with a backtracking facility in Algorithm 3. The parameter max-stale determines how much backtracking is done based on serial nonimproving nodes. We consider nonimproving nodes before ending the search in our current implementation. We used the same parameter value because of our SCFS. Algorithm 1 is a very effective algorithm used with the help of basic classifiers, such as C4.5 [37] and naive Bayes [38]. We calculated the worst case time complexity of this algorithm as $O(n^2)$ (n is the available features in the dataset). We also

Algorithm 4 SCFS

```
Input: Training data, A-: feature set, maxstale-: non-improving nodes, q-addjudge
Output: SCFS- built classifier
Create set Φ for feature set (open, closed, best)
    k=0, and p = sizeof (A)
if (k<maxstale) then
    q = false
    v = max Merit(model, w) from Φ = open
for i=1 to m,
    do Fi = ith feature from A
if v or Fi are not in open and closed, then
Create tempMerit= Merit(Model, v∪Fi)
    open=open+ v∪Fi
if(tempMerit>bestMerit) then
    k=0, q=true
    bestMerit=tempMerit
    Best= v∪Fi
endif
endif
End for
if(!q) then
K++
end if
create a classifier with selected features as Best
Return built classifier
```

applied the resubstituting accuracy to lessen the time complexity. However, this approach is not a good indicator of accurate classification of fresh data during moderate processing.

VI. Experimental Results and Its Analysis

Based on the proposed framework, methodology, and algorithm, we evaluated various parts of the model, formulas, equations, and procedures of algorithms. As mentioned in Tables I–III in the Appendix for evaluation, we considered different datasets. Since our approaches are different, all datasets are not used for standard methods. Based on the approach, suitable datasets are evaluated for good performance. The dataset is also considered as per applied approaches. Thus, different types of tables are considered based on datasets for the complementary approach. For example, Table I is evaluated with dataset characteristics, such as several features, classes, and instances with additional characters i.e., clusters.

A. Datasets and Parameters

We collected different datasets from the UCI machine learning repository to conduct the experiments. The suggested algorithm can address various problems by using different datasets with different features, classes, and instances. We considered the dataset as 70% and 30% of data for each dataset on training and test sets. Using Weka [39] to discretize the training data is necessary because the clustering and MI-based works are evaluated through discrete data. Each dataset only needs to go through the statistical clustering method once, as shown in Table I.

We considered two selection approaches as stepwise forward selection (SFS) and greedy stepwise backward selection (GSBS) [9]. With Weka [39], the experiments for LFS and GSBS are run using only the default settings. Furthermore, we considered PSO-based approaches, such as PSOFS, PSO2S, PSO42, and PSOBE from [13] and [40]. The following parameter values are entered into the PSOFS, PSO2S, PSO42, and PSOBE methods as w is equal to 0.7298, c1 equals c2 as 1.49618, and the threshold value θ is 0.6. There are 40 population sizes with a maximum of 100 iterations.

TABLE I
DATASETS

Dataset	# Features	#Cluster	#Classes	#Instances
Wine	13	6	3	178
Vehicle	18	6	4	846
Ionosphere	34	11	2	351
Sonar	60	12	2	208
Musk Version 1 (Musk1)	166	14	2	476
Arrhythmia	279	15	16	452
Madelon	500	11	2	4400
Multiple Features	649	15	10	2000

B. PSO-Based Standard Approaches

We considered two standard greedy search-based FS algorithms, such as LFS [9] and GSBS, for searching techniques in our experiments. LFS reduces the number of evaluations by limiting the number of features based on the forward selection step. As a result, LFS creates better performance than others [9]. This backward selection method begins with all features and continues until the accuracy of classification decreases due to removing any of them. Two PSO-based algorithms are called PSOFS and PSO2S, and the third is called PSO42. PSOFS selects features using a standard continuous PSO. The details of PSO2S's two-stage algorithm can be found in [13]. There is a difference between PSO2S in [13] because it is consistent with other PSO2S implementations, such as PSOBE. There are two primary updating mechanisms in PSO42: pbest and gbest. PSO42's specifics can be gleaned from [40]. The error rate of accuracy is utilized in the fitness function in PSOFS, PSO2S, and PSO42, just like it is in PSOBE.

C. PSO-Based Result Analysis

The performances of PSOBE- and PSO-based FS approaches are mentioned in fig 4, and the comparison performance is mentioned in Fig. 5.

A PSO approach selects an average numeral of options from a list of all available options, represented in Fig. 4 by the term "All." The values for "Best," "Ave," and "Std" are based on 40 independent runs, and these values represent the best results from the 40 separate tests. We considered the implication experiments of accuracy for several methods, such as "All," PSOFS, PSO2S, PSO42, and PSOBE methods, under the PSO approach. The different outputs are shown in Fig. 4(a)–(c).

D. Performance of PSOBE

As per the evaluation of PSOBE, the classification accuracy is good on two datasets by PSOBE compared to other datasets, as shown in Fig. 4. PSOBE only used about a quarter of the available features in each case. When working with datasets with many features, such as Arrhythmia, Madelon, and multiple, PSOBE removed 90% of the features while also improving classification accuracy significantly. The FS algorithm reduces the dimensionality of the data using PSOBE while improving the classification accuracy.

