Statistics and Applications {ISSN 2454-7395 (online)}
Volume 24, No. 1, 2026 (New Series), pp 155–176
http://www.ssca.org.in/journal

Semi-supervised Feature Selection using Maximum Mutual Information and Minimum Correlation through Augmented Learning

Arghya Kusum Das[1], Saptarsi Goswami[2], Amlan Chakrabarti[3] and Basabi Chakraborty[4]

[1]*Department of Computer Science and Engineering, Techno International New Town, Kolkata 700156, India*
[2]*Department of Computer Science, Bangabasi Morning College, Kolkata, 700012, India*
[3]*A.K. Choudhury School of Information Technology, University of Calcutta, Kolkata 700009, India*
[4]*Faculty of Software and Information Science, Prefectural University, Takizawa City, Iwate, Japan*
[4]*School of Computer Science, Madanapalle Institute of Technology and Science, AP, India*

Received: 18 February 2025; Revised: 26 July 2025; Accepted: 29 July 2025

Abstract

Feature selection is a critical pre-processing step in machine learning. For supervised problems, class labels are used to identify important features. However, labeling or annotating the data is labor-intensive and hence costly. Consequently, there is an abundance of unlabeled data and limited labeled data. Therefore, semi-supervised learning is very pertinent in this case. The problem of feature selection is equally relevant to semi-supervised learning. In this research work, a novel semi-supervised method of feature selection, **R**epeated **S**ampled **S**emi-supervised **F**eature **S**election(RSSFS), is proposed with a time complexity of O(nd), which is significantly better than other existing algorithms. This follows an unbiased and neutral strategy in producing a final feature set that has less redundancy and frequently occurring feature components, thereby inducing better stability in the feature set. In the first step, a decision tree classifier is used for labeling the unlabeled portion of the data and augmenting the training set. Repeated sampling is done from the pseudo-labeled portion, to generate multiple augmented training sets. The top k features are selected based on mutual information from each augmented training set. While choosing the features, it is ensured that the features are not redundant by applying a correlation coefficient based threshold. A voting-based approach is used to combine these multiple features into the final feature set. The proposed algorithm is compared with a) a benchmark, using the full feature set, and b) top k % supervised feature selection on the labeled portion. Comparing these three methods across 18 datasets, it was found that RSSFS outperformed the supervised method based on F1 scores by 2.79% and the benchmark by 0.36%. Thus, the proposed algorithm would prove impactful in applications where there is a plethora of unlabeled data compared to labeled data.

Corresponding Author: Arghya Kusum Das
Email: arghya.das@gmail.com

Key words: Semi-supervised learning; Feature selection; Stability; Mutual information; Correlation.

AMS Subject Classifications: 68T01 The video recording of the paper made under the SSCA's Online Lecture series is available at the Youtube channel URL `https://youtu.be/0WQ_FuYt55E`.

1. Introduction

The curse of dimensionality [*c.f.* Bellman and Kalaba (1959)] has been a very important problem in data mining/machine learning applications as it requires high processing power and storage capacity which causes the model to overfit and increases the error rate of the learning algorithm. A Feature X_i is said to be an irrelevant feature if the output Y is not conditionally dependent on X_i [*c.f.* John *et al.* (1994)]. The features that are highly associated with each other are called redundant features [*c.f.* Kumar and Minz (2014)]. In high-dimensional datasets, feature reduction or selection is used as a dimensionality reduction method and is employed in machine learning and data mining. Here, a subset of the features is selected for the learning algorithm that leaves out the irrelevant and redundant features.

Feature selection can also be categorized into three types based on the presence of class label information: supervised feature selection, unsupervised feature selection and semi-supervised feature selection. In supervised feature selection,feature importance is calculated based on the degree of association between the class and the feature [*c.f.* Benabdeslem and Hindawi (2011)], [*c.f.* Song *et al.* (2016)] and [*c.f.*Chang *et al.* (2014)]. However, one challenge with this approach is the costly and potentially unreliable process of labeling the data using external knowledge [*c.f.* Kalakech *et al.* (2011)].

Traditionally, the performance of supervised approaches has been compared to unsupervised approaches [*c.f.* Yarowsky (1995)]. With the abundance of unlabeled data and the cost associated with labeling, the importance of semi-supervised learning has increased. Though there has been a lot of theoretical work in this domain [*c.f.* Zhao and Liu (2007)], the exploration of semi-supervised feature selection has been relatively limited. However, the introduction of semi-supervised feature selection has expanded the range of applications [*c.f.* Rosenberg *et al.* (2005)], [*c.f.* Dópido *et al.* (2013)], [*c.f.* Wu *et al.* (2012)] and [*c.f.* Kok *et al.* (2021)]. Contemporary research work focused on this domain covers a wide range of applications, from detecting liver diseases [*c.f.* Tran *et al.* (2019)] to semiconductor materials and manufacturing [*c.f.* Liu *et al.* (2022)]. Typically, in semi-supervised learning setups, a classifier trained on the labeled data is used to assign pseudo-labels to the unlabeled portion of the dataset. A dilemma exists regarding whether to fully utilize the pseudo-labeled data, as this may introduce bias if the classifier assigns incorrect labels.

In this paper, a simple semi-supervised feature selection method is proposed. In different iterations, various samples of the pseudo-labeled data are selected, creating multiple augmented training sets. On each of these augmented training sets, a simple mutual information based feature selection is applied, with a threshold for feature redundancy. Hence, in n iterations, n such feature sets are retrieved. However, the selection of the final feature set from the n feature sets each having k features is determined using a voting mechanism based

on the most repetitive and non-redundant features. The feature set produced in this manner outperforms all features set and also the partially labeled dataset after applying supervised selection in terms of both F1-Score and feature stability.

The contributions of the paper are as follows:

i. A novel and simple sampling-based feature selection technique for semi-supervised learning is proposed.

ii. An intuitive voting mechanism is employed to derive the final feature subset by aggregating multiple intermediate subsets, ensuring the selection of the most relevant and non-redundant features.

iii. With only 20% of the features, the proposed algorithm outperforms the supervised feature selection by 2.79% and the benchmark by 0.36% based on the F1 Score evaluated across 18 benchmark datasets.

iv. The proposed algorithm exhibits a time complexity of O(nd), which is significantly lower than that of existing semi-supervised models.

v. As the secondary focus of the proposed algorithm was on stability by similarity of the feature sets, the feature set of the proposed algorithm achieved a mean similarity of 39.4% with benchmark, and 42.64% with the supervised model feature sets.

The remainder of this work is organized into the following sections. In the literature survey *i.e.* Section 2, the related work in semi-supervised learning is discussed. The preliminaries Section provides an overview of fundamental concepts related to feature selection and semi-supervised learning. In Section 4, the proposed methodology is explained through a flow diagram followed by the algorithm used. The simulation experiment Section details the experimental setup and environmental considerations. In the result and discussion Section, the experimental results are presented, along with an analysis. The proposed work is summarized in the conclusions Section, which also outlines future scope.

