Subject

Revisiting Feature Selection with Data Complexity for Biomedicine

Ngan Thi Dong *,Lisa Winkler and Megha Khosla

[1] L3S Research Center, Leibniz University Hannover, Germany

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

The identification of biomarkers or predictive features that are indicative of a specific biological or disease state is a major research topic in biomedical applications. Several feature selection methods ranging from simple univariate methods to recent deep-learning methods have been proposed to select a minimal set of the most predictive features. In this work, we perform a comparative study of 11 feature selection methods over 8 publicly available datasets evaluated using stability measure as well as classification and feature reconstruction tasks. We provide several guidelines to choose suitable feature selection methods for a given dataset based on its data 5 complexity measurements. We also investigate deep learning methods which were largely ignored in previous comparative studies. Despite small sample sizes, a recently proposed multilayer perceptron based deep learning method show competitive performance on multiple datasets pointing to the need of more investigations in such methods.

1 Introduction

One of the core issues in applying machine learning and data mining techniques to biomedical domain is the so called *curse of dimensionality*. This refers to the phenomena observed in biomedical data: small number of instances with high dimensionality (features), leading to high sparsity in data, which adversely affects algorithms designed for low-dimensional space [1]. In addition, with large number of features, learning models tend to overfit hence leading to a drop in performance on unseen data. Consider for example, gene micro-array analysis [2], where data might contain thousands of variables in which many of them could be exceedingly correlated. Generally, for a pair of perfectly correlated features, keeping one is sufficient to retain the descriptive power of the pair. These redundant but relevant features can contribute significantly to the over-fitting of a model. In addition, there could exist some noisy features (e.g, the ones having no correlation to the class) leading to erroneous class separation. In such cases, feature selection[3], as a dimension reduction technique, has proven to be effective in preparing the data or selecting the most relevant features for performing downstream machine learning tasks such as classification. In addition, it plays a critical role in *biomarker discovery* [4] for diagnosis and treatment of complex diseases.

Feature selection (FS) has been a widely applied in bioinformatics [3, 5–15] and can be broadly classified into filter, wrapper and embedded methods. While filter methods evaluate the relevance of features by considering only the intrinsic properties of the data, the wrapper method selects a feature subset by iteratively selecting features based on the classifier performance. The embedded methods combines feature selection and classifier construction using an integrated model building process. Feature selection has attracted strong research interest in the past several decades and a huge number of methods have been proposed. Nevertheless, the main question of *which FS method to use when* remains unanswered. In particular, it would be useful to develop a guideline for non machine learning specialists to aid them in the choice of a particular feature selection method for their given dataset. In this work, we prepare such a guideline by studying and comparing different aspects for feature selection methods. Our first aspect is covered by the following research question, where we investigate the suitability of FS methods with respect to properties of the dataset.

RQ 1. *Which properties of data and the FS algorithm make a particular FS method more suitable than others?*

In order to answer the above question, we conduct a comprehensive empirical comparative study using 11 FS methods, including representatives of (i) filter, (ii) wrapper, (iii) embedded (iv) and recently proposed deep learning approaches on 8 biological datasets with varying properties. Table 1 presents a summary of those 11 selected techniques. We evaluate the features returned from those methods on two machine learning tasks, namely *classification* and *feature reconstruction*. We note that previous works have largely ignored evaluation on *feature reconstruction*, where we attempt to **use the subset of selected features to reconstruct the original feature**

Method	Method Type	Learning Type	Link
mRMR	filter	supervised	link
RELIEFF	filter	supervised	link
MCFS	filter	unsupervised	link
LASSO	embedded	supervised	link
ELASTIC NET	embedded	supervised	link
HSIC-LASSO	embedded	supervised	link
SVM-RFE	wrapper	supervised	link
BORUTA	wrapper	supervised	link
TSFS	deep learning	unsupervised	link
CAE	deep learning	unsupervised	link
MLP	deep learning	supervised	link

Table 1. List of evaluated FS methods with link to used implementation.

set. We argue that this task has a practical significance in a number of situations. For example, in a medical dataset, the features correspond to various patient tests results. Although knowing the most important features of biomarkers for a disease can help the medical practitioner to diagnose using fewer tests, she might still need the values corresponding to other tests/features to administrate drugs.

In addition to presenting statistically best and worst performing algorithms, we also relate the performance difference of various methods to (a) the properties of their respective objective functions and (b) the dataset properties determined by several data complexity[16, 17] measures as explained in Section 3. To the best of our knowledge, this is the first work which applies data complexity measures to understand the suitability of FS methods. As a result, we provide a guideline which lists suitable FS methods for a particular dataset based on data complexity and the properties of the FS method.

One of the issues in evaluating FS methods is that the optimal size of relevant features is not known. A FS method usually returns either a subset of features or a list of ranked/weighted features. For ranking methods, the higher the weight/rank, the more important the feature is. For methods which return a subset of features, all features in the returned list are considered important. An approach taken by most of the previous works is to evaluate the classification performance of FS methods on a small range of pre-specified selected features. However, depending on the dataset characteristics, the number of important/indispensable features in each dataset is different. Yet it could also happen that for a simple dataset with high overlap in the feature value, a given method reaches its peak performance with a significantly smaller number of features (for example, 1%) whereas more features might be required for another dataset. Therefore, evaluating feature selection methods on merely some subsets of the key features in each dataset might result in biased judgement. Additionally, for some datasets with noisy features, adding more features could degrade its performance considerably. This reveals an issue in ranking, where noisy features are erroneously ranked higher. Such an issue is easily revealed when we use a larger range of selected features, whereby for some methods increasing the number of selected features downstream task performance drops much faster than others. Therefore, an important factor in declaring a method superior to other is to check its reliability while using a wider range of its returned features. More specifically we investigate the following research question.

RQ 2. *Which FS method is more reliable in terms of the returned list of important features?*

To determine the reliability of a method, we evaluate on a wider range, i.e., from 1% to 19% of the total features, instead of evaluating for a particular predefined %. Moreover, we do a five-fold cross validation for each dataset, instead of only using one train/test split set. This also creates the need to rank a method based on its *stability*, which is quantified by the overlap of returned features on using slightly different training data. So, the third aspect of our guideline is motivated by the following research question.

RQ 3. *Which feature selection method is more stable as regards to small changes in the training data?*

Another important property of biological datasets is the small number of instances. It is believed that deep learning methods, in particular, should not be applied to these types of datasets because of very small sample sizes. Nevertheless, several deep learning methods for feature selection[10, 18, 19] have been devised in the recent years, some of which have also been applied to biomedical datasets. In order to provide a more complete picture, we include various deep learning (DL) methods in our study and investigate the following research question empirically.

RQ 4. *Do deep learning methods also offer advantages in small instance size problems?*

Our results show that deep learning methods with proper regularization might be able to avoid over-fitting for small instance size problems. For example, MLP outperforms several other methods on a number of datasets. In particular, for feature reconstruction, 2 of the studied deep learning methods show consistently good performance.

1.1 Related Work

In this section we provide a brief overview of related reviews and comparative studies and their differences to the present work. We also point to various works which have studied data complexity measures either to quantify difficulty in classification or deciding cut-off thresholds for feature selection methods.

Degenhardt et al. [14] studied and compared various Random Forest based methods on two high dimensional real word biological datasets with respect to classification performance, stability and run time. But their focus is limited to a particular type of methods. Moreover, the number of datasets considered is also quite small.