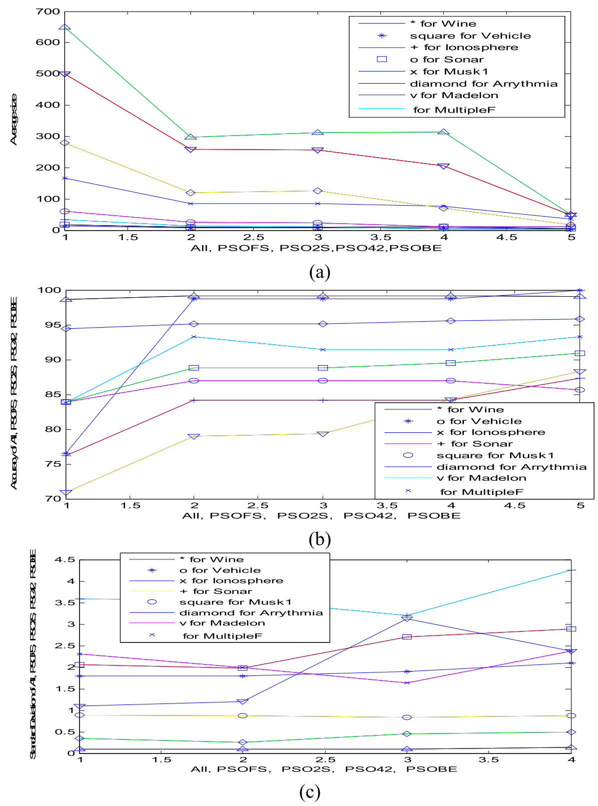

Fig. 4. (a) Evaluation of the PSO approach on different datasets for average size. (b): Accuracy as per the PSO approach. (c) Standard deviation values as per the PSO approach.

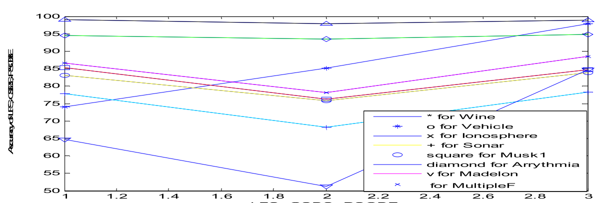

Fig. 5. Performance of LFS and GSBS.

E. Comparing Evaluation on PSO-Based Algorithms

We experimented with different algorithms under PSO for FS and compared its classification accuracy, as shown in Fig. 4. The PSOBE-based FS is significantly fewer than PSOFS FS, as shown in Fig. 4. PSOBE performed well in classification accuracy than PSOFS on datasets such as Vehicle and MultipleF datasets. However, the quantity of features is much smaller. PSOFS selected an average of 297.07 features on the MultipleF dataset. When we decrease the number of features for PSOBE, such as 51 features, then there is only a 0.14% decrease in average classification accuracy. When comparing PSOBE to PSO2S and PSO42, it is easy to see that PSOBE has selected fewer features than either PSO2S or PSO42. Only on the Ionosphere datasets, PSOBE selects slightly more features than PSO42, but PSOBE outperformed PSO42 in classification accuracy.

F. Compared With Existing Approaches

We have taken a comparison performance among LFS, GSBS, and PSOBE, as shown in Fig. 5. When using LFS or GSBS, only one solution is generated with two evaluation items, such as Size and Acc columns, which identify the number of features and classification accuracy. Size and accuracy metrics for PSOBE can be found in the columns labeled "Size" and "Acc." PSOBE's classification accuracy increases with the smaller number of sizes. As shown in Fig. 5, PSOBE selected more features than LFS, but it was much more accurate in its classifications than LFS in the vast majority of cases. LFS outperformed PSOBE for classification in the multiple datasets, but PSOBE's classification accuracy is higher than LFS's (as shown in Fig. 4). When comparing classification accuracy and the number of features, PSOBE consistently outperformed GSBS. Using PSO and "BE," the results show that traditional methods, such as LFS and GSBS, cannot adequately discover a better feature set solution.

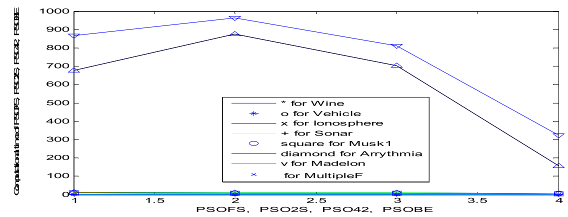

Fig. 6. Computational time of PSOFS, PSO2S, PSO42, and PSOBE.

G. Comparing the Computational Times of Different Algorithms

We considered the average computational time for all approaches, such as PSOFS, PSO2S, PSO42, and PSOBE in one time run, as shown in Fig. 6. Minutes are used to express the values in Fig. 6, which shows that all four algorithms under PSO completed their execution within 10 min, excluding more features and records. All four algorithms are wrapper approaches, which is why they take so long on large datasets. Their computational time was mainly devoted to evaluating the fitness of feature subsets to determine the training classification error rate. A more significant number of features or records necessitate a more extended evaluation period to assess their suitability.

Fig. 6 shows that PSOBE took less time than the other approaches, such as PSOFS, PSO2S, and PSO42, and this can be seen from the results. Even though PSOBE requires more "BE" steps than the other three algorithms, it is computationally less expensive, supporting our hypothesis. This pattern can be seen especially on the two datasets: Madelon and MultipleF. These two datasets took more time due to a larger dataset. Due to LFS's narrower focus on features, PSOBE runs slower than LFS and GSBS. However, with many more features, such as Madelon and multiple, PSOBE is faster than GSBS. GSBS initiates with all features and necessitates a lengthy evaluation process for each one. While the number of evaluations in PSOBE is fixed, the number of evaluations in GSBS increases as datasets get more extensive.

VII. Performance on Clustered-Based Feature Selection

We considered clustered-based FS's performance with five gene expression datasets, as shown in Table II for experiments.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
FIVE GENE EXPRESSION DATASETS

Name	Abbr.	No. of Genes	No. of Patients	No. of Classes
Beast.3.class	BR3	4.869	96	3
SRBCT	ST	2.308	83	4
Brain-tumor2	BT2	10.367	50	4
11-tumors	11T	12.533	174	11
14-tumors	14T	15,009	308	26

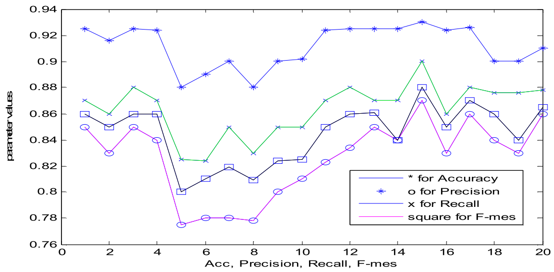

Fig. 7. Feature clustering outputs of SFC versus η on D1.