2. Literature survey

Semi-supervised feature selection methods can be classified based on different perspectives. The viewpoint discussed here is based on the fundamental nomenclature of feature selection methods, which categorizes semi-supervised feature selection methods into several categories depending on how they interact with the learning algorithm. The first way of classification is similar to regular feature selection categories, namely a) filter b) wrapper c) hybrid d) embedded. More specific categorizations from the perspective of semi-supervised feature selection are i) Co-training ii) Self-Training iii) Entropy Based methods *etc.*

Co-Training: This is a method of semi-supervised learning. In the first step, the entire feature set is divided into two subsets, let us call them f_1 and f_2. They are mutually exclusive and exhaustive. Hence $f_1 \cup f_2 = f$, and, $f_1 \cap f_2 = \phi$. Next, two classifiers C1 and C2 are trained on the f1 and f2 view of the labeled data. Now the algorithm proceeds iteratively. In each iteration, an unlabeled portion of the data, on which C1 is most confident, is added to the training set of C2 and vice versa. These training sets are often called augmented training sets, let us denote them by T_1 and T_2. A simpler way to extend this to feature selection is to select top k features separately from T_1 and T_2 based on measures like feature

selection and take a union from them.

Self-Training: Similar to Co-Training, Self-Training is also a kind of semi-supervised learning. Initially, a particular classifier is selected. Then iteratively predicted points from the unlabeled set are added to the training set based on confidence of the prediction. Similar to co-training, such a training set is called the augmented training set. Feature selection now can be done in the usual manner from the augmented training set. The major developments in semi-supervised learning are summarized in Table 1 shown next.

In this section, the different models of semi-supervised learning that have evolved are summarized, along with their significance or drawbacks, discussed in Table 1. This study has helped to draw the motivation for this work, as discussed next.

The time complexity of existing methods like Laplacian Score(semi-supervised) [*c.f.* He *et al.* (2005)] is O(n^2d), while Semi-Supervised Feature Selection via Spectral Analysis(SSFS) [*c.f.* Cai *et al.* (2007)] is O(n^2d+d^3) and Semi-supervised Feature Selection using Max-margin Criterion (SSMMC) [*c.f.* Wu *et al.* (2013)] is O(nd^2+d^3) where n is the number of samples and d is the number of features. Hence, improving the time complexity was the foremost motivation and hence the algorithm was proposed with a complexity of O(nd).

This work was inspired by the self-training and co-training methods of wrapper-based approaches. This research is different in the sense that it does not only add the most confident points based on a confidence measure of prediction but, through sampling, adds a portion of the predicted samples to the partially labeled dataset.This is done to ensure that there is no bias against the confident points. It operates by taking the predicted labels, which, in turn, depend on the classifier's performance. With this augmented training set, feature selection is implemented and compared with the supervised mode, which is discussed in the results and discussion section to justify this approach. The stability of the selected feature subset is also studied to assess the quality of the feature selection process adopted.

3. Preliminaries

In this Section, the factors/criteria for the selection of features (mutual information and correlation coefficient), the evaluation method of feature sets based on stability, and the confidence of prediction using confidence score are discussed.

3.1. Mutual information

Mutual Information(MI) of a feature is the degree of dependence between the feature and the class variable. Mutual information of a feature, denoted as F, concerning the class variable C, is the difference between the entropy of the class $H(C)$ and the conditional entropy of the class variable given the value of the feature variable $H(C|F)$. The following equations are used to calculate the feature-to-class relationship:

$$I(F;C) = H(C) - H(C|F) \tag{1}$$

$$I(F;C) = H(C) + H(F) - H(C.F) \tag{2}$$

Table 1: Summary of semi-supervised methods based on the taxonomy of feature selection methods

Related Works		
Methodology	**Method**	**Conclusion/Remarks**
Graph theory and cluster assumption [*c.f.* Zhao and Liu (2007)] Type-Filter.	• It constructs a neighborhood graph and transforms each feature vector into a cluster indicator, which is evaluated based on separability and consistency. • Laplacian score is associated with normalized mutual information.	• The method overlooks the correlation between features and evaluates the features one by one.
Laplacian score [*c.f.* Cheng *et al.* (2011)] and [*c.f.* Zhao *et al.* (2008)] Type-Filter.	• It constructs a within-class graph and a between-class graph. • The features are estimated through their degree of preserving the graph structures.	• The method disregards the correlation among features and evaluates the features one by one.
Constraint score [*c.f.* Kalakech *et al.* (2011)] and [*c.f.* Benabdeslem and Hindawi (2011)] Type-Filter.	• The metric uses some supervision information in the form of pairwise constraints. • It constructs two graphs using pairwise constraints and unlabeled data. • Finally, it evaluates the features based on their locality and constraint-preserving ability.	• It depends on the subsets of pairwise constraints created by the user. • The constraints can be redundant or incoherent. The method evaluates the features individually, ignoring the correlation among features.
Fisher criterion [*c.f.* Chen *et al.* (2010)], [*c.f.* Lv *et al.* (2013)], [*c.f.* Liu *et al.* (2013)] and [*c.f.* Liu *et al.* (2010)] Type-Filter.	• This metric utilizes the Fisher criterion and considers the local structure of both labeled and unlabeled data. • The features are estimated based on their discriminant and locality-preserving abilities.	• The method disregards the correlation among features and evaluates the features one by one.

Methodology	Method	Conclusion/Remarks
Sparse-based filter methods [*c.f.* Han *et al.* (2014)] Type-Filter.	• It combines two supervised and unsupervised scatter matrices. • It preserves the discriminant information from labeled data and the local geometric structure from both labeled and unlabeled data. • It adds a l2 norm to the objective function, making it suitable for feature selection. • The method utilizes an iterative algorithm to solve the objective function.	• The features are estimated jointly while considering the correlation among the features. The objective function is non-smooth and difficult to solve.
Single Learner [*c.f.* Ren *et al.* (2008)] Type-Wrapper.	• The initial labeled training set is augmented with predicted unlabeled data. • Data is randomly selected from unlabeled set to create new training sets. • Next, it adds the most frequently selected feature to the feature subset during each iteration	• The method ignores confidence measures for unlabeled data. • On adding mislabeled data, it may degrade performance. • The method neglects the discriminative power of feature combinations. • It takes high computational time.
Ensemble learning [*c.f.* Bellal *et al.* (2012)], [*c.f.* Han *et al.* (2011)] and [*c.f.* Barkia *et al.* (2011)] Type-Wrapper.	• The method uses ensemble learning with self-training or co-training to predict the labels of unlabeled data.	• It considers the reliance on features. • It uses a confidence measure to select predicted unlabeled data. • But if the confidence measures are inaccurate, this may lead to mislabeling of data.

Methodology	Method	Conclusion/Remarks
		• It enhances generalization ability using an ensemble classifier. • It takes high computational time.
Sparse-based Embedded methods [*c.f.* Song *et al.* (2016)], [*c.f.* Ma *et al.* (2012)], [*c.f.* Shi *et al.* (2014)] and [*c.f.* Ma *et al.* (2011)].	• The method directly learns the classifiers during feature selection. • It constructs a graph using both labeled and unlabeled data.	• This combines the strengths of joint feature selection and semi-supervised learning.
SVM-based [*c.f.* Yang and Wang (2007)], [*c.f.* Xu *et al.* (2010)] and [*c.f.* Dai *et al.* (2013)] Type-Embedded.	• Select the features by maximizing the margin between different classes and at the same time exploiting the local structure of both labeled and unlabeled data.	• The objective function is difficult to solve.

