IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2019, accepted October 8, 2019, date of publication October 16, 2019, date of current version October 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947701

Combining Multiple Feature-Ranking Techniques and Clustering of Variables for Feature Selection

ANWAR UL HAQ[1], DEFU ZHANG[1], (Member, IEEE), HE PENG[1], AND SAMI UR RAHMAN[2]
[1]School of Information Science and Engineering, Xiamen University, Xiamen 361005, China
[2]Department of Computer Science, University of Malakand, Malakand, Pakistan

Corresponding author: Defu Zhang (dfzhang@xmu.edu.cn)

ABSTRACT Feature selection aims to eliminate redundant or irrelevant variables from input data to reduce computational cost, provide a better understanding of data and improve prediction accuracy. Majority of the existing filter methods utilize a single feature-ranking technique, which may overlook some important assumptions about the underlying regression function linking input variables with the output. In this paper, we propose a novel feature selection framework that combines clustering of variables with multiple feature-ranking techniques for selecting an optimal feature subset. Different feature-ranking methods typically result in selecting different subsets, as each method has its own assumption about the regression function linking input variables with the output. Therefore, we employ multiple feature-ranking methods having disjoint assumption about the regression function. The proposed approach has a feature ranking module to identify relevant features and a clustering module to eliminate redundant features. First, input variables are ranked using regression coefficients obtained by training $L1$ regularized Logistic Regression, Support Vector Machine and Random Forests models. Those features which are ranked lower than a certain threshold are filtered-out. The remaining features are grouped into clusters using an exemplar-based clustering algorithm, which identifies data-points that exemplify the data better, and associates each data-point with an exemplar. We use both linear correlation coefficients and information gain for measuring the association between a data-point and its corresponding exemplar. From each cluster the highest ranked feature is selected as a delegate, and all delegates from the three ranked lists are combined into the final feature set using union operation. Empirical results over a number of real-world data sets confirm the hypothesis that combining features selected using multiple heterogeneous methods results in a more robust feature set and improves prediction accuracy. As compared to other feature selection approaches evaluated, features selected using linear correlation-based multi-filter feature selection achieved the best classification accuracy with 98.7%, 100%, 92.3% and 100% for Ionosphere, Wisconsin Breast Cancer, Sonar and Wine data sets respectively.

INDEX TERMS Classification, clustering of variables, feature selection, filter methods, random forests.

I. INTRODUCTION

For a given classification problem, machine learning algorithms use discriminative abilities of features to categorize observations into different classes, where each feature is an individual characteristic of the process under observation. In recent years, machine learning data sets have become very large, and in some cases the number of input variables even exceeds that of the samples. Performance of a machine learning model not only depends on model specific factors, but also on factors related to input data, such as total number of variables, correlation among input variables and signal-to-noise ratio of data. In high-dimensional data sets, all features may not be equally important and some of these may be redundant, irrelevant or noise. The presence of redundant, irrelevant or noise variables not only leads to increased computational cost, but may also affect predictive performance of the learning model. Machine learning models, which do not have an embedded feature selection mechanism, will accumulate small noisy contribution for each noise variable to the predicted variable. In case of a large number of small noisy contributions, all will add up and result in higher prediction error [2]. Performance of machine learning models having embedded dimensionality reduction mechanism such as Deep

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

Neural Networks (DNN) may also be improved by using an external feature selection module [3]. In high-dimensional settings, it is pertinent to assume that a low-dimensional subset of input features exists, which can efficiently represent the entire data set. Identifying relevant features also provides valuable insight into the underlying relationship between input and output variables, and a more general concept can be yielded [42].

In the past, many methods have been proposed to transform high dimensional data into low dimensional space, as training classification or regression algorithms using all input features could lead to results no better than a random guess as shown in [2]. Most dimensionality reduction techniques perform feature space reduction by deriving compact features through selection or extraction of features in a supervised or unsupervised manner [4], [5]. Feature selection techniques assume that discriminative features can be independent of irrelevant or redundant features and tend to keep a subset of the original set [6], while feature extraction approaches create new variables as combination of existing input features to reduce dimensionality of the feature space [7]. Typically in some real-world problems, the objective is not only to predict the associated class for an observation, but also to identify input features which are responsible for a specific behavior. For example, for a given gene expression profile the first objective could be to predict whether a patient responds to a specific treatment or not; whereas the second objective could be identifying part of the genome responsible for a good or bad response. Therefore, feature extraction methods such as PCA [8] and autoencoders [9] may not be suitable for such problems. Unlike feature extraction methods, feature selection methods do not alter the original representation of data [10], and are considered to provide better readability and interpretability. However, feature selection methods are generally not scalable and their performance usually decline sharply, when used on data sets distinct to those used for developing them. Majority of the studies focus on finding a good solution for a specific problem settings and none of them can be presented as a so-called ''best-method'' [34]. Therefore, new feature selection methods are constantly appearing using different strategies such as combining feature selection with other techniques [39], combining multiple feature selection approaches [54], reinterpreting existing methods [12] and creating new methods to achieve better performance.

In this paper, we propose a novel feature selection framework that ranks input features using multiple feature-ranking algorithms and then clusters them into groups. Employing multiple heterogeneous feature-ranking techniques will exploit disjoint assumptions about the regression function linking input variables with target variable(s). The proposed approach considers both the predictive abilities of input variables as computed by the feature-ranking methods, and their correlations with each other. First, input features are ranked by training three feature ranking approaches namely $L1$ regularized Logistic Regression ($L1$-LR) [14], Support Vector Machines (SVM) [15] and Random Forests (RF) [57] and their computed regression coefficients are used as variable importance measures. Each of these methods measures variable importance differently based on their assumption about the regression function linking the predictors with the predicted variable. This will result in three differently ordered lists for each data set. The lowest ranked features are dropped from each list based on a given elimination threshold. The remaining features are clustered into groups using affinity propagation, an exemplar-based correlation technique. It identifies data-points that best exemplify the data set, and associates each data-point with an exemplar based on their similarity. We use both linear correlation coefficients and information gain as similarity measures, which results in two slightly different techniques. After clustering, the highest ranked feature from each cluster is selected as a delegate. The three sets of delegates are combined using union operation to obtain the final feature set. Empirical results over a number of real-world data sets were compared with that of the original data set and other feature selection techniques such as Information Gain, ReliefF [12], SVM-GA [65], PSO-SVM [66], Sequential Forward Selection (SFS), Sequential Floating Forward Selection (SFFS) [68], PCA and CoV/VSURF [39]. The results confirm that generally feature selection improves classification accuracy and combining features selected using multiple heterogeneous feature-ranking techniques results in more robust feature set. Features selected using correlation-based multi-filter feature selection (MFFS) approach proved to have better predictive performance as compared to other feature selection approaches evaluated.

The rest of the paper is organized in the following manner. In section II, a detailed literature review is presented. Section III briefly discusses related concepts including techniques used for feature-ranking and correlation-based clustering of variables using affinity propagation. Section IV outlines the overall framework of the proposed approach. In Section V, simulation setup and the results obtained are discussed in comparison with other feature selection methods evaluated followed by concluding remarks in Section VI.