Different gene expression data with several genes, the number of patients, and classes are mentioned in Table II. We executed different experiments on the dataset, as shown in Table II. It shows the system's capabilities and investigates how it ranks features based on SFR performance. As per the experimental setup, we considered the dataset D1 as 100 rows and 100 columns. D1 is capable of being divided into 16 equal blocks. D1 was used to investigate the SFC algorithm's subspace weights in the experiments. Because the data contain four coclusters, we chose $L = 4$ and 20 real values for η. To sum it up, we gathered over 2000 results to examine the influence of parameters on structured feature clustering for the final coclustering result.

A. Use of η on C

As per clustering output, we calculated the average entropy of C. When η is small, the average entropy of C decreases. It grew when multiplied by η and then shrank back down. As a result, it proliferated as η increased. Entropy regularizes forces' weights to be more evenly distributed, so the overall average entropy of C does not update more when it is high.

B. Effects of η on the Results of Feature Clustering

All feature clustering results are evaluated using the five most widely used evaluation indices. Considering that the clustering result depended on initial clusters, we averaged out 100 evaluations and presented the average results. When η is small, everything is low, and then, they quickly increase. The results of confusion evaluation based on η parameter are $\{2^{-17}, 2^{-16}, \ldots, 2^{1}, 2^{0}\}$, where the parameter values are considered for all evaluation matric items, such as accuracy, precision, recall, and F-measure, as mentioned in Fig. 7. When the parameter values are considered in the range of 2^{-13}–2^{-6}, the evaluation result affects all items, i.e., decreasing during this range; other values do not affect results.

C. Assessment of Result and Analysis

We considered six methods of FSs that are compared to substantiate the efficiency of SFR, including Relief-F [17], [41], RFS [42], MRMR [43], Fisher Score [44], SVM-RFE-CBR [45], and UGL [46]. We used the same set of parameters for all methods to ensure fair experiments, ranging from 10^{-5} to 10^{5}. In both UGL and SVM-RFE-CBR, we used thresholds ranging from 0.6 to 0.9 for highly correlated feature pairs. After 60 features are removed from SVM-RFE-CBR, half of them are removed in each iteration until all 60 features have been removed. We chose the numbers between [1, 10] and λ for [0.1, 1] to run SFR on all datasets. For our evaluation, the quantity of clustering reps is set to 20.

We used seven supervised FS approaches to pick out various features from Table II. We then performed a fourfold SVM on the feature set data. Fig. 8 shows the maximum accuracy against FS as seven approaches on five datasets. The proposed method SFR is performed well compared with all other methods in terms of accuracy, such as BR3, BT2, and 14T datasets. We improved 8% accuracy as SFR on the 14T dataset compared to Relief-F, the runner-up. SFR had the best result on the ST dataset on only 20 features, followed by 100 on the BR3 dataset and 140 on the 14T dataset. For the most part, SFR performed admirably across all datasets.

Three parameters l, η, and λ are tested in this experiment to see how they affect SFR's performance. We begin by looking at how l affects SFR's performance. Based on five datasets, this shows that the overview of SFC for FS certainly supports choosing better features for classification on all datasets, as only one feature cluster yields the lowest accuracy. On most datasets, the accuracy increased with an increase in l, as shown in Fig. 7. The accuracy improved with increasing on the ST and BT2 datasets, which both have incredibly high dimensions. From this table, it is easy to see that the classification accuracies were stable at 0.90 when set to 0. We also see that for all datasets, $\eta = 1$ yields the lowest accuracy. From (18), we know that the conventional ranking method degenerates to the structured weighting feature ranking with $\eta = 1$. These results demonstrate that using a feature ranking system to classify features improved the process.

VIII. RESULT ANALYSIS ON COMPUTATIONAL COST-BASED FEATURE SELECTION

This section describes the proposed wrapper framework's experimental validation and effectiveness evaluation. Specifically, we experimented to find out the computational cost for FS and classification accuracy. We considered the evaluation of the total cost for classification accuracy and FS.

A. Evaluation of Test Cost Experiments

The performance of our experiments is developed on the different datasets in Table III as in the Appendix, which contains the number of records, features, classes, and datasets collected from UCI machine repository datasets [35]. Our experiments used datasets based on both numerical and nominal features.

We considered all datasets designed for test-cost-insensitive, which meant that they did not include any of the input features' intrinsic test costs. As a result, several test costs were created as their input features. We used β distribution on the interval [0, 1] to obtain test cost as [8, 48] for each feature. We noticed

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE III
DATASETS FOR EXPERIMENTS

Dataset Name	Instance number	Number of features	Number of classes
Anneal	898	38	5
Audiology	226	69	24
Breast Cancer	699	9	2
Diabetes	768	8	2
Glass	214	9	7
Hepatitis	155	19	2
Ionosphere	351	34	2
Iris	150	4	3
Mushroom	8124	22	2
Sonar	208	60	2
Vehicle	846	18	4
Vowel	990	10	3

that SCFS performed well compared to a uniform distribution with the same interval. However, any distributions did not affect SCFS for the test costs. Because the results were so close, this study does not detail their differences.

We used standard data mining algorithms as [37] and [38] to test our SCFS approach empirically. It is compared to two well-established competitors: the test-cost-insensitive FS wrapper framework [36] and the SCFS filter framework [8]. We will now go over the tested algorithms and their abbreviations that we used in this article as follows.

1) The classifier C4.5 uses a decision tree to determine the best classification.
2) *NB:* Naive Bayes classifier [38].
3) *FS:* Existing computational insensitive FS.
4) *CS-CFS:* It is a used framework for filtering out SCFS [8] with correlation approaches.
5) *SCFS:* It defines an evaluation function as per the proposed function.