3.2. Correlation coefficient

Correlation coefficient is a statistical measure of the strength of the relationship between two variables. The correlation coefficient, denoted as (ρ), between two variables x and y is defined as follows:

$$\rho(x,y) = \frac{cov(x,y)}{\sqrt{var(x)var(y)}} \tag{3}$$

where $cov(x,y)$ refers to the covariance between x and y , and $var(x)$ is the variance of x.

$$cov(x,y) = \sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y}) \tag{4}$$

where $\bar{x}$ and $\bar{y}$ are the means of x and y respectively.

$$var(x) = \sum_{i=1}^{n}(x_i - \bar{x})^2 \tag{5}$$

3.3. Stability of feature sets

The stability of a feature selection algorithm results in a persistent feature subset when new training samples are added or removed [*c.f.* Xin *et al.* (2015)]. Stability can be categorized into three groups: stability by rank, stability by weight, and stability by

similarity [*c.f.* Khaire and Dhanalakshmi (2022)] and [*c.f.* Chelvan and Perumal (2016)]. In this work, we consider stability by similarity between feature sets. In this case, similarity is estimated by using the ratio of the intersection to the union of two selected feature subsets or the amount of overlap between the overall subset of selected features [*c.f.* Khaire and Dhanalakshmi (2022)] and [*c.f.* Yu *et al.* (2011)].

Let X and Y be two different feature sets such that

$X = \{x_1, x_2, x_3, x_4, x_5\}$ and $Y = \{y_1, y_2, y_3, y_4, y_5\}$ where $x_1 \ldots\ldots x_5$ and $y_1 \ldots\ldots y_5$ are the individual feature components such that $x_1 = y_2$ and $x_3 = y_5$

$$Similarity = \frac{|X \cap Y|}{|X \cup Y|} \tag{6}$$

$|X \cup Y| = 8, |X \cap Y| = 2, Similarity = \frac{2}{8} = 0.25$

3.4. Confidence score

The confidence score is used in predictions made by models. For binary classification or a yes-no answer, it predicts 0/1 based on a score known as the confidence score. A confidence score is a number between 0 and 1, representing the likelihood that the output of a machine learning model is correct and will satisfy a user's request. One way to interpret it, is by considering a 'yes' or '1' if the value is > 0.5, where 0.5 serves as the minimum confidence score or threshold. Increasing the threshold will result in lower recall and improved precision.

4. Proposed methodology

The working of the proposed method is explained in Section 4.1 followed by the algorithm in Section 4.2 and finally the complexity analysis in Section 4.3.

4.1. Working of the algorithm

The step-by-step working of the algorithm is mentioned below:
Step 1 (Divide Dataset): To simulate the semi-supervised setup, a part of the dataset is unlabeled and the labeled portion is divided into train and test as shown in Figure 1.

Figure 1: Splitting the dataset

Step 2 (Train using classifier and add label to unlabeled): The selected classifier is trained on the training dataset and the unlabeled portion is now labeled using the classifier as shown in Figure 2.

Step 3 (Construct different sample sets from the pseudo label and form augmented training set) : This sample of pseudo labeled data is added to the training

Figure 2: Labeling the unlabeled to pseudo label

set each time to form an augmented training set as shown in Figure 3.

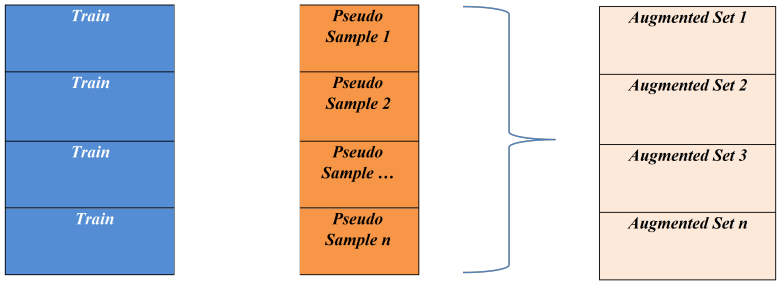


Figure 3: Construction of Augmented set

Step 4 (Select *K* features from each augmented set based on MI and Redundancy Level): Using the set, feature selection is performed based on MI and redundancy threshold to form n feature sets into the Master Feature Set(MFS) as shown in Figure 4.

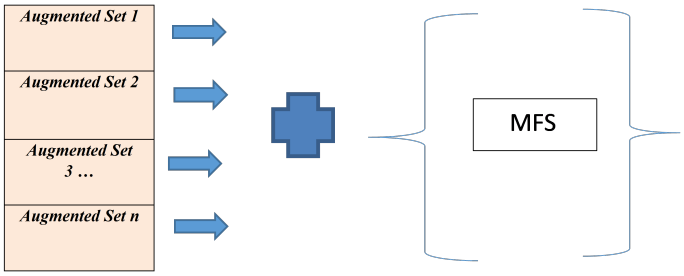


Figure 4: Construction of Master Feature Set

Step 5 (Select *K* features into FFS based on frequency count and Redundancy Level): Once these n feature sets are obtained in MFS, voting has been done to select the final k features into Final Feature Set(FFS) as shown in Figure 5. Here, the redundant features are eliminated.

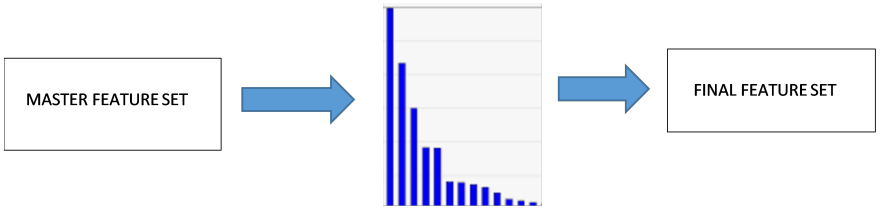


Figure 5: Retrieval of the Final Feature Set

4.2. Algorithm RSSFS (Repeated Sampled Semi-Supervised Feature Selection)

Input (D_L: Labeled Dataset,D_U: Unlabeled Dataset, C: Classifier, α: Redundancy Threshold, I: No of Iterations, p:Randomness, K: No of Features (can be an absolute value or a percentage))
Output: *FFS* // Final Feature set
Initialization :
$MFS \rightarrow \phi$,$FFS \rightarrow \phi$// MFS and FFS are set to empty set
Step 1: Divide D_L into Train + Test
Step 2: Train C on Train and predict the labels for D_U and Assign to *D_Pseudo*.
Step 3:Construct *D_Augment*
For iteration (i) 1 to I
Step 3a: p % of observations selected from *D_Pseudo* and Assigned to $D\_PseudoSample_i$
Step 3b: $D\_PseudoSample_i$ are added to Train to form $D\_Augment\_i$
Step4: Select K features from $D\_Augment\_i$ to form Master Feature Set(*MFS*)
Step 4a: Set K to 0
For iteration (i)= 1 to I
Step 4b: The First feature is added to MFS_i based on maximum mutual information with the target variable
Step 4c: The next feature is added which has the next highest mutual information and does not cross the redundancy Threshold α, with any of the selected features in MFS_i.
Step 4d: Increment K, Continue up to K features
Step 5: Compose the final feature set *FFS* from the *MFS*
Step 5a: Out of the $I * k$ features in *MFS*, count the frequency of the features
Step 5b: The first feature is added to *FFS*, based on the highest frequency
Step 5c: Set K =1
Step 5d: The next feature is added which has the next highest frequency count and does not cross the redundancy Threshold α, with any of the selected features in *FFS*
Step 5e: Increment K, Continue up to K features
Return *FFS*

