A General Framework for Class Label Specific Mutual Information Feature Selection Method

Deepak Kumar Rakesh and Prasanta K. Jana, *Senior Member, IEEE*

***Abstract*— Information theory-based feature selection (ITFS) methods select a single subset of features for all classes based on the following criteria: 1) minimizing redundancy between the selected features and 2) maximizing classification information of the selected features with the classes. A critical issue with selecting a single subset of features is that they may not represent the feature space in which individual class labels can be separated exclusively. Existing methods fail to provide a way to select the feature space specific to the individual class label. To this end, we propose a novel feature selection method called class-label specific mutual information (CSMI) that selects a specific set of features for each class label. The proposed method maximizes the information shared among the selected features and target class label but minimizes the same with all classes. We also consider the dynamic change of information between selected features and the target class label when a candidate feature is added. Finally, we provide a general framework for the CSMI to make it classifier-independent. We perform experiments on sixteen benchmark data sets using four classifiers and found that the CSMI outperforms five traditional, two state-of-the-art ITFS (multi-class classification), and one multi-label classification methods.**

***Index Terms*— Feature selection, filter method, information theory, class label specific mutual information, classification.**

I. Introduction

THERE has been an explosion in data generation from industry, business transactions, sensors, smart devices, and social media [1]. Such a high volume of data needs to analyze for retrieving useful information that can be used for decision support, forecasting, recommendation, and business intelligence. High-dimensional feature space is common in all these applications. However, any such feature space may include many redundant and irrelevant features, which increases the data processing complexity. Feature selection (FS) is one of the most effective tools which can eliminate such irrelevant and redundant features while retaining the relevant ones [2], [3]. It refers to the process of selecting a subset of features from the original data set by preserving the actual characteristics of the data [4]. However, in the course of FS, the relationship between the selected features, the candidate feature (i.e., a new feature to be selected), and the class labels tends to change, which may affect the data analysis and thus need to be addressed properly.

FS methods are broadly classified into three groups: Wrapper methods [5], [6], Embedded methods [5] and Filter methods [7], [8]. The filter methods have been paid much attention due to their generalization capability and low cost of computation. These methods select the most informative features based on the statistical properties of the data. As of now, many filter-based feature selection methods have been proposed based on different selection criteria, such as correlation [9], [10], consistency [11] and mutual information [7], [12]. Among all, information theory-based filters are extensively studied [13]. These methods are effective in evaluating linear as well as nonlinear relations among features [14], [15].

Information theory-based feature selection (ITFS) methods search for an optimal subset of features to maximize the mutual information between the selected features and the class labels. However, they work satisfactorily with lower feature dimensions, they are not much reliable with high-dimensional data sets. Hence, researchers [16]–[19] made efforts to improve the ITFS based on the following optimizing criteria:

1) Maximizing relevancy between the features and the class labels.
2) Minimize redundancy among the features.
3) Maximizing conditional redundancy of the features, given the class labels.

Combining all these criteria, Brown et al. [20] proposed the following general framework for the ITFS:

$$J(f_k) = I(f_k;C) \; - \; \alpha \sum_{f_j \in S} I(f_j;f_k) \; + \; \beta \sum_{f_j \in S} I(f_j;f_k|C) \tag{1}$$

where f_k is a candidate feature, f_j represents a selected feature, and C represents the class labels, α and β are two nonnegative parameters, whose values belong to [0, 1]. The first, second, and third terms of the equation represent the concepts of *relevancy*, *redundancy*, and *conditional redundancy*, respectively. It is worth noticing that J decreases as the redundancy increases; nevertheless, when the value of conditional redundancy increases, J increases. This concludes that the inclusion of correlated features are beneficial if the correlation across the classes is stronger than the overall correlation [20]. Table V presents various ITFS methods proposed so far, most

Manuscript received 13 May 2021; revised 13 May 2022; accepted 25 June 2022. Date of publication 6 July 2022; date of current version 22 November 2022. *(Corresponding author: Deepak Kumar Rakesh.)*

The authors are with the Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, Dhanbad 826004, India (e-mail: rakesh.deepak12@gmail.com).

Communicated by C. Suh, Associate Editor for Machine Learning and Statistics.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIT.2022.3188708.

Digital Object Identifier 10.1109/TIT.2022.3188708

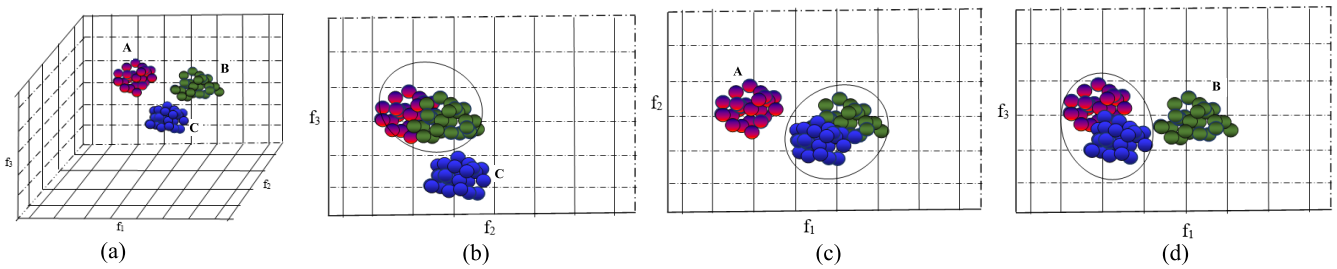


Fig. 1. Conceptual visualization of the proposed class label-based feature space selection.

TABLE I
TOY DATA

f_1	f_2	f_3	C
1	1	0	0
0	1	0	1
0	0	1	0
1	0	0	1
1	0	1	0

TABLE II
MIM-BASED FS

All classes	Class 0	Class 1
$J_{MIM}(f_1) = 0.3219$	$J_{MIM}(f_1) = 0.2516$	$J_{MIM}(f_2) = 0.3113$
$J_{MIM}(f_3) = 0.4200$	$J_{MIM}(f_2) = 0.2516$	$J_{MIM}(f_3) = 0.3113$

TABLE III
COMPARISON ITFS METHODS WITH CSMI

ITFS Method	FS criteria			
	Feature Relevancy	*Feature Redundancy	Conditional Feature Information	Dynamic Change of Features' Information
MIM	All	None	None	None
CIFE	All	All	All	None
JMI	All	All	All	None
MIFS	All	All	None	None
mRMR	All	All	None	None
MRI	All	All	All	None
DCSF	All	All	All	All
CSMI (Proposed)	All & Individual	All	All & Individual	Individual

***Legend*:**
All: Criteria considered with respect to all classes.
Individual: Criteria considered with respect to individual class label.
None: Criteria not considered.

* *Note: Feature redundancy is independent of class labels.*

of which basically adjust the values of α and β as per their FS criteria. All such methods consider the evaluation of mutual information (MI) that is shared among the features with all classes. However, the amount of information shared by the features with individual class labels differs from that with all classes. This is illustrated with an example as follows.

An Illustrative Example: Let us consider a hypothetical data set with three features f_1, f_2, f_3 and a binary class label C as shown in Table I. Let us also consider an ITFS method Mutual Information Maximization (MIM) [16] which calculates the MI between the candidate feature f_k and all classes. Here, f_1, f_2, and f_3 are the candidate features. Then, it decides the importance of features as per their value sorted in descending order. According to the framework 1, MIM takes the value of α and β as 0, as shown in the Eq. 2. In this equation, the MI value J_{MIM} is computed using the difference between the entropy $H(f_k)$ and conditional entropy $H(f_k|C)$. $H(f_k)$ and $H(f_k|C)$ are computed based on probability distribution of f_k and conditional probability distribution of f_k given C, respectively. We give basics of information theory in Section III.

$$J_{MIM}(f_k) = I(f_k; C). \tag{2}$$

Based on Table I, the values of J_{MIM} are calculated with respect to all as well as different class labels (i.e., class 0 and class 1) which are shown in Table II for two high-ranked features. We select all instances of the target class label and one random instance of another class label to calculate J_{MIM} for different class labels. Table II shows f_1 and f_3 are selected for all classes which are different from class 0 and class 1 (i.e., f_1 and f_2 for class 0, and f_2 and f_3 for class 1). Therefore, the feature space selected on all classes may affect the classification accuracy of the final model.

Next, we show the advantage of the proposed class label-based FS method through Fig. 1 as follows. Consider a data set presented in Fig. 1(a) for class labels A, B, and C and features f_1, f_2, and f_3. As shown in Fig. 1(b), C can be exclusively separated from A and B in feature space (f_2, f_3). Similarly, A can be exclusively separated in (f_1, f_2) (Fig. 1(c)) and B in (f_1, f_3) (Fig. 1(d)). However, we cannot select two features out of three in which all three classes can be well separated from each other as in traditional FS approaches. Thus, intuitively we can say that the class label-based feature selection approach can select features specific to the target class label and increase the performance of the final ML model. This motivates us to design an FS method that incorporates mutual information shared by the features not only with all classes but also with an individual class label. The rationale behind this is as follows. By considering all classes, the single subset of selected features may not represent the feature space in which individual class labels can be exclusively separated. Hence, it may affect the performance of the machine learning (ML) model for classification. For the sake of explanation, consider the task of text categorizing, where the aim is to classify documents relating to medical, political, economic, and other topics. In this scenario, features associated with the keywords hospital, patient, and discharge would be favored in distinguishing medical documents from non-medical ones. Similarly, features associated with the keywords market, currency, GDP, and taxation would be

favored in distinguishing economic documents from those of non-economic. However, it might be challenging to find a single subset of features associated with common keywords that can exclusively separate the documents into different groups.

Incorporating the above idea, we propose in this paper a novel ITFS method called Class-label Specific-Mutual-Information (CSMI). The proposed method has the following novelty and differences from the existing works: 1) It considers the feature redundancy, the conditional mutual information of features with all classes as well as individual class labels, and dynamic change of features' information with an individual class label. To the best of our knowledge, we are the first to consider the class-label-specific information shared by the features in the ITFS methods. 2) We apply a general framework to the proposed method, which combines the different subsets of features generated for each class label in order to predict the class label of a new test sample. 3) Unlike the earlier works [21]–[24], the proposed method selects the feature space in multi-class data sets (i.e., one instance belongs to one class label) instead of multi-label data sets (i.e., one instance belongs to the set of class labels). Table III presents the difference between the proposed CSMI with existing ITFS methods with respect to some important criteria. Note that these ITFS methods are popular, which are also used in earlier experiments [4], [25]–[27]. Our contributions are summarized as follows:

- We present a general framework that has four stages, namely class binarization, class balancing, CSMI, and classification, which are successively briefed.
- First, we convert a given data set with m class labels into m data sets with binary class labels using a one-against-all class binarization approach. This produces an imbalanced data set. Therefore, we apply oversampling technique through repeating instances to balance the data set.
- Next, we propose the CSMI FS method based on maximizing mutual information of the features with individual class labels and minimizing the same with all classes. This context employs feature redundancy, conditional mutual information, and dynamic change of information of selected features. The CSMI produces a subset of features for each class label.
- We then present an ensemble-based classification process for deciding the winner class label of a new sample by building a training model for each class label with the selected features specific to that class label.
- We perform extensive experiments and compare the results with various multi-class classification methods, i.e., five traditional ITFS and two state-of-the-art methods, namely MRI [20] and DCSF [26]. The experiments are carried out on 16 benchmark data sets using four classifiers, namely k-Nearest Neighbour (kNN), Support Vector Machine (SVM), Naive-Bayes (NB), and eXtreme Gradient Boosting (XGBoost). We also show the comparison results with one multi-label classification method called LIFT [21]. The results show that the proposed method outperforms the existing methods, which is also statistically validated through the paired two-tailed t-test.

TABLE IV
NOTATION OF DIFFERENT PARAMETERS OF THE EQUATIONS

Notation	Description
$F = \{f_i \mid i = 1, 2, ..., \lvert F \rvert\}$	Set of all features
$C = \{c_j \mid j = 1, 2, ..., m\}, m \geq 2$	Set of different class labels present in the data
f_k	Candidate feature: feature to be selected
f_j	Feature selected
S	Selected features on all classes
S_{c_i}	Selected features for a specific class label c_i
D_{c_i}	Re-sampled data set for the class label c_i
$H(X)$	Entropy of variable X
$H(X/Y)$	Conditional entropy of X given Y
$I(X;Y) = H(X) - H(X\|Y)$	MI between variable X and Y
$I(X, Y; Z)$	Joint MI of variables X and Y with Z
$I(X;Y\|Z)$	Conditional MI of variable X with Y given Z
$J_M(f_k)$	Value of J as per FS method M on the candidate feature f_k

This may be noted that the class label specific FS can be very effective in real-world quantification problems such as opinion mining [28], network-behavior analysis [29], quality control [30], vehicular accident prediction [31], credit scoring [32], and among others. For instance, there is an increasing demand for automatic methods to track overall consumer opinions [33]. The goal is to answer questions like what are the features of a product which satisfy the consumer? This goal can be achieved by selecting the subset of features specific to the satisfied consumers.

The rest of the paper is organized as follows. Section II gives a survey on the ITFS methods. Section III briefs about the basics of information theory. The proposed framework is described in Section IV, which includes class binarization, class balancing, class-label specific mutual information feature selection method, and classification process. Section V shows the experimental results through various tables and graphs, which are analyzed for performance evaluation. The paper is concluded in Section VI. For ease of readability, the notations used throughout the paper are summarized in Table IV.