Taking a broader perspective, Neto et al. [20] construct a large scale study on simulated data to investigate the effects of sample size, number of features, true model sparsity, signal-to-noise ratio, and feature correlation on predictive performance of ridge regression, ELASTIC NET and LASSO methods. Through diverse, carefully designed

experiments, they focus on the strengths and weaknesses of only those three methods under very particular conditions.

Urbanowicz et al. [13] evaluated 13 existing and newly proposed Relief-Based algorithms for a genetic simulation study. They run experiments on 2280 simulated datasets cover a wide range of problems and types. Nevertheless, the question of whether the findings can be applied to the real-world datasets is still left open.

Wang and Barbu [21] try to answer the questions of (i) whether filter methods help improve classification model and (ii) how existing filter methods are different from each other in terms of predictive capabilities. They construct experiments on five regression and five classification datasets. They measure the classification performance of FS methods on 40 different runs with 30 different numbers of selected features. In addition to limitation to filter methods, the focus is also not primarily on biological datasets. The survey conducted by Li et al. [22] seems to be the most comprehensive one, which presented a general summary of existing works on FS methods from different data type perspectives. Nevertheless, deep learning methods are not included in their study.

Data complexity measures for feature overlap are used in [23] to choose the feature cut-off threshold for the ranked list. In [12], complexity measures were used to quantify the difficulty of classification on two different gene expression datasets. Another work [24] relates complexity measurements to the classification performance of Support Vector Machines on cancer gene expression data.

We conclude that previous works are either (i) too narrow, focusing on a particular class of FS methods or/and using only simulated datasets, or are (ii) too broad, meaning that their comparisons are not focused solely on biological data. In addition, none of these works compare deep learning methods. From the data complexity perspective, none of the works study our proposed problem, i.e., whether one can use a set of complexity measures to guide the choice of a particular feature selection method.

In the next section, we summarize our approach and main findings. Sections 3 and 4 provide details about the complexity measures used and FS methods respectively. Finally, we present our experimental set-up and results followed by conclusion and a feature selection method guideline.

2 Approach and Findings

We perform a large scale empirical study using a wide range of FS methods and biological datasets. The goal is to characterize FS methods based on the dataset characteristics. Specifically, to answer RQ 1 we experiment with feature overlap, topological and dimensionality-based data complexity measures[16, 17] and empirically show that several of these simple-to-compute measures indicate the suitability of a particular FS method for finding the most relevant features. For example, our results suggest that neighborhood-based filter methods like RELIEFF work best for datasets with a low nearest neighbor classifier error rate. With our findings, we support the use of five simple-to-compute complexity measures (detailed in Section 3) for the selection of a more suitable FS method. Moreover, using a combination of two or more of these measures, we explain the relative higher performance scores observed for some datasets as compared to others when a comparable number of selected features are used in a task.

In order to answer RQ 2, we evaluate the FS methods over a wider range of returned relevant features ranging from 1% to 19% (with intervals of 1%) of the total features. For each specified $\%$, we compute the average scores using 5 train/test splits. We then present a summary (see Section 6) of these average scores over the complete range of 19 sets of selected features. To understand the rational behind our strategy, consider two FS methods A and B, which show high performance for classification when we use its top 10 features. Now we increase the number of used features to 12. Let both A and B assign non-zero importance to more than 12 features. Assume we see a drop in the performance score of method A, while B shows the same performance for 10 or 12 features. This would imply that the additional 2 features used by B might be redundant, as the performance does not increase, but also does not decrease. However, for FS method A, the additional 2 features are noisy features, which do cause a drop in performance. We report top and worst performing methods (with statistical significance scores) for each dataset based on such a comparison over a wider range of selected features.

As regards to RQ 3, we compute stability score of a method over five train splits (folds). Formally, the stability of a feature selection method is calculated as:

$$S = \frac{2\sum S_{ij}}{k(k-1)} \tag{1}$$

where $k = 5, S_{ij}$ is the similarity between two subsets returned from the FS method corresponding to folds i and j, $i \neq j$. We use Tanimoto[25] to measure similarity between two feature subsets. Formally, $S_{ij} = \frac{|S_i \cap S_j|}{|S_i \cup S_j|}$, where S_i, S_j are the two selected feature subsets from fold i and fold j, respectively. For each dataset, we report only the average results across five folds. We argue that such a measure of stability might not be adequate as it only considers feature overlap rather than correlations of returned features. Moreover it does not take into account the degree of change in the trained data. As a second measure of *task-specific stability* we look at standard deviations observed in the performance on classification and feature reconstruction tasks. Higher the deviation from the mean value, less stable is the method. For example, embedded methods like LASSO and ELASTIC NET obtain higher stability scores than deep learning based methods MLP but with lower standard deviation MLP shows more stability in performance over a task computed over 5 train/test splits.

With respect to RQ 4, we experimented using 3 recent deep learning methods. We find that even for datasets with a small number of instances, MLP performs consistently well for many datasets with respect to both classification and feature reconstruction tasks. We believe that its good performance stems from the right kind of

Name	Features	Samples	Classes	Source	F1	N3	T4	T2	C2
CARCINOM	9,182	174	11	link	0.003	0.138	0.015	52.770	0.026
CHIN	22,215	118	2	link	0.356	0.229	0.004	188.263	0.137
CLL-SUB-111	11,340	111	3	link	0.155	0.387	0.008	102.162	0.143
CNS	7,129	60	2	link	0.828	0.450	0.006	118.817	0.165
COLON	2,000	62	2	link	0.601	0.290	0.016	32.258	0.156
LSVT	309	126	2	link	0.719	0.270	0.116	2.460	0.200
PROSTATE-GE	5,966	102	2	link	0.389	0.128	0.010	58.490	0.001
TOX-171	5,748	171	4	link	0.386	0.123	0.022	33.614	0.002

Table 2. Dataset statistics. Links to sources are provided. The attribute/feature type of data for all data sets is genomic, except for LSVT, which contains dysphonia measurements.

regularization employed in its objective. In addition, one of the other deep learning based method also shows comparable performance on feature reconstruction task.

We also discover that BORUTA , HSIC-LASSO though return only a small subset of features, still perform quite well in term of classification performance. And when it comes to stability they also gain moderate performance.

3 Data Complexity Measures

In this section, we briefly describe the data complexity measures that we use in our analysis. Data complexity measures [16] have been traditionally used to study the intrinsic difficulty of a classification task on a given dataset. In this work, we relate the data complexity characterized by 5 such measures to the suitability of a particular FS method. We use m to denote the number of features and n to denote the number of samples. The definitions and implementation of these measures have been adapted from [17].

1. **Maximum Fischer's Discriminant Ratio (F1)[26]:** F1 represents the maximum discriminating power of the features. Discrimination is measured by the overlap of feature values between different classes. Formally,

$$F1 = \max_{i=1}^{m} \frac{\sum_{j=1}^{n_c} n_{c_j}(\mu_{c_j}^{f_i} - \mu^{f_i})^2}{\sum_{j=1}^{n_c}\sum_{l=1}^{n_{c_j}}(x_{li}^{j} - \mu_{c_j}^{f_i})^2} \quad (2)$$

   where n_{c_j} is the number of samples of class c_j , $\mu_{c_j}^{f_i}$ is the mean of feature i for class j , μ^{f_i} is the mean of feature i across all classes, and x_{li}^{j} is the value of feature i for sample l of class c_j . The lower the value of F1, the higher the overlap between feature values among different classes, which makes the class-separation problem more complex.
2. **Error Rate of Nearest Neighbor (NN) classifier (N3):** N3 is measured by the error rate of the 1-NN classifier using leave-one-out cross validation. Formally, $N3 = \frac{\sum_{i=1}^{n} I(NN(x_i) \neq y_i)}{n}$, where $NN(x_i)$ is the predicted target value for sample x_i using all other samples as the training set. High N3 scores indicate that instances of different classes are close together.
3. **Ratio of the PCA dimension to the original dimension (T4) [27]**: T4 is the ratio of the number of PCA (Principal Component Analysis) components needed to represent 95% of data variability on the total number of features.