4.3. Case study

Let us assume that we have a dataset with thirty features, denoted as f_1 to f_{30}, for binary classification. T represents the redundancy threshold, and our goal is to select the top 20 % of non-redundant augmented features, which starts as an empty set. So, by the end of step five, after six iterations, the features selected by RSSFS, *i.e. ssl_set*, and the supervised set, *i.e. sl_set*, are mentioned using their column indices which are as follows:

ssl_set = [[1,3, 7, 11, 13,21], [1,3, 10, 12, 20,22], [1,3, 9, 11, 23,25], [1,3, 7, 19, 17,30], [1,3, 7, 11, 20,29], [1,5, 7, 11, 23,30]]
sl_set = [[3.7,11,17,21,28], [3.7,11,17,21,28], [3.7,11,17,23,19], [3.7,11,18,25,29], [3.7,9,16,27,22], [3.6,10,12,24,26]]

The most frequent features from *ssl_set* and *sl_set* are enlisted in the *freq_ssl_set* and *freq_sl_set* respectively as shown next.

freq_ssl_set = [1,3,7,11,23, 30]
freq_sl_set = [3.7,11,17,21,28]

Find the similarity between the frequent feature sets of semi-supervised and supervised.

Similar features = [3,7,11]
$|Similar features| = 3$
Total features = [1,3,7,11,17,21,23,28,30]
$|Total features| = 9$

$$Similarity = \frac{|Similar features|}{|Total features|} = 3/9 = 33.33\% \tag{7}$$

4.4. Computational complexity

With n being the number of samples and d being the number of attributes, time complexity of feature selection is O(nd). Finding the frequent feature set from the feature sets has a complexity of O(nd). The complexity of evaluating the precision and recall of the frequent feature sets is O(n). Hence, the overall complexity is of the quadratic order as evident in equation 9 shown next.

$$T(n) = O(nd) + O(nd) + O(n) \tag{8}$$

$$T(n) = O(nd) \tag{9}$$

5. Simulation experiment

In this Section, the working environment is described in four subsections which comprises of the dataset description, the experimental setup, the parameter sensitivity analysis and the performance metrics used.

5.1. Dataset description

The datasets used are available in public online repositories of the University of California, Irvine [*c.f.* Lichman *et al.* (2013)]. The proposed algorithm is compared with the benchmark having all the features and the supervised feature selection having the top k% features. These three methods were conducted on eighteen datasets. Among these, six datasets have binary classes, while twelve datasets have multi-class labels, with the isolet dataset having a maximum of twenty six classes. The dataset with the highest number of features used is six hundred and seventeen in the isolet dataset. The dataset with the maximum number of patterns used is 42,158 in the Dry Bean dataset. Details of the datasets are provided in Table 2.

While working, it has been observed that some of the datasets were similar to each other in terms of input size, where the input size of a dataset is the product of the number of records and the number of features/attributes in the dataset. These datasets are then categorized into the same group. Ultimately, this categorization resulted in the creation of three distinct categories based on input size. The first category *i.e.* Group A has an input size up to 50000, whereas the input size for the second group, B, is from 50001-300000, and for the third group C, it ranges from 300001-1000000. The main reason behind this approach is to study and analyze the performance of the three methods across these three different categories having varying input size.

Table 2: Details of the datasets used

Sl No.	Dataset	Records	Features	InputSize	Category
1	lung_cancer	32	57	1824	Group-A/1 (0-50000)
2	wine	178	13	2314	
3	lymphography	147	19	2793	
4	cleave	297	13	3861	
5	sonar	208	60	12480	
6	Vehicle	846	18	15228	
7	wbdc	569	30	17070	
8	ctg	2126	34	72284	Group-B/2 (50001- 300000)
9	Colon	62	2000	124000	
10	Arythmia	452	279	126108	
11	mfeat-karhunen	2000	64	128000	
12	texture	5500	40	220000	
13	Spambase	4601	57	262257	
14	digits	1593	256	407808	Group-C/3 (300001-1000000)
15	Dry Bean	42158	16	674528	
16	ECG	4998	140	699720	
17	isolet	1559	617	961903	
18	Madelon	2000	500	1000000	

5.2. Experimental setup

i. The processor used is a Core i5, with 8GB of memory.

ii. The operating system used in this context was Windows 10, 64-bit, and the computing environment Python 3.6 was used for the experiments. Various Python libraries were also utilized.

iii. The classifier used is the decision tree classifier.

iv. This experiment was conducted having different proportions of unlabeled, training, and test data. Initially, the dataset was divided into a ratio of 50% unlabeled, 30% training, and 20% test data. The dataset was trained using a random 30% of the samples. This knowledge was then used to produce pseudo-labels for the remaining 50% of unlabeled data.

v. A separate test was performed where the proposed algorithm was compared against Supervised Feature Selection (SFS) which was retrieved from the labeled samples only having an equal number of features compared to RSSFS, and the Benchmark having all the features.

vi. While performing feature selection for all the three methods the top k% non-redundant features were selected. The value of k is chosen to be twenty in this case, as ten would be very insignificant and thirty would be much more significant.

vii. The final feature set for RSSFS and SFS is retrieved by selecting the most frequent features from n feature sets retrieved in n iterations with different seed values. The value of n is chosen to be one hundred in this case, as it would provide an average of the results on different random samples.

viii. While selecting features it is ensured that the correlation between the features in the feature subset is less than 0.67.

ix. Additionally, the goal was to assess the category-wise performance, which is the average accuracy of these three methods that is calculated for all the datasets within a particular category.

x. For each dataset, feature selection for RSSFS is performed using the mixed training set, which combines original and pseudo labels, to select the top $k\%$ of original non-redundant features. From this feature set, metrics are calculated for the semi-supervised set. Similarly, using the original training set, feature selection for SFS is conducted to select the top $k\%$ of original non-redundant features, and classification metrics are calculated for the supervised set. This process is repeated for one hundred iterations. Next, from the two feature sets, each composed of one fundred feature sets, each of size k, the frequency of each feature in both sets is determined. Based on the frequency of the features, two feature sets are finally composed, each containing the k most frequent features.

5.3. Parameter sensitivity analysis

In the adopted strategy RSSFS, the parameters chosen are redundancy threshold α, number of iterations I and randomness parameter p, and fine-tuning these parameters may affect the proposed algorithm to a certain extent as discussed next.

i. **Redundancy threshold** α: The redundancy threshold used is 0.67. This is because if the value is increased to 0.75 or 0.72, sometimes in an iteration a fixed number of k features cannot be retrieved, as it goes down to less than k features say *k-1* or *k-2* features which results in a different number of feature components in different iterations. Hence the strategy is sensitive to the redundancy threshold.

ii. **Number of iterations** $\boldsymbol{I}$: The number of iterations used uniformly for all datasets was one hundred. However, for certain datasets, it was tested with one hundred and twenty but the metrics calculated such as precision, recall and F1-score did not vary much. Hence, the method is not sensitive to a much higher value of I.

iii. **Randomness parameter** $\boldsymbol{p}$: The randomness parameter p used here is 60% which suggests that for a distribution of one hundred samples with 50% unlabeled samples only 60% *i.e.* thirty pseudo samples are added to the earlier training set. If the randomness is further increased, then there is a chance of picking the majority of the pseudo samples. Also if the randomness is decreased then again there is a chance of picking less number of augmented samples into the training set. In both ways, the earlier training set augmented may be affected, where either the pseudo samples may dominate, or, the training set from label data may have more preference. Hence the method is sensitive to randomness parameter.