II. LITERATURE REVIEW

Many ITFS methods (refer to Table V) have been developed that are based on the framework 1. MIM [16] is the simplest one among them, which considers the feature relevancy only with the class labels. As compared to MIM, the MIFS [17] is based on the notion that the features are generally dependent on each other as per the framework 1. Thus, the method considers the redundancy among features to select the most informative feature subset, in addition to feature relevancy with the class labels. MIFS sets the parameter α as 0. Thus it ignores conditional redundancy of the candidate features with

TABLE V
INFORMATION THEORY-BASED FEATURE SELECTION METHODS. SECTION II SHOWS THEIR RELATION WITH THE FRAMEWORK 1

ITFS method	Full name	Year
MIM	Mutual Information Maximisation [16]	1992
MIFS	Mutual Information Feature Selection [17]	1994
KS	Koller-Sahami metric [9]	1996
JMI	Joint Mutual Information [19]	1999
MIFS-U	Mutual Iinformation Feature Selection-'Uniform' [34]	2002
IF	Informative Fragments [35]	2003
FCBF	Fast Correlation-based Filter [36]	2004
AMIFS	Adaptive MIFS [37]	2004
CMIM	Conditional Mutual Info Maximisation [38]	2004
mRMR	Max-Relevance Min-Redundancy [7]	2005
ICAP	Interaction Capping [39]	2005
CIFE	Conditional Infomax Feature Extraction [40]	2006
DISR	Double Input Symmetrical Relevance [41]	2006
MINRED	Minimum Redundancy [42]	2006
IGFS	Interaction Gain Feature Selection [43]	2008
SOA	Second Order Approximation [40]	2009
CMIFS	Conditional MIFS [44]	2011
MRI	Max-Relevance and Max-Independence [26]	2017
DCSF	Dynamic Change of Selected Feature [27]	2018

the class labels and takes the following form:

$$J_{MIFS}(f_k) = I(f_k;C) - \beta \sum_{f_j \in S} I(f_k;f_j). \tag{3}$$

MIFS-U [34] further improved MIFS. In which, the authors proposed to calculate the mutual information between the candidate feature plus already selected features and the class labels, $I(f_k, S; C)$, where S is the set of already selected features. Although, MIFS and MIFS-U were able to identify the redundant features, their performance degrades with the increasing number of selected features because the feature redundancy outperforms the effect of feature relevancy. mRMR [7] addressed this problem by setting the value of β as the inverse of the number of selected features as follows:

$$J_{mRMR}(f_k) = I(f_k;C) - \frac{1}{|S|} \sum_{f_j \in S} I(f_k;f_j). \tag{4}$$

As a result, it minimizes the increasing magnitude of feature redundancy and reduces the chance of selecting redundant features. CIFE [40] employed the concept of conditional mutual information between the candidate feature and the selected features, given the class labels. CIFE set the values of α and β as 1 and took the following criterion:

$$J_{CIFE}(f_k) = I(f_k;C) - \sum_{f_j \in S} I(f_k;f_j) + \sum_{f_j \in S} I(f_k;f_j|C). \tag{5}$$

Earlier studies [27], [45] showed that mRMR obtains better results compared to other ITFS methods, such as Correlation-based Feature Selection (CFS) [46], FCBF [47], ReliefF [48], MIFS [17] and CIFE [40].

Unlike discussed FS methods, JMI [19] advocated the use of joint mutual information of the candidate feature and the selected features with the class labels. The criterion of JMI is shown as follows:

$$J_{JMI}(f_k) = \sum_{f_j \in S} (f_j, f_k; C) \tag{6}$$

The drawback of JMI is that it overestimates the significance of some features, for example, if the candidate feature is fully correlated with one or more selected features but at the same time is almost independent of the selected features. In such a scenario, the value of J_{JMI} would be high despite the redundancy of the candidate feature with some of the selected features. Joint Mutual Information Maximization (JMIM) [25] addressed this problem by employing the *maximum of the minimum* approach. Here, the candidate feature is selected if it produces the maximum of the minimum joint mutual information values with already selected features with respect to the class labels. JMIM is similar to Conditional Mutual Information Maximization (CMIM) [38]. However, it prefers selecting those feature which maximizes the conditional mutual information with the class labels.

The limitations of the aforementioned methods [7], [16], [17], [19], [25], [38], [40] are that they minimize the feature redundancy or maximize the feature relevancy with the class labels. The methods fail to provide the proper relation between both of these criteria. MRI [26] addressed this problem through independent classification information. In which, the authors considered the new classification information plus the information preserved in earlier classification. The criterion of MRI is shown as follows:

$$J_{MRI}(f_k) = I(f_k;C) + \sum_{f_j \in S} ICI(C;f_j,f_k) \tag{7}$$

where $ICI(C;f_j,f_k)$ consists of two terms, $I(C;f_k|f_j)$ as the new classification information and $I(C;f_j|f_k)$ as the earlier classification information which is negatively correlated with $I(C;f_j;f_k)$. Although, the efficacy of the aforesaid methods is desirable, they do not consider the dynamic change of information between selected features and the class labels when the candidate feature is added and hence addressed in DCSF [20]. DCSF takes the value of α and β as $\frac{3}{|S|}$ and $\frac{2}{|S|}$, respectively. Thus, the criterion for DCSF is as follows:

$$J_{DCSF}(f_k) = I(f_k;C) - \frac{3}{|S|} \sum_{f_j \in S} I(f_k;f_j) + \frac{2}{|S|} \sum_{f_j \in S} I(f_k;f_j|C). \tag{8}$$

In light of the above analysis, we propose to select a subset of features separately for each class label keeping in view that the single subset of selected features on all classes can not represent the feature space in which all classes can be separated.

III. BASICS OF INFORMATION THEORY

In [49], the author had presented the concept of Information theory. It measures the uncertainty present in the random variable distribution X and is measured as Entropy $H(X)$. $H(X)$ is calculated based on the probability distribution of possible values of X as follows:

$$H(X) = -\sum_{x_i \in X} p(x_i)\ log\ p(x_i) \tag{9}$$

Here, x_i denotes the possible values of the random variable X. For two random variables X and Y, the definition of joint

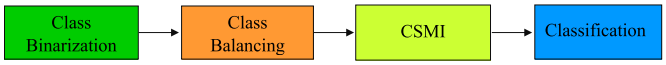


Fig. 2. General framework.

entropy and conditional entropy are defined as follows:

$$H(X,Y) = -\sum_{x_i \in X}\sum_{y_i \in Y} p\,(x_i, y_i)\, log\; p\,(x_i,\; y_i)\,, \quad (10)$$

$$H(X|Y) = -\sum_{y_i \in Y} p(y_i) \sum_{x_i \in X} p(x_i|y_i)\; log\; p(x_i|y_i) \quad (11)$$

where $p(x_i, y_i)$ and $p(x_i|y_i)$ represent the joint probability and conditional probability of x_i and y_i, respectively. The joint entropy measures uncertainty in two (or more) variables. Whereas, the conditional entropy measures the uncertainty left with one variable after knowing the uncertainty of other variables.

Based on the uncertainty of two variables, MI measures the difference between entropies $H(X)$ and $H(X|Y)$, as shown in Eq. 12. Thus, the MI can be intuitively defined as the amount of information provided by one variable about another variable as follows:

$$I(X;Y) = H(X) - H(X|Y) \quad (12)$$

Similarly, Conditional mutual information and Joint mutual information are defined in Eq. 13 and Eq. 14, respectively.

$$I(X;Y|X) = H(X|Z) + H(Y|Z) - H(X,Y|Z) \quad (13)$$

Conditional mutual information measures the mutual information between two variables X and Y when the third variable Y is already known. In comparison, Joint mutual information measures the mutual information between X and Y combined with the third variable Z.

$$I(X,Y;Z) = H(X;Y) - H(X;Y|Z) \quad (14)$$

In terms of MI, Eq. 14 can be represented as [50]:

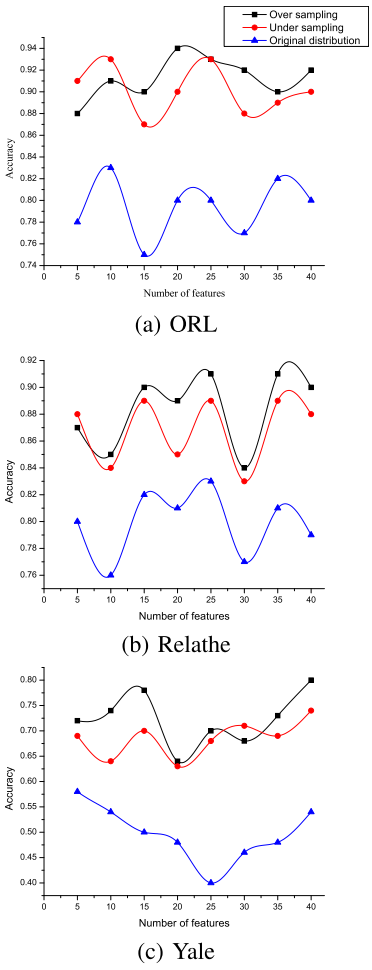
$$I(X,Y;Z) = I(X;Z) + I(Y;Z|X) \quad (15)$$

IV. Proposed Work

Here, we propose a general framework for the CSMI. The framework proposed consists of four stages: class binarization, class balancing, CSMI, and classification, as shown in Fig. 2. In this framework, we apply a proper classification strategy for measuring the final performance of the method. This is because conventional classifiers perform classification on a single subset of features, which is not in our case. Next, we discuss the various stages of the proposed framework with reference to Fig. 2, showing the FS process for each class label.

A. Class Binarization

In the first stage, we transform a data set with m class labels into m data sets with binary class labels by applying one-against-all class binarization approach. In this approach, for each class label $c_i,\ i = 1, \ldots, m$, a binary class label



(a) ORL

(b) Relathe

(c) Yale

Fig. 3. Comparing over-sampling and under-sampling with the original distribution on three data sets (ORL, Relathe, and Yale.).

$\langle c_i, \rho_i \rangle$ (refer to Fig. 4) is created where $\rho_i = \bigcup_{j=1, j\neq i}^{m} w_j$. The samples of the target class label c_i are labeled as 0 and the samples belonging to rest of the class labels ρ_i are labeled as 1 (refer to Fig. 7).

B. Class Balancing

The one-against-all class binarization technique applied in stage 1 generates an imbalanced data set [51]. Therefore, we apply an oversampling technique to balance the newly generated data set. Many oversampling methods [52]–[54] have been proposed. Fig. 3 shows the comparison of two basic sampling techniques, i.e., over-sampling and under-sampling, with the original distribution of three data sets, i.e., ORL, Relathe, and Yale (refer Table VI). It can be observed that over-sampling shows better performance than under-sampling and the original distribution. Hence, for our study, we use over-sampling through repeating instances because it produces the best results, which is also used in paper [55]. The procedure applied for each class label $c_i,\ i = 1, \ldots, m$ is as follows. The difference between number of training samples of the class labels ρ_i and c_i is calculated, i.e., $\delta_i = |\rho_i| - |c_i|$. If $\delta_i > 0$ then the samples of the class label c_i is repeated until the δ_i becomes 0 (refer to Fig. 7). The algorithm for calculating

Algorithm 1 : To Calculate Classification Accuracy of Sampling Techniques

Input: m re-sampled data sets with binary class labels.
Output: Accuracy.

1: **for** $c_i, i = 1$ to m **do**
2: Calculate δ_i as $|\rho_i| - |c_i|$ *// for ρ_i see section IV-A*
3: **If** *Oversampling* **do**
4: Repeat the random samples of c_i till δ_i becomes 0.
5: Calculate classification accuracy on increasing number of selected features.
6: *// The features are increased as such 5, 10, 15, …*
7: **end if**
8: **If** *Undersampling* **do**
9: Delete the random samples of ρ_i till δ_i becomes 0. *// for ρ_i see section IV-A*
10: Calculate classification accuracy on increasing number of selected features.
11: *// The features are increased as such 5, 10, 15, …*
12: **end if**
13: **If** *Nosampling* **do**
14: Calculate classification accuracy on increasing number of selected features.
15: *// The features are increased as such 5, 10, 15, …*
16: **end if**
17: **end for**
18: **for all** *Oversampling, Undersampling, Nosampling* **do**
19: Calculate average accuracy of features over all classes.
20: **end for**

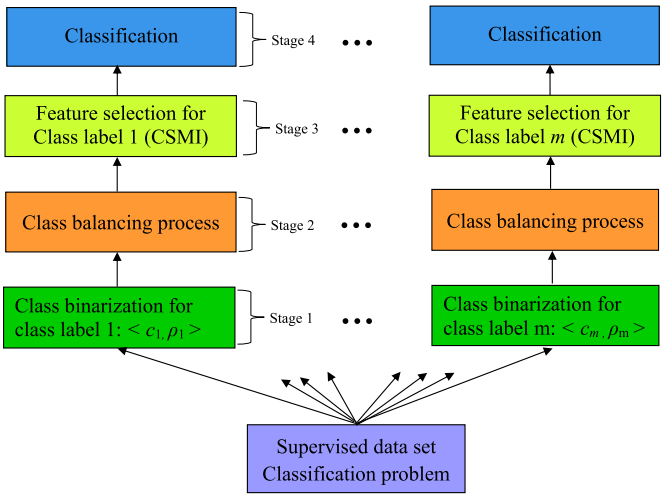


Fig. 4. Conversion of m class labels data set into m binary class label data sets.

the classification accuracy of sampling techniques is shown in Algorithm 1. For each class label (line 1), we first find the difference between the target class label c_i and the rest of the class labels ρ_i (line 2). Then, we balance the training sample as per over-sampling (line 4). We apply the ML classification model to measure the accuracy of the increasing number of selected features (line 5). Here, our objective is to observe and compare the classification ability of the sampling techniques.

TABLE VI
BENCHMARK DATA SETS DESCRIPTION

No.	Data sets	Instances	Features	Classes	Types
1.	RELATHE	1427	4332	2	Discrete
2.	lymphoma	96	4026	9	Discrete
3.	ORL	400	1024	40	Discrete
4.	WarpAR10P	130	2400	10	Discrete
5.	WarPIE10P	210	2420	10	Discrete
6.	Orlraws10P	100	10304	10	Discrete
7.	Yale	165	1024	15	Discrete
8.	PCMAC	1943	3289	2	Discrete
9.	Lung_discrete	73	325	7	Discrete
10.	CLL-SUB-111	111	11340	3	Continuous
11.	TOX-171	171	5748	4	Continuous
12.	Movement_libras	360	90	15	Conitnuous
13.	Arrhythmia16	452	279	16	Continuous
14.	Isolet	1560	617	26	Continuous
15.	Air	359	64	3	Continuous
16.	COIL20	1440	1024	20	Continuous

Thus, the features are chosen from feature 1 to feature 5, then feature 1 to feature 10, and so on up to feature 40. We repeat the same process for under-sampling (lines 8-12). We calculate the accuracy of the original distribution in lines 13-16. We next calculate the average accuracy over all classes on each selected feature subset for over-sampling, under-sampling, and original distribution (no-sampling) in lines 18-20. In this way, we calculate the accuracy using four different classifiers (NB, kNN, SVM, and XG) and finally take the average value to ensure that the observed behavior is not biased toward a particular classifier.