II. LITERATURE REVIEW

Feature selection has been widely studied and a large number of methods have been developed. These methods have been successful in dealing with real-world problems such as medical image processing [17], customer churn prediction [20], gene microarray analysis [24], malware detection [25], [26], intrusion detection [27], stock trend prediction [28], text categorization [29], information retrieval [30], including image retrieval [31] and music retrieval [32].

Feature selection methods can be supervised, unsupervised and semi-supervised. In this paper, we focus on supervised feature selection for classification and use the terms class labels and target variable interchangeably. These methods are categorized into two main groups: individual evaluation and subset evaluation methods. Individual evaluation

measures the relevance of each variable with the target variable and assigns importance or rank according to its relevance, while subset evaluation selects a subset of variables for model construction based on some search strategy. Besides this classification, these methods are categorized into filters, wrappers, embedded and hybrid approaches based on their selection strategy [33]. We suggest [1], [34], [35] for detailed discussion on the subject.

Filter methods use variable importance or feature ranking as a principle criteria for variable selection. A typical filter method consists of two steps. In first step, variables are ranked according to some feature evaluation criteria, such as discriminative ability of a variable to classify samples or its correlation with target variables. In individual evaluation approach, each feature is individually assessed and importance measure assigned independent of other variables, while in subset evaluation approach multiple variables are ranked together. In second step, features deemed less important or irrelevant are filtered out. These methods are independent of the learning algorithm. Therefore, the selected features may not be the best feature set for the learning algorithm used for classification. It is common for variables to be ranked differently by different filters, as each method use different assumption about the relevance of a variable. One major drawback of these methods is their inability to consider multicollinearity among the predictors, which results in redundancy [18], [19]. Variable discriminative ability to classify samples [36], [38]–[40], feature correlation [41], [42], information gain [43], [44] are some of the common evaluation criteria used by filter methods.

Wrappers utilize a predefined learning algorithm to evaluate predictive abilities of variables. The predictor is used as a black box and its performance as an objective function to evaluate relevance of variables. Given a specific wrapper, it first selects a subset of variables and then evaluates the subset using the predefined algorithm. Both steps are repeated, until the highest performing subset is found or the stop criteria is met. As wrappers evaluate feature subsets based on the performance of a learning algorithm, they tend to select a better feature subset. A known issue with wrappers is that the search space for a **d**-dimensional data set is 2^d, which is impractical when **d** is very large [1]. Therefore, search algorithms are utilized to find an optimal subset heuristically. These search methods are divided into deterministic and non-deterministic search algorithms. Deterministic methods such as SFS, SFFS use certain strategy for searching subset, while non-deterministic methods such as Genetic Algorithms (GA) [46] and Particle Swarm Optimization (PSO) based methods randomly select subsets [48]. The main disadvantage of wrappers is their higher computational cost and overfitting, which results in poor generalization for unseen data [10]. Moreover, due to their inability to scale to extremely high-dimensional data, wrappers are rarely used in practice.

Embedded methods incorporate feature selection as part of the training process and are considered more efficient. These methods provide a trade-off between filters and wrappers. Like wrappers they take into account interaction with the learning algorithm and are more efficient as they do not evaluate feature subsets iteratively. These methods iteratively create new classifiers by discarding a small proportion of features, until the smallest subset of features and highest predictive performance is achieved. However, selected features are classifier dependent and may not perform as well on other algorithms. Regularization models [49], a type of embedded approach, tend to fit a model by minimizing model fitting error and forcing variable coefficients to be small. Random forests [50] create multiple decision trees by drawing random samples from the data set along with different averaging techniques to improve accuracy. Recursive feature elimination for SVM (RFE-SVM) [52], recursively train an SVM classifier with current feature set and remove the least important features as evaluated by the classifier.

Majority of the existing studies have focused on improving individual methods, and have been proved to be effective on feature selection for training and testing data [3]. However, researchers suggest that there is no "one fit all" solution and majority of the efforts are focused on finding an optimal solution for a specific problem settings. Therefore, new methods are constantly appearing using different approaches. A small number of recent studies combined different feature selection techniques to improve results. In [54], features selected using PCA, Genetic Algorithm and Decision Trees are combined using union, intersection and multi-intersection for stock prediction, without providing insight as to why these particular methods are selected. In [3], Maximal Information Coefficient (MIC), Linear regression with L1 regularization and Group Interaction Lasso were combined for high-dimensional data, without taking into account multicollinearity among the predictors or trade-off between efficiency and accuracy. These studies prove, that combining feature subsets identified by different methods improves the overall quality of the selected features, and hence achieve higher predictive performance. In this study, we expand on the concept of multi-filter feature selection and develop a novel framework that takes into account feature relevance, redundancy and also provide a flexible mechanism for achieving a balance between efficiency and accuracy.

III. RELATED CONCEPTS

First, we formally describe the feature selection process for a given data set with a single response variable; followed by a brief overview of the feature selection techniques used and correlation-based clustering of variables. Consider a given data set $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$, where $\boldsymbol{x_i}$ is a d-dimensional vector and $\boldsymbol{y_i}$ is the associated class label. We denote the i^{th} instance of the data set by $x_1^i, x_2^i, ...x_d^i, y^i$, and consider we have n such i.i.d. instances. The goal of feature selection is to find a subset of features $\boldsymbol{x_i'}$, a d'-dimensional vector of variables associated with class label $\boldsymbol{y_i}$, where $d' << d$. For two variables $\{x_j, x_k\}$ in $\boldsymbol{x_i}$, x_j is considered more important, if it explains more variability in $\boldsymbol{y_i}$ than x_j.

A. LOGISTIC REGRESSION-BASED FEATURE RANKING

$L1$-LR is a well known regression model used in data analysis settings where the response variable is discrete. Recently, it has received significant attention as a feature selection method. In this study, we use the regression coefficients computed by $L1$-LR as variable importance measure to evaluate predictive abilities of input features. $L1$-LR uses linear combinations of predictors to learn class membership probability and construct an easily understandable linear model. It selects a subset of predictors and measures their predictive contribution in the model. The $L1$ penalty term is used to shrink the estimates of regression coefficients to zero and set some of them to zero. Therefore, it is considered a natural candidate for feature selection settings, where many features are considered irrelevant. Many studies have shown it to be an effective feature selection tool [55], [56]. $L1$ logistic regression is formulated as:

$$min_{(\alpha,\beta)}\ell = \frac{1}{n}\sum_{i=1}^{n}\log(1+\exp(-y_i(\beta^T x_i+\alpha_i))) + \lambda\sum_{i=1}^{n}|\beta_i| \tag{1}$$

where $\sum_{i=1}^{n}|\beta_i|$ denotes the $L1$ norm and $\lambda > 0$ is the given regularization parameter. β^T represents the regression coefficients and serve as variable importance measure for feature selection. $L1$-LR typically produces a sparse vector β, which means that β has few nonzero values. When $\beta_j = 0$, the concerned logistic model will discard the *jth* item of the input vector. Thus, sparse β can be considered as the selection of relevant features.