C4.5 is used to compare the computational cost for selected features, and the results are shown in Figs. 9 and 10. Classification accuracy (C4.5) and its area under the curve (AUC) are shown in Fig. 11. Comparing the two sets of results reveals that the experimental findings are nearly identical to those of the C4.5 experiments. Based on these findings, it appears that the proposed SCFS is unaffected by the base classifier selected. Although our SCFS framework is based on the minimal sequential optimization (SMO) algorithm, we have also tested it with other cutting-edge classifiers, such as the k-nearest neighbor classification (KNN). Because the results were so close, the comparisons in this article are omitted to this article.

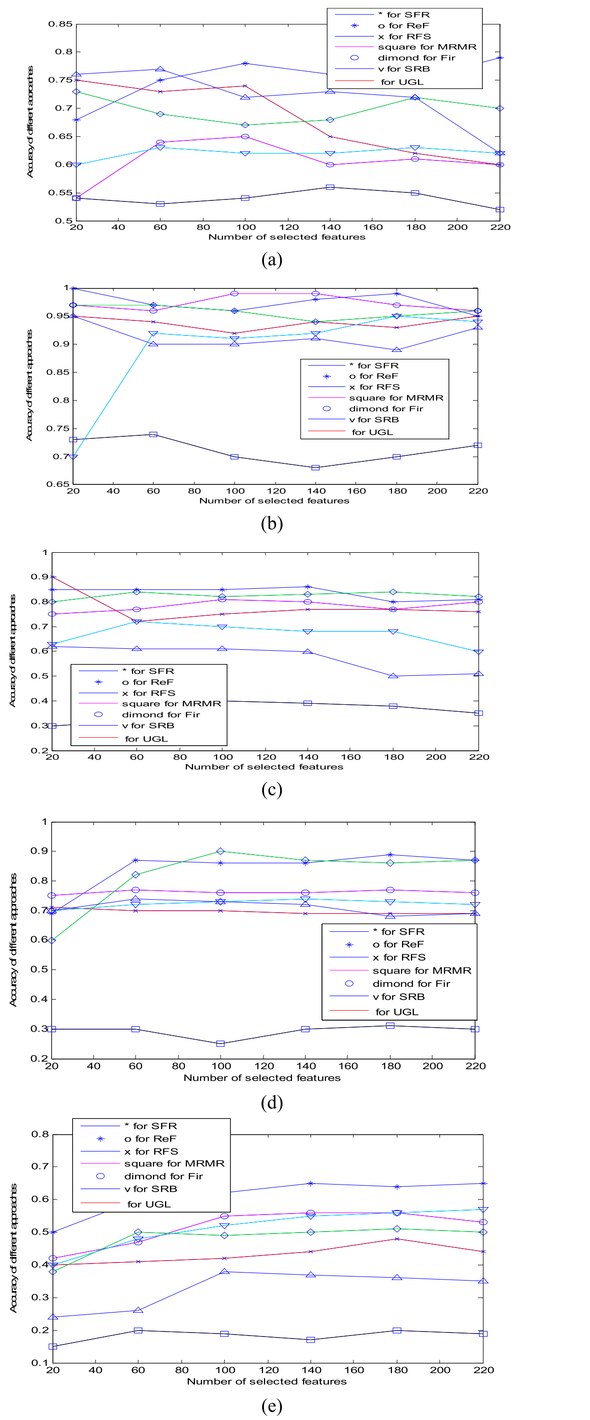

Fig. 8. Accuracy performed by different approaches on various datasets. (a) Performance on BR3. (b) Performance on the ST dataset. (c) Performance on the BT2 dataset. (d) Performance on the 11T dataset. (e) Performance on the 14T dataset.

IX. Evaluation of Wrapper Method Using Forward Selection and Recursive Feature Elimination

Although we have taken different datasets for our proposed model, we considered three datasets: the Mobile dataset, the Heart dataset, and the Diabetes Dataset for a sample of evaluation using wrapper methods. We considered two wrapper methods, forward selection and RFE, for further analysis for accuracy and computational cost.

A. Wrapper Method—Forward Selection

In this part, we considered seven features for the step forward FS (SFS). Then, the accuracy score performance for each feature is developed by random forest classifiers (RFCs). The several accuracies as per the feature with computation time are shown in Table IV.

Here, we considered only training dataset with seven features from 20 features. Each feature is tested from seven feature datasets using SFS shown in Table IV. If the features of mobile data are {“battery_power,” “blue,” “clock_speed,” “dual_sim,” “fc,” “four_g,” “int_memory,” “m_dep,” “mobile_wt,” “n_cores,” “pc,” “px_height,”

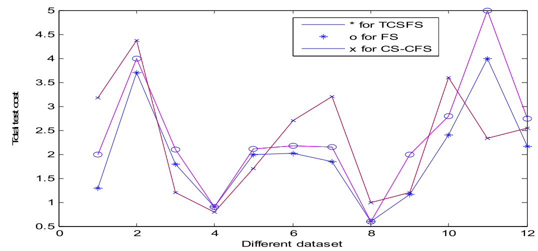

Fig. 9. Comparison of computational cost among SCFS, FS, and CS-CFS.

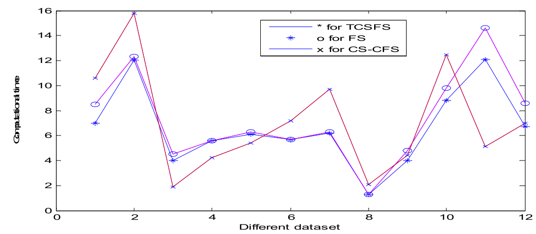

Fig. 10. Comparison of selected features of SCFS, FS, and CS-CFS performances.

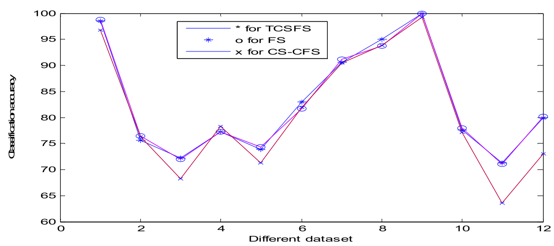

Fig. 11. Comparison of the classification accuracy of SCFS, FS, and CS-CFS performances.