C. CSMI Method

In this stage, we describe the proposed CSMI as follows. The CSMI employs mutual and conditional mutual information to find the optimal subset of features for the target class label. It also considers the dynamic change of information shared between the selected features and the class label when a candidate feature is added.

1) Basic Terminologies: Before discussing CSMI, we brief about the concepts of feature relevancy and feature redundancy in the context of the individual class label as follows. Consider a data set D with the total number of features F. After applying binarization (stage 1) and balancing (stage 2) on the data set, we obtain the resampled data set D_{c_i} for the class label c_i,

$$D_{c_i} = F \times P_i \tag{16}$$

where P_i represents the binary class labels $\langle c_i, \rho_i \rangle$ concerning to the class label c_i. Let S_{c_i} be the subset of selected features for the class label c_i and f_k is the candidate feature. Then, f_k should be selected in such a way that it maximizes the joint mutual information of f_k and S_{c_i} with P_i. According to the Eq. 15, it is expressed as

$$I(f_k, S_{c_i}; P_i) = I(f_k; P_i|S_{c_i}) + I(S_{c_i}; P_i) \tag{17}$$

For the two candidate features f_{k_1} and f_{k_2}, we have the equations as follows:

$$I(f_{k_1}, S_{c_i}; P_i) = \quad I(f_{k_1}; P_i|S_{c_i}) + I(S_{c_i}; P_i) \tag{18}$$
$$I(f_{k_2}, S_{c_i}; P_i) = \quad I(f_{k_2}; P_i|S_{c_i}) + I(S_{c_i}; P_i) \tag{19}$$

By Eq. 18 and Eq. 19,

$$I(f_{k_1}, S_{c_i}; P_i) - I(f_{k_1}; P_i|S_{c_i}) = I(f_{k_2}, S_{c_i}; P_i) - I(f_{k_2}; P_i|S_{c_i}) \tag{20}$$

If $I(f_{k_1}, S_{c_i}; P_i) > I(f_{k_2}, S_{c_i}; P_i)$:

$$I(f_{k_1}; P_i|S_{c_i}) > I(f_{k_2}; P_i|S_{c_i}) \tag{21}$$

else

$$I(f_{k_1}; P_i|S_{c_i}) < I(f_{k_2}; P_i|S_{c_i}) \tag{22}$$

Next, we define the terms feature relevancy and the feature redundancy as follows.

a) Feature relevancy: Given selected features S_{c_i}, f_{k_1} is more relevant to the binary class label P_i than f_{k_2} if the conditional mutual information between f_{k_1} and P_i is greater than the conditional mutual information between f_{k_2} and P_i, i.e., $I(f_{k_1}; P_i|S_{c_i}) > I(f_{k_2}; P_i|S_{c_i})$.

b) Feature redundancy: f_{k_1} is more redundant than f_{k_2}, if the mutual information between f_{k_1} and S_{c_i} is greater than the mutual information between f_{k_2} and S_{c_i}, i.e., $I(f_{k_1}; S_{c_i}) > I(f_{k_2}; S_{c_i})$.

2) Feature Selection Process: We have already discussed that all the ITFS methods select an optimal subset of features that share maximal information with all classes. Hence, there has been a lack of considering the amount of information shared between the features and individual class labels. However, the information shared between feature and individual class labels differs from that shared between feature and all classes. Therefore, we compare the conditional mutual information of the candidate feature (f_k) and the target class label (P_i) with all classes (C), given selected features (f_j) in the following three cases:

1) The candidate feature shares more information with the target class label than all classes, i.e.,
$$I(f_k; P_i|f_j) > I(f_k; C|f_j), \tag{23}$$
2) The candidate feature share equal information with the target class label and all classes, i.e.,
$$I(f_k; P_i|f_j) = I(f_k; C|f_j), \tag{24}$$
3) The candidate feature shares less information with the target class label than to all classes, i.e.,
$$I(f_k; P_i|f_j) < I(f_k; C|f_j), \tag{25}$$

Based on the above cases, we conclude that the candidate feature is more important to the class label, the greater the value of $I(f_k; P_i|f_j) - I(f_k; C|f_j)$ is.

Considering this analysis and dynamic change of information of selected features with the target class label and the traditional feature redundancy, the criterion for CSMI is as follows:

$$J(f_k) = \sum_{f_j \in S_{c_i}} \{I(f_k; P_i|f_j) - I(f_k; C|f_j) + I(f_j; P_i|f_k) - I(f_j; f_k)\} \tag{26}$$

In Eq. 26, $I(f_k; P_i|f_j)$ maximizes the information content of the candidate feature for the target class label at the same time and $I(f_k; C|f_j)$ minimizes information content of the same for all classes. Thus, it ensures that the subset of selected features is specific to the target class label. Further, $I(f_j; P_i|f_k)$ estimates the effect on the mutual information between the selected features and the target class label when the candidate feature is introduced. Note that maximizing $I(f_j; P_i|f_k)$ ultimately maximizes $I(f_k, f_j; P_i)$ as per Eq. 15. Finally, $I(f_j; f_k)$ shows the feature redundancy in terms of mutual information between the selected features and the candidate feature; hence it is reduced. We reformulate CSMI more intuitively as follows:

$$J(f_k) = \sum_{f_j \in S_{c_i}} \{I(f_k; P_i) + 2I(f_k; f_j|P_i) + I(f_j; P_i) - \{I(f_k; C) + I(f_k; f_j|C)\} - 2I(f_j; f_k)\} \tag{27}$$

The proof of the above equation is given in Appendix A.

The term $\sum_{f_j \in S_{c_i}} I(f_j; P_i)$ is constant term with respect to f_k as shown in paper [20]. Therefore, it can be omitted and Eq. 27 can be rewritten as follows.

$$\begin{aligned} J(f_k) &= \sum_{f_j \in S_{c_i}} \{I(f_k; P_i) + 2I(f_k; f_j|P_i) - \{I(f_k; C) + I(f_k; f_j|C)\} - 2I(f_j; f_k)\} \\ &= |S_{c_i}| \times I(f_k; P_i) + \sum_{f_j \in S_{c_i}} 2I(f_k; f_j|P_i) - \sum_{f_j \in S_{c_i}} \{I(f_k; C) + I(f_k; f_j|C)\} - \sum_{f_j \in S_{c_i}} 2I(f_j; f_k) \end{aligned}$$

$$\begin{aligned} \frac{J(f_k)}{|S_{c_i}|} = I(f_k; P_i) &+ \frac{2}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} I(f_k; f_j|P_i) - \\ &\frac{1}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} \{I(f_k, C) + I(f_k; f_j|C)\} - \\ &\frac{2}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} I(f_j; f_k) \end{aligned} \tag{28}$$

Further, the term $\frac{J(f_k)}{|S_{c_i}|}$ is proportional to $J(f_k)$ with respect to f_k and will not affect the ranking of candidate features. Hence, Eq. 28 can be rewritten as:

$$\begin{aligned} J(f_k) = I(f_k; P_i) &+ \frac{2}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} I(f_k; f_j|P_i) - \\ &\frac{1}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} \{I(f_k, C) + I(f_k; f_j|C)\} - \frac{2}{|S_{c_i}|} \sum_{f_j \in S_{c_i}} I(f_j; f_k) \end{aligned} \tag{29}$$

As we can observe from Eq. 29 that the feature space of original class structure may not be preserved during the feature selection process. Nevertheless, our intention is to select the feature space that contains more information related to the particular class label, c_i and at the same time contains less information related to all classes, C.

The algorithm for CSMI presented as Algorithm 2 is explained as follows. We select a specified threshold T which means CSMI selects a subset of top T performing features. Max is a variable that stores the maximal value of the objective function $J(f_k)$ for obtaining the most informative features for the class label c_i. The selected feature at each iteration is represented by f_j. First, CSMI initializes the set

Algorithm 2 : Class-Label Specific Mutual Information (CSMI) FS Method

Input: D_{c_i}: Balanced training sample for class c_i prepared from original data set D; User-specified threshold T for number of selected features.

Output: $\{S_{c_i}\}$: Class label specific selected features.

1: $S_{c_i} \leftarrow \phi$;
2: $t \leftarrow 0$;
3: $Max \leftarrow \phi$;
4: **for** i = 1 to $|F|$ **do**
5: Calculate the mutual information $I(f_i; P_i)$;
6: **end for**
7: **While** $t < T$ **do**
8: **If** t == 0 **do**
9: Select the feature f_j with the largest $I(f_i; P_i)$;
10: $t = t + 1$;
11: $S_{c_i} = S_{c_i} \cup f_j$;
12: $F = F - f_j$;
13: **end if**
14: **for** each candidate feature $f_{k_i} \in F$ **do**
15: $Max[i] + = I(f_{k_i}; P_i|f_j) - I(f_{k_i}; C|f_j) + I(f_j; P_i|f_{k_i}) - I(f_j; f_{k_i})$;
16: **end for**
17: Select the feature f_j with the largest $Max[i]$;
18: $S_{c_i} = S_{c_i} \cup f_j$;
19: $F = F - f_j$;
20: $t = t + 1$;
21: **end while**

S_{c_i}, the threshold T, and Max (lines 1 - 3). Next, it calculates the mutual information between each feature and the binary class label P_i (lines 4 - 6). Finally, according to criteria (26), CSMI selects the most informative feature concerning to class label c_i until t reaches T (lines 7 - 21). Fig. 5 shows the flowchart of the CSMI algorithm with respect to the target class label, c_i.

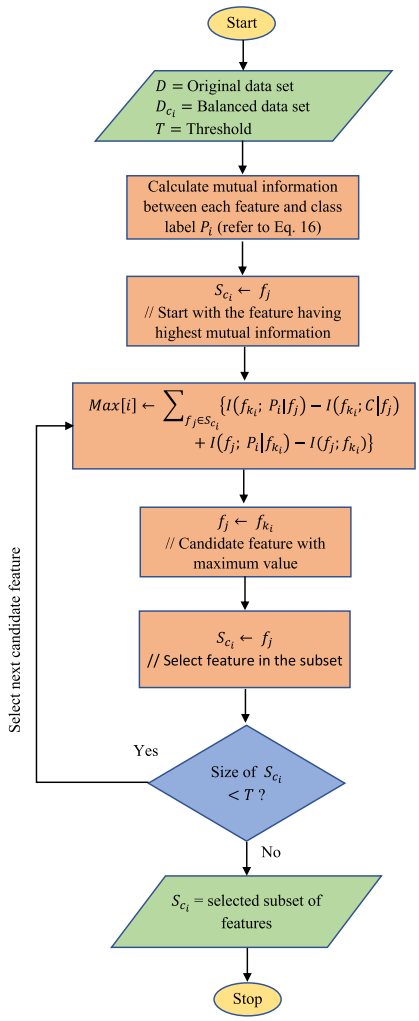


Fig. 5. Flowchart of CSMI algorithm.

D. Classification Process

A set of selected features S_{c_i} has been generated for each class label c_i using Algorithm 2 (CSMI). Now, here we apply a majority-based ensembled approach to identify the class label of a new sample. Hence, we prepare a training model M_i for the class label c_i from the original data set D, but, with selected features S_{c_i}. In this way, a set of m training models $\{M_1, M_2, \ldots, M_m\}$ are prepared where each model is trained for each class label $c_i \in C$. To classify a new sample O, its original dimension is reduced to S_{c_i} used by the model M_i. Next, each model assigns a class label to a new sample O. To decide the winner class label, the following rules are followed (refer to Decision rules in Fig. 6):

1) If a model M_i predicts the class label c_i on the subset S_{c_i}, i.e. on the same subset for which M_i is trained, c_i is assigned to O. In case of two or more models show the same behavior, the class label of the sample O is assigned through the majority voting of the models.

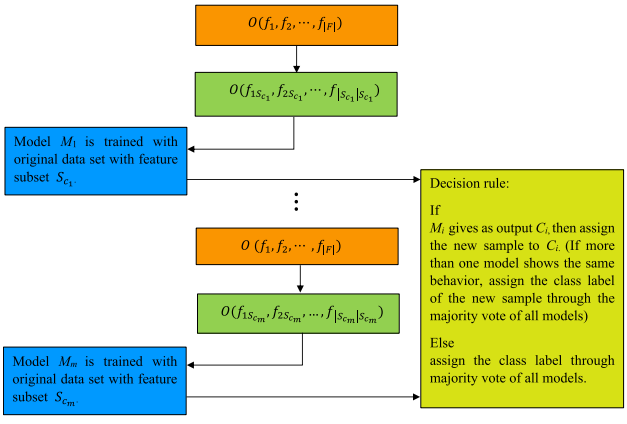


Fig. 6. Ensembled-based classification process of the proposed framework.

2) If no models are able to give the same class label c_i, for which the model is trained, the class label of the sample, O is assigned through the majority voting of the models.