5.4. Performance metrics

In the case of RSSFS and SFS, on computing the mutual information for all original and extracted features, the top 20% of features were selected. To validate the proposed methodology, three perspectives or views were utilized, as described below:

i. The classification performance metrics used for comparing the performance of the semi-supervised feature set against the supervised feature set included precision, recall and F1 scores.

ii. For visualizing graphically, sensitivity and specificity were also considered from where the AUC(Area Under the Curve) was computed.

iii. To assess the stability of the feature sets, stability by similarity was calculated. This involved determining the percentage of common feature components in both the sets.

iv. As the proposed algorithm was compared against the benchmark and Supervised Feature Selection(SFS), a t-test was conducted to analyze the statistical significance.

6. Results and discussion

In this Section, the results of the research work conducted are provided, followed by a comprehensive discussion. The result analysis is mainly divided into four Subsections namely comparison test, t-test, analysis of similarity of feature sets and AUC analysis.

6.1. Comparison test: performance of RSSFS against Benchmark and SFS

In each iteration, the performance is evaluated thrice, first using the entire feature set that sets the benchmark, secondly, using a semi-supervised training set *i.e.* the proposed algorithm (RSSFS), and third using a portion of the train set with labels which is the supervised mode of feature selection. After one hundred iterations, the average precision, recall and F1 Score are recorded. Here, the proposed algorithm with 20% features is compared against two methods, firstly the benchmark having 100 % features, and secondly, the Supervised Feature Selection (SFS) with 20% features.

The results of the performance of these methods are shown next in Table 3 for all the 18 datasets used individually. Also, the average scores of the performance metrics used *i.e.* precision, recall and F1 Score for each of the three categories A, B and C have been computed to draw a conclusion. From Table 3, the observations are mentioned as follows :

- In Group A, the proposed algorithm outperforms supervised feature selection by 1.68%, 1.58%, and 1.65% in precision, recall, and F1 Score, respectively. Also, it lags behind the benchmark by a marginal amount *i.e.* 0.44%, 0.54% and 0.34% in precision, recall and F1 Score respectively.
- In Group B, the proposed algorithm outperforms supervised feature selection by 5.18%, 4.38%, and 4.85% in precision, recall, and F1 Score, respectively. Also, it exceeds the benchmark by 1.7%, 2.15% and 1.92% in precision, recall and F1 Score respectively.
- In Group C, the proposed algorithm outperforms supervised feature selection by 1.52% in precision, 2.26% in recall, and 1.89% in F1 Score. Also, it lags behind the benchmark by 0.85%,0.17% and 0.51% in precision, recall and F1 Score respectively.
- In each of the three categories, the proposed algorithm has outperformed the supervised feature selection, and is at par with the benchmark.

Table 3: Comparison of the performance of the proposed algorithm against Benchmark and Supervised Feature Selection

		Benchmark			RSSFS			Supervised Feature Selection		
Sl No.	Dataset	Avg.Pr.	Avg.Re.	F1	Avg.Pr.	Avg.Re.	F1	Avg.Pr.	Avg.R.	F1
1	lung_cancer	56.42	38.57	45.82	65.07	52.85	58.33	64.81	51.42	57.35
2	wine	97.56	97.22	97.39	90.67	89.58	90.13	90.43	88.19	89.3
3	lymphography	72.66	74.85	73.74	77.27	74	75.6	75.41	72.66	74.01
4	cleave	57.21	59.5	58.34	52.94	55.16	54.03	50.93	54.66	52.73
5	Sonar	79.48	78.57	79.03	72.19	72.02	72.11	71.9	71.42	71.66
6	Vehicle	68.08	68.52	68.3	69.94	69.67	69.81	65.32	66.29	65.81
7	wbdc	94.35	94.3	94.33	94.58	94.52	94.55	92.15	92.1	92.13
Group-A Average		**75.11**	**73.08**	**73.85**	**74.67**	**72.54**	**73.51**	**72.99**	**70.96**	**71.86**
8	ctg	91.03	90.61	90.82	90.08	90.07	90.08	86.7	86.9	86.8
9	Colon	77.12	74.61	75.85	77.11	75.38	76.24	71.35	68.46	69.88
10	Arythmia	71.13	72.97	72.04	68.33	71.78	70.02	50.26	59.88	54.65
11	mfeat-karhunen	92.56	92.37	92.47	91.89	91.62	91.76	88.67	88	88.34
12	texture	78.75	78.78	78.77	93.04	92.99	93.02	92.74	92.69	92.72
13	Spambase	92.54	92.51	92.53	92.91	92.9	92.91	92.53	92.5	92.52
Group-B Average		**83.86**	**83.64**	**83.75**	**85.56**	**85.79**	**85.67**	**80.38**	**81.41**	**80.82**
14	digits	86.31	86.01	86.16	86.8	87.28	87.04	81.92	81.27	81.6
15	Dry Bean	90.71	90.68	90.7	90.15	90.11	90.13	89.77	89.73	89.75
16	ECG	98.61	98.62	98.62	98.67	98.67	98.67	98.42	98.42	98.42
17	isolet	85.88	84.58	85.23	84.1	85.41	84.75	84.51	83.05	83.78
18	Madelon	69.77	69.65	69.71	67.32	67.22	67.27	64.82	64.91	64.87
Group-C Average		**86.26**	**85.91**	**86.08**	**85.41**	**85.74**	**85.57**	**83.89**	**83.48**	**83.68**

Table 4: Comparative summary of the performance of the proposed algorithm against Benchmark and Supervised Feature Selection

		Benchmark			RSSFS			Supervised Feature Selection		
Sl No.	Dataset	Avg.Pr.	Avg.Re.	F1	Avg.Pr.	Avg.Re.	F1	Avg.Pr.	Avg.R.	F1
1-7	Group-A	75.11	73.08	73.85	74.67	72.54	73.51	72.99	70.96	71.86
8-13	Group-B	83.86	83.64	83.75	85.56	85.79	85.67	80.38	81.41	80.82
14-18	Group-C	86.26	85.91	86.08	85.41	85.74	85.57	83.89	83.48	83.68
	Avg.	**81.74**	**80.87**	**81.22**	**81.88**	**81.35**	**81.58**	**79.08**	**78.61**	**78.78**

To further summarize the performance of the proposed algorithm against the Benchmark and Supervised Feature Selection(SFS) a study is done as shown next in Table 4.

Analyzing Table 4, the observations are mentioned as follows:

- Across eighteen datasets, the proposed algorithm, RSSFS outperforms supervised feature selection (SFS) by 2.79% in precision, 2.74% in recall, and 2.79% in F1 Score.
- Also, the proposed algorithm, RSSFS excels over the benchmark by 0.13%,0.48% and 0.36% in precision, recall, and F1 Score, respectively.

To compare the behavior of F1 Scores across different input sizes categorized into distinct classes, please refer to Figure 6. Analyzing Figure 6, it is evident that the proposed algorithm has dominated over the benchmark and SFS in Group-B and Group-C. For an alternative perspective, the digit dataset was used to capture multiple readings of F1-scores through a box plot as displayed in Figure 7 next.