TABLE IV
FEATURE EVALUATION FOR ACCURACY WITH COMPUTATIONAL TIME

S.No.	No. of features evaluated	Accuracy score	Computational time
	Mobile data set		
1	1/7	0.678	17.6s
2	2/7	0.805	16.1s
3	3/7	0.878	15.7s
4	4/7	0.9075	14.4s
5	5/7	0.9075	13.9s
6	6/7	0.905	12.7s
7	7/7	0.899	11.8s
	Heart dataset		
1	1/5	0.832	8.4s
2	2/5	0.855	7.1s
3	3/5	0.844	7.2s
4	4/5	0.855	5.7s
5	5/5	0.860	5.1s
	Diabetes Dataset		
1	1/5	0.7608	6.2s
2	2/5	0.8925	4.5s
3	3/5	0.9133	5.0s
4	4/5	0.91	3.6s
5	5/5	0.9141	3.2s

“px_width,” “ram,” “sc_h,” “sc_w,” “talk_time,” “three_g,” “touch_screen,” “wifi,” “price_range”}, the selected features are (“battery_power,” “blue,” “pc,” “px_height,” “px_width,” “ram,” “touch_screen”) with feature id (0, 1, 10, 11, 12, 13, 18). Furthermore, we evaluate the standard deviation error from above features, as shown in Table V.

The features are collected as per standard deviation error during evaluation.

The DS-SFS accuracies are obtained using different classifiers, as shown in Table VI.

TABLE V
FEATURES ARE COLLECTED AS PER STANDARD DEVIATION ERROR

S.No.	Feature ID	standard deviation error
	Mobile data set	
1	(13)	0.0102253
2	(0, 13)	0.00420813
3	(0, 11, 13)	0.00534
4	(0, 11, 12, 13)	0.0053033
5	(0, 1, 11, 12, 13)	0.00444878
6	(0, 1, 10, 11, 12, 13)	0.00467707
7	(0, 1, 10, 11, 12, 13, 18)	0.0037326
	Heart dataset	
1	(11)	0.0194647
2	(10, 11)	0.0348838
3	(4, 10, 11)	0.0272309
4	(4, 6, 10, 11)	0.0232875
5	(4, 6, 8, 10, 11)	0.0234925
	Diabetes Dataset	
1	(6)	0.01117
2	(1,6)	0.00629153
3	(1,4,6)	0.003849
4	(1,2,4,6)	0.00490653
5	(1,2,4,6,7)	0.00786165

TABLE VI
DS-SFS ACCURACIES ARE EVALUATED BY USING DIFFERENT CLASSIFIERS

S.No.	Classifier name	Wrapper(Step Forward Selection)		
		Mobile data set	Heart dataset	Diabetes Dataset
0	Nearest_Neighbors	0.93750	0.81667	0.86250
1	Linear_SVM	0.95250	0.76667	0.79500
2	Polynomial_SVM	0.96000	0.61667	0.78750
3	RBF_SVM	0.23000	0.61667	0.97750
4	Guassian_Process	0.28000	0.83333	0.98500
5	Gradient_Boosting	0.91750	0.83333	0.97750
6	Decision_tree	0.82500	0.76667	0.82250
7	Extra_Trees	0.92500	0.80000	0.99000
8	Random_Forest	0.85250	0.85000	0.84500
9	Neural_Net	0.47000	0.83333	0.74750
10	AdaBoost	0.73750	0.80000	0.83000
11	Naïve_Bayes	0.82500	0.80000	0.77000
12	QDA	0.95500	0.73333	0.77500
13	SGD	0.49000	0.65000	0.69500

Similarly, we considered eight feature data for a similar evaluation; then, we got accuracy with the computational time based on eight features evaluations. The selected features are (“battery_power,” “px_height,” “px_width,” “ram”). From the Heart dataset, we considered five out of 12 features and 179 training datasets for SFS methods. Thus, we get accuracy with computational time based on five features evaluations, which are given as follows. The selected features are (“ejection_fraction,” “platelets,” “serum_sodium,” “smoking,” “time”) with feature Id (4, 6, 8, 10, 11) from whole features {“age,” “anaemia,” “creatinine_phosphokinase,” “diabetes,” “ejection_fraction,” “high_blood_pressure,” “platelets,” “serum_creatinine,” “serum_sodium,” “sex,” “smoking,” “time,” “DEATH_EVENT”}. Furthermore, we evaluate the standard deviation error from the above features, as shown in Table V. Similarly, we considered

TABLE VII
COMPUTATIONAL TIME FOR MOBILE DATASET

S.No.	Items	With RFC	Non-RFC	With RFE	Without RFE
CPU Times	User times	412 ms	485 ms	406 ms	476 ms
	System times	47.9 ms	39.3 ms	36.5 ms	37.6 ms
	Total Times	460 ms	524 ms	442.5 ms	513.6 ms
	Wall Times	448 ms	453 ms	443 ms	445 ms
	Accuracy	0.915	0.8575	0.895	0.8575

six features' data for similar evaluation; then, we get accuracy with computational time based on six features evaluations, as shown in Table IV. The selected features are ("ejection_fraction," "high_blood_pressure," "platelets," "serum_sodium," "smoking," "time"). The accuracies for different classifiers are given in Table VI. From the Diabetes dataset, we considered six features out of nine features and 1200 training datasets for SFS methods. Thus, we get accuracy with computational time based on six features evaluations, which are given as follows. The selected features are ("Glucose," "Blood Pressure," "Insulin," "Diabetes Pedigree Function," "Age") with feature Id (1, 2, 4, 6, 7) from whole features {"Pregnancies," "Glucose," "Blood Pressure," "Skin Thickness," "Insulin," "BMI," "Diabetes Pedigree Function," "Age"}. Furthermore, we evaluate the standard deviation error from the above features. Similarly, we considered six features' data for a similar evaluation; then, we get accuracy with computational time based on six features evaluations, as shown in Table VII. The selected features are ("Glucose," "Blood Pressure," "Skin Thickness," "Insulin," "Diabetes Pedigree Function," "Age"). The accuracies for different classifiers are given in Table VI.