Finally, Fig. 7 shows the complete data flow diagram of the proposed framework. In Stage 1, the original data set is binarized for each class label keeping the target class label

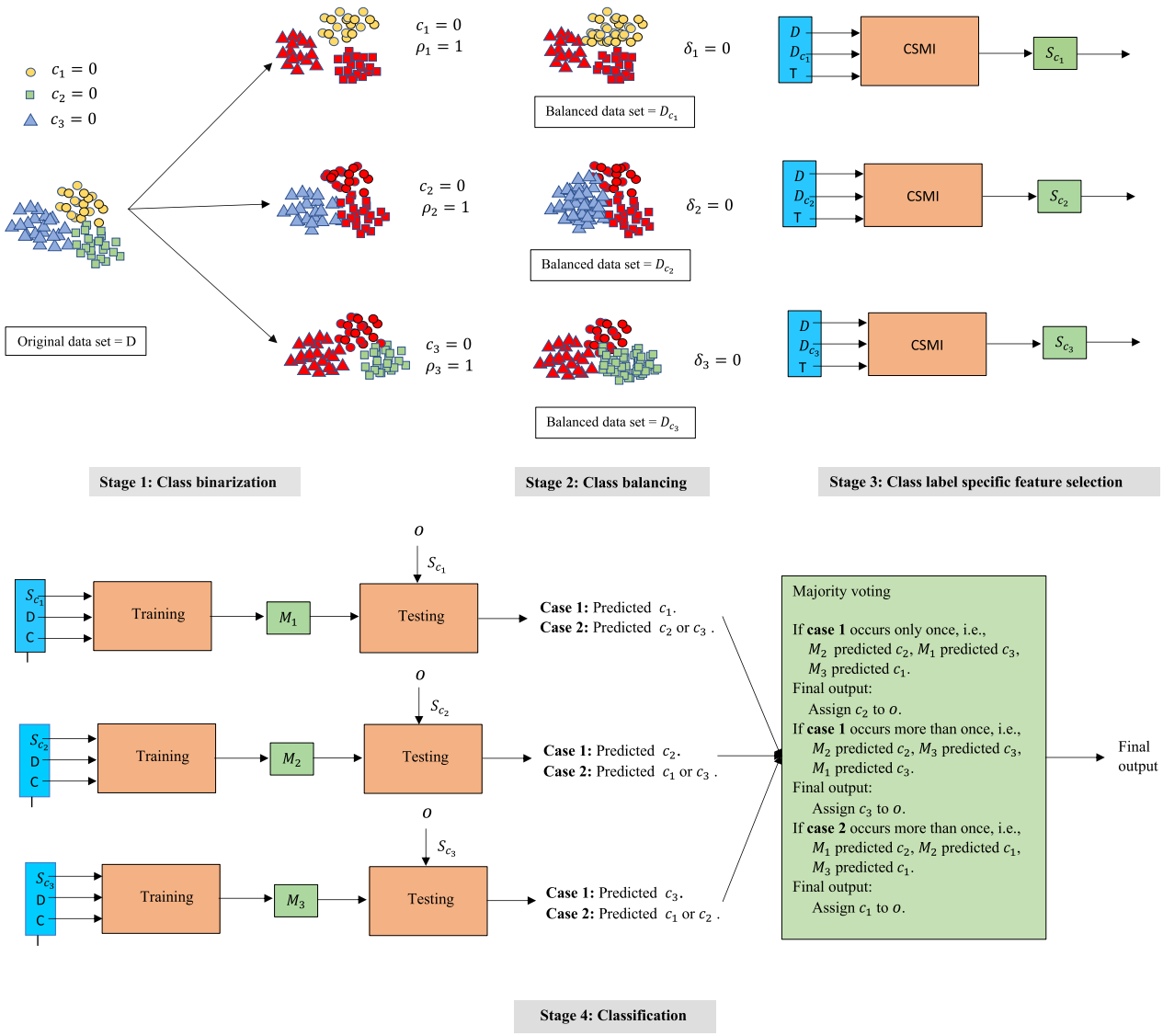


Fig. 7. Complete data flow diagram of the proposed framework.

as 0 and the rest class labels as 1. Stage 2 balances the newly generated binarized data sets through oversampling technique. Algorithm 2 is then applied to the balanced data sets to select features specific to the target class label in Stage 3. Finally, Stage 4 performs the classification. Here, Case 1 indicates that M_i predicts the class C_i, i.e., on the same feature subset for which M_i is trained. Whereas Case 2 indicates that the M_i predicts the class other than C_i. If Case 1 occurs for more than one model, then the class of a new sample is decided based on the majority vote of all models. However, if only one model satisfies Case 1, only that class is assigned to the new sample. Furthermore, if no models satisfy Case 1, the class of a new sample is assigned through the majority vote of all models.

1) Time Complexity Analysis: As T features are selected from the F features on N samples, the time complexity for all, i.e., mutual information, conditional mutual information, and joint mutual information, is $O(N)$ as all samples are estimated based on probability estimation [27]. As a result, MIM obtains the best time complexity as $O(FN)$ and the time complexity of DCSF, CIFE, JMI, MIFS, mRMR, and MRI is $O(TFN)$ for each [27]. As the CSMI follows similar steps like DCSF for each class label but works with m class labels, the time complexity is $O(mTFN)$. As m cannot be large enough in normal cases, this is acceptable.

V. Experimental Results and Analysis

The proposed method was tested through extensive experiments on a personal computer with the Windows 10 operating system, Intel Core i3-5005 CPU processor, and 8 GB RAM. We used the "Scikit-learn" package of the python environment for the implementation [56]. The data sets used for the experiments are taken from UCI ML repository [57] and literature [4]. These data sets are from different fields, such as text data (Relathe), biological data (Lung_discrete), face image data (ORL). Note that similar data sets are used in previous literatures [4], [25], [58]. The missing values of the data sets were handled by imputing the average value of the corresponding column. Further, we discretized continuous data

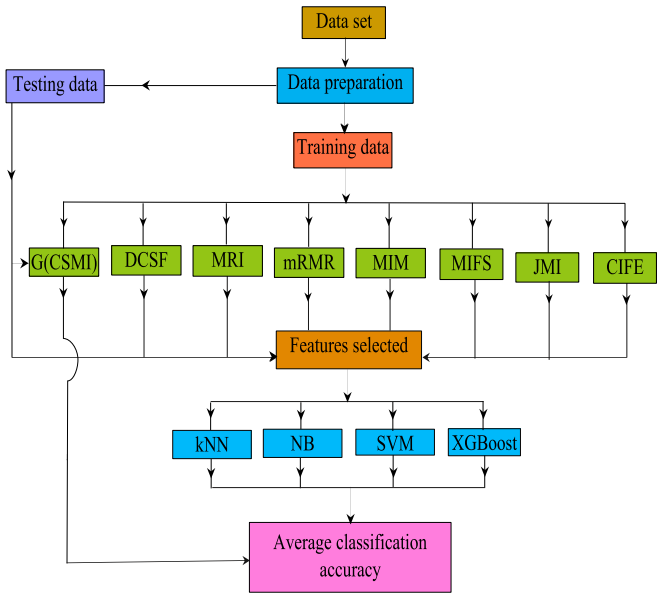


Fig. 8. Evaluation framework.

into five equal-interval groups, and the nominal data were transformed into integers. Although, data discretization results in a significant loss of information that may affect classification performance, it makes the features understandable and easier to interpret. Moreover, it has been frequently shown that discretization is time-saving for classification problems [59]. Thus, we discretize the continuous data similar to earlier literature [7], [27]. The details of the data sets used are shown in Table VI.

We used four classifiers, namely kNN (k=3), SVM, NB, and XGBoost, to measure the performance of the model. We applied the K-fold cross-validation to mitigate the effect of overfitting [26]. To find the optimum value of K, we performed experiments varying its value from 1 to 10 on four data sets (Isolet, ORL, Relathe, and Yale) using the above four classifiers. In Appendix B, Fig. 13, Fig. 14, Fig. 15, and Fig. 16 show the classification performance (accuracy) of training and test set on four data sets using classifier NB, kNN, SVM, and XGBoost, respectively. It can be observed from the figures that the accuracy difference between training and test set is at most five at K=10 in eleven out of sixteen cases. Further, Table XVII summarizes and shows the minimum, maximum, and average accuracy differences between training and test set as Min., Max., and Ave., respectively. The table shows the Max. and Ave. are minimum at the K value 10. Thus, it can be safely concluded that in most cases, 10-fold cross-validation (K=10) does not show overfitting. However, the classification performance of an ML model is dependent on the data set and the classifier used [25], [60]. We used the evaluation framework for all the experiments same as that the previous works [25], [27], [61] as shown in Fig. 8 where G(CSMI) refers to general framework for CSMI proposed in this article. As we study the filter-based FS methods, it is necessary to establish a threshold cutoff to obtain a practical subset of features. Most of the studies in the literature [50], [62] use thresholds that retain different percentages of features or fixed number of features, especially on high dimensional data sets. Similarly, we set the threshold $T = 40$ to evaluate the performance of the models keeping the view that the value T should not be too small or large because smaller T will have fewer feature alternatives, and the selected features will have lesser diversity. On the other hand, the larger T will increase the computational time of the models. We also perform experiments on the automatic threshold of the data sets. Earlier attempts [63], [64] have been made in developing a general automatic threshold which may vary depending on the data set. This study decides the threshold value based on complexity measure [65] of each feature since they can estimate the relevancy of features concerning class labels. Therefore, it can help in establishing a threshold cutoff for the ranking of features generated by FS methods [63]. The formula for calculating the complexity value of each feature is taken from [66] as,

$$e = \gamma \times \frac{1}{r} + (1-\gamma) \times \eta \tag{30}$$

where γ is the parameter in the range $[0, 1]$ that balances the importance of both the error obtained and the number of features retained, the η is the percentage of features retained, and r is the fisher discriminant ratio [67]. Similar to [63], this work takes $\gamma = 0.75$ empirically and $\eta = \log_2(|F|)$ where $|F|$ is the number of all features. Note that we explicitly use the Intel Core i7-10750H CPU processor and 24 GB RAM to generate results for the automatic threshold of features of the data sets. Finally, we applied a paired two-tailed t-test to indicate that the improvement does not occur by chance at the p-value 0.05. The data sets and auxiliary codes are available at http://featureselection.asu.edu/ [4]. Now we compare the proposed method with multi-class and multi-label classification as follows. However, since our scheme is based on multi-class classification, we show the comparison results and their analysis in detail with respect to the accuracy, Area Under the Curve (AUC), and F-score with seven benchmark multi-class methods. We also compare the experimental results with a multi-label classification method, namely, LIFT [21] with respect to accuracy only.

A. Comparison Results With Multi-Class Classification

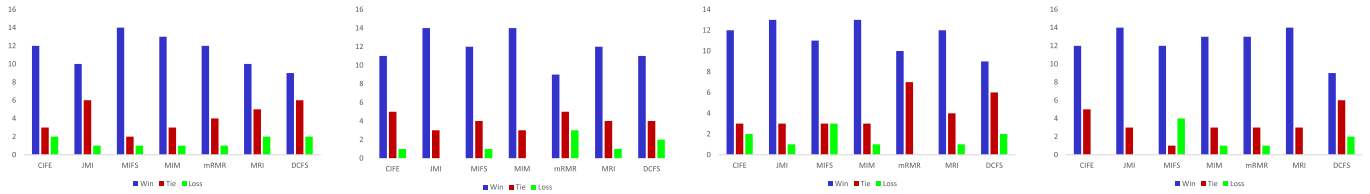
To evaluate the performance of the proposed method, we compare CSMI with CIFE [40], JMI [19], MIFS [17], MIM [16], and mRMR [7], and two state-of-the-art methods proposed recently MRI [26] and DCSF [20]. The reason behind choosing these methods is that they are the traditional ITFS methods and used in earlier experiments [27], [25], [4], [26]. We already discussed these methods in Section I summarizing their criteria in Table 3. Next, we give an analysis of the results generated through the experiments. Tables VII - X show the average accuracy and standard deviation of four classifiers on different data sets. The accuracy is generated on 100 random initialization trials. The "Average" shows the average accuracy of eight methods on 16 data sets and the "Number" shows the number of times the corresponding method obtains better accuracy than the

TABLE VII
AVERAGE ACCURACY (MEAN ± STD.) WITH STATISTICAL SIGNIFICANCE ON NB

Data sets	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	61.39±1.28(+)	68.32±2.05(+)	65.23±3.56(+)	59.17±4.14(+)	84.92±2.38(=)	82.46±3.26(=)	76.34±1.12(+)	83.10±3.52
Lymphoma	71.69±4.87(+)	69.42±3.23(+)	94.56±5.41(=)	75.55±4.29(+)	87.60±2.47(+)	79.10±2.73(+)	89.00±1.16(+)	95.68±1.89
ORL	75.13±2.89(-)	72.43±3.01(=)	68.38±2.17(+)	65.20±3.81(+)	70.36±4.01(+)	72.58±0.75(=)	73.47±1.42(=)	73.11±3.51
WarpAR10P	68.25±3.71(+)	53.49±2.72(+)	68.47±2.59(+)	42.59±0.69(+)	72.01±2.39(+)	68.19±1.10(+)	65.09±1.56(+)	69.31±0.29
WarPIE10P	63.91±0.52(+)	78.37±1.95(+)	69.82±1.48(+)	65.72±2.01(+)	74.31±1.24(+)	59.21±1.24(+)	88.45±1.95(=)	89.19±1.42
Orlraws10P	73.15±4.29(+)	69.48±3.11(+)	79.28±3.56(+)	70.77±1.94(+)	85.91±1.28(=)	76.26±2.36(+)	81.40±2.12(+)	87.00±2.71
Yale	39.01±1.69(+)	62.81±1.32(=)	57.40±0.66(+)	34.41±1.94(+)	51.01±0.19(+)	62.23±1.16(=)	62.07±0.72(=)	61.89±1.88
Lung_discrete	64.80±1.06(+)	62.00±2.29(+)	68.19±0.18(+)	71.28±3.01(=)	58.75±0.32(+)	76.27±1.48(-)	77.59±1.59(-)	72.21±0.37
PCMAC	79.53±1.17(=)	76.84±2.65(=)	68.41±4.71(+)	75.07±2.14(+)	72.03±4.81(+)	75.59±2.39(+)	70.81±3.54(+)	78.41±1.54
CLL-SUB-111	69.62±2.18(+)	65.48±0.59(+)	75.20±2.10(+)	86.83±2.34(-)	83.59±1.61(=)	72.09±2.39(+)	78.05±3.15(+)	82.46±2.41
TOX-171	79.23±0.72(=)	75.46±2.68(+)	78.13±1.59(=)	69.28±2.10(+)	74.03±1.56(+)	77.19±0.62(=)	79.92±0.91(-)	78.02±1.08
Movement_libras	76.95±1.81(-)	78.74±1.59(-)	75.13±1.84(-)	38.23±2.17(+)	48.43±0.84(+)	77.73±1.32(-)	63.52±1.72(+)	72.20±2.31
Arrhythmia16	43.52±3.24(+)	58.27±2.01(=)	25.03±3.40(+)	58.10±3.69(+)	55.40±2.68(+)	47.63±3.12(+)	43.26±4.22(+)	57.19±2.14
Isolet	57.95±2.41(+)	62.31±4.91(+)	53.09±2.84(+)	68.37±1.83(=)	48.32±2.24(+)	68.83±2.86(=)	66.20±3.37(=)	67.64±4.57
COIL20	59.65±5.42(+)	55.31±4.52(+)	62.64±2.10(+)	54.24±3.78(+)	79.27±4.09(=)	74.02±0.57(+)	75.43±2.13(+)	78.61±2.59
Air	58.26±3.21(+)	65.05±1.81(=)	63.91±2.63(+)	53.46±3.26(+)	57.10±1.71(+)	50.23±2.25(+)	66.43±0.91(=)	64.81±1.27
Numbers	2	3	1	3	3	3	7	10
Average	63.82	65.29	65.06	60.25	67.35	67.95	70.78	73.46