On analyzing the box plot on the digit dataset as shown in Figure 7 above, the observations are mentioned as follows:

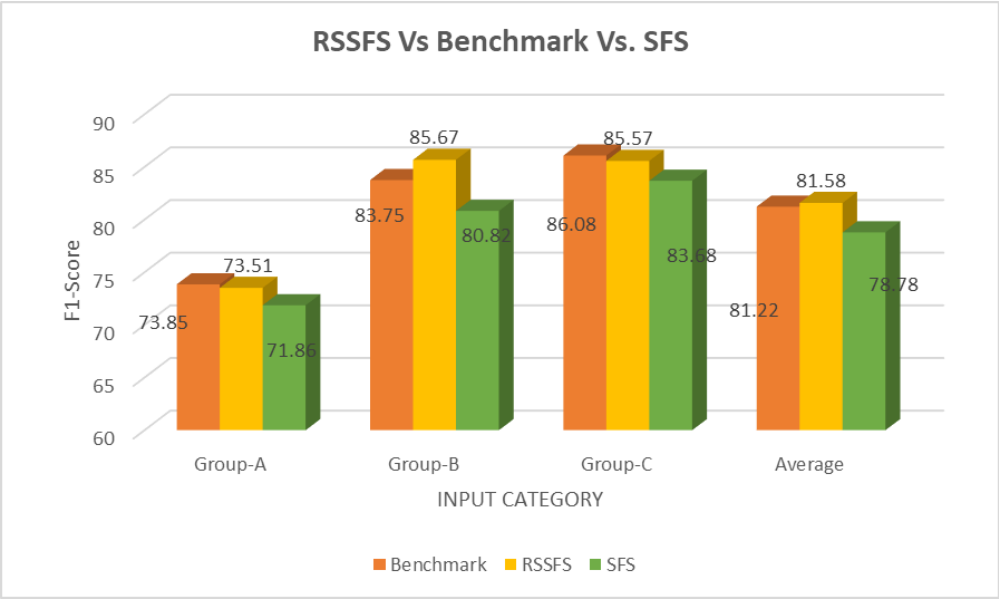


Figure 6: Comparison of performance of RSSFS against benchmark and SFS

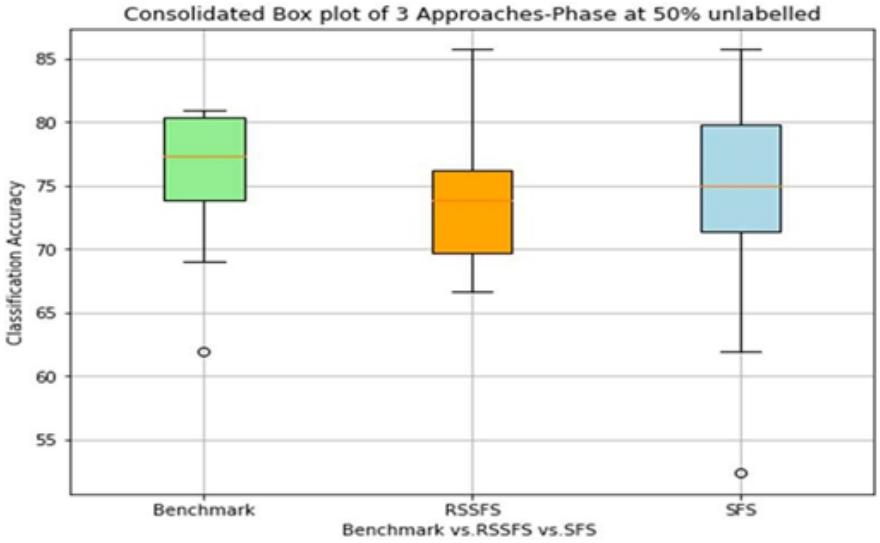


Figure 7: Box plot of F1 scores of the most frequent feature set for digit dataset

- The benchmark and SFS have performed better than the proposed algorithm concerning accuracy based on F1 Scores.
- The Interquartile Range(IQR) for the proposed algorithm is around 6%, whereas that of the benchmark and SFS is around 7% and 9% respectively, which suggests a lesser dispersion for the proposed algorithm.

6.2. Test of statistical significance through t-test

Since the number of samples was eighteen that is than thirty, a t-test was used instead of a z-test. The t-test was applied to compare the performance of our proposed algorithm against Supervised Feature Selection (SFS). The resulting p-value was found to be less than

Table 5: Comparison of similarity of frequent feature sets between proposed algorithm and SFS, and, proposed algorithm and benchmark

Sl No.	Dataset	Similarity between RSSFS and SFS (%)	Similarity between RSSFS and Benchmark (%)
1	lung_cancer	41.17	33.33
2	wine	50	50
3	lymphography	33.33	60
4	cleave	50	50
5	Sonar	33.33	20
6	Vehicle	100	60
7	wbdc	33.33	20
	Group-A Average	**48.74**	**41.9**
8	ctg	55.55	75
9	Colon	66.07	43.9
10	Arythmia	45.45	12
11	mfeat-karhunen	85.71	85.67
12	texture	60	30
13	Spambase	84.61	71.42
	Group-B Average	**66.23**	**53**
14	digits	33.33	42.46
15	Dry Bean	60	100
16	ECG	60	40
17	isolet	39.35	32.62
18	Madelon	23.45	11.73
	Group-C Average	**43.23**	**45.36**
	Mean	**42.64**	**39.4**
	Standard Deviation	15.07	26.91

0.05, indicating that the 2.79% improvement achieved by our proposed algorithm over SFS is statistically significant. In contrast, the 0.36% reduction in performance compared to using all features was not statistically significant.

6.3. Analysis of stability using the similarity of feature sets

The extent of similarity between the feature components found in the feature sets of the proposed algorithm and those of supervised feature selection and the benchmark, is a crucial aspect of the study, thus serving as an indicator of stability. Therefore, a detailed comparison is made between the most frequent semi-supervised feature set along with the most frequent supervised feature set retrieved after n operations, and the full feature set considered for the benchmark is presented in Table 5 next.

From Table 5, it can be observed that the mean similarity between the benchmark and RSSFS is 39.4%, and the mean similarity between SFS and RSSFS is 42.64%. The similarity between the most frequent feature sets from the two methods, namely the proposed algorithm and the supervised mode of feature selection along with the full feature set of the benchmark, are also evaluated as they provide insight into the stability of the feature sets.

This is pictorially represented in Figure 8 as shown next.