   Higher T4 scores indicate a larger portion of the original features set is necessary to preserve the information of the dataset, thus, implying the need to use a larger number of features for a given task.
4. **Sparsity (T2):** Sparsity is defined as $T2 = \frac{n}{m}$. Highly sparse datasets can be difficult for classification since learning process can be hindered in the low density regions.
5. **Class Imbalance(C2):** The imbalance ratio measures the differences in the number of instances per class in the dataset and is computed as:

$$C2 = 1 - \frac{1}{ci}, \text{ where } ci = \frac{n_c - 1}{n_c}\sum_{i=1}^{n_c}\frac{n_i}{n - n_i},$$

   where n_i is the number of samples in class i and n_c is the number of classes. Higher values of C2 indicate higher imbalance.

4 Compared Feature Selection Methods

In this work, we compare representatives of a wide range of unsupervised and supervised FS methods, including filter, embedded, wrapper and deep learning (DL) based methods. The choice of our models is based on recommendations from previous works, as well as our own initial set of experiments that we conducted. We included a larger number of methods from each category, from which we choose a subset of the best-performing approaches. Unlike previous evaluation works, we have included recent deep learning methods in our study. Experimental results claim that even small instance size problems can benefit from the non-linearity learnt by deep methods. A summary of the compared methods is provided in Table 1.

Maximum Relevance Minimum Redundancy (mRMR)[28]. mRMR selects features, which are highly correlated with the class attribute (maximum relevance), but weakly correlated with other features (minimum redundancy). In other words, the algorithm selects relevant features which are maximally dissimilar using Mutual Information Scores[29].

RELIEFF [30]. RELIEFF is based on the Relief algorithm[31] whose main idea is to estimate features according to how well their values distinguish between neighboring samples. Each feature is weighted according to the relationship of n random samples to their nearest neighbor(s). For a given sample, RELIEFF selects k nearest

samples (hits) from the same class and k nearest samples (misses) from each of the other classes.

Multi-Cluster Feature Selection (MCFS) [32]. MCFS aims to select the most informative features by selecting features, which preserve the clustering structure of the data. It works in two stages. The first stage is responsible for constructing a k-nearest neighbor weighted graph from the dataset as well as learning a low-dimensional representation (embedding) for each node (instance) by solving the generalized eigen problem [33]. The second stage is subjected to extracting the importance of each feature by solving a L1 regularized least squares problem, such that the clustering structure of the data is preserved.

Least Absolute Shrinkage and Selection Operator (LASSO)[34]. LASSO is a linear regression method with an L1-penalty (regularization), which shrinks many of the coefficients to zero. Features with non-zero coefficients are considered important features by LASSO . However, for a set of correlated features, LASSO tends to randomly pick only one feature.

ELASTIC NET [35]. ELASTIC NET addresses the drawback of LASSO , by incorporating both the L1 and L2 regularization penalties. Like LASSO , Elastic Net simultaneously produces a model and performs automatic variable selection via shrinkage; however, it is also able to account for subsets of correlated features by using the L2 penalty.

HSIC-LASSO [36]. HSIC-LASSO extends LASSO by finding non-linear feature dependencies. In particular, it finds non-redundant features with strong statistical dependence on the output classes using kernel-based independence measures such as the Hilbert-Schmidt independence criterion (HSIC)[37].

Support Vector Machine - Recursive Feature Elimination (SVM-RFE)[9]. SVM-RFE uses an iterative elimination process to remove redundant features in two phases. During the first phase, while the SVM classifier is trained on the original dataset, weights for all features are also computed. In the second phase, the least important features (with lowest weights) are removed from the dataset. This process is repeated until only one feature remains. SVM-RFE then returns a ranked list of all of the features (according to their removal order).

BORUTA[38]. The key idea behind this approach is to compare the importance of every feature with those of random or *shadow* variables using statistical testing and several runs of Random Forest. A shadow variable is created for every feature by permuting its original value. After that, a Random Forest classifier model is trained on the extended dataset, while the importance scores/weights of all of the attributes, including the shadow variables, are calculated at the same time. Since the shadow variables are designed to be random, their weights are expected to be close to zero. Boruta uses the highest importance score of all shadow variables as a threshold to determine whether a feature is truly important or redundant.

Deep Feature Selection with Multi Layer Perceptron (MLP) [10]. This model is constructed from an multilayer perceptron by adding a sparse one-to-one linear layer between its input layer and the first hidden layer. The weights of this one-to one layer are considered as the importance of the corresponding features. The model parameters (including those of the one-to-one layer) are optimized by log likelihood loss function, along with an ELASTIC NET based regularization for the feature importance weights.

Teacher Student Feature Selection (TSFS) [18]. In the teacher-student feature selection (TSFS) method a 'teacher' (a deep neural network or a manifold learning method) is first employed to learn the best representation of data in a low dimension. Then a 'student' network (a simple neural network) is used to perform feature selection by minimizing the reconstruction error of the low-dimensional representation using a row-sparsity constraint. In our experiments, we use MDS [39] as the teacher network.

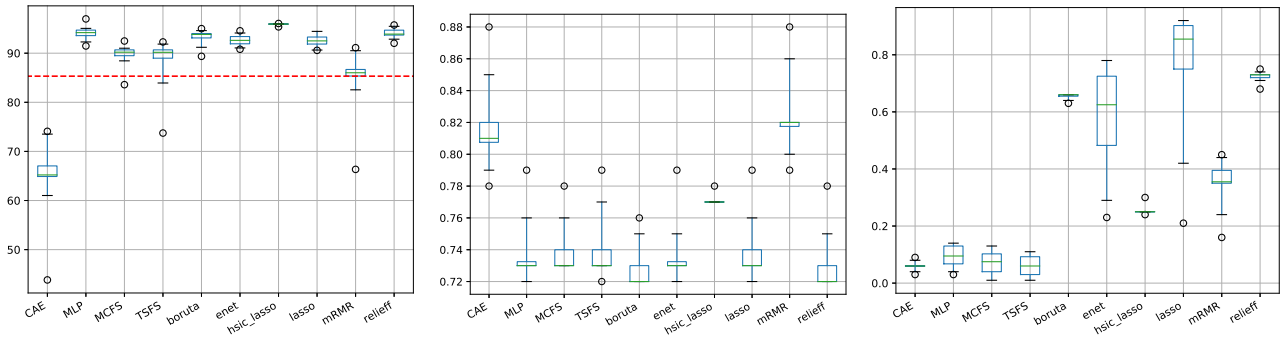
Concrete autoencoder (CAE) [19]. CAE is based on the standard autencoder, but uses a concrete selector layer as the encoder and a standard neural network as the decoder. The concrete selector layer consists of k nodes, where k is the predefined number of features to be selected. For each of the k nodes in the concrete selector layer, a concrete random variable, which is a relaxation of discrete feature weight regulated by a temperature parameter, is sampled. The kth node in the selector layer then outputs the linear combination of features at a higher temperature. As the temperature of the selector layer is gradually decreased and approaches zero, each node of the concrete selector layer outputs exactly one input feature. Once the model has been trained, the concrete selector layer is replaced with a discrete arg max layer for the testing phase. One of the drawbacks of this method is that the number of required features needs to be explicitly specified.