TABLE VIII
AVERAGE ACCURACY (MEAN ± STD.) WITH STATISTICAL SIGNIFICANCE ON kNN (k = 3)

Data sets	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	78.29±2.81(-)	65.14±0.89(+)	71.32±1.75(+)	60.23±3.13(+)	75.01±0.32(-)	64.15±4.13(+)	66.17±3.32(+)	73.40±1.23
Lymphoma	79.13±2.13(+)	75.02±1.24(+)	92.34±2.35(-)	65.41±1.32(+)	91.42±2.01(-)	81.12±0.51(+)	79.12±3.31(+)	90.81±3.54
ORL	69.48±0.56(=)	68.39±4.08(=)	69.36±2.35(=)	57.92±2.45(+)	69.15±3.56(=)	66.05±2.17(+)	66.34±2.50(+)	69.82±3.21
WarpAR10P	37.27±2.57(+)	32.45±2.05(+)	39.13±0.75(+)	41.94±2.01(+)	69.23±0.12(-)	64.39±1.27(=)	43.75±2.07(+)	64.83±2.92
WarPIE10P	79.57±0.91(+)	72.57±1.13(+)	81.10±1.64(+)	65.01±1.23(+)	80.26±1.37(+)	78.66±1.83(+)	93.74±2.18(=)	93.23±1.07
Orlraws10P	73.57±1.48(+)	57.37±1.86(+)	85.49±0.82(=)	69.18±1.82(+)	82.41±2.05(+)	84.91±0.88(=)	77.37±2.82(+)	85.32±1.32
Yale	37.29±2.12(+)	41.16±0.81(+)	43.27±1.28(+)	45.53±0.53(+)	49.66±1.86(+)	59.21±0.47(+)	57.17±1.47(=)	58.51±1.49
Lung_discrete	71.39±3.85(=)	65.55±1.15(+)	69.84±4.64(+)	61.85±2.94(+)	69.23±2.34(+)	63.14±3.50(+)	70.00±3.69(+)	72.31±0.22
PCMAC	73.51±2.20(+)	78.19±3.90(+)	78.32±3.12(+)	84.12±3.13(=)	84.52±0.51(=)	78.49±3.42(+)	84.95±4.28(=)	85.14±0.90
CLL-SUB-111	82.11±1.84(=)	81.20±1.23(=)	70.81±0.21(+)	78.17±1.72(+)	69.39±1.67(+)	66.85±0.36(+)	80.19±0.23(+)	82.32±0.52
TOX-171	63.14±1.91(+)	59.18±2.75(+)	75.25±0.59(+)	68.17±3.01(+)	79.19±0.91(=)	65.36±2.48(+)	77.56±1.69(+)	79.91±2.45
Movement_libras	59.67±1.04(+)	61.50±0.29(+)	69.01±2.45(=)	45.82±2.84(+)	70.27±0.53(=)	69.00±4.57(=)	70.11±1.10(=)	69.35±0.28
Arrhythmia16	69.21±0.98(+)	52.57±2.12(+)	60.00±1.31(+)	59.19±3.81(+)	67.63±3.19(+)	60.16±2.38(+)	68.30±2.38(+)	68.89±1.12
Isolet	57.24±4.60(+)	64.52±3.31(+)	80.90±2.46(=)	64.41±5.23(=)	66.34±3.37(+)	81.44±2.75(-)	73.72±2.91(+)	79.15±5.63
COIL20	79.32±3.25(=)	78.71±2.84(=)	73.61±1.56(+)	67.13±3.07(+)	78.40±0.61(=)	74.19±3.16(+)	72.52±1.94(+)	79.83±1.39
Air	53.12±4.19(+)	60.12±2.39(+)	58.16±2.01(+)	54.42±2.38(+)	60.77±1.68(+)	56.73±1.77(+)	69.73±2.83(-)	68.00±1.95
Number	2	3	1	3	4	3	5	12
Average	66.01	62.57	68.80	61.66	71.55	68.94	71.58	75.30



(a) NB (b) kNN (c) SVM (d) XGBoost

Fig. 9. Classification performance comparison of CSMI with other methods. Win/tie/loss means number of times CSMI performs better than/equal to/worse than compared methods on four classifiers (a) NB, (b) kNN (k=3), (c) SVM, and (d) XGBoost.

CSMI. The notations '+,' '=', and '−' indicate that the CSMI performs 'better,' 'equal to,' and 'worse' than the other methods. This is also explicitly shown by means of bar charts in Fig. 9 in which '+,' '=', and '−' are indicated by Win/tie/loss on four classifiers, NB, kNN (k = 3), SVM, and XGBoost.

It can be observed from Tables VII - X that CSMI has the highest average accuracy of 73.46%, 75.30%, 79.19%, and

TABLE IX
AVERAGE ACCURACY (MEAN ± STD.) WITH STATISTICAL SIGNIFICANCE ON SVM

Data sets	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	76.31±2.54(+)	71.67±1.06(+)	85.01±3.07(=)	72.92±1.25(+)	80.10±1.52(+)	78.34±2.08(+)	87.27±1.01(-)	85.41±2.72
Lymphoma	84.51±3.52(+)	79.37±1.43(+)	87.16±2.49(+)	73.12±4.85(+)	88.50±2.96(+)	81.35±3.65(+)	89.18±2.06(+)	95.31±4.17
ORL	63.94±1.44(+)	79.67±0.12(=)	72.05±1.42(+)	77.49±1.76(=)	79.11±0.71(=)	72.85±2.02(+)	74.60±3.77(+)	78.19±1.01
WarpAR10P	55.26±1.45(+)	61.57±1.03(+)	79.65±0.43(-)	60.01±3.89(+)	74.17±2.90(=)	78.31±3.91(-)	72.81±2.18(=)	74.09±1.09
WarPIE10P	87.14±2.10(=)	81.19±3.68(+)	79.61±4.50(+)	69.14±2.97(+)	84.70±1.31(=)	78.16±4.37(+)	83.40±3.91(+)	86.02±2.43
Orlraws10P	69.26±5.81(+)	65.12±4.63(+)	79.42±3.29(+)	80.01±2.43(+)	83.59±5.30(+)	80.81±4.71(+)	92.71±2.73(=)	91.19±2.10
Yale	69.30±1.29(+)	60.41±2.56(+)	64.18±0.97(+)	63.10±2.07(+)	73.18±2.79(=)	72.12±1.79(=)	66.90±3.76(+)	73.71±1.10
Lung_discrete	78.12±4.93(-)	71.92±5.45(+)	77.33±3.14(=)	65.12±5.06(+)	77.28±4.49(=)	70.57±3.91(+)	69.10±4.01(+)	76.01±3.34
PCMAC	75.80±3.06(+)	84.01±0.96(+)	79.45±2.13(+)	74.81±4.93(+)	87.20±1.32(+)	88.41±1.32(=)	83.50±2.64(+)	89.27±0.78
CLL-SUB-111	87.06±5.12(=)	71.16±4.57(+)	78.10±3.12(+)	72.05±3.52(+)	80.29±6.95(+)	73.44±4.10(+)	85.46±5.01(=)	86.51±2.31
TOX-171	61.08±1.82(+)	78.54±4.39(=)	62.28±2.09(+)	73.48±3.61(=)	64.14±4.27(+)	59.13±2.05(+)	67.42±2.40(+)	72.17±0.97
Movement_libras	78.10±7.36(=)	53.71±4.72(+)	65.27±6.51(+)	41.04±8.75(+)	68.27±5.90(+)	72.11±6.01(+)	74.84±7.12(+)	79.30±5.05
Arrhythmia16	49.63±5.90(+)	69.41±6.32(-)	60.53±4.90(+)	65.82±3.20(=)	61.66±4.10(+)	64.12±4.60(=)	64.27±3.40(=)	65.17±5.61
Isolet	74.11±3.05(+)	77.81±2.71(+)	85.60±3.62(=)	77.02±4.66(+)	85.04±5.09(=)	70.43±2.07(+)	86.19±5.81(=)	85.55±6.55
COIL20	78.51±3.70(+)	73.96±2.08(+)	85.09±4.56(-)	67.46±3.51(+)	79.28±2.45(+)	82.58±3.17(=)	85.20±3.07(-)	82.62±2.19
Air	65.07±1.35(+)	62.13±2.68(+)	76.67±1.33(-)	76.15±1.52(-)	67.92±2.09(+)	63.50±2.61(+)	68.22±0.71(+)	71.89±1.41
Number	5	1	4	2	3	2	4	9
Average	71.46	70.25	74.38	67.91	75.80	72.69	76.63	79.19

TABLE X
AVERAGE ACCURACY (MEAN ± STD.) WITH STATISTICAL SIGNIFICANCE ON XGBOOST

Data sets	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	54.24±3.76(+)	67.12±4.32(=)	61.44±4.91(+)	56.20±3.44(+)	60.02±2.26(+)	55.50±5.44(+)	66.92±2.05(+)	78.38±1.56
Lymphoma	91.65±1.22(+)	89.00±1.62(+)	88.06±1.19(+)	87.26±2.18(+)	91.58±2.29(+)	83.25±1.39(+)	92.15±2.44(=)	94.26±1.56
ORL	55.56±6.13(=)	45.64±5.40(+)	43.29±4.15(+)	46.52±4.75(+)	50.95±5.22(+)	57.80±4.26(=)	57.36±4.26(=)	56.86±3.78
WarpAR10P	51.38±1.22(+)	78.91±1.75(=)	48.32±0.87(+)	73.52±1.75(+)	56.48±1.80(+)	41.10±1.01(+)	71.48±0.34(+)	78.15±0.34
WarPIE10P	72.19±4.76(=)	52.14±3.81(+)	62.73±4.54(+)	54.76±4.22(+)	67.03±3.29(+)	64.58±3.49(+)	69.39±3.56(+)	73.64±3.05
Orlraws10P	85.40±0.11(+)	75.69±1.01(+)	89.30±1.89(+)	81.95±1.28(+)	92.28±1.77(=)	89.85±0.26(+)	94.23±1.24(=)	92.47±0.83
Yale	64.63±4.51(+)	58.28±3.09(+)	75.82±3.42(=)	75.79±2.57(=)	68.82±4.05(+)	66.68±2.67(+)	67.03±1.39(+)	75.10±3.73
Lung_discrete	69.04±5.68(=)	59.47±6.56(+)	77.49±4.04(-)	68.36±5.21(=)	60.68±4.97(+)	71.44±5.27(+)	70.16±4.31(=)	69.57±3.40
PCMAC	72.76±6.48(+)	80.41±4.17(=)	84.91±3.20(-)	79.26±5.22(=)	78.71±6.29(=)	80.42±5.62(=)	71.52±4.31(+)	79.64±3.62
CLL-SUB-111	79.92±2.82(+)	74.13±4.15(+)	82.33±3.03(+)	77.04±3.89(+)	85.29±4.31(+)	78.59±2.40(+)	89.61±2.33(=)	89.92±2.07
TOX-171	69.07±1.62(+)	51.60±1.80(+)	72.42±1.85(+)	75.86±2.02(+)	78.48±0.19(+)	71.30±1.94(+)	91.54±2.67(=)	92.67±1.24
Movement_libras	78.58±1.83(=)	60.73±1.09(+)	68.43±1.38(+)	54.98±0.65(+)	78.13±1.77(=)	51.82±0.19(+)	73.14±1.69(+)	79.62±0.84
Arrhythmia16	54.76±2.70(+)	59.83±2.12(+)	57.79±3.05(+)	54.11±2.93(+)	68.01±3.38(+)	62.98±2.38(+)	61.38±2.07(+)	70.79±3.47
Isolet	56.08±5.97(+)	71.26±6.11(+)	83.40±5.05(-)	60.69±6.20(+)	78.34±4.19(-)	73.75±3.56(+)	79.29±6.00(-)	76.89±5.91
COIL20	78.71±2.61(=)	73.03±2.19(+)	88.56±1.32(-)	81.43±1.57(-)	71.50±1.08(+)	79.71±1.22(=)	70.79±0.17(+)	78.35±2.49
Air	57.61±4.83(+)	60.91±3.04(+)	53.67±2.93(+)	61.84±2.37(+)	70.47±4.73(+)	62.47±3.61(+)	66.53±3.57(-)	69.84±2.17
Number	3	1	5	1	2	1	4	11
Average	67.47	65.67	70.16	66.74	71.52	67.23	73.38	77.25

77.26% over all the four classifiers, NB, kNN, SVM, and XGBoost, respectively. CSMI also generates better average accuracy on 10, 12, 09, and 11 data sets out of 16 using the same classifiers, respectively. In Table VII, DCSF achieves the second-best accuracy followed by MRI, mRMR, JMI, MIFS, CIFE, and MIM. In Table VIII, CSMI outperforms MIM, JMI, CIFE, MIFS, MRI, mRMR, and DCSF by up to 18.11%, 16.90%, 12.33%, 8.63%, 8.44%, 4.98%, and 4.94%, respectively. In Table IX, CSMI shows 16.61%, 12.72%, 10.84%, 6.46%, 8.94%, 5.16%, and 3.34% better performance than MIM, JMI, CIFE, MIFS, MRI, mRMR, and DCSF, respectively. Similarly, in Table X, CSMI outperforms JMI, MIM, MRI, CIFE, MIFS, mRMR, and DCSF by up to 14.99%, 13.60%, 12.97%, 12.66%, 9.17%, 7.41%, and 5.09%, respectively. The performance of mRMR is better than MIFS for all classifiers, which is consistent with earlier study [25], [27]. Here, the CSMI shows significant improvement over the seven methods compared on four different classifiers.