B. Wrapper Method—Recursive Feature Elimination

In this part, we apply RFC based on the embedded method from the training dataset. We get four supporting selected features from the whole features of the Mobile dataset. Four true values are selected from above methods, such as {"battery_power," "px_height," "px_width," "ram"}. Thus, the length of selected features is 4, and its mean is 0.05. However, the accuracy of all features of the Mobile dataset is (0.07564013, 0.00690401, 0.02941371, 0.00674471, 0.0238337, 0.00634795, 0.03572272, 0.02406503, 0.03962903, 0.02420826, 0.02930681, 0.05670975, 0.05691962, 0.47976244, 0.02799993, 0.02788505, 0.03031405, 0.0051026, 0.00724727, 0.00624325). RFE uses two kinds of evaluation on the Mobile dataset as follows.

Furthermore, we apply RFC based on the wrapper method from the training dataset. We get eight supporting selected features from the whole features of the Mobile dataset. Thus, the selected features are {"battery_power," "int_memory," "mobile_wt," "pc," "px_height," "px_width," "ram," "talk_time"}. Furthermore, when we apply random forest elimination (RFE) with estimator RFC, then accuracy with the computational time is varied as table. Furthermore, when we apply RFE with the estimator gradient boosting classifier, then we get seven supporting selected features from the whole features of the Mobile dataset. Thus, selected features are {"battery_power," "int_memory," "mobile_wt," "n_cores," "px_height," "px_width," "ram"}. The corresponding accuracy with computational time is varied, as shown in Table VII. The accuracies of different classifiers are given in Table VIII.

TABLE VIII
EVALUATION RESULTS OF THREE DATASETS USING DS-RFE

S.No.	Classifier name	Wrapper (Recursive Feature Elimination)		
		Mobile Dataset	Heart dataset	Diabetes dataset
0	Nearest_Neighbors	0.93750	0.46667	0.83250
1	Linear_SVM	0.97750	0.75000	0.77250
2	Polynomial_SVM	0.92500	0.61667	0.77500
3	RBF_SVM	0.23000	0.61667	0.97250
4	Guassian_Process	0.28500	0.61667	0.97500
5	Gradient_Boosting	0.92250	0.86667	0.99500
6	Decision_tree	0.83000	0.78333	0.80500
7	Extra_Trees	0.86500	0.78333	0.97750
8	Random_Forest	0.84750	0.85000	0.83250
9	Neural_Net	0.62750	0.61667	0.74000
10	AdaBoost	0.79500	0.80000	0.80000
11	Naïve_Bayes	0.82750	0.76667	0.77250
12	QDA	0.97500	0.73333	0.75750
13	SGD	0.46250	0.61667	0.68250

TABLE IX
COMPUTATIONAL TIME FOR HEART DATASET

S.No.	Items	With RFC	Without RFC	With RFE	Without RFE
CPU Times	User times	209 ms	208 ms	217 ms	210 ms
	System times	36.9 ms	37.7 ms	32.4 ms	25.6 ms
	Total Times	246 ms	246 ms	249 ms	236 ms
	Wall Times	348 ms	345 ms	352 ms	345 ms
	Accuracy	0.817	0.85	0.867	0.85

The Heart dataset considered 239 training and 60 test data with 12 features. We get supporting selecting features using the SelectFromModel method based on the training datasets RFC. Thus, five selected features are identified as true from {"age," "creatinine_phosphokinase," "ejection_fraction," "serum_creatinine," "time"}, and its mean is 0.0833. The accuracy score of all features are {0.09079284, 0.0152904, 0.09123371, 0.01620469, 0.11295166, 0.01231782, 0.08022786, 0.13546353, 0.08104264, 0.01619514, 0.01052158, 0.33775813}. Furthermore, we apply RFC based on the wrapper method from the training dataset to get accuracy with the computational time mentioned in Table IX.

Furthermore, we get seven supporting selecting features using RFC from the training dataset from {True, False, True, False, True, False, True, True, True, False, False, True}. Thus, selected features are {"age," "creatinine_phosphokinase," "ejection_fraction," "platelets," "serum_creatinine," "serum_sodium," "time"}. Here, we evaluate the accuracy with the computational time, as shown in the table. When we apply RFE on the gradient boosting classifier estimator, then we also select seven supporting features as {"age," "creatinine_phosphokinase," "ejection_fraction," "platelets,"

TABLE X
COMPUTATIONAL TIME FOR DIABETES DATASET

S.No.	Items	With RFC	Without RFC	With RFE	Without RFE
CPU Times	User times	364 ms	349 ms	338 ms	361 ms
	System times	28.3 ms	31.4 ms	35.5 ms	22.6 ms
	Total Times	393 ms	380 ms	374 ms	384 ms
	Wall Times	444 ms	449 ms	374 ms	384 ms
	Accuracy	0.995	0.995	0.99	0.995

"serum_creatinine," "serum_sodium," "time"}. Here, we get different accuracies with the computational time value compared to previous feature evaluations on the same selected features. The accuracies of different classifiers are given in Table VIII.

The Diabetes dataset considered 239 training and 60 test data with 12 features. We get supporting selecting features using the SelectFromModel method based on the training datasets, RFC. Thus, four selected features are identified as true form {"Glucose," "BMI," "DiabetesPedigreeFunction," "Age"}, and its mean is 0.125. The accuracy scores of all features are {0.08473198, 0.24702528, 0.08780013, 0.06942873, 0.07836758, 0.16477351, 0.12729925, 0.14057354}. Furthermore, we apply RFC based on the wrapper method from the training dataset to get accuracy with the computational time mentioned in Table X.