B. SVM-BASED FEATURE RANKING

SVM classifier is a supervised learning algorithm which classify samples by finding the maximum margin between bounding planes of different classes. Margin maximization is treated as an optimization problem which finds optimal hyperplane or decision boundary. Although, it can handle non-linear decision boundaries, this study is limited to linear SVM. SVM uses a small subset of training set called ''support vectors'' for training. Support vectors are training examples which are the closest to the decision boundary, and are also used to compute the weights w of the decision function. Considering our data set $\{\boldsymbol{x_i}, \boldsymbol{y_i}\}$, the SVM-train algorithm tend to minimize over α_k using the given equation:

$$J = \frac{1}{2}\sum_{jk} y_j y_k \alpha_j \alpha_k (x_j \cdot x_k + \lambda\delta_{jk}) - \sum_k \alpha_k$$

$$where \quad \sum_{k=1}\alpha_k y_k = 0 \; and \; 0 \le \alpha_k \le C \tag{2}$$

The summation run over all training samples x_i, where $x_i \cdot x_j$ denotes scalar product, δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if i = j, otherwise 0) and λ and C are soft margin parameters.

The resulting decision function of input vector $\boldsymbol{x}$ is given:

$$f(x) = w \cdot \boldsymbol{x} + b$$

$$where \quad w = \sum_k \alpha_k y_k x_k \; and \; b = y_k - w \cdot x_k \tag{3}$$

The weight matrix w of SVM classifier represents the regression coefficients of input variables and are used for feature ranking [38], [52], [53].

C. RANDOM FORESTS-BASED FEATURE RANKING

RF is an ensemble learning approach, which combines multiple decision trees (*ntree*) and are commonly used for classification and regression problems [57]. RF uses bagging to grow trees, in which a training set is randomly drawn from sample data without replacement. At each split a small subset of predictors (*mtry*) is randomly selected from the set of predictors, to be considered as candidates to split the tree. Feature that produces the best result is selected from the subset to split the tree. CART methodology is used to grow trees to maximum size without pruning. Observations which are left out during the tree construction are called out-of-bag (OOB) samples, and are used to estimate prediction error. Mode of classes and mean prediction of individual trees is used for classification and regression respectively. Although, RFs are considered computationally intensive, but has drawn much attention due to its non-parametric nature and its capacity to handle high-dimensional data [58].

Random forests also provide useful internal estimates of error, strength, correlation and variable importance according to their predictive abilities. The two commonly used importance measures are Gini and permutation importance or ''mean decrease in accuracy''. Permutation importance is considered to be more reliable and is calculated by change in prediction error when any association between the target variable and the concerned predictor is eliminated by permuting the values of a predictor. Less important variables will result in small or no change in prediction error, while more important variables will result in a larger change. As given in [59], variable importance is computed by comparing the prediction error before and after permuting the values of a variable.

$$VI_i = \frac{1}{ntree}\sum_{t=1}^{ntree}\frac{1}{|OOB_t|}\sum_{i\in OOB_t}\{E(y_i \neq \hat{y}_{it}^{*}) - E(y_i \neq \hat{y}_{it})\} \tag{4}$$

where E(.) denotes the error estimation function, OOB_t denotes the set of indices of observations which are out-of-bag for tree $t \in \{1, \ldots, ntree\}$, and $\hat{y}_{it}$ and $\hat{y}_{it}^{*}$ denote the predictions by the *t-th* tree before and after permuting the values of variable X_j, respectively.

D. CORRELATION-BASED CLUSTERING

The above mentioned techniques rank features according to their relevance to the output variable. However, they do not consider the presence of redundant features. Therefore, we construct clusters of variables having similar predictive abilities to identify any redundant features in the data set. We use affinity propagation [60], an exemplar-based clustering approach, which identifies data-points that best exemplify the data. Unlike k-means clustering [22], affinity propagation does not require the number of clusters to be prespecified,

but takes a preference parameter $\boldsymbol{p}$ for each data-point and simultaneously considers all features as potential exemplars. Each data point is considered a node in a network, and real-valued messages are recursively transmitted along the edges of the network until a good set of exemplars and their corresponding clusters emerge. Messages are updated on the basis of simple formulas that search for minima of an appropriately chosen energy function such as squared error. The preference parameter could be the median of input similarities or its multiple. A large value of $\boldsymbol{p}$ will result in more clusters, while a small value will produce few clusters. The number of clusters is influenced by the input preference and the message passing procedure. This method also requires a function or a collection of real-valued similarities as the criterion to measure correlation between each pair of features.

We use both linear correlation coefficients and information gain as similarity measures between variables to cluster them into groups, as variables in real-world problems may have different types of relationship. For a pair of random variables (X, Y), linear correlation coefficient r is given as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{5}$$

where $\bar{x}$ is the mean of x and $\bar{y}$ is the mean of y. The correlation coefficient is measured on a scale that varies from -1 to $+1$. A complete correlation is denoted by $+1$ or -1 and absence of correlation by 0.

Linear correlation is only sensitive to linear relationship between two variables, which may not be the case in some real-world examples. Therefore, we also adopt information gain to quantify non-linear relationship. In information theory, the expected value of information gain about one random variable in a pair (X, Y) is obtained by observing the other. The calculation of information gain is based on information-theoretic concept of entropy, which for a given variable X is defined as:

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2(P(x_i)) \tag{6}$$

and the entropy of X given Y is defined as:

$$H(X|Y) = -\sum_{i}\sum_{j} P(y_j)P(x_i|y_j) log_2(P(x_i|y_j)) \tag{7}$$

where $P(y_j)$ are the probabilities of y and $P(x_i|y_j)$ are the probabilities of X given Y. With the definition of $H(X)$ and $H(X|Y)$ information gain can be given as:

$$I(X|Y) = H(X) - H(X|Y) \tag{8}$$

From the above equation, we conclude that information gain is the amount by which the uncertainty or entropy of X decreases, by obtaining additional information about X by observing Y. Information gain is symmetric, thus $I(X|Y) = I(Y|X)$.

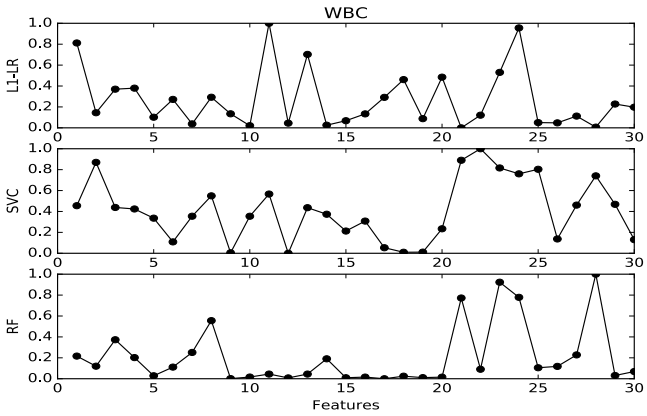

FIGURE 1. *L*1-LR, SVC and RF assigned [11,24,1], [22,21,2] and [28,23,24] the highest importance respectively, while input features [10,28,21], [18,9,12] and [12,9,17] are assigned the lowest importance.

IV. METHODOLOGY

As previously mentioned, the quality of input features can be assessed by their relevance to the output variable and lack of redundancy in the feature set. Therefore, we develop a two-staged modular approach, which consists of a feature-ranking module and a clustering module. The ranking module computes the relevance of features by identifying features which can clearly discriminate between different classes, whereas the clustering module produces clusters of features having similar information to identify any redundant features. Feature ranking is used as a principle criteria by many feature selection methods, and is commended for its simplicity, scalability, and good empirical results [42]. Different feature-ranking methods evaluate variable importance differently based on their assumption about the regression function linking the predicted variable and the predictors. Therefore, to encompass multiple different assumptions about the underlying regression function, we select $L1$-LR, SVM and RF for feature ranking.