The results in the tables also show that there is a difference in the classifiers used in these experiments. For instance, on the Relathe data set, CSMI obtains the best average accuracy compared to the other seven methods on kNN. On the other hand, in the case of SVM the performance of CSMI is lower than DCSF, higher than CIFE, JMI, MIM, mRMR, and MRI, and comparable to MIFS in terms of average accuracy. Similarly, the classifiers show the diverse classification performance for the rest of the methods on different data sets. Earlier studies indicate that the classification

TABLE XI
HIGHEST AVERAGE ACCURACY (%) OF FOUR CLASSIFIERS WITH EIGHT METHODS

Data sets	All	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	81.64	70.16	70.14	74.07	65.12	76.63	73.84	76.05	**82.33**
Lymphoma	78.56	84.66	79.33	93.44	81.87	90.59	83.28	89.61	**96.81**
ORL	**74.56**	68.78	69.69	65.78	66.55	69.27	69.65	70.93	71.62
WarpAR10P	70.87	55.28	58.49	61.06	59.89	71.06	64.82	64.82	**72.76**
WarPIE10P	85.56	78.45	71.78	76.01	69.74	77.51	72.88	86.56	**87.51**
Orlraws10P	86.66	78.27	69.57	81.37	78.96	87.8	78.48	88.66	**90.74**
Yale	62.50	54.96	57.61	65.58	59.21	62.38	68.32	65.13	**69.35**
Lung_discrete	**79.27**	74.72	68.6	76.49	73.61	68.2	77.7	75.11	74.26
PCMAC	**87.56**	78.63	82.78	80.99	84.37	83.85	80.24	81.39	84.83
CLL_SUB_111	67.64	82.67	75.63	74.51	84.3	83.28	78.73	86.01	**87.13**
TOX-171	72.11	69.65	69.10	75.4	72.21	75.69	69.86	81.03	**82.13**
Movement_libras	72.27	76.35	65.59	70.92	54.72	68.54	70.69	73.31	**77.24**
Arrhythmia	**70.25**	57.49	63.16	54.10	61.90	66.59	61.84	62.32	68.6
Isolet	**94.65**	65.36	73.24	79.24	74.31	73.23	76.42	79.41	82.98
COIL20	**98.84**	77.79	73.28	79.86	73.24	79.17	79.66	77.81	82.02
Air	69.25	61.91	64.53	65.33	61.94	66.62	60.79	69.73	**70.34**
Average	77.32	70.15	68.51	71.92	69.08	73.79	71.52	75.49	78.53

TABLE XII
AVERAGE F - SCORE VALUE OF FOUR CLASSIFIERS WITH EIGHT METHODS

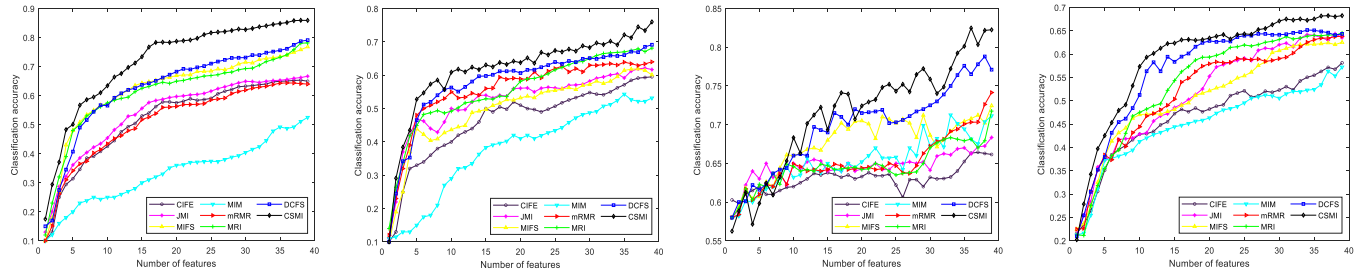
Data sets	All	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Relathe	83.58	66.35	76.14	72.31	73.42	79.68	74.65	79.26	**84.10**
Lymphoma	63.24	26.70	48.53	43.32	44.48	50.44	68.79	70.41	**72.02**
ORL	**76.47**	49.89	65.35	51.91	45.60	60.93	64.94	62.53	66.41
WarpAR10P	68.79	46.91	54.42	36.76	62.68	55.47	49.04	50.19	**70.52**
WarPIE10P	86.77	80.13	82.45	80.04	81.20	85.95	81.63	88.45	**91.26**
Orlraws10P	83.54	74.35	82.24	58.00	80.98	55.69	69.21	83.54	**87.54**
Yale	55.75	23.16	56.76	31.80	40.62	39.36	61.98	58.16	**70.57**
Lung_discrete	73.66	76.52	61.62	73.57	70.43	68.18	**79.72**	73.83	76.93
PCMAC	**87.95**	76.68	83.25	72.43	81.31	84.24	79.21	82.49	84.54
CLL_SUB_111	58.61	50.75	55.64	47.47	65.04	55.60	63.40	68.85	**74.36**
TOX-171	71.37	44.07	57.00	45.23	55.95	55.26	73.69	77.92	**79.05**
Movement_libras	67.17	57.53	68.28	61.97	46.24	69.49	67.56	70.53	**76.83**
Arrhythmia	26.20	28.09	21.60	16.89	24.27	21.58	33.12	36.48	**41.78**
Isolet	**83.82**	56.37	60.72	43.50	48.61	62.54	68.78	75.35	80.30
COIL20	**93.96**	54.30	84.09	41.06	88.45	87.17	76.59	82.81	90.84
Air	62.10	58.90	51.11	46.29	52.89	53.73	59.35	64.38	**71.56**
Average	71.02	54.60	63.38	52.03	59.31	61.77	68.42	69.20	74.47

TABLE XIII
AVERAGE ACCURACY (MEAN $\pm$ STD.) OF FOUR CLASSIFIERS WITH EIGHT METHODS ON AUTOMATIC THRESHOLD

Data sets	T	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Yale	10	26.17±7.13	27.35±5.95	22.60±17.90	19±9.5	33.30±21.40	14.535±10.45	17.65±13.25	49.06±8.02
PCMAC	11.6	76.11±3.89	77.09±4.79	75.47±4.53	77.09±4.79	82.91±1.45	78.04±8.47	81.09±0.59	85.75±1.25
Orlraws10P	13.3	70±6	74±14	73.69±6.69	75.99±7.99	57.99±15.99	54.66±14.66	54±18	80.20±5.20
Isolet	9.6	16.02±3.20	43.15±4.25	45.20±3.90	40.76±0.51	52.01±0.81	45.23±9.89	39.96±1.06	59.21±0.90
Relathe	12.0	67.47±0.87	70.4±2.9	76.75±0.19	73.06±1.64	76.19±0.56	69.3±0.15	73.83±0.39	75.78±0.84
TOX-171	12.4	46.50±9.30	46.51±4.65	43.60±15.42	37.20±6.97	36.04±1.16	36.71±0.82	43.01±3.48	53.77±9.49
warpAR10P	11.2	30.28±6.08	45.45±0	44.84±8.29	41.51±3.03	47.92±1.47	40.16±11.05	31.78±1.51	48.54±2.00
COIL20	10	45.41±8.19	72.91±12.64	67.91±0.08	69.99±18.05	71.24±10.41	72.43±11.73	74.8±12.97	77.56±13.91
warPIE10P	11.2	35.83±1.87	78.29±6.61	69.57±1.38	59.4±2.8	67.91±5.67	50.15±8.47	51.85±2.85	79.11±5.46
CLL-SUB-111	13.4	55.32±1.82	60.48±8.91	49.17±10.17	62.49±5.35	48.17±8.97	54.99±0.99	57.14±0	67.55±7.44

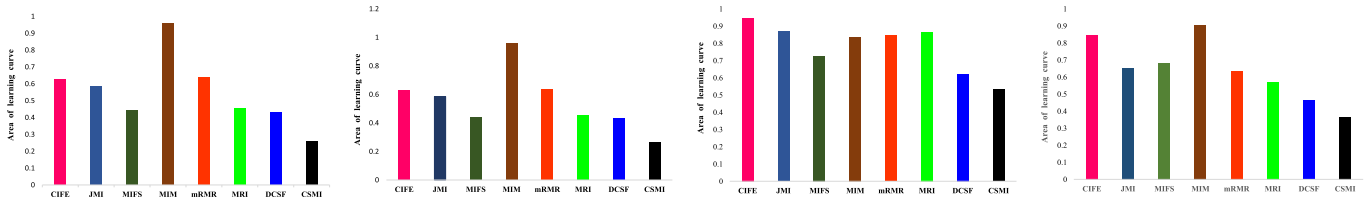
performance is dependent on the data set and classifier [25], [60]. Table XI shows the maximum highest accuracy over four classifiers as indicated by bold. The last row shows the average value over the data sets, which asserts that the CSMI has 14.62%, 13.67%, 11.43%, 9.80%, 9.19%, 6.42%, and 4.02% improved results over JMI, MIM, CIFE, MRI, MIFS, mRMR, and DCSF, respectively. Similarly, Table XII shows the average F - Score value over four classifiers, and the best performance is indicated by bold. It can be observed from the table CSMI shows better performance as compared to other methods.

Further, Table XIII and Table XIV show the average accuracy and computational time for the automatic thresholds of eight methods on ten high dimensional data sets, respectively. It can be observed from Table XIII that the CSMI shows significant improvement in accuracy on seven data sets out of 10. However, it shows comparable performance with the other three data sets. Note that the classification performance



(a) Isolet (b) ORL (c) Relathe (d) Yale

Fig. 10. Average classification accuracy achieved on four representative data sets.



(a) Isolet (b) ORL (c) Relathe (d) Yale

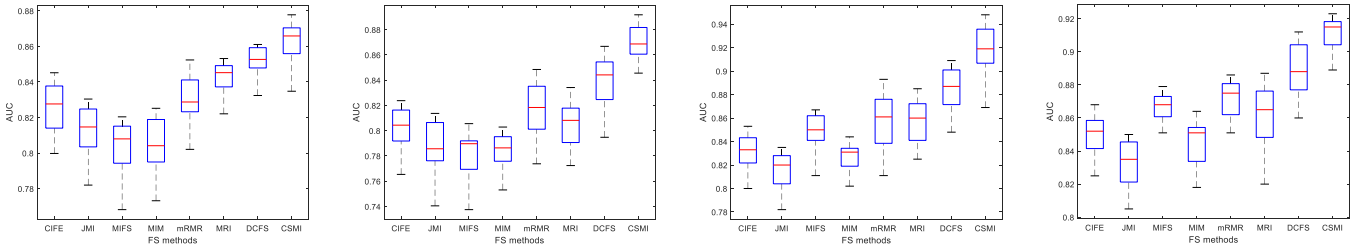
Fig. 11. Goodness-of-fit interpretation of FS methods based on learning curves on four representative data sets shown in Fig. 10.

TABLE XIV
COMPUTATIONAL TIME OF EIGHT METHODS (IN SECONDS)

	T	CIFE	JMI	MIFS	MIM	mRMR	MRI	DCSF	CSMI
Yale	10	4.80	5.00	4.90	5.10	5.30	5.40	4.90	5.10
	40	21.34	21.24	21.72	22.12	22.44	25.11	23.20	24.59
PCMAC	11.6	132	130	129	129	131	134	130	139
	40	496	495	479	501	457	471	497	479
Orlraws10P	13.3	44.80	45.90	45.10	46.00	43.10	43.90	44.50	46.30
	40	110	74	179	130	136	180	174	192
Isolet	9.6	26.70	29.50	29.40	29.40	25.90	28.10	28.20	30
	40	119	135	171	130	120	119	135	189
Relathe	12.0	141	130	132	129	135	142	132	150
	40	496	492	496	490	493	483	487	503
TOX-171	12.4	41.90	41.30	42.00	42.40	42.50	42.40	42.50	42.60
	40	151	153	141	147	152	140	153	162
warpAR10P	11.2	10.10	11.10	11.50	11.20	10.90	11.10	10.60	11.70
	40	42.30	45.10	40.50	45.50	43.30	41.25	44.40	44.55
COIL20	10	36.20	46.10	45.90	45.20	38.50	42.30	43.80	44.30
	40	155	192	170	192	178	180	190	212
warPIE10P	11.2	16.30	17.70	18.10	18.10	16.10	17.20	16.90	18.20
	40	67	71	69	71	67	70	69	85
CLL-SUB-11	13.4	60	81	73	89	60	87	90	81
	40	207	201	194	200	205	180	201	224

of an ML model is dependent on the data set and classifier used [25], [60]. Finally, Table XIV shows the computational time of eight methods on the fixed threshold 40 and automatic threshold. For the CSMI, we show the computational time for a single subset of features averaged over all subsets of features selected for different class labels for a given data set.

We chose four representative data sets, namely Isolet, ORL, Yale, and Relathe, for more effective, precise, and further performance analysis. The data sets contain continuous and discrete values. Fig. 10(a), Fig. 10(b), Fig. 10(c), and Fig. 10(d) show the learning curve of different FS methods with respect to increasing number of features on Isolet, ORL, Relathe, and Yale data sets, respectively. It can be observed from these figures that CSMI outperforms all the other methods. Specifically, CSMI has the highest, and MIM has the lowest performance for Isolet data set (Fig. 10(a)). The CSMI outperforms DCSF, MRI, mRMR, MIM, JMI and CIFE by up to 9.03%, 10.35%, 15.78%, 30.04%, 20.91%, 18.81% and 21.66%, respectively as per Fig. 10(b). However, for the Relathe data set, CSMI initially performs poorer, i.e., upto 5 features, comparable to DCSF for fifteen and nineteen feature subset and for the rest, it outperforms the other seven methods as shown Fig. 10(c). Note that the increasing number of features does not guarantee the improved performance of the ML model as it depends on the type of information shared by the new features with predicting class labels. Earlier studies also have similar characteristics of the ML model [27], [50]. Table XI shows that CSMI obtains 82.33%, 71.62%, 69.35%, and 82.98% on Relathe, ORL, Yale, and Isolet, respectively, in terms of the highest average accuracy on four classifiers. It can be observed from Fig. 10 that the CSMI shows a more steep learning curve as compared to others, especially on the selection of the first few features. This shows that the CSMI is more efficient than other FS methods as more the steepness of a learning curve, the better is the FS method [71]. The steepness of the learning curve can also be directly measured in terms of their "goodness-of-fit" [71]. The goodness-of-fit is measured as the region covered between the highest accuracy and the learning curve associated with the FS algorithm. The lesser the area under goodness-of-fit, the better is the algorithm. It is clear from Fig. 11 that the CSMI has the minimum area for all the four representative data sets. We have normalized the area under learning curve between 0 and 1. Finally, Fig. 12 shows the performance of FS methods using box plots that display the AUCs. Note that the higher the AUC the better is the FS method. The average classification performance of the CSMI on four classifiers is better, which means that the selected features by the proposed approach provide a better generality and are independent of the individual classifier [71].