Furthermore, we get six supporting selecting features using RFC from the training dataset. Thus, selected features are {"Pregnancies," "Glucose," "BloodPressure," "BMI," "DiabetesPedigreeFunction," "Age"}. Here, we evaluate the accuracy with computational time, as shown in the table. When we apply RFE on the gradient boosting classifier estimator, then we also select six supporting features as {"Pregnancies," "Glucose," "Insulin," "BMI," "Diabetes Pedigree Function," "Age"}. Here, we get different accuracy with computational time value compared to previous feature evaluations on the same selected features. The accuracies of different classifiers are given in Table X.

X. Discussion

Different approaches have been applied to solve social problems where social connectivity items have promoted the quality of social life, such as SIoT, smartphones, and wearable devices in the HITL system. Thus, everyone gets benefits from social system mechanisms although there are no certain mechanisms that are used for all social items. For example, health issues are different than sports issues, any kind of incident issues, fresh technology issues, and so on. The mechanism for health analysis is different from the smart devices' analysis. However, we tried to solve how various datasets are analyzed through our proposed generic approach.

In this article, we considered various approaches, such as the PSO approach, BE, structured weighting feature ranking, and computational cost-based FS. When various approaches are used on different kinds of datasets, its results are also varied as per the datasets. We considered different kinds of datasets (as shown in Tables I–III) to test our proposed model. Since all experiments are explained in Section VI, it does not repeat that information once again here. Since various data are analyzed through our proposed model, it helps to solve social problems partially as per the proposed model and the dataset.

Moreover, we will continue to work on our framework by testing it with different wrapper methods in the future to fulfill the social problem. SFC will be improved by introducing new techniques, such as ensemble learning. Using SFR in real-world applications, we will be working on a public health system in the future.

XI. Conclusion

This article has emphasized the multiple approaches to different datasets for FS. We used the filter methods in the PSO-based approach for FS. Although the PSO-based approach makes good classification, LFS and GSBS approaches generate good accuracy on a large dataset. Furthermore, the SFR approach is used for cluster ranking features in large volume data. We proposed a structured weighting feature ranking method to find out the feature's rank from various feature clusters. According to the proposed model on most datasets, SFR performed better than the other six feature ranking methods. The new method selects informative and diverse features, as demonstrated experimentally. Again, we considered two objectives in the SCFS problem: improving classification accuracy and reducing test costs. We used an optimal model for testing features data. After a large amount of data testing, we found that the proposed model chooses an optimal feature subset with the lowest test cost while maintaining a high level of classification accuracy.

Appendix

See Tables I–III.

References

[1] M. Dash and H. Liu, "Feature selection for classification," *Intell. Data Anal.*, vol. 1, nos. 1–4, pp. 131–156, 1997.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.

[3] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evol. Comput. IEEE World Congr. Comput. Intell.*, May 1998, pp. 69–73.

[4] H. Xie, L. Zhang, C. P. Lim, Y. Yu, and H. Liu, "Feature selection using enhanced particle swarm optimisation for classification models," *Sensors*, vol. 21, no. 5, p. 1816, Mar. 2021.

[5] H. K. Bhuyan, L. R. Kumar, and K. R. Reddy, "Optimization model for sub-feature selection in data mining," in *Proc. Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Nov. 2019, pp. 1212–1216.

[6] M. A. Esseghir, G. Goncalves, and Y. Slimani, "Adaptive particle swarm optimizer for feature selection," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn. (IDEAL)*. Berlin, Germany: Springer-Verlag, 2010, pp. 226–233.

[7] X. Chen, J. Z. Huang, Q. Wu, and M. Yang, "Subspace weighting co-clustering of gene expression data," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 16, no. 2, pp. 352–364, Mar./Apr. 2019.

[8] V. Bolón-Canedo, I. Porto-Díaz, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A framework for cost-based feature selection," *Pattern Recognit.*, vol. 47, no. 7, pp. 2481–2489, Jul. 2014.

[9] M. Gutlein, E. Frank, M. Hall, and A. Karwath, "Large-scale attribute selection using wrappers," in *Proc. IEEE Symp. Comput. Intell. Data Mining*, Mar. 2009, pp. 332–339.

[10] H. Sirajul, H. K. Bhuyan, and S. R. Pattanaik, "Gradient descent based feature selection for classification," *J. Appl. Sci. Comput.*, vol. 6, no. 2, pp. 913–918, Feb. 2019.

[11] H. K. Bhuyan and S. K. Pani, "Video usefulness detection in big surveillance systems," in *Applications of Machine Learning in Big-Data Analytics and Cloud Computing*. Denmark: River Publishers, 2021, ch. 14, pp. 289–308.

[12] H. K. Bhuyan, M. Mohanty, and S. R. Das, "Privacy preserving for feature selection in data mining using centralized network," *Int. J. Comput. Sci. Issues (IJCSI)*, vol. 9, no. 3, pp. 434–440, Apr. 2012.

[13] H. K. Bhuyan and S. K. Pani, "Crime predictive model using big data analytics," in *Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications*. Hoboken, NJ, USA: Wiley, 2021, ch. 3, pp. 57–78.

[14] H. K. Bhuyan, "Large sensing data flows using cryptic techniques," in *Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications*. Hoboken, NJ, USA: Wiley, 2021, ch. 13, pp. 269–289.

[15] Z. X. Zhu, Y. S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, Cybern., B (Cybern.)*, vol. 37, no. 1, pp. 70–76, Feb. 2007.

[16] S.-W. Lin, K.-C. Ying, S.-C. Chen, and Z.-J. Lee, "Particle swarm optimization for parameter determination and feature selection of support vector machines," *Expert Syst. Appl.*, vol. 35, no. 4, pp. 1817–1824, Nov. 2008.

[17] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Proc. 9th Int. Workshop Mach. Learn.*, 1992, pp. 249–256.

[18] R. Chen, N. Sun, X. Chen, M. Yang, and Q. Wu, "Supervised feature selection with a stratified feature weighting method," *IEEE Access*, vol. 6, p. 15087–15098, 2018.

[19] H. K. Bhuyan and C. V. M. Reddy, "Sub-feature selection for novel classification," in *Proc. 2nd Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Apr. 2018, pp. 20–21, doi: 10.1109/ICICCT.2018.8473206.