5 Evaluation Set Up

A summary of the compared FS methods is given in Table 1 and the datasets are summarized in Table 2. For each dataset, we use z-score[1] to normalize the feature values. After that, we divided the data into 5 folds randomly. On each of the 5 runs, 4 folds were used to train the feature selection model to obtain the relevant features and one fold was held out as test set. These relevant features were then used to train the classification or feature reconstruction (regression) model using the training data. The performance was then tested on the test fold using only the returned relevant features (on training set). For classification, average F_1 scores (harmonic mean of precision and recall) are reported and for feature reconstruction[2], mean square error (MSE) is reported. As baseline for classification, we use the complete feature space and average F_1 score on different test splits is shown as a red dashed line in the corresponding plots. We use the same vanilla set up with default parameters for all

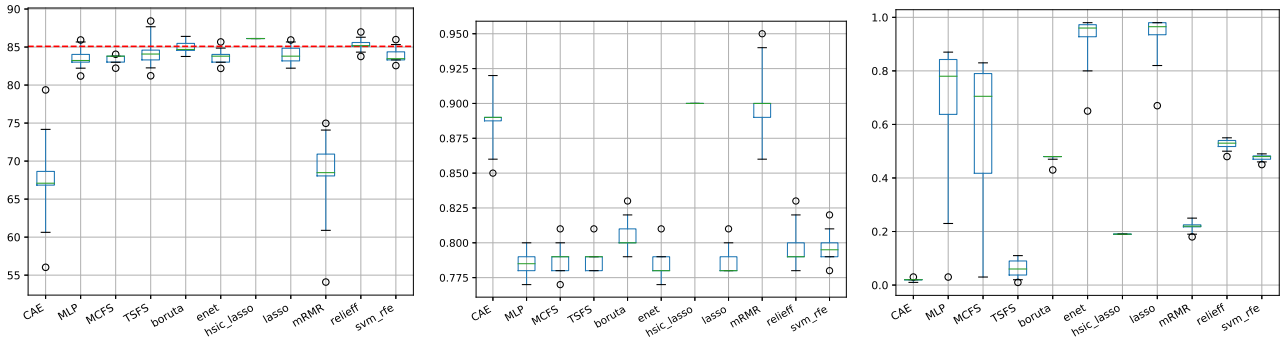
[1] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html

[2] For feature reconstruction, we adapt the implementation from https://github.com/alimirzaei/TSFS



(a) Classification (b) Reconstruction (c) Stability

Fig. 1: CARCINOM . y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b) .



(a) Classification (b) Reconstruction. (c) Stability

Fig. 2: Chin. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b)

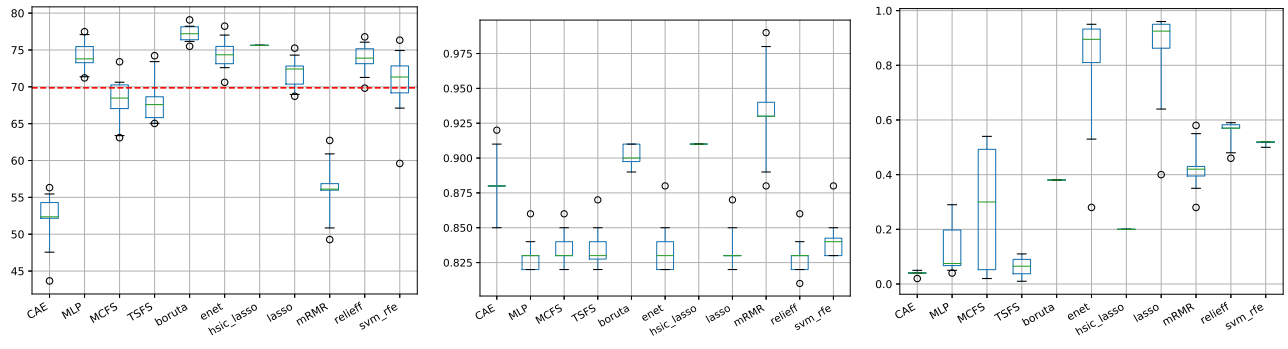
of the feature selection algorithms and learning models. For computing statistical significance of performance differences, we perform Welch's paired t-tests. Tables 4 - 19 in the supplementary material present the statistical significance test results of all pairs of methods with regards to classification performance and reconstruction errors for our selected datasets.

For all methods, except CAE and mRMR , we vary the range of selected features from 1% to 19%. However, for CAE and mRMR , we had to use a different feature range, as it would have taken weeks to compute the results for the 1% to 19% range. Instead of choosing a specified $\%$ of features, we chose a range of a fixed number of features and restricted the number of selected features to 100. Because the number of features in LSVT is only several hundreds, we ran CAE and mRMR on a range of features from 5 to 50 with a step increase of 5. For the other datasets, we ran CAE and mRMR on a range of features from 10 to 100 with a step increase of 10. We also note that SVM-RFE did not converge for CARCINOM , which has 11 classes, with the vanilla set-up. We point out that comparing the methods based on their space and time complexities is out of scope of this work.

6 Results

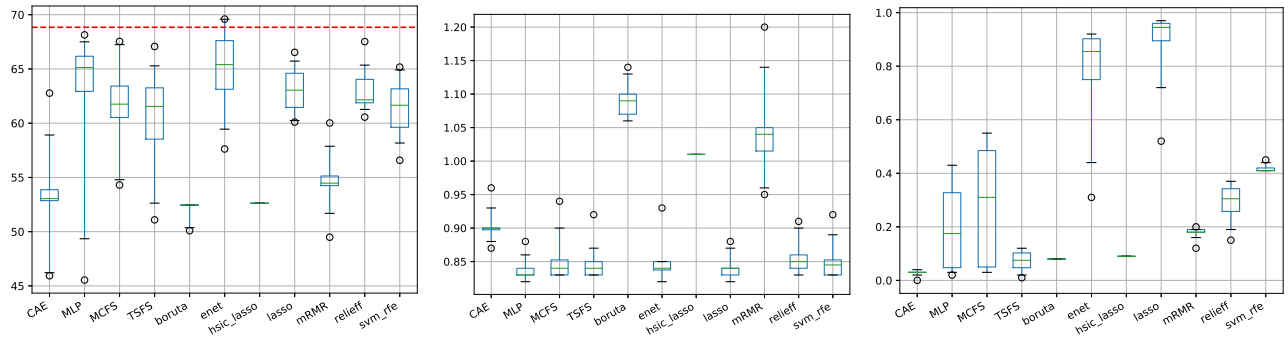
In Table 2, we present the complexity measurements for all of the datasets. In addition, we also measure complexity for all training and test splits and present the summarized results in the Figures 10-12 in the supplementary material. In all of the plots, we divide the values of T2 by 1000 for display purposes. High variation of the obtained complexity scores within/among train/test splits suggests high complexity of the dataset. The summarized results for classification(F_1), reconstruction error and stability, where the number of selected features range from 1% to 19% of the total number of features, are presented in the first three plots of Figures 1-8. Detailed plots for top performing methods (for each dataset) over the complete range of selected features are also provided in the supplementary material. Each bar represent a range of values according to the performance of different methods on a range of different number of selected features. The two points indicate the lowest and highest values while the green line represents the average value. The longer the bar, the higher the variance in those values. In addition, we also observe that HSIC-LASSO and BORUTA return a limited number of features which are sometimes less than the desired percentage. Table 1 in the supplementary file presents the number of returned features from those two methods compared to the desired ones taken by others.