(a) NB (b) kNN (c) SVM (d) XGBoost

Fig. 12. AUC (the higher the AUC the better) for all benchmark data sets.

TABLE XV
ACCURACY COMPARISON OF STATE-OF-THE ART CLASSIFIERS AND PROPOSED METHOD ON FOUR DATA SETS

Data set	State-of-the-art Classifier	NB-CSMI	kNN-CSMI	SVM-CSMI	XGBoost-CSMI
Arrhythmia16	64.30 [68]	57.10	68.89	65.17	**70.79**
Yale	**81.33** [69]	61.89	58.51	73.71	75.10
CLL-SUB-111	81.62 [70]	82.46	82.32	86.51	**89.92**

TABLE XVI
AVERAGE ACCURACY (MEAN ± STD.) OF LIFT [21] AND CSMI ON FOUR CLASSIFIERS (NB, kNN, SVM, AND XGBOOST)

	NB		kNN		SVM		XGBoost	
Data set	LIFT	CSMI	LIFT	CSMI	LIFT	CSMI	LIFT	CSMI
Yale	35.93±0.29	**61.89±1.88**	53.35±1.07	**58.51±1.10**	37.81±0.58	**73.71±1.10**	25.57±19.57	**75.10±3.73**
PCMAC	50.73±0.06	**78.41±1.54**	**87.09±4.49**	85.14±0.90	54.76±0.10	**89.27±0.78**	**90.49±0.48**	79.64±3.62
Orlraws10P	80.90±0.99	**87.00±2.71**	83.82±0.23	**85.32±1.32**	78.90±0.73	**91.19±2.10**	26.92±14.85	**73.64±0.83**
Isolet	54.02±1.07	**67.64±4.57**	**81.10±0.80**	79.15±5.63	71.31±1.05	**85.55±6.55**	14.92±3.48	**76.89±5.91**
Relathe	54.84±0.06	**83.10±3.52**	**85.36±0.75**	73.40±1.23	56.06±0.06	**85.41±2.72**	**89.77±0.23**	78.38±1.56
TOX-171	45.08±1.87	**78.02±1.08**	72.92±2.67	**79.91±2.45**	57.77±6.29	**72.17±0.97**	**98.15±0.74**	92.67±1.24
warpAR10P	25.76±0.65	**69.31±0.29**	56.30±1.65	**64.83±2.92**	38.38±0.67	**86.02±2.43**	**84.00±12.63**	78.15±0.34
COIL20	**82.56±0.77**	78.61±2.59	**97.93±0.20**	79.83±1.39	**91.62±0.23**	82.62±2.19	25.94±17.20	**78.35±2.49**
warPIE10P	31.28±2.59	**89.19±1.42**	68.00±2.34	**93.23±1.07**	42.80±2.51	**86.02±2.43**	69.95±8.82	**73.64±3.05**
CLL-SUB-111	58.73±0.37	82.46±2.41	77.56±0.51	**82.32±0.52**	54.05±0.97	**86.51±2.31**	**95.85±1.54**	89.92±2.07
Number	1	9	4	6	1	9	5	5
Average	51.98	**77.56**	76.53	**78.16**	58.34	**83.84**	61.15	**79.63**

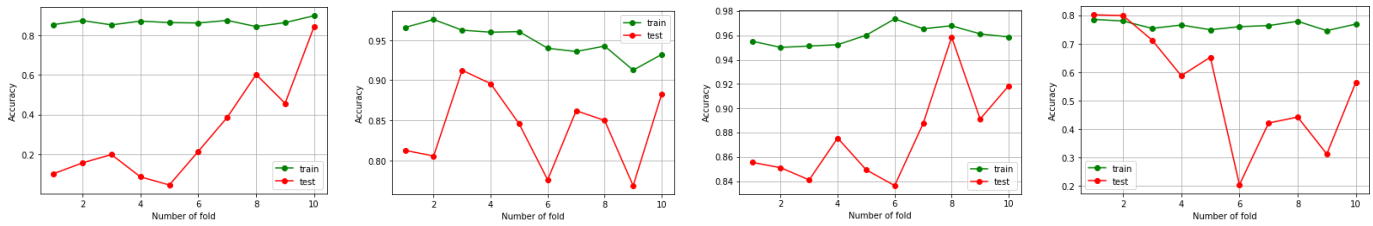
Comparison With the State-of-the-Art Classifiers: We now compare the accuracy of CSMI generated after applying four classifiers, i.e., NB, kNN, SVM, and XGBoost with the recently proposed state-of-the art classifiers as shown in Table XV. It can be observed that CSMI shows better accuracy on the data set Arrhythmia16 and CLL-SUB-111. However, CSMI outperformed by the state-of-the art classifier as shown in [69] on the Yale data set. This may be noted that the accuracy of the proposed CSMI is obtained using top 40 (threshold) selected features, rather than all and thus reduces the execution time of the final ML model.

B. Comparison Results With Multi-Label Classification

Here, we show the comparison results and their analysis with respect to the multi-label classification method LIFT [21]. A multi-label classification learning deals with the problems where single instance is associated with a set of class labels, whereas in multi-class learning single instance is associated with one class label. Therefore, for the sake of comparison, we map LIFT into the multi-class environment as follows. Let us consider a multi-class training set $D = \{(\mathbf{x_i}, c_j)|1 \leq i \leq n, 1 \leq j \leq m\}$, where $\mathbf{x_i}$ is the feature vector of the input data and c_j is the corresponding class. Firstly, the class specific instances are constructed as,

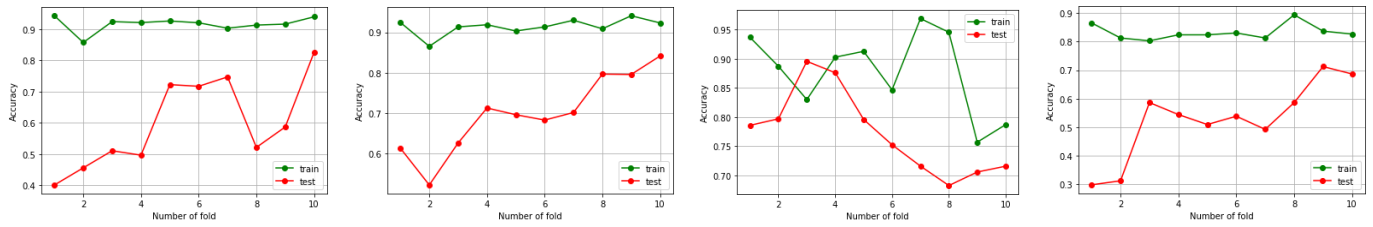
$$I^+ = \{\mathbf{x_i}|(\mathbf{x_i}, c_j) \in D, c_j = c_k\}$$
$$I^- = \{\mathbf{x_i}|(\mathbf{x_i}, c_j) \in D, c_j \neq c_k\}$$

which means I^+ are the instances having class label c_k and I^- are the instances belonging to all other classes. Similar to LIFT, k-means clustering algorithm is employed to I^+ and I^- to generate I_m^+ and I_n^- number of clusters, respectively. Then, we follow the same procedure of the LIFT (as discussed in Section III of [21]) to generate class-specific features. Finally, we apply the classification models over the generated features. Table XVI shows the average accuracy of LIFT and CSMI on ten high dimensional data sets using four classifiers separately.



(a) Isolet (b) ORL (c) Relathe (d) Yale

Fig. 13. Classification performance of data sets (Isolet, ORL, Relathe, and Yale) for K-fold cross-validation (K $= \{1, 2, \ldots, 10\}$ using NB.



(a) Isolet (b) ORL (c) Relathe (d) Yale

Fig. 14. Classification performance of data sets (Isolet, ORL, Relathe, and Yale) for K-fold cross-validation (K $= \{1, 2, \ldots, 10\}$ using kNN.

The row labeled with "Number" in the table shows the number of times the corresponding method outperforms the other. The row labeled with "Average" shows the average accuracy of the methods over all data sets. The best performances are marked with bold values in the table. It can be observed that the average accuracy of CSMI is far better than LIFT for all classifiers except kNN. Specifically, the CSMI outperforms LIFT by 32.98%, 30.41%, and 23.20% using NB, SVM, and XGBoost respectively.

The reason for the better performance of the proposed framework is that the CSMI can generate a distinct feature subspace for each class label in which the target class label can be effectively separated from the rest of the class labels. For a better explanation, let us consider a data set with the female and male as the two class labels. In this case, a single subset of features, including body strength and voice pitch range, as selected features are sufficient to separate the class labels. However, if the data set has three class labels: female, male, and transgender, selecting the single subset of features to separate these class labels can be misleading. Hence, based on the data set, selecting different subsets of features that can separate the female or male, or transgender from the rest of the class labels would be efficient. We apply the same concept in this study and get better results for most of the data sets.

VI. Conclusion

In this article, we have proposed a novel feature selection method called CSMI. We have also presented a general framework for the complete classification process with three processing stages, i.e., class binarization, class balancing, and feature selection with the proposed CSMI, and the final stage of classification. Through extensive experiments on sixteen benchmark data sets using four classifiers, we have shown that the proposed CSMI outperforms five traditional ITFS (i.e., CIFE, JMI, MIFS, MIM, and mRMR) and two state-of-the-art methods (i.e., MRI and DCSF) with respect to average accuracy, highest average accuracy, and AUC. CSMI shows an average improvement of 4.39%, 10.26%, 6.63%, 18.95%, 10.02%, 15.78%, and 13.55% accuracy over DCSF, MRI, mRMR, MIM, MIFS, JMI, and CIFE, respectively. CSMI also shows an average improvement of 22.29% accuracy over multi-label classification method LIFT. We have also shown that the CSMI has acceptable time complexity like all these methods. However, we have explored the over-sampling and under-sampling techniques for balancing the data set and the feature threshold is also fixed while preparing the ML model. This is noteworthy that multi-label classification problems are of great importance in natural language processing domain such as sarcasm detection. So, our future efforts will be made to formulate the label-specific information theory as per multi-label classification environment.

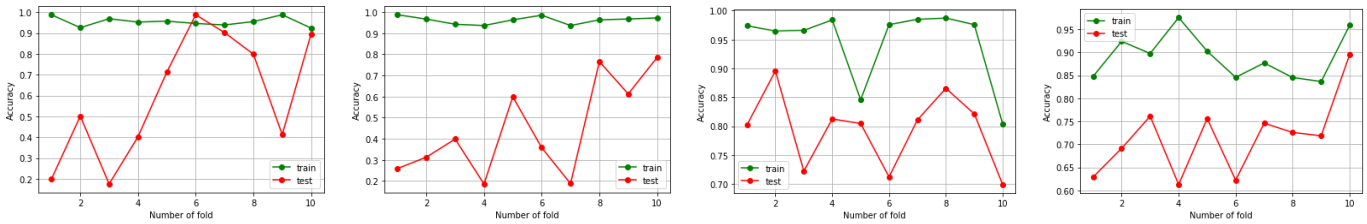
Appendix A
Proof of Equation 27

The following proof make use of the identity, $I(A;B|C) - I(A;B) = I(A;C|B) - I(A;C)$ discussed in paper [20], *Section IV-A*.

$$I(f_k;P_i|f_j) = I(f_k;P_i) - I(f_k;f_j) + I(f_k;f_j|P_i) \quad (31)$$
$$I(f_k;C|f_j) = I(f_k;C) - I(f_k;f_j) + I(f_k;f_j|C) \quad (32)$$
$$I(f_j;P_i|f_k) = I(f_j;P_i) - I(f_j;f_k) + I(f_j;f_k|P_i) \quad (33)$$

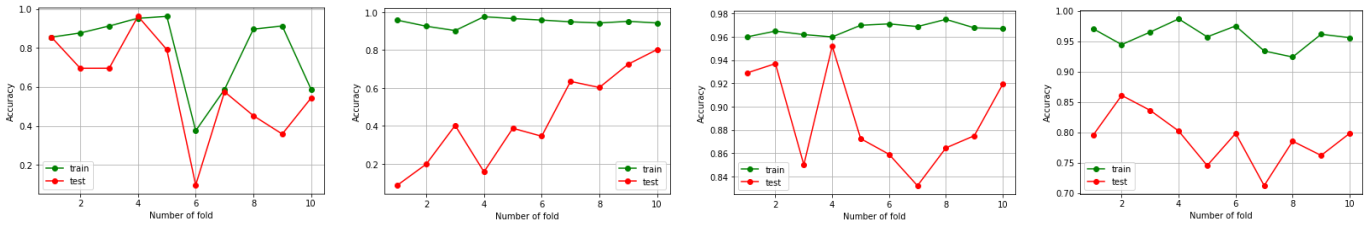
We can rewrite Eq. 26 as:

$$J(f_k) = \sum_{f_j \in S_{c_i}} \{I(f_k;P_i) - I(f_k;f_j) + I(f_k;f_j|P_i) - \{I(f_k;C) - I(f_k;f_j) + I(f_k;f_j|C)\} + I(f_j;P_i) - I(f_j;f_k) + I(f_j;f_k|P_i) - 2I(f_j;f_k)\} \quad (34)$$



(a) Isolet (b) ORL (c) Relathe (d) Yale

Fig. 15. Classification performance of data sets (Isolet, ORL, Relathe, and Yale) for K-fold cross-validation (K $= \{1, 2, \ldots, 10\}$ using SVM.



(a) Isolet (b) ORL (c) Relathe (d) Yale

Fig. 16. Classification performance of data sets (Isolet, ORL, Relathe, and Yale) for K-fold cross-validation (K $= \{1, 2, \ldots, 10\}$ using XGBoost.