Although, the selected feature-ranking approaches have better performance and are considered to have small number of hyperparameters, their selection is primarily based on having disjoint assumptions about the regression function linking the predicted variable with predictors [3]. For example $L1$-LR assumes the regression function to be linear, while RF considers variables which may statistically interact in their effect on the target variable [3]. In case of $L1$-LR and SVM, there is a partial overlapping about the assumption as both assumes linearity. However, the resulting variable importance is not similar. Fig. 1 shows Wisconsin Breast Cancer (WBC) [62] data set ranked by the three component methods of the proposed approach.

Before training $L1$-LR, SVM and RF, each data set is split into train and test sets. Input features of each train set are scaled to [0, 1], and the same scaling parameters are applied to their corresponding test sets. For each data set, the three filter methods are independently trained using the training set, and a variable importance $VI_m \in \mathbb{R}^d$ is computed by $\boldsymbol{FeatureRanking_m}$, where the value $|VI_{m,j}|$ represents the

relevance of *jth* feature to output variable $\boldsymbol{y}$, computed by filter method m.

After the features are ranked, it is important to establish a criteria to drop features which are considered less important. To filter out less important features, [63] used the largest gap between consecutive ranked features, [64] dropped features whose weights are two variances further than the mean. Filtering out less important features is an important step and have significant effect on the subsequent steps. Instead of using variance in variable importance as a threshold, we use a user given integer value $\boldsymbol{e}$, as the number of lowest ranked features being dropped from each of the three ranked lists. This approach is simple and will ensure fairness by returning the same number of features in each list. Elimination of low-ranked features will produce slightly different sets for each ranking method and will result in constructing different clusters as well [34]. For extremely high-dimensional data sets, we suggest normalization of VI and using variance or its multiple as a threshold for filtering-out irrelevant features.

Algorithm 1 Multi-Filter Feature Selection Algorithm

Input: Input data $S := \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is a d-dimensional vector

Output: Output data $S' := \{\{(x_1, y_1), \ldots, (x_n, y_n)\}$ where x_i is a d'-dimensional vector ▷ where $d' << d$

procedure MFFS(S, $\boldsymbol{e}$, $\boldsymbol{p}$, $\boldsymbol{c}$)
  delegates $\leftarrow \emptyset$
  $\boldsymbol{SF} \leftarrow \emptyset$ ▷ selected features
  $xtrain, ytrain, xtest, ytest \leftarrow$ preprocess(S)
  $VI_{lr} \leftarrow \boldsymbol{FeatureRanking_{lr}}$ ($xtrain, ytrain$)
  $VI_{svm} \leftarrow \boldsymbol{FeatureRanking_{svm}}$ ($xtrain, ytrain$)
  $VI_{rf} \leftarrow \boldsymbol{FeatureRanking_{rf}}$ ($xtrain, ytrain$)
  for each $VI \in [VI_{lr}, VI_{svm}, VI_{rf}]$ **do**
    $xtrain \leftarrow xtrain$.drop($VI, \boldsymbol{e}$)
    $\boldsymbol{sm} \leftarrow$ Correlation($xtrain, \boldsymbol{c}$)
    clusters $\leftarrow$ AffinityPropagation($\boldsymbol{sm}, \boldsymbol{p}$)
    delegates $\leftarrow$ get-delegates(VI, clusters)
    for each item in delegates **do**
      if item $\notin$ SF **then**
        SF.append(item)
      end if
    end for
  end for
  $xtrain' \leftarrow xtrain[\boldsymbol{SF}]$
  $xtest' \leftarrow xtest[\boldsymbol{SF}]$
  $S' \leftarrow$ concatenate($xtrain', xtest'$)
  return S'
end procedure

Feature ranking methods are mainly criticized for its poor handling of redundant variables [18], [19]. Therefore, after ranking input features and subsequent elimination of low-ranked features, the remaining features are divided into groups based on their correlation among themselves, so that features with similar characteristics are grouped together. For this purpose, we first compute a similarity matrix $\boldsymbol{sm}$ based on correlation type $\boldsymbol{c}$. The similarity matrix $\boldsymbol{sm}$ and preference parameter $\boldsymbol{p}$ are used as inputs to affinity propagation for clustering variables. The highest ranked feature is then selected from each cluster as a delegate, and the three sets of delegates are combined together using union operation. Thus, the selected features are not strongly correlated with each other, while have high predictive ability as compared to its peers in the same cluster. Together, the elimination parameter $\boldsymbol{e}$ and preference $\boldsymbol{p}$ provide a flexible mechanism to achieve a trade-off between efficiency and accuracy by controlling the number of features dropped and clusters produced. Algorithm 1. presents a step-by-step functioning of the proposed approach.

$\boldsymbol{e}$: number of low-ranked features dropped.
$\boldsymbol{p}$: preference parameter of affinity propagation.
$\boldsymbol{sm}$ is the similarity matrix computed using correlation type $\boldsymbol{c}$.

TABLE 1. Summary of data sets.

Data Set	Instances	Features	Classes
Ionosphere	351 (200/151)	34	2
WBC	569	30	2
Sonar	208 (104/104)	60	2
Wine	178 (142/36)	13	3
Vowels	990 (528/462)	10	11

Note x/y: Number of train samples / Number of test samples

V. SIMULATIONS AND RESULTS

In this study, we use five data sets namely Ionosphere, Wisconsin Breast Cancer (WBC), Sonar, Wine, and Vowels, downloaded from UCI machine learning database [62]. These data sets have been predominantly used in machine learning studies and cover a variety of different real-world problems. Prior to feature ranking the data sets were split into training and testing sets using the ratio given by the contributors of the data sets and then scaled to [0, 1]. In case of WBC, where no ratio is given 80% of the samples are used for training and 20% for testing. Table 1. provides the total number of instances, number of features and number of classes for each data set.

As per our proposed approach, input features are first ranked using the three selected filter methods. The value of parameter *ntree* for RF and C for $L1$-LR and SVC are tuned using k-fold cross-validation, where k = 10. In order to include negatively correlated features, we consider their absolute values. Although, the value of $\boldsymbol{e}$ can range from 0 to $\boldsymbol{d} - 1$, but in practice eliminating up to 5% of the lowest features can achieve optimal results, as more features are dropped in the clustering step.

The remaining features are clustered into groups based on their correlation using affinity propagation. In clustering module the preference parameter $\boldsymbol{p}$ is the median of correlation coefficients for each variable and influence the number of clusters produced. Variables with large value of $\boldsymbol{p}$ are more likely to be selected as exemplars, while variables with small

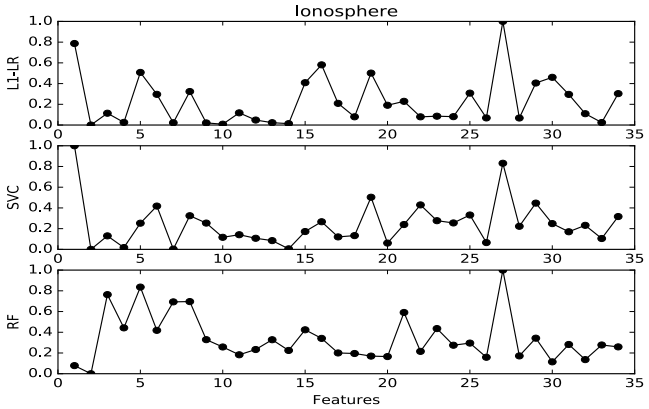

FIGURE 2. *L*1-LR, SVC and RF ranked [27,1,16], [1,27,9] and [27,5,3] the most important input features and ranked [2,10,14], [2,7,14] and [2,1,30] the least important.