In the following sections, we analyze the effect of a dataset's characteristics with respect to the choice of a FS method. We quantify the characteristics of the datasets using the complexity measures. Complexity scores of the experimented datasets are presented in Table 2.



(a) Classification (b) Reconstruction. (c) Stability

Fig. 3: CLL-SUB-111. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b).



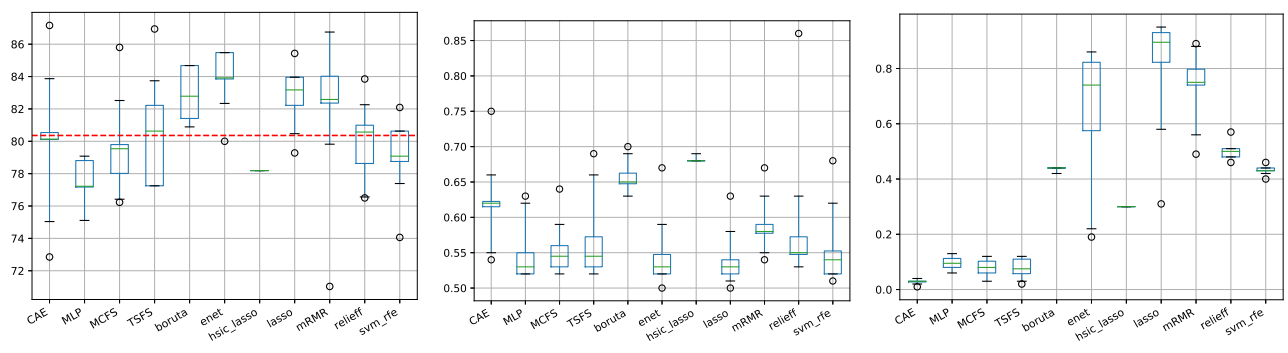
(a) Classification (b) Reconstruction (c) Stability

Fig. 4: CNS. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b).

6.1 Effect of F1 and T4

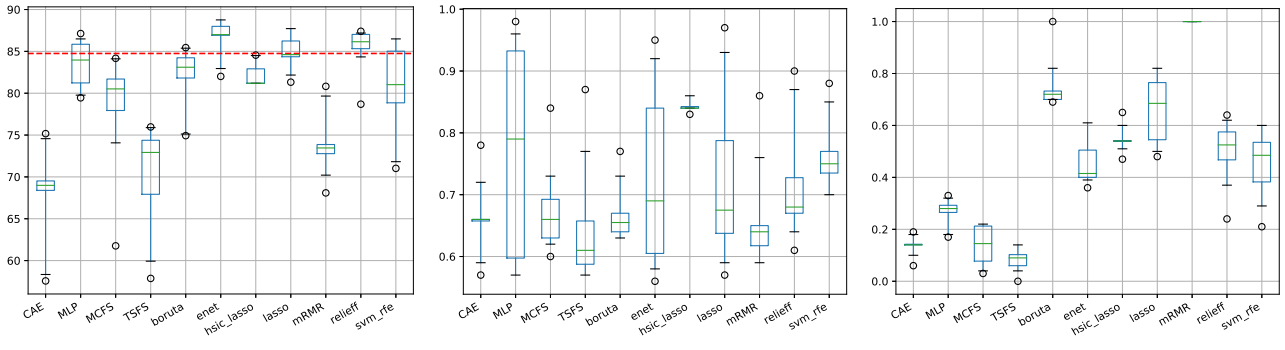
Low F1 values imply high overlap in the feature values of instances of different classes. A low T4 value signifies that only a small fraction of the original features set is needed to explain the data. Our hypothesis is that a combination of low F1 and low T4 indicate the presence of noisy features which, if removed, can make the classification problem easier. CARCINOM is the representative dataset that has this characteristic. We can see in plot 1a, most methods outperform the baseline classification performance with the largest gain of around 10%. At the same time, we observe the smallest variation in train/test complexity plot. That might be the underlying reason why we observe the smallest variation in classification performance of all methods in this dataset. Moreover, it also explains why though CARCINOM has more than $9K$ features and HSIC-LASSO uses at most 200 features, the method still gains the highest classification performance (statistically significant with p-value $= 0$) compared with other methods.

CHIN is the dataset with smallest value of T4. Small T4 value indicates only a small portion of the data needed to represent 95% of the data. HSIC-LASSO by using around 50 relevant features out of $22K$ is one of the best performing methods. Another best method on this dataset, RELIEFF , also obtains its best performance by using only 1% of the features.



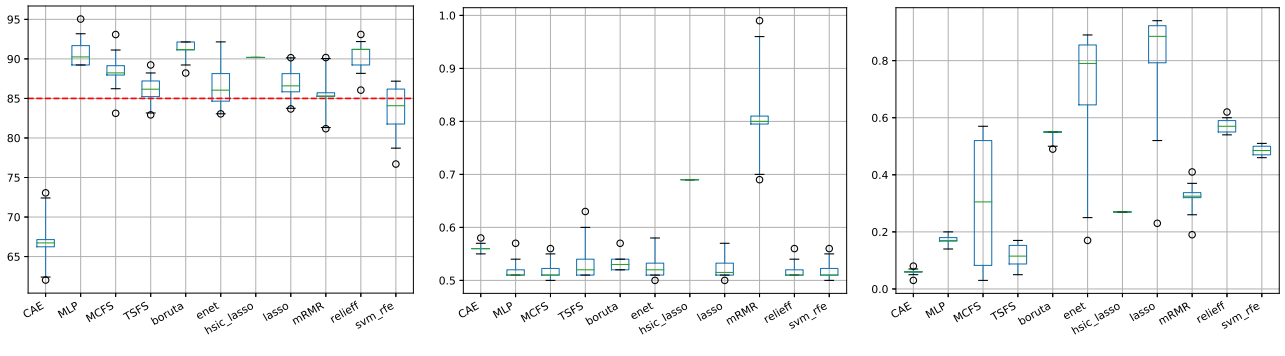
(a) Classification (b) Reconstruction. (c) Stability

Fig. 5: Colon. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b)



(a) Classification (b) Reconstruction. (c) Stability

Fig. 6: LSVT. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b.)



(a) Classification (b) Reconstruction. (c) Stability

Fig. 7: Prostate-GE. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b).