TABLE XVII

SUMMARY OF DIFFERENCE BETWEEN CLASSIFICATION PERFORMANCE (ACCURACY) OF TRAINING AND TEST SET ON DATA SETS (ISOLET, ORL, RELATHE, YALE) USING CLASSIFIERS NB, kNN, SVM, AND XGBOOST

	NB				kNN				SVM				XGBoost			
K-fold cross-validation	Min.	Max.	Ave.	Std.	Min.	Max.	Ave.	Std.	Min.	Max.	Ave.	Std.	Min.	Max.	Ave.	Std.
1	1.6	75.03	25.47	33.51	15.09	56.68	39.27	19.77	17.15	78.88	47.69	32.68	0	86.96	26.90	40.76
2	1.89	71.57	25.08	31.59	9.01	40.09	33.37	17.47	6.89	42.44	34.51	25.23	2.8	72.67	25.5	32.08
3	4.11	65.24	21.33	29.42	6.56	41.29	24.56	14.49	13.63	79.29	42.90	29.76	11.2	49.99	23.95	17.95
4	6.42	78.33	27.55	34.23	2.67	42.4	23.40	16.51	17.12	55.03	45.88	24.86	0.07	81.78	25.51	38.41
5	9.65	81.69	28.47	35.48	20.35	31.4	21.05	8.07	4.1	36.48	19.92	13.81	9.75	57.82	26.47	21.42
6	13.7	64.6	37.55	26.27	9.42	29.14	20.49	08.25	4.18	62.76	28.90	24.54	11.22	61.22	29.51	22.22
7	7.37	48.78	24.54	20.46	15.57	32	23.94	6.77	3.66	17.38	27.26	32.27	1.19	31.42	17.13	12.85
8	0.09	33.67	16.98	14.66	11.24	39.11	26.84	11.68	11.94	19.78	14.87	3.68	11.06	44.44	25.85	16.06
9	7	43.31	26.32	18.31	5.12	92.89	16.24	11.81	11.8	57.5	30.06	21.05	9.28	55.46	26.83	19.21
10	4.02	**20.42**	**10.28**	**7.42**	8.2	**13.98**	**10.16**	**3.10**	2.8	**18.71**	**9.58**	**6.84**	4.78	**15.78**	**8.71**	**7.81**

$$= \sum_{f_j \in S_{c_i}} \{I(f_k; P_i) + 2I(f_k; f_j|P_i) + I(f_j; P_i) - \{I(f_k; C) + I(f_k; f_j|C)\} - 2I(f_j; f_k)\}$$

$$J(f_k) = \sum_{f_j \in S_{c_i}} \{I(f_k; P_i) + 2I(f_k; f_j|P_i) + I(f_j; P_i) - \{I(f_k; C) + I(f_k; f_j|C)\} - 2I(f_j; f_k)\} \quad (35)$$

APPENDIX B
OVERFITTING

See Figs. 13–16 and Table XVII.

REFERENCES

[1] M. S. Mahmud, J. Z. Huang, S. Salloum, T. Z. Emara, and K. Sadatdiynov, "A survey of data partitioning and sampling methods to support big data analysis," *Big Data Mining Anal.*, vol. 3, no. 2, pp. 85–101, Jun. 2020.

[2] R. Sheikhpour, M. A. Sarram, S. Gharaghani, and M. A. Z. Chahooki, "A survey on semi-supervised feature selection methods," *Pattern Recognit.*, vol. 64, pp. 141–158, Apr. 2017.

[3] Z. Zhang, L. Bai, Y. Liang, and E. Hancockc, "Joint hypergraph learning and sparse regression for feature selection," *Pattern Recognit.*, vol. 63, pp. 291–309, Mar. 2017.

[4] J. Li *et al.*, "Feature selection: A data perspective," *ACM Comput. Surv.*, vol. 50, no. 6, pp. 1–45, Dec. 2017.

[5] J.-B. Yang and C.-J. Ong, "Feature selection using probabilistic prediction of support vector regression," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 22, no. 6, pp. 954–962, Jun. 2011.

[6] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.

[7] 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.

[8] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.

[9] D. Koller and M. Sahami, "Toward optimal feature selection," Stanford InfoLab, Stanford, CA, USA, Tech. Rep., 1996.

[10] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[11] Y. Saeys, I. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007.

[12] P. E. Meyer, C. Schretter, and G. Bontempi, "Information-theoretic feature selection in microarray data using variable complementarity," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 3, pp. 261–274, Jun. 2008.

[13] V. Bolón-Canedo, N. Sánchez-Maroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, "A review of microarray datasets and applied feature selection methods," *Inf. Sci.*, vol. 282, pp. 111–135, Oct. 2014.

[14] J. Zhao, Y. Zhou, X. Zhang, and L. Chen, "Part mutual information for quantifying direct associations in networks," *Proc. Nat. Acad. Sci. USA*, vol. 113, no. 18, pp. 5130–5135, May 2016.

[15] A. Dionisio, R. Menezes, and D. A. Mendes, "Mutual information: A measure of dependency for nonlinear time series," *Phys. A, Stat. Mech. Appl.*, vol. 344, nos. 1–2, pp. 326–329, Dec. 2004.

[16] D. D. Lewis, "Feature selection and feature extract ion for text categorization," in *Proc. Speech Natural Lang., Workshop Harriman*, New York, NY, USA, Feb. 1992, pp. 1–6.

[17] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[18] D. Lin and X. Tang, "Conditional infomax learning: An integrated framework for feature extraction and fusion," in *Proc. Eur. Conf. Comput. Vis.* Springer, May 2006, pp. 68–82.

[19] H. H. Yang and J. Moody, "Data visualization and feature selection: New algorithms for nongaussian data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 687–693.

[20] G. Brown, A. Pocock, M.-J. Zhao, and M. Luján, "Conditional likelihood maximisation: A unifying framework for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 27–66, Jan. 2012.

[21] M.-L. Zhang and L. Wu, "LIFT: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jul. 2014.

[22] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3309–3323, Dec. 2016.

[23] J. Ma, H. Zhang, and T. W. Chow, "Multilabel classification with label-specific features and classifiers: A coarse-and fine-tuned framework," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 1028–1042, Aug. 2019.

[24] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Mar. 2013.

[25] M. Bennasar, Y. Hicks, and R. Setchi, "Feature selection using joint mutual information maximisation," *Expert Syst. Appl.*, vol. 42, pp. 8520–8532, Sep. 2015.

[26] J. Wang, J. Wei, Z. Yang, and S. Wang, "Feature selection by maximizing independent classification information," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 4, pp. 828–841, Apr. 2017.

[27] W. Gao, L. Hu, and P. Zhang, "Class-specific mutual information variation for feature selection," *Pattern Recognit.*, vol. 79, pp. 328–339, Jul. 2018.

[28] A. Esuli and F. Sebastiani, "Sentiment quantification," *IEEE Intell. Syst.*, vol. 25, no. 4, pp. 72–75, Jan. 2010.

[29] L. Tang, H. Gao, and H. Liu, "Network quantification despite biased labels," in *Proc. 8th Workshop Mining Learn. Graphs (MLG)*, Jul. 2010, pp. 147–154.

[30] L. Sánchez, V. González, E. Alegre, and R. Alaiz, "Classification and quantification based on image analysis for sperm samples with uncertain damaged/intact cell proportions," in *Proc. Int. Conf. Image Anal. Recognit.* Springer, Jun. 2008, pp. 827–836.

[31] H. Kim, K. Lee, G. Hwang, and C. Suh, "Predicting vehicle collisions using data collected from video games," *Mach. Vis. Appl.*, vol. 32, no. 4, pp. 1–15, Jul. 2021.

[32] A. D. Pozzolo, G. Boracchi, O. Caelen, C. Alippi, and G. Bontempi, "Credit card fraud detection: A realistic modeling and a novel learning strategy," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 8, pp. 3784–3797, Sep. 2017.

[33] W. Gao and F. Sebastiani, "Tweet sentiment: From classification to quantification," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining*, Aug. 2015, pp. 97–104.

[34] N. Kwak and C.-H. Choi, "Input feature selection for classification problems," *IEEE Trans. Neural Netw.*, vol. 13, no. 1, pp. 143–159, Jan. 2002.

[35] M. Vidal-Naquet and S. Ullman, "Object recognition with informative features and linear classification," in *Proc. ICCV*, vol. 3, Oct. 2003, p. 281.

[36] G. Gulgezen, Z. Cataltepe, and L. Yu, "Stable and accurate feature selection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Springer, Sep. 2009, pp. 455–468.

[37] M. Tesmer and P. A. Estevez, "AMIFS: Adaptive feature selection by using mutual information," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jul. 2004, pp. 303–308.

[38] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, pp. 1531–1555, Nov. 2004.

[39] A. Jakulin, "Machine learning based on attribute interactions," Ph.D. dissertation, Dept. Fac. Comput. Inf. Sci., Univerza V Ljubljani, 2005.

[40] B. Guo and M. S. Nixon, "Gait feature subset selection by mutual information," *IEEE Trans. Syst., Man, Cybern., A, Syst. Humans*, vol. 39, no. 1, pp. 36–46, Dec. 2008.

[41] P. E. Meyer and G. Bontempi, "On the use of variable complementarity for feature selection in cancer classification," in *Proc. Workshops Appl. Evol. Comput.* Springer, Apr. 2006, pp. 91–102.

[42] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature Extraction: Foundation Application*, vol. 207. Springer, 2008.

[43] A. El Akadi, A. El Ouardighi, and D. Aboutajdine, "A powerful feature selection approach based on mutual information," *Int. J. Comput. Sci. Netw. Secur.*, vol. 8, no. 4, p. 116, 2008.

[44] G. Cheng, Z. Qin, C. Feng, Y. Wang, and F. Li, "Conditional mutual information-based feature selection analyzing for synergy and redundancy," *ETRI J.*, vol. 33, no. 2, pp. 210–218, 2011.

[45] Z. Zeng, H. Zhang, R. Zhang, and C. Yin, "A novel feature selection method considering feature interaction," *Pattern Recognit.*, vol. 48, no. 8, pp. 2656–2666, 2015.

[46] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, Dept. Comput. Sci., Univ. Waikat, Hamilton, New Zealand, 1999.

[47] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, Aug. 2011.

[48] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Proc. Eur. Conf. Mach. Learn.* Springer, Apr. 1994, pp. 171–182.

[49] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, Jan. 2001.

[50] G. Wei, J. Zhao, Y. Feng, A. He, and J. Yu, "A novel hybrid feature selection method based on dynamic feature importance," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106337.

[51] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intell. Data Anal.*, vol. 6, no. 5, pp. 429–449, Jan. 2002.

[52] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "Experimental perspectives on learning from imbalanced data," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 935–942.

[53] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[54] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-smote: A new over-sampling method in imbalanced data sets learning," in *Proc. Int. Conf. Intell. Comput.* Springer, Aug. 2005, pp. 878–887.

[55] B. B. Pineda-Bautista, J. A. Carrasco-Ochoa, and J. F. Martınez-Trinidad, "General framework for class-specific feature selection," *Expert Syst. Appl.*, vol. 38, no. 8, pp. 10018–10024, 2011.

[56] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[57] A. Asuncion and D. Newman, "UCI machine learning repository," Tech. Rep., 2007.

[58] N. X. Vinh, S. Zhou, J. Chan, and J. Bailey, "Can high-order dependencies improve mutual information based feature selection?" *Pattern Recognit.*, vol. 53, pp. 46–58, May 2016.

[59] A. Tillander, "Effect of data discretization on the classification accuracy in a high-dimensional framework," *Int. J. Intell. Syst.*, vol. 27, no. 4, pp. 355–374, 2012.

[60] J. Che, Y. Yang, L. Li, X. Bai, S. Zhang, and C. Deng, "Maximum relevance minimum common redundancy feature selection for nonlinear data," *Inf. Sci.*, vols. 409–410, pp. 68–86, Oct. 2017.

[61] X. Sun, Y. Liu, M. Xu, H. Chen, J. Han, and K. Wang, "Feature selection using dynamic weights for classification," *Knowl. Based Syst.*, vol. 37, pp. 541–549, Jan. 2013.

[62] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowl. Based Syst.*, vol. 86, pp. 33–45, Sep. 2015.

[63] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "Using data complexity measures for thresholding in feature selection rankers," in *Proc. Conf. Spanish Assoc. Artif. Intell.* Springer, Sep. 2016, pp. 121–131.
[64] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, "On developing an automatic threshold applied to feature selection ensembles," *Inf. Fusion*, vol. 45, pp. 227–245, Jan. 2019.
[65] M. Basu and T. K. Ho, *Data Complexity in Pattern Recognition*. Springer, 2006.
[66] L. Morán-Fernández, V. Bolón-Canedo, and A. Alonso-Betanzos, "A time efficient approach for distributed feature selection partitioning by features," in *Proc. Conf. Spanish Assoc. Artif. Intell.* Springer, Nov. 2015, pp. 245–254.
[67] C. M. Biship, *Pattern Recognition and Machine Learning* (Information Science and Statistics), 2007.
[68] M. N. Adnan, R. H. Ip, M. Bewong, and M. Z. Islam, "BDF: A new decision forest algorithm," *Inf. Sci.*, vol. 569, pp. 687–705, Aug. 2021.
[69] C. Peng and Q. Cheng, "Discriminative ridge machine: A classifier for high-dimensional data or imbalanced data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2595–2609, Jul. 2020.
[70] S. Liu, D. C. Mocanu, A. R. R. Matavalam, Y. Pei, and M. Pechenizkiy, "Sparse evolutionary deep learning with over one million artificial neurons on commodity hardware," *Neural Comput. Appl.*, vol. 33, no. 7, pp. 2589–2604, Apr. 2021.
[71] G. Taşkın, H. Kaya, and L. Bruzzone, "Feature selection based on high dimensional model representation for hyperspectral images," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2918–2928, Jun. 2017.

Deepak Kumar Rakesh received the B.Tech. degree in information science and engineering from RNSIT, Bengaluru, in 2013, and the master's degree (M.Tech.) in computer science and engineering from NIT Patna in 2015. He is currently pursuing the Ph.D. degree in CSE with the Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad. His research interests include machine learning, NLP, and ontology, with a current focus on feature selection algorithms for supervised classification problems.

Prasanta K. Jana (Senior Member, IEEE) received the M.Tech. degree in computer science from the University of Calcutta in 1988 and the Ph.D. degree from Jadavpur University in 2000. Currently, he is a Professor with the Department of Computer Science and Engineering, Indian Institute of Technology (ISM) Dhanbad, India. He has contributed 206 research publications in his credit, coauthored six books, and four book chapters. He has also produced 16 Ph.D. degree students. His current research interests include wireless sensor networks, cloud computing, fog computing, and machine learning. As a recognition of his outstanding research contributions, he has been awarded Senior Member of IEEE in 2009. He was a recipient of the Canara Bank Research Publication Award in 2015 and 2017. He is also Among the World ranking of top 2% (all fields) Indian scientists in artificial intelligence surveyed by Stanford University, USA, for the years 2020 and 2021.