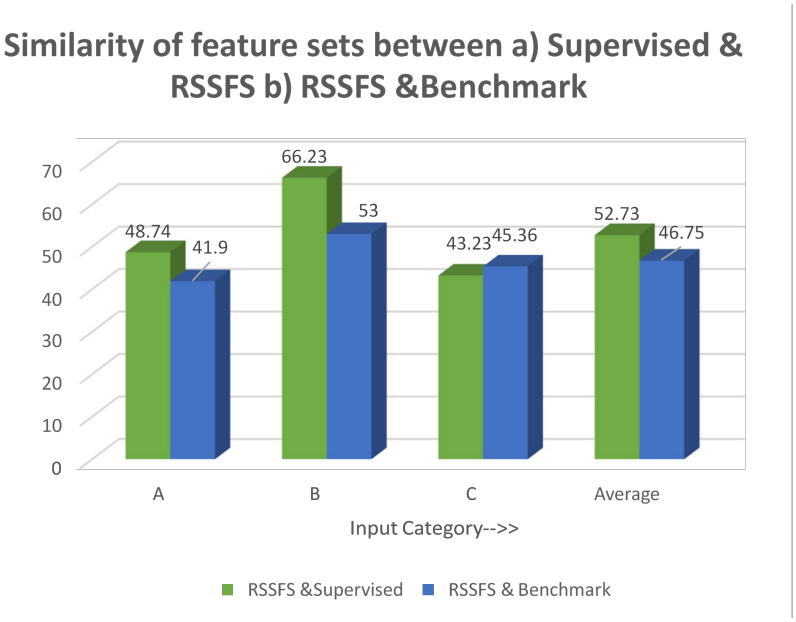


Figure 8: Similarity of the feature sets of the proposed algorithm against benchmark and SFS in all categories

From Figure 8, it can be seen that the similarity of the feature components of the proposed algorithm with the benchmark and that of the supervised feature selection is in the range of 40-60% approximately in all the three categories.

6.4. Analysis of the Area Under the Curve (AUC) from the RoC curve

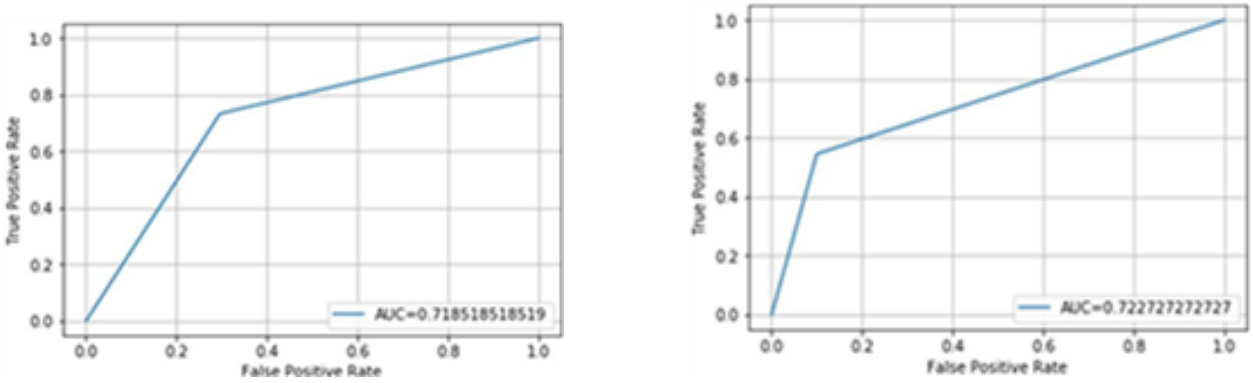
Another way to justify the proposed algorithm is by considering metrics such as sensitivity and specificity. In this analysis, the AUC was calculated from the RoC [*c.f.* Fawcett (2006)]. The performance of the proposed algorithm is compared against Supervised Feature Selection (SFS) using the sonar dataset as shown next in Figure 9 and the isolet dataset as shown in Figure 10. The classifier used in these tests is decision tree.

On analyzing both the figures 9 and 10, the observations are mentioned as follows:

i. For the sonar dataset, the proposed algorithm outperforms SFS, exceeding it by 0.42% in terms of AUC, as shown in Figures 9 (a) and (b).
ii. For the isolet dataset, the proposed algorithm again outperforms SFS significantly, exceeding it by 6.67% in terms of AUC, as shown in Figures 10 (a) and (b). Hence for both the categories A/1 and category C/3 the proposed algorithm excels over SFS.

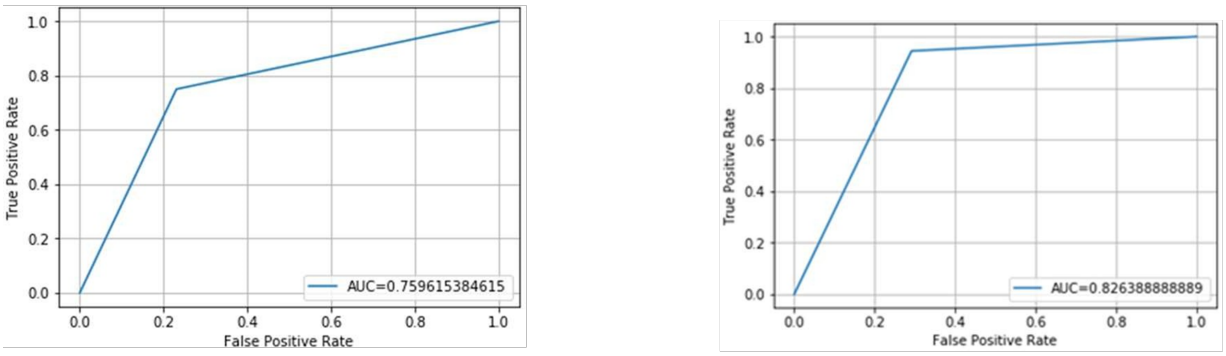
7. Conclusions

The proposed work aimed to implement feature subset selection in partially labeled datasets, with the majority portion being unlabeled. This methodology was evaluated on 18 datasets and compared against all features and supervised feature selection . For comparison,



(a) SFS **(b) RSSFS**

Figure 9: sonar dataset–RoC curve and the AUC



(a) SFS **(b) RSSFS**

Figure 10: isolet dataset–RoC curve and the AUC

conventional performance metrics and similarity measures were used. On analyzing across 18 datasets,the major observations are summarized as follows:

i. 20% of selected features of the proposed algorithm outperformed the 20% features of the supervised set and the benchmark having the full feature set, by 2.79% and 0.36% respectively based on the F1 Score.

ii. The mean similarity of the feature components of the feature sets among benchmark and RSSFS is 39.4%, while the mean similarity among the feature sets of SFS and RSSFS is 42.64%.

In datasets where the number of labeled data is much less, maybe 5-10% at most, with 90-95% having unlabeled data, it is challenging for the proposed algorithm, and here, the performance metrics may be affected. Also, in datasets where there is a clear imbalance in the number of classes /label representations, then the proposed method is prone to failure as it may learn in a biased environment given the dataset.

While the proposed work has been validated on representative datasets, future research will focus on improving the model's performance in scenarios where labeled data is extremely limited. Also in datasets, where there is a class imbalance problem, this presents an intriguing opportunity to develop a feasible solution.

Apart from predicting the labels, this work can also be applied in those domains where the availability of data on relevant features is scarce. In conclusion, it is anticipated that the significant gap between clustering and classification will greatly diminish. In essence, both methods can be viewed as special cases of semi-supervised learning, where either only labeled data or only unlabeled data is available.

Acknowledgements

We are indeed grateful to the Editors for their guidance and counsel. We are very grateful to the reviewer for the valuable comments and suggestions of generously listing many useful references.

Conflict of interest

The authors do not have any financial or non-financial conflict of interest to declare for the research work included in this article.

References

Barkia, H., Elghazel, H., and Aussem, A. (2011). Semi-supervised feature importance evaluation with ensemble learning. In *2011 IEEE 11th International Conference on Data Mining*, 31–40.