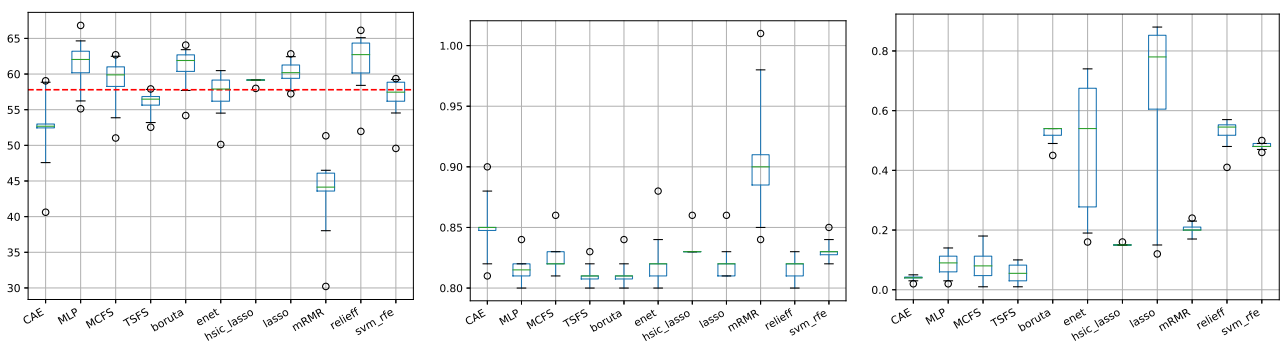
A relatively high T4 value for LSVT indicates that we need more features to describe the data. This effect is prominent in the feature reconstruction results, where we observe a huge variation in performance across all of the methods. Basically, all methods witness a large mean square error, when using very small number of features. The error decreases, when a higher % of features is used.

6.2 Effect of N3

A low N3 value suggests that neighboring instances belong to the same class. For this reason, neighborhood-based algorithms, like RELIEFF , perform better for datasets with low N3 scores, compared with other datasets. CARCINOM , PROSTATE-GE and TOX-171 have the lowest N3 scores where RELIEFF is one of the best performing methods for classification. For the datasets with high N3 values, COLON and CNS , one can observe weaker performance for RELIEFF in classification.

6.3 Effect of C2 and T2 with N3, F1

A high C2 value indicates high class imbalance, which results in overall limited performance for any method. We recognize that whenever there is a high variance in C2 score among the test sets, there is also a high



(a) Classification (b) Reconstruction. (c) Stability

Fig. 8: TOX-171. y-axis for plot (a) is the F_1 score (in %) and mean square error for plot (b).

variance in the classification performance among different feature selection methods as well as different number of selected features for the same method. COLON , LSVT , CNS are some of the most prominent datasets with this characteristic. ELASTIC NET seems to be the best choice of all of these datasets.

High values for T2 imply data sparsity. Higher values for C2 and T2 make the dataset more complex than others. CNS falls into this category. We notice most FS methods get lower classification performance compared to the base line. In addition, CNS has high nearest neighbor error rate (N3). If one compares the classification scores for all of the datasets, CNS comes in last. Moreover, the minimum error in feature reconstruction is much more than what is observed for other datasets, i.e., it is more difficult to reconstruct the feature space from a small set of selected features in CNS than for other datasets. The overall average stability score for most of the FS methods is also one of the lowest.

Another example of a dataset with high sparsity (T2) and imbalance is CHIN . Despite these scores, it is still an easier dataset, because of its low N3 score. HSIC-LASSO and RELIEFF are the best performing methods. We found out that though HSIC-LASSO only uses around 50 selected features (much smaller than the number of selected features in other methods), it already gains the second best position in term of classification performance. Moreover, both CNS and CHIN had high Fischer discriminant ratio (F1), indicating a generally low feature overlap among classes in the original feature space. That explains why using the whole set of features for classification achieves competitive performance compared to that of using a smaller subset of important features.

6.4 Feature Reconstruction

For feature reconstruction, CAE , mRMR and HSIC-LASSO are the worst performing methods. HSIC-LASSO which uses much less number of features than others is not able to reconstruct the original feature space with a very small number of features. The performance of all other methods is comparable except BORUTA whose performance drops for some datasets. For example, it returns a maximum of 13 relevant features for CNS which is insufficient for reconstruction task.

6.5 Stability

In terms of overlap among the returned feature set for different training splits, ELASTIC NET and LASSO are in the leading position for most of the datasets. However, this does not translate to task specific stability, where they show a larger standard deviation in task performance as compared to MLP which always has low stability scores.

6.6 Deep Learning Methods

The multilayer perceptron based deep FS method shows high performance for many datasets in regards to classification, as well as feature reconstruction. MLP , **which uses EN type regularization**, shows better performance than EN in classification for all datasets except COLON ,CNS and LSVT , possibly because of its ability to find non-linear feature dependencies. The teacher-student network based FS method, which is trained using a feature reconstruction objective, shows good performance in feature reconstruction tasks.

7 Conclusion

We conducted an extensive comparative study of 11 FS methods for 8 biological datasets with varying properties. Our evaluation set up consisted of using the returned set of features for two learning tasks (i) classification and (ii) feature reconstruction. We divided our datasets randomly into five-folds and used 4 folds as training data and 1 fold as test data in each of the 5 runs. This ensures that train and test sets does not have any common datapoints and each datapoint is used as a test point. We ran stability tests over the returned features from these 5 runs on training data. As the optimal number of features is not known in prior, we tested over a wider range where the number of selected features were varied from 1% to 19% of the total features. This helped us not to only to test the performance on the two learning tasks but also the reliability of returned ranking or weights of features. We note that BORUTA and HSIC-LASSO were exceptions here as they returned a small number of relevant features. Inspite of this, they performed quite well in several datasets for classification task but failed in feature reconstruction due to the use of a few number of features. Moreover, due to high computational requirements of CAE and mRMR , we were able to test these methods only on a smaller range of features. In addition we evaluated 3 deep learning methods and our findings encourage future investigations of such method even for small sample size problems. We also argued about the effect of 5 complexity measures which can be used to describe the data and their effect on performance of a given FS method. In conclusion, we present our key points which are summarized as a guideline for selecting suitable feature selection methods in Figure 9.

References

[1]Trevor Hastie, Robert Tibshirani, Jerome Friedman, et al. The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2):83–85, 2005.

[2]Cosmin Lazar, Jonatan Taminau, Stijn Meganck, et al. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM TCBB*, 9(4):1106–1119, 2012.

[3]Zena M Hira and Duncan F Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015, 2015.

[4]Mickael Leclercq, Benjamin Vittrant, Martin-Magniette, et al. Large-scale automatic feature selection for biomarker discovery in high-dimensional omics data. *Frontiers in genetics*, 10:452, 2019.

[5]Yvan Saeys, Iñaki Inza, and Pedro Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

Key Points

- Ratio of PCA dimension to actual number of features (T4) provides a useful lower bound for the fraction of features required to represent the data.
- For datasets with low maximum Fischer's discriminant ratio (F1) and low T4, supervised methods BORUTA and HSIC-LASSO are recommended to build predictive classification models.
- For data with low error rate of nearest neighbor classifier (N3), RELIEFF is recommended.
- Deep learning based MLP , which uses ELASTIC NET type regularization is a good candidate for both the tasks.
- ELASTIC NET outperforms other methods in classification in presence of high class imbalance in data.
- HSIC-LASSO and BORUTA might return a very small number of relevant features which limits their applicability to datasets where a larger fraction of relevant features are required (for example for cases with high T4).
- For cases where stability of the returned feature set is a concern for example for biomarker discovery,ELASTIC NET and LASSO are recommended.
- Though we did not provide exact running times of the methods, we observed in our experiments that CAE and mRMR required much larger amount of time. Moreover, CAE requires that the number for desired features is prespecified . This makes it difficult in biomedical applications were such a number is not known in advance.

Fig. 9: Guidelines for selecting a suitable Feature Selection Method

[6]Huan Liu and Hiroshi Motoda. *Computational methods of feature selection*. CRC Press, 2007.