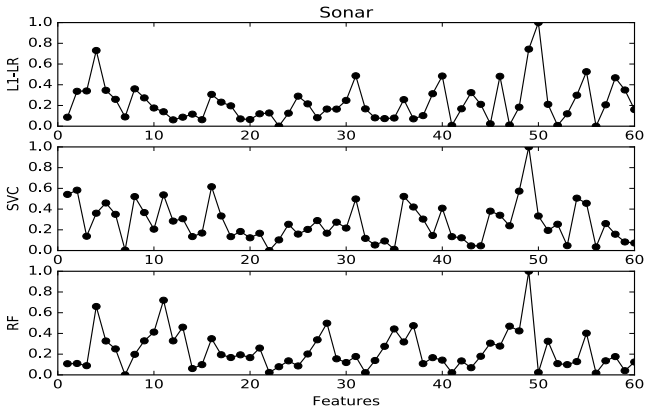

FIGURE 3. *L*1-LR, SVC and RF ranked [50,49,4], [49,16,2] and [49,11,4] the most important input features and ranked [56,47,23], [19,3,52] and [56,33,59] the least important.

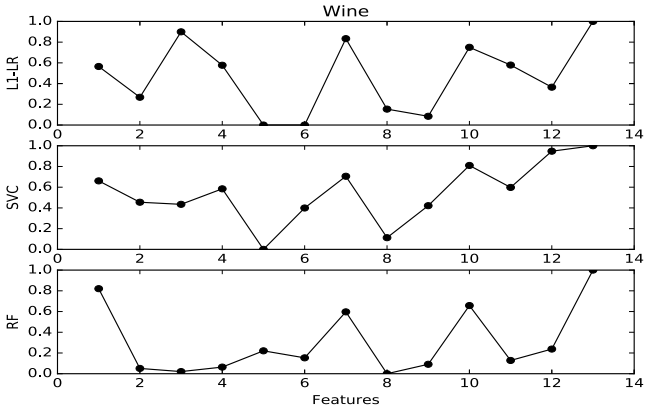

FIGURE 4. *L*1-LR, SVC and RF ranked [13,3,7], [13,12,10] and [13,1,10] the most important input features and ranked [5,6,9], [5,8,6] and [8,3,2] the least important.

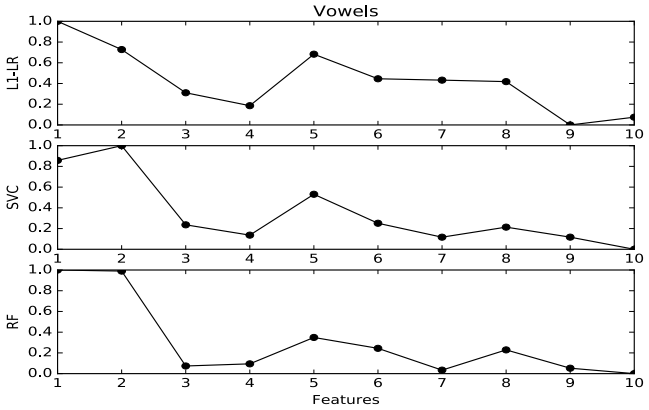

FIGURE 5. *L*1-LR, SVC and RF ranked [1,2,5], [1,2,5] and [1,2,5] the most important input features and ranked [9,10,4], [10,7,9] and [10,7,9] the least important.

$\boldsymbol{p}$ are less likely. Values of parameter p and e provides a flexible mechanism for increasing or decreasing the number of features to be selected. Unlike SFS, SFFS or PCA, it does not need to be given the exact number of features to be selected, while providing flexibility to increase or decrease the number of features to be selected. Based on the variable importance vector VI, the highest ranked feature is selected from each cluster as a delegate and included in the final feature set.

A. RESULTS

Results of the proposed approach over the five selected data sets are presented. The results confirm our hypothesis, that combining features selected by methods relying on disjoint assumptions about the regression function linking the input variables and the class label produces a more reliable feature set and improves prediction accuracy. In this section, correlation-based linear correlation-based Multi-Filter Feature Selection is denoted by MFFS-*lr* and information gain-based Multi-Filter Feature Selection is denoted by MFFS-*IG*, Similarly, the three feature ranking methods with corresponding similarity measure are also abbreviated.

Variable importance for Ionosphere, Sonar, Wine and Vowel data sets are presented in Fig. 2, Fig. 3, Fig. 4, and Fig. 5 respectively. The difference in variable importance as computed by each method can be clearly observed. The difference is more evident in high-dimensional data sets as compared to low-dimensional sets. The lowest ranked features are filtered-out and the remaining features are clustered into groups.

Finally, the highest ranked feature from each cluster is selected and added to the final feature set. Table 2. provides selected features for each of the five data sets along with the parameters used to obtain them. $\mathbf{e}$, p' and $\boldsymbol{d'}$ is the number of low-ranked features dropped, multiplier of preference parameter for constructing clusters and the reduced number of features respectively. Parameters $\boldsymbol{e}$ and $\boldsymbol{p'}$ are user provided parameters and are used to control the number of features selected.

In order to demonstrate the effectiveness of our proposed approach, we perform comparative analysis with each of the three component methods and other feature selection techniques such as Information Gain, ReliefF [12], SVM-GA [65], PSO-SVM [66], SFS, SFFS [68], PCA and CoV/VSURF [39]. Although, it is impractical to compare

TABLE 2. Selected features.

Data Set	FS Approach	e	p'	d'	Selected Features
Ionosphere	MFFS-lr	2	2	15	1,4,6,7,8,14,15,16,20,21,22,24,27,30,34
	MFFS-IG	3	5	18	1,3,5,6,8,9,15,16,17,18,19,21,22,23,24,27,30,34
WBC	MFFS-lr	0	2	23	2,5,6,7,8,9,10,11,12,14,15,16,17,18,19,20,22,24,25,27,28,29,30
	MFFS-IG	0	2	15	2,5,6,7,11,12,14,16,17,20,22,24,25,29,30
Sonar	MFFS-lr	2	3	30	4,5,6,8,9,10,11,16,17,21,22,24,25,26,28,30,31,32,36,37,39,40,43,44,46,49,50,55,58,59
	MFFS-IG	2	5	37	1,2,3,4,5,6,8,10,11,12,14,16,17,21,22,25,27,28,30,31,36,37,38,39,40,42,43,46,49,50,51,52,53,54,55,57,59
Wine	MFFS-lr	2	1	6	3,4,5,7,12,13
	MFFS-IG	2	2	7	1,3,5,9,10,11,13
Vowels	MFFS-lr	2	8	8	1,2,3,4,5,6,8,9
	MFFS-IG	2	8	9	1,2,3,4,5,6,7,8,9

e is the number of lowest ranked features dropped
p' is a real number to multiply preference parameter p

TABLE 3. Prediction accuracy.