[20] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, Jan. 2014.

[21] H. K. Bhuyan and V. K. Ravi, "Analysis of subfeature for classification in data mining," *IEEE Trans. Eng. Manage.*, early access, Aug. 4, 2021, doi: 10.1109/TEM.2021.3098463.

[22] H. Bhuyan, D. C. Chakraborty, S. Pani, and V. Ravi, "Feature and subfeature selection for classification using correlation coefficient and fuzzy model," *IEEE Trans. Eng. Manag.*, early access, Apr. 19, 2021, doi: 10.1109/TEM.2021.3065699.

[23] H. K. Bhuyan, N. K. Kamila, and S. K. Pani, "Individual privacy in data mining using fuzzy optimization," *Eng. Optim.*, pp. 1–19, May 2021.

[24] L. Jiang, Z. Cai, H. Zhang, and D. Wang, "Not so greedy: Randomly selected naive Bayes," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11022–11028, 2012.

[25] R. Archibald and G. Fann, "Feature selection and classification of hyperspectral images with support vector machines," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 674–677, Oct. 2007.

[26] L. Jiang, G. Kong, and C. Li, "Wrapper framework for test-cost-sensitive feature selection," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 3, pp. 1747–1756, Mar. 2021.

[27] H. Gao, K. Xu, M. Cao, J. Xiao, Q. Xu, and Y. Yin, "The deep features and attention mechanism-based method to dish healthcare under social IoT systems: An empirical study with a hand-deep local-global net," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 336–347, Feb. 2022.

[28] G. Mariammal, A. Suruliandi, S. P. Raja, and E. Poongothai, "Prediction of land suitability for crop cultivation based on soil and environmental characteristics using modified recursive feature elimination technique with various classifiers," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 5, pp. 1132–1142, Oct. 2021.

[29] Z. Zheng, S. Mumtaz, R. Mohammad Khosravi, and G. Varun Menon, "Linked data processing for human-in-the-loop in cyber–physical systems," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 5, pp. 1238–1248, Oct. 2021.

[30] T. Wang, X. Shen, M. S. Obaidat, X. Liu, and S. Wan, "Edge-learning-based hierarchical prefetching for collaborative information streaming in social IoT systems," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 1, pp. 302–312, Feb. 2022.

[31] S. Ghosh and T. Anwar, "Depression intensity estimation via social media: A deep learning approach," *IEEE Trans. Computat. Social Syst.*, vol. 8, no. 6, pp. 1465–1474, Dec. 2021.

[32] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory*, vol. 9, no. 1, pp. 11–17, Jan. 1963.

[33] L. Cervante, B. Xue, M. Zhang, and L. Shang, "Binary particle swarm optimisation for feature selection: A filter based approach," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2012, pp. 881–888.

[34] H. B. Nguyen, B. Xue, I. Liu, and M. Zhang, "Filter based backward elimination in wrapper based PSO for feature selection in classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Beijing, China, Jul. 2014, pp. 3111–3118.

[35] M. A. Hall, "Correlation-based feature selection for discrete and numeric class machine learning," in *Proc. 17th Int. Conf. Mach. Learn.*, 2000, pp. 359–366.

[36] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[37] J. R. Quinlan, *Programs for Machine Learning*, 1st ed. San Mateo, CA, USA: Morgan Kaufmann, 1993, ch. 4.5.

[38] P. Langley, W. Iba, and K. Thompson, "An analysis of Bayesian classifiers," in *Proc. 10th Nat. Conf. Artif. Intell.*, San Jose, CA, USA, 1992, pp. 223–228.

[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 931–934, 2009.

[40] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.

[41] H. Liu and H. Motoda, *Computational Methods of Feature Selection* (Data Mining and Knowledge Discovery Series). London, U.K.: Chapman & Hall, 2007.

[42] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint $\ell 2,1$-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[43] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[44] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2010.

[45] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015.

[46] D. Kong, J. Liu, B. Liu, and X. Bao, "Uncorrelated group lasso," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 1765–1771.

[47] G. Kong, L. Jiang, and C. Li, "Beyond accuracy: Learning selective Bayesian classifiers with minimal test cost," *Pattern Recognit. Lett.*, vol. 80, pp. 165–171, Sep. 2016.

Hemanta Kumar Bhuyan received the M.Tech. degree from Utkal University, Bhubaneswar, India, in 2005, and the Ph.D. degree in computer science and engineering from Sikshya 'O' Anusandhan (SOA) University, Bhubaneswar, in 2016.

He is currently an Associate Professor with the Department of Information Technology, Vignan's Foundation for Science, Technology & Research (Deemed to be University), Guntur, India. He has published many research papers in Science Citation Index Expanded (SCIE)-indexed journals, such as Elsevier, Springer, and IEEE TRANSACTIONS. He has also published many Elsevier, Springer, and IEEE conference proceedings and edited book chapters. His research interests include data mining, big data analytics, machine learning, fuzzy system, and image processing.

Dr. Bhuyan is also a reviewer of many SCIE-indexed journals.

Chinmay Chakraborty (Senior Member, IEEE) completed his post-doctoral research at the Federal University of Piauí, Teresina, Brazil. He is currently an Assistant Professor of electronics and communication engineering with the Birla Institute of Technology, Mesra, Jharkhand, India. He has published more than 150 papers in reputed international journals, conferences, book chapters, more than 20 books, and more than 16 special issues. His main research interests include the Internet of Medical Things, wireless body sensor networks, wireless networks, telemedicine, m-health/e-health, and medical imaging.

Dr. Chakraborty is also an editorial board member of the different journals and conferences. He received the Best Session Runner-Up Award, the Young Research Excellence Award, the Global Peer Review Award, the Young Faculty Award, and the Outstanding Researcher Award. He was also selected as one of the top 2% of scientists in the world in the field of Artificial Intelligence and Image Processing at Stanford University, USA.