[7]Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. *Briefings in bioinformatics*, 9(5):392–403, 2008.

[8]Thomas Abeel, Thibault Helleputte, Yves Van de Peer, et al. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, 2009.

[9]Xiaohui Lin, Chao Li, Yanhui Zhang, et al. Selecting feature subsets based on svm-rfe and the overlapping ratio with applications in bioinformatics. *Molecules*, 23:52, 12 2017.

[10]Yifeng Li, Chih-Yu Chen, and Wyeth W Wasserman. Deep feature selection: theory and application to identify enhancers and promoters. *Journal of Computational Biology*, 23(5):322–336, 2016.

[11]Chad A Davis, Fabian Gerick, Volker Hintermair, et al. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics*, 22(19):2356–2363, 2006.

[12]V. Kamath, T. J. Yeatman, and S. A. Eschrich. Toward a measure of classification complexity in gene expression signatures. In *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5704–5707, 2008.

[13]Ryan J Urbanowicz, Randal S Olson, Peter Schmitt, et al. Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of biomedical informatics*, 85:168–188, 2018.

[14]Frauke Degenhardt, Stephan Seifert, and Silke Szymczak. Evaluation of variable selection methods for random forests and omics data sets. *Briefings in bioinformatics*, 20(2):492–503, 2017.

[15]Zahra Mungloo-Dilmohamud, Gary Marigliano, Yasmina Jaufeerally-Fakim, et al. A comparative study of feature selection methods for biomarker discovery. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2789–2791, 2018.

[16]Mitra Basu and Tin Kam Ho. *Data Complexity in Pattern Recognition (Advanced Information and Knowledge Processing)*. 2006.

[17]Ana C. Lorena, Luis P. F. Garcia, Jens Lehmann, et al. How complex is your classification problem? a survey on measuring classification complexity, 2018.

[18]Ali Mirzaei, Vahid Pourahmadi, Mehran Soltani, et al. Deep feature selection using a teacher-student network. *ArXiv*, 2019.

[19]Abubakar Abid, Muhammad Fatih Balin, and James Y. Zou. Concrete autoencoders for differentiable feature selection and reconstruction. In *ICML*, 2019.

[20]Elias Chaibub Neto, J Christopher Bare, and Adam A Margolin. Simulation studies as designed experiments: the comparison of penalized regression models in the "large p, small n" setting. *PloS one*, 9(10):e107957, 2014.

[21]Mingyuan Wang and Adrian Barbu. Are screening methods useful in feature selection? an empirical study. *arXiv preprint arXiv:1809.05465*, 2018.

[22]Jundong Li, Kewei Cheng, Suhang Wang, et al. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2018.

[23]Borja Seijo-Pardo, Verónica Bolón-Canedo, and Amparo Alonso-Betanzos. Using data complexity measures for thresholding in feature selection rankers. In *Conference of the Spanish Association for Artificial Intelligence*, pages 121–131, 2016.

[24]Marcilio CP de Souto, Ana C Lorena, Newton Spolaôr, et al. Complexity measures of supervised classifications tasks: a case study for cancer gene expression data. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7.

[25]Y Chien. *Pattern classification and scene analysis*, volume 19. IEEE, 1974.

[26]José Martínez Sotoca, Ramón Alberto Mollineda, and José Salvador Sánchez. A meta-learning framework for pattern classication by means of data complexity measures. *Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial*, 10(29):31–38, 2006.

[27]Ana C Lorena, Ivan G Costa, Newton Spolaör, et al. Analysis of complexity indices for classification problems: Cancer gene expression data. *Neurocomputing*, 75(1):33–42, 2012.

[28]Hanchuan Peng, Fuhui Long, and C. Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[29]Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[30]Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *European conference on machine learning*, pages 171–182, 1994.

[31]Kenji Kira and Larry A Rendell. A practical approach to feature selection. In *Machine Learning Proceedings 1992*, pages 249–256. 1992.

[32]Deng Cai, Chiyuan Zhang, and Xiaofei He. Unsupervised feature selection for multi-cluster data. In *KDD*, pages 333–342, 2010.

[33]Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2002.

[34]Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[35]Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005.

[36]Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, et al. High-dimensional feature selection by feature-wise kernelized lasso. *Neural Computation*, 26(1):185–207, 2014.

[37]Arthur Gretton, Olivier Bousquet, Alex Smola, et al. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77, 2005.

[38]Miron Kursa and Witold Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software, Articles*, 36(11):1–13, 2010.

[39]Forrest W Young. *Multidimensional scaling: History, theory, and applications*. Psychology Press, 2013.

8 Supplementary Material

8.1 Number of selected features

We ran experiments over a wide range of number of selected features ranging from 1% to 19% of the total features. But BORUTA and HSIC-LASSO returned much less number of features even after parameter adjustments. the exact number of maximum number of features returned by these two methods and the maximum of 19% of the total features as returned by other methods is given in Table 3.

8.2 Statistical Significance Tests

In order to compare various methods based on their performance in two learning tasks, we performed Welch's paired t-tests and reported the corresponding t and p values. Tables 4 - 19 provide the results for all datasets on the two tasks. The test is conducted on each pair of method and the corresponding p values are given in brackets.

8.3 Detailed Plots for top performing methods

For each dataset, we present the line plots for the top 3 methods which have the highest average classification performance over our selected feature range. In all of the plots, red, blue, green lines represent the first, second and third best performing approach respectively. For each FS method, the middle line show the average score while the other thin, dashed lines present the 95% confidence intervals. Moreover, in term of classification performance, figure 16 provide additional insights into the number of selected features where the top-performing FS methods (i) get the highest performance (figure 16a, (ii) out-perform the baseline (figure 16c as well as the highest F1 score that they can get (figure 16b).

8.4 Complexity Plots

Figures 13-15 presents the complexity scores of the training and testing portion for each dataset over 5 splits (training/test split created as explained in the main paper). Those values are represented as a bar with (i) the points denote the highest and lowest values (ii) the green line exhibits the average score, (iii) the length of the bar is an relative approximation of the variance between different splits.

Dataset	HSIC-LASSO					BORUTA					others
	f1	f2	f3	f4	f5	f1	f2	f3	f4	f5	
PROSTATE-GE	23	23	20	24	24	145	202	158	184	168	1134
CNS	35	30	34	29	35	2	8	2	2	13	1355
LSVT	21	16	19	23	24	50	59	59	51	54	59
chin	54	53	48	51	45	456	370	415	428	298	4221
CLL	48	56	51	64	53	89	71	83	85	59	2155
colon	22	26	21	21	22	33	28	26	38	51	381
Car	170	178	219	196	192	538	546	515	533	498	1745
TOX	79	93	77	86	76	400	373	351	340	376	1093

Table 3. Maximum number of selected feature for each dataset. f1-f5 denotes the 5 training splits.