Approach	Ionosphere		WBC		Sonar		Wine		Vowels	
	d'	Accuracy(%)	d'	Accuracy(%)	d'	Accuracy(%)	d'	Accuracy(%)	d'	Accuracy(%)
Originial Set	34	98.0	30	98.2	60	83.6	13	97.2	10	62.3
GA	16	98.0	16	98.2	27	79.8	6	97.2	8	61.5
PSO-SVM	25	98.0	20	97.3	33	79.8	10	97.2	9	63.8
SFS	30	98.0	15	95.6	37	77.9	10	**100**	9	60.0
SFFS	29	98.0	13	99.1	35	81.7	10	**100**	9	60.0
PCA	25	98.0	17	97.3	34	85.6	6	97.2	9	62.8
CoV/VSURF	6	98.0	8	97.3	7	87.5	8	**100**	8	63.8
InfoGain	23	98.0	18	98.2	32	86.5	6	**100**	9	**64.7**
ReliefF	18	98.0	26	99.1	30	86.5	8	**100**	8	63.0
$L1$-LR-lr	25	97.3	26	98.2	15	91.3	6	**100**	5	51.7
SVM-lr	12	97.3	12	98.2	14	90.4	3	**100**	3	53.2
RF-lr	18	98.0	13	99.1	16	81.7	4	**100**	7	62.3
Logit-IG	9	96.7	10	97.4	14	89.4	6	**100**	8	61.2
SVM-IG	10	98.0	12	98.2	20	80.8	3	**100**	6	62.7
RF-IG	10	96.7	10	99.1	17	85.6	4	**100**	8	62.3
MFFS-lr	15	**98.7**	23	**100**	30	**92.3**	6	**100**	7	62.3
MFFS-IG	18	97.3	15	99.1	37	90.3	7	**100**	9	**64.7**

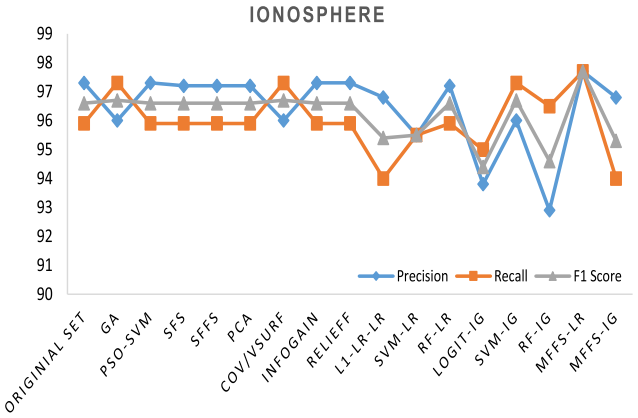

FIGURE 6. MFFS-*lr*: Precision 97.7, Recall 97.7 and F1 Score 97.7.

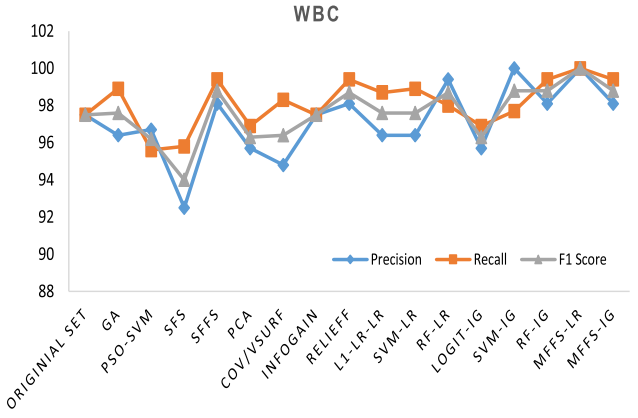

FIGURE 7. MFFS-*lr*: Precision 100, Recall 100 and F1 Score 100.

with every existing techniques, we have selected some of the commonly used techniques.

Classification accuracy, precision, recall and F1 score are used as evaluation metrics [70]. From Table 3., the results demonstrate that MFFS-*lr* has achieved the best results for Ionosphere, WBC, Sonar and Wine data sets. MFFS-*IG* has achieved highest accuracy for Wine and Vowels data set and is the second best for WBC with 0.9% decrease in accuracy. Information gain based feature selection has performed well for Wine and Vowels, while having the second best performance for Ionosphere and WBC, slightly lower than MFFS-*IG*. Feature selection methods, which randomly select feature subsets such as GA and PSO-SVM achieved the lowest accuracy as compared to other techniques. Overall,

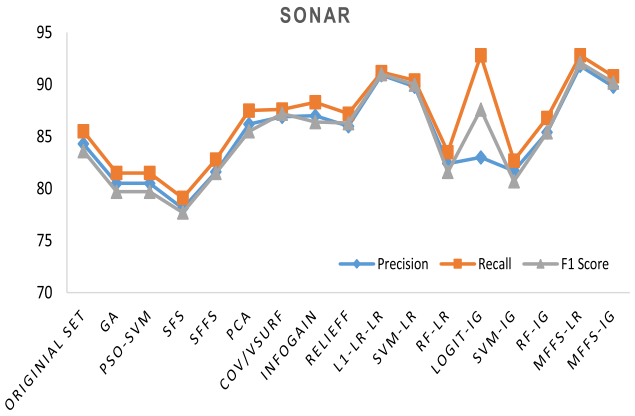

FIGURE 8. MFFS-*lr*: Precision 91.8, Recall 92.8 and F1 Score 92.1.

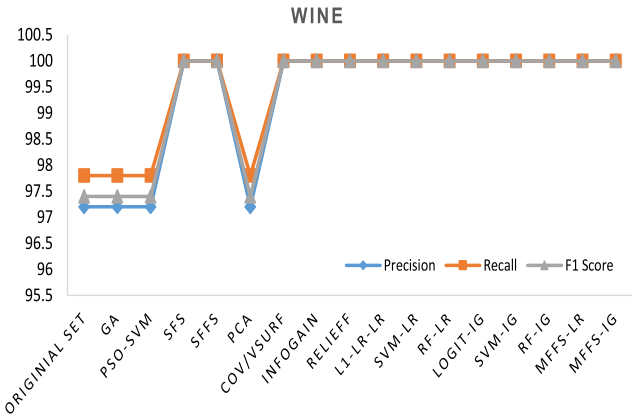

FIGURE 9. MFFS-*lr*: Precision 100, Recall 100 and F1 Score 100.

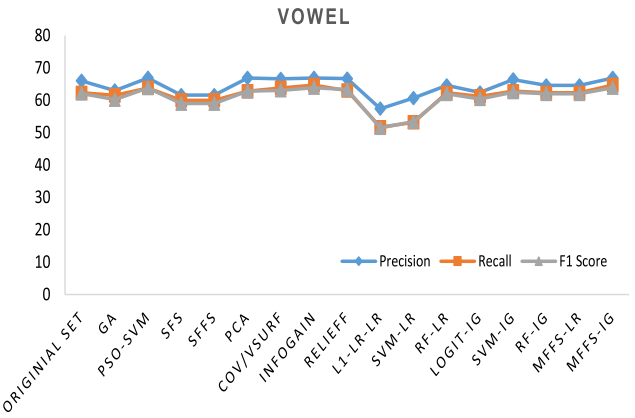

FIGURE 10. MFFS-*lr*: Precision 64.6, Recall 62.3 and F1 Score 62.0.

correlation-based feature selection approaches have performed better than other methods. Although, CoV/VSURF has selected the lowest number of features, but it does not provide flexible mechanism for controlling the number of features to be selected, which results in lower prediction accuracy. MFFS-*lr* not only proved to outperform other approaches in terms of classification accuracy, but also performed consistently well for majority of the data sets. Precision, recall and F1 score for the five data sets are given in Fig. 6, Fig. 7, Fig. 8, Fig. 9 and Fig. 10. These evaluation measures also demonstrate better predictive performance of MFFS-*lr* and MFFS-*IG* as compared to other feature selection techniques evaluated.