Bellal, F., Elghazel, H., and Aussem, A. (2012). A semi-supervised feature ranking method with ensemble learning. *Pattern Recognition Letters*, **33**, 1426–1433.

Bellman, R. and Kalaba, R. (1959). A mathematical theory of adaptive control processes. *Proceedings of the National Academy of Sciences*, **45**, 1288–1290.

Benabdeslem, K. and Hindawi, M. (2011). Constrained laplacian score for semi-supervised feature selection. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5-9, 2011. Proceedings, Part I 11*, 204–218.

Cai, D., He, X., and Han, J. (2007). Semi-supervised feature selection via spectral analysis. In *Proceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, 199–207.

Chang, X., Shen, H., Wang, S., Liu, J., and Li, X. (2014). Semi-supervised feature analysis for multimedia annotation by mining label correlation. In *Advances in Knowledge Discovery and Data Mining: 18th Pacific-Asia Conference, PAKDD 2014, Tainan, Taiwan, May 13-16, 2014. Proceedings, Part II 18*, 74–85.

Chelvan, P. M. and Perumal, K. (2016). A study on selection stability measures for various feature selection algorithms. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 1–4. IEEE.

Chen, L., Huang, R., and Huang, W. (2010). Graph-based semi-supervised weighted band selection for classification of hyperspectral data. In *2010 International Conference on Audio, Language and Image Processing*, 1123–1126.

Cheng, H., Deng, W., Fu, C., Wang, Y., and Qin, Z. (2011). Graph-based semi-supervised feature selection with application to automatic spam image identification. In *Computer Science for Environmental Engineering and EcoInformatics: International Workshop, CSEEE 2011, Kunming, China, July 29-31, 2011, Proceedings, Part II*, 259–264.

Dai, K., Yu, H.-Y., Li, Q., et al. (2013). A semisupervised feature selection with support vector machine. *Journal of Applied Mathematics*, **2013(1)**, 416320.

Dópido, I., Li, J., Marpu, P. R., Plaza, A., Dias, J. M. B., and Benediktsson, J. A. (2013). Semisupervised self-learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, **51**, 4032–4044.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, **27**, 861–874.

Han, Y., Park, K., and Lee, Y.-K. (2011). Confident wrapper-type semi-supervised feature selection using an ensemble classifier. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, 4581–4586.

Han, Y., Yang, Y., Yan, Y., Ma, Z., Sebe, N., and Zhou, X. (2014). Semisupervised feature selection via spline regression for video semantic recognition. *IEEE Transactions on Neural Networks and Learning Systems*, **26**, 252–264.

He, X., Cai, D., and Niyogi, P. (2005). Laplacian score for feature selection. In *Advances in Neural Information Processing Systems (NeurIPS)*, **18**, 507–514.

John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In Cohen, W. W. and Hirsh, H., editors, *Machine Learning Proceedings 1994*, 121–129. Morgan Kaufmann, San Francisco (CA).

Kalakech, M., Biela, P., Macaire, L., and Hamad, D. (2011). Constraint scores for semi-supervised feature selection: A comparative study. *Pattern Recognition Letters*, **32**, 656–665.

Khaire, U. M. and Dhanalakshmi, R. (2022). Stability of feature selection algorithm: A review. *Journal of King Saud University-Computer and Information Sciences*, **34**, 1060–1073.

Kok, T. T., Krempl, G., and Schnack, H. G. (2021). Implementation of and experimental software for active selection of classification features. *Software Impacts*, **9**, 100103.

Kumar, V. and Minz, S. (2014). Feature selection. *SmartCR*, **4**, 211–229.

Lichman, M. et al. (2013). UCI machine learning repository.

Liu, D.-Y., Xu, L.-M., Lin, X.-M., Wei, X., Yu, W.-J., Wang, Y., and Wei, Z.-M. (2022). Machine learning for semiconductors. *Chip*, **1**, 100033.

Liu, Y., Nie, F., Wu, J., and Chen, L. (2010). Semi-supervised feature selection based on label propagation and subset selection. In *2010 International Conference on Computer and Information Application*, 293–296.

Liu, Y., Nie, F., Wu, J., and Chen, L. (2013). Efficient semi-supervised feature selection with noise insensitive trace ratio criterion. *Neurocomputing*, **105**, 12–18.

Lv, S., Jiang, H., Zhao, L., Wang, D., and Fan, M. (2013). Manifold based fisher method for semi-supervised feature selection. In *2013 10th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 664–668.

Ma, Z., Nie, F., Yang, Y., Uijlings, J. R., Sebe, N., and Hauptmann, A. G. (2012). Discriminating joint feature analysis for multimedia data understanding. *IEEE Transactions on Multimedia*, **14**, 1662–1672.

Ma, Z., Yang, Y., Nie, F., Uijlings, J., and Sebe, N. (2011). Exploiting the entire feature space with sparsity for automatic image annotation. In *Proceedings of the 19th ACM International Conference on Multimedia*, 283–292.

Ren, J., Qiu, Z., Fan, W., Cheng, H., and Yu, P. S. (2008). Forward semi-supervised feature selection. In *Advances in Knowledge Discovery and Data Mining: 12th Pacific-Asia Conference, PAKDD 2008 Osaka, Japan, May 20-23, 2008 Proceedings 12*, 970–976.

Rosenberg, C., Hebert, M., and Schneiderman, H. (2005). Semi-supervised self-training of object detection models. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)*, **1**, 29–36.

Shi, C., Ruan, Q., and An, G. (2014). Sparse feature selection based on graph Laplacian for web image annotation. *Image and Vision Computing*, **32**, 189–201.

Song, X., Zhang, J., Han, Y., and Jiang, J. (2016). Semi-supervised feature selection via hierarchical regression for web image classification. *Multimedia Systems*, **22**, 41–49.

Tran, T. N., Vu, D. M., Tran, M. T., and Le, B. D. (2019). The combination of fuzzy min–max neural network and semi-supervised learning in solving liver disease diagnosis support problem. *Arabian Journal for Science and Engineering*, **44**, 2933–2944.

Wu, Q., Wang, J., and Zhang, J. (2013). Semi-supervised feature selection using max-margin criterion. *IEEE Transactions on Neural Networks and Learning Systems*, **24**, 1279–1291.

Wu, Z., Wu, J., Cao, J., and Tao, D. (2012). HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 985–993.

Xin, B., Hu, L., Wang, Y., and Gao, W. (2015). Stable feature selection from brain sMRI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, **29**.

Xu, Z., King, I., Lyu, M. R.-T., and Jin, R. (2010). Discriminative semi-supervised feature selection via manifold regularization. *IEEE Transactions on Neural networks*, **21**, 1033–1047.

Yang, L. and Wang, L. (2007). Simultaneous feature selection and classification via semi-supervised models. In *Third International Conference on Natural Computation (ICNC 2007)*, **1**, 646–650.

Yarowsky, D. (1995). Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, 189–196.

Yu, L., Han, Y., and Berens, M. E. (2011). Stable gene selection from microarray data via sample weighting. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **9**, 262–272.

Zhao, J., Lu, K., and He, X. (2008). Locality sensitive semi-supervised feature selection. *Neurocomputing*, **71**, 1842–1849.

Zhao, Z. and Liu, H. (2007). Semi-supervised feature selection via spectral analysis. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, 641–646.