VI. CONCLUSION

In this paper, we present multi-filter feature selection approach (MFFS), which combines multiple feature-ranking techniques with clustering of variables to select an optimal features set. Three different filter methods are used for computing variable importance as filter methods are computationally less intensive and are independent of the classification model. Different feature ranking methods rank features differently as each method has different assumption about the underlying regression function linking input variables with the output. Therefore, combining multiple heterogeneous methods for feature ranking exploits different assumptions about the underlying relationship between input variables and class label. Those features which are ranked lower than the given threshold are filtered-out. The remaining features are grouped into clusters using an exemplar-based clustering algorithm called affinity propagation. Affinity propagation requires a similarity measure between variables and a preference parameter to group features into clusters. We use both linear correlation coefficients and information gain as similarity measures between input features. The number of clusters can be controlled using the preference parameter and the number of features dropped, which eventually effect the number of features to be selected. The highest ranked variable is selected from each cluster and included in the final subset. Experimental results confirm that the proposed approach selects features, which are more informative and diverse resulting in improved accuracy of the prediction model. Elimination of less important features is an important step and in some cases highly effect the prediction accuracy. Therefore, threshold criteria need to be further studied. The union operation also results in larger subset and need further consideration. Overall, the study validates the hypothesis that combining features selected using different feature selection methods results in an efficient subset and improves predictive performance of the learning model.

REFERENCES

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

[2] J. Fan and Y. Fan, "High dimensional classification using features annealed independence rules," *Ann. Statist.*, vol. 36, no. 6, pp. 2605–2637, 2008.

[3] A. M. Mihaela, W. Shicai, and G. Yike, "Combining multiple feature selection methods and deep learning for high-dimensional data," *Trans. Mach. Learn. Data Mining*, vol. 9, no. 1, pp. 27–45, 2016.

[4] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.

[5] J. M. Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognit.*, vol. 43, no. 6, pp. 2068–2081, 2010.

[6] M. H. C. Law, M. A. T. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1154–1166, Sep. 2004.

[7] Z. M. Hira and D. F. Gillies, "A review of feature selection and feature extraction methods applied on microarray data," *Adv. Bioinf.*, vol. 2015, May 2015, Art. no. 198363.
[8] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[9] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.
[10] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, 2007.
[11] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint ℓ_2, 1-norms minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.
[12] Y. Sun and J. Li, "Iterative RELIEF for feature weighting," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 913–920.
[13] B. Chidlovskii and L. Lecerf, "Scalable feature selection for multi-class problems," in *Machine Learning and Knowledge Discovery in Databases* (Lecture Notes in Computer Science). 2008, pp. 227–240.
[14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., B (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[15] V. Vapnik, *The Nature of Statistical Learning Theory*. 1995.
[16] M. Rong, D. Gong, and X. Gao, "Feature selection and its use in big data: Challenges, methods, and trends," *IEEE Access*, vol. 7, pp. 19709–19725, 2019.
[17] S. Vajda, A. Karargyris, S. Jaeger, K. C. Santosh, S. Candemir, Z. Xue, S. Antani, and G. Thoma, "Feature selection for automatic tuberculosis screening in frontal chest radiographs," *J. Med. Syst.*, vol. 42, p. 146, Aug. 2018.
[18] R. Genuer, J. M. Poggi, and C. Tuleau-Malot, "VSURF: An R package for variable selection using random forests," *R J., R Found. Stat. Comput.*, vol. 7, no. 2, pp. 19–33, 2015.
[19] Y.-W. Chang and C.-J. Lin, "Feature ranking using linear SVM," in *Causation and Prediction Challenge*, vol. 31. 2008, pp. 53–64.
[20] A. Amin, S. Anwar, A. Adnan, M. Nawaz, K. Alawfi, A. Hussain, and K. Huang, "Customer churn prediction in the telecommunication sector using a rough set approach," *Neurocomputing*, vol. 237, pp. 242–254, May 2017.
[21] S. Vajda and K. C. Santosh, "A fast k-nearest neighbor classifier using unsupervised clustering," in *Proc. Int. Conf. Recent Trends Image Process. Pattern Recognit.*, 2016, pp. 185–193.
[22] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.
[23] A. Amin, B. Shah, A. M. Khattak, F. Joaquim, L. Moreira, G. Ali, A. Rocha, and S. Anwar, "Cross-company customer churn prediction in telecommunication: A comparison of data transformation methods," *Int. J. Inf. Manage.*, vol. 46, pp. 304–319, Jun. 2019.
[24] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinformat.*, vol. 9, no. 4, pp. 1106–1119, Jul./Aug. 2012.
[25] B. M. Khammas, A. Monemi, J. S. Bassi, I. Ismail, S. M. Nor, and M. Marsono, "Feature selection and machine learning classification for malware detection," *J. Teknologi*, vol. 77, no. 1, pp. 243–250, 2015.
[26] Q. Jiang, X. Zhao, and K. Huang, "A feature selection method for malware detection," in *Proc. IEEE Int. Conf. Inf. Automat.*, Jun. 2011, pp. 890–895.
[27] K.-C. Khor, C.-Y. Ting, and S.-P. Amnuaisuk, "A feature selection approach for network intrusion detection," in *Proc. Int. Conf. Inf. Manage. Eng.*, Apr. 2009, pp. 133–137.
[28] Y. Xu, Z. Li, and L. Luo, "A study on feature selection for trend prediction of stock trading price," in *Proc. Int. Conf. Comput. Inf. Sci.*, Jun. 2013, pp. 579–582.
[29] J. Chen, H. Huang, S. Tian, and Y. Qu, "Feature selection for text classification with Naïve Bayes," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 5432–5435, 2009.
[30] O. Egozi, E. Gabrilovich, and S. Markovitch, "Concept-based feature generation and selection for information retrieval," in *Proc. 23rd AAAI Conf. Artif. Intell. (AAAI)*, 2008, pp. 1132–1137.
[31] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen, "Unsupervised feature selection applied to content-based retrieval of lung images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 3, pp. 373–378, Mar. 2003.
[32] J. Pickens, "A survey of feature selection techniques for music information retrieval," in *Proc. 2nd Int. Symp. Music Inf. Retr. (ISMIR)*, vol. 124, Sep. 2001, pp. 1–6.
[33] N. Hoque, D. Bhattacharyya, and J. K. Kalita, "MIFS-ND: A mutual information-based feature selection method," *Expert Syst. Appl.*, vol. 41, no. 14, pp. 6371–6385, 2014.
[34] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, 2013.
[35] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, 2016, Art. no. 94.
[36] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proc. Nat. Conf. Artif. Intell.*, vol. 2, 1992, pp. 129–134.
[37] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992*. 1992, pp. 249–256.
[38] B. Li, Q. Wang, and J. Hu, "Feature subset selection: A correlation-based SVM filter approach," *IEEJ Trans. Elect. Electron. Eng.*, vol. 6, no. 2, pp. 173–179, 2011.
[39] M. Chavent, R. Genuer, and J. Saracco, "Combining clustering of variables and feature selection using random forests," *Commun. Statist.-Simul. Comput.*, pp. 1–20, Feb. 2019.
[40] J. Tang, X. Hu, H. Gao, and H. Liu, "Discriminant analysis for unsupervised feature selection," in *Proc. SIAM Int. Conf. Data Mining*, 2014, pp. 938–946.
[41] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. ICML*, 1995, pp. 284–292.
[42] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, no. 3, pp. 1157–1182, Mar. 2003.
[43] A. Shishkin, A. Bezzubtseva, A. Drutsa, I. Shishkov, E. Gladkikh, G. Gusev, and P. Serdyukov, "Efficient high-order interaction-aware feature selection based on conditional mutual information," in *Proc. NIPS*, 2016, pp. 4637–4645.
[44] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural Comput. Appl.*, vol. 24, no. 1, pp. 175–186, Jan. 2014.
[45] S. H. Huang, "Supervised feature selection: A tutorial," *Artif. Intell. Res.*, vol. 4, no. 2, p. 22, 2015.
[46] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, nos. 2–3, pp. 95–99, 1988.
[47] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov./Dec. 1995, pp. 1942–1948.
[48] J. Vashishtha, "Particle swarm optimization based feature selection," *Int. J. Comput. Appl.*, vol. 146, no. 6, pp. 11–17, 2016.
[49] S.-I. Lee, H. Lee, P. Abbeel, and A. Y. Ng, "Efficient L_1 regularized logistic regression," in *Proc. AAAI*, vol. 6, 2006, pp. 401–408.
[50] R. Diaz-Uriarte and S. A. de Andrés, "Gene selection and classification of microarray data using random forest," *BMC Bioinf.*, vol. 7, Jan. 2006, Art. no. 3.
[51] R. Genuer, J.-M. Poggi, and C. Tuleau-Malot, "Variable selection using random forests," *Pattern Recognit. Lett.*, vol. 31, no. 14, pp. 2225–2236, Oct. 2010.
[52] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Mach. Learn.*, vol. 46, pp. 1–3, pp. 389–422, 2002.
[53] A. Rakotomamonjy, "Variable selection using SVM-based criteria," *J. Mach. Learn. Res.*, vol. 3, pp. 1357–1370, Mar. 2003.
[54] C.-F. Tsai and Y.-C. Hsiao, "Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches," *Decis. Support Syst.*, vol. 50, no. 1, pp. 258–269, 2010.
[55] A. Y. Ng, "Feature selection, L_1 vs. L_2 regularization, and rotational invariance," in *Proc. 21st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 78.
[56] R. Zakharov and P. Dupont, "Ensemble logistic regression for feature selection," in *Pattern Recognition in Bioinformatics* (Lecture Notes in Computer Science). 2011, pp. 133–144.
[57] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
[58] B. F. F. Huang and P. C. Boutros, "The parameter sensitivity of random forests," *BMC Bioinf.*, vol. 17, no. 1, 2016, Art. no. 331.
[59] S. Janitza, E. Celik, and A.-L. Boulesteix, "A computationally fast variable importance test for random forests for high-dimensional data," *Adv. Data Anal. Classification*, vol. 12, no. 4, pp. 885–915, 2018.

[60] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[61] I. Kononenko, "Estimating attributes: Analysis and extensions of RELIEF," in *Proc. Eur. Conf. Mach. Learn.*, 1994, pp. 171–182.
[62] D. Dua and C. Graff, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, CA, USA, Tech. Rep., 2019. [Online]. Available: http://archive.ics.uci.edu/ml
[63] L. A. Belanche and F. F. González, "Review and evaluation of feature selection algorithms in synthetic problems," 2011, *arXiv:1101.2320*. [Online]. Available: https://arxiv.org/abs/1101.2320
[64] M. Mejía-Lavalle, E. Sucar, and G. Arroyo, "Feature selection with a perceptron neural net," in *Proc. Int. Workshop Feature Selection Data Mining*, 2006, pp. 131–135.
[65] G. R. Kumar, G. A. Ramachandra, and K. Nagamani, "An efficient feature selection system to integrating SVM with genetic algorithm for large medical datasets," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 4, no. 2, pp. 272–277, 2014.
[66] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, Feb. 2015.
[67] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1100–1103, Sep. 1971.
[68] P. Pudil, J. Novovičová, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognit. Lett.*, vol. 15, no. 11, pp. 1119–1125, 1994.
[69] D. W. Aha, "Incremental, instance-based learning of independent and graded concept descriptions," *Proc. 6th Int. Workshop Mach. Learn.*, 1989, pp. 387–391.
[70] A. Amin, S. Anwar, A. Adnan, M. Nawaz, N. Howard, J. Qadir, A. Hawalah, and A. Hussain, "Comparing oversampling techniques to handle the class imbalance problem: A customer churn prediction case study," *IEEE Access*, vol. 4, pp. 7940–7957, 2016.

ANWAR UL HAQ received the bachelor's degree in computer science from the Department of Computer Science, University of Peshawar, in 2005, and the master's degree from the Department of Computer Science, University of Manchester, U.K., in 2007. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Xiamen University. His research interests include machine learning, decision support systems, and financial data mining.

DEFU ZHANG (M'16) received the bachelor's and master's degrees in computational mathematics from Xiangian University, in 1996 and 1999, respectively, and the Ph.D. degree in computer software and its theory from the School of Computer Science, Huazhong University of Science and Technology. He was a Senior Researcher with the Shanghai Jinxin Financial Engineering Academy, from 2002 to 2003. He was a Postdoctoral Researcher with the Longtop for Financial Data Mining Group, from 2006 to 2008. From 2008 to 2016, he visited the City University of Hong Kong, the University of Wisconsin–Madison, and the University of Macao. He has also developed an Internet and a big data platform. He has supervised the ACM/ICPC Team, Xiamen University. He is currently a Professor with the Department of Computer Science, Xiamen University. He has published over 40 journal articles. His research interests include computational intelligence, data mining, big data, online decision optimization, and food security. He has participated in the World Final Contest, in 2007. He has received three gold medals and eight silver medals, from 2004 to 2009.

HE PENG received the master's degree in bioinformatics from the School of Life Sciences, Sun Yat-sen University, Guangzhou, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Information Science and Engineering, Xiamen University. His research interests include representation learning and its applications in bioinformatics.

SAMI UR RAHMAN received the bachelor's degree in computer science from the University of Peshawar, Pakistan, in 2001, the master's degree in image processing from Albert–Ludwig University, Germany, in 2009, and the Ph.D. degree in medical imaging from the University of Technology Darmstadt, Germany, in 2012. He joined the University of Malakand, as a Lecturer, in 2003, where he became an Assistant Professor, in 2013. In 2016, he taught with Stratford University, USA, as a Visiting Faculty. His current research interests include image processing, medical imaging, and education visualization.