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff
7	CAE	NaN	-18.74(∼0)	-15.60(∼0)	-12.98(∼0)	-18.12(∼0)	-17.83(∼0)	-20.24(∼0)	-17.76(∼0)	-10.49(∼0)	-18.86(∼0)
2	MLP	18.74(∼0)	NaN	9.00(∼0)	5.41(∼0)	1.98(0.06)	4.14(∼0)	-6.87(∼0)	3.90(∼0)	6.90(∼0)	-0.06(0.96)
5	MCFS	15.60(∼0)	-9.00(∼0)	NaN	1.03(0.31)	-6.95(∼0)	-6.14(∼0)	-15.28(∼0)	-5.90(∼0)	3.26(∼0)	-9.56(∼0)
5	TSFS	12.98(∼0)	-5.41(∼0)	-1.03(0.31)	NaN	-4.54(∼0)	-3.97(∼0)	-7.50(∼0)	-3.93(∼0)	1.98(0.06)	-5.49(∼0)
3	boruta	18.12(∼0)	-1.98(0.06)	6.95(∼0)	4.54(∼0)	NaN	1.71(0.10)	-8.47(∼0)	1.63(0.11)	6.19(∼0)	-2.19(0.03)
4	enet	17.83(∼0)	-4.14(∼0)	6.14(∼0)	3.97(∼0)	-1.71(0.10)	NaN	-13.85(∼0)	0.01(0.99)	5.72(∼0)	-4.68(∼0)
1	hsic	20.24(∼0)	6.87(∼0)	15.28(∼0)	7.50(∼0)	8.47(∼0)	13.85(∼0)	NaN	12.15(∼0)	8.59(∼0)	8.39(∼0)
4	lasso	17.76(∼0)	-3.90(∼0)	5.90(∼0)	3.93(∼0)	-1.63(0.11)	-0.01(0.99)	-12.15(∼0)	NaN	5.69(∼0)	-4.34(∼0)
6	mRMR	10.49(∼0)	-6.90(∼0)	-3.26(∼0)	-1.98(0.06)	-6.19(∼0)	-5.72(∼0)	-8.59(∼0)	-5.69(∼0)	NaN	-6.97(∼0)
2	relieff	18.86(∼0)	0.06(0.96)	9.56(∼0)	5.49(∼0)	2.19(0.03)	4.68(∼0)	-8.39(∼0)	4.34(∼0)	6.97(∼0)	NaN

Table 4. Welch's test for classification performance between experimented FS methods for Carcinom dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff
4	CAE	NaN	12.44(∼0)	12.77(∼0)	11.91(∼0)	14.98(∼0)	12.54(∼0)	8.36(∼0)	12.29(∼0)	-0.91(0.37)	13.68(∼0)
1	MLP	-12.44(∼0)	NaN	-0.67(0.51)	-0.50(0.62)	1.85(0.07)	-0.10(0.92)	-9.58(∼0)	-0.31(0.76)	-14.66(∼0)	1.15(0.26)
2	MCFS	-12.77(∼0)	0.67(0.51)	NaN	0.11(0.91)	2.93(0.01)	0.57(0.57)	-11.00(∼0)	0.34(0.74)	-15.31(∼0)	1.99(0.05)
2	TSFS	-11.91(∼0)	0.50(0.62)	-0.11(0.91)	NaN	2.37(0.02)	0.40(0.69)	-8.62(∼0)	0.20(0.84)	-14.04(∼0)	1.65(0.11)
1	boruta	-14.98(∼0)	-1.85(0.07)	-2.93(0.01)	-2.37(0.02)	NaN	-2.02(0.05)	-16.23(∼0)	-2.23(0.03)	-17.86(∼0)	-0.61(0.55)
1	enet	-12.54(∼0)	0.10(0.92)	-0.57(0.57)	-0.40(0.69)	2.02(0.05)	NaN	-9.87(∼0)	-0.21(0.84)	-14.82(∼0)	1.29(0.21)
3	hsic	-8.36(∼0)	9.58(∼0)	11.00(∼0)	8.62(∼0)	16.23(∼0)	9.87(∼0)	NaN	9.39(∼0)	-10.96(∼0)	12.05(∼0)
2	lasso	-12.29(∼0)	0.31(0.76)	-0.34(0.74)	-0.20(0.84)	2.23(0.03)	0.21(0.84)	-9.39(∼0)	NaN	-14.54(∼0)	1.48(0.15)
4	mRMR	0.91(0.37)	14.66(∼0)	15.31(∼0)	14.04(∼0)	17.86(∼0)	14.82(∼0)	10.96(∼0)	14.54(∼0)	NaN	16.13(∼0)
1	relieff	-13.68(∼0)	-1.15(0.26)	-1.99(0.05)	-1.65(0.11)	0.61(0.55)	-1.29(0.21)	-12.05(∼0)	-1.48(0.15)	-16.13(∼0)	NaN

Table 5. Welch's test for reconstruction error between experimented FS methods for Carcinom dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
7	CAE	NaN	-12.87(∼0)	-13.08(∼0)	-13.10(∼0)	-14.13(∼0)	-13.20(∼0)	-15.30(∼0)	-13.37(∼0)	-0.56(0.58)	-14.54(∼0)	-13.29(∼0)
4	MLP	12.87(∼0)	NaN	0.17(0.86)	-1.67(0.10)	-4.17(∼0)	-0.67(0.51)	-9.60(∼0)	-1.55(0.13)	13.12(∼0)	-5.76(∼0)	-1.11(0.27)
5	MCFS	13.08(∼0)	-0.17(0.86)	NaN	-2.03(0.05)	-6.69(∼0)	-1.23(0.23)	-23.41(∼0)	-2.28(0.03)	13.37(∼0)	-9.25(∼0)	-1.81(0.08)
3	TSFS	13.10(∼0)	1.67(0.10)	2.03(0.05)	NaN	-1.11(0.28)	1.33(0.19)	-4.16(∼0)	0.55(0.58)	13.33(∼0)	-2.21(0.03)	0.97(0.34)
3	boruta	14.13(∼0)	4.17(∼0)	6.69(∼0)	1.11(0.28)	NaN	4.32(∼0)	-7.14(∼0)	2.57(0.01)	14.50(∼0)	-2.04(0.05)	3.52(∼0)
4	enet	13.20(∼0)	0.67(0.51)	1.23(0.23)	-1.33(0.19)	-4.32(∼0)	NaN	-12.37(∼0)	-1.10(0.28)	13.49(∼0)	-6.31(∼0)	-0.55(0.58)
1	hsic	15.30(∼0)	9.60(∼0)	23.41(∼0)	4.16(∼0)	7.14(∼0)	12.37(∼0)	NaN	8.22(∼0)	15.79(∼0)	4.41(∼0)	10.54(∼0)
4	lasso	13.37(∼0)	1.55(0.13)	2.28(0.03)	-0.55(0.58)	-2.57(0.01)	1.10(0.28)	-8.22(∼0)	NaN	13.67(∼0)	-4.25(∼0)	0.58(0.57)
7	mRMR	0.56(0.58)	-13.12(∼0)	-13.37(∼0)	-13.33(∼0)	-14.50(∼0)	-13.49(∼0)	-15.79(∼0)	-13.67(∼0)	NaN	-14.95(∼0)	-13.59(∼0)
2	relieff	14.54(∼0)	5.76(∼0)	9.25(∼0)	2.21(0.03)	2.04(0.05)	6.31(∼0)	-4.41(∼0)	4.25(∼0)	14.95(∼0)	NaN	5.41(∼0)
4	svm_rfe	13.29(∼0)	1.11(0.27)	1.81(0.08)	-0.97(0.34)	-3.52(∼0)	0.55(0.58)	-10.54(∼0)	-0.58(0.57)	13.59(∼0)	-5.41(∼0)	NaN

Table 6. Welch's test for classification performance between experimented FS methods for Chin dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
4	CAE	NaN	20.86(∼0)	20.62(∼0)	21.07(∼0)	16.55(∼0)	21.08(∼0)	-2.45(0.02)	21.21(∼0)	-1.41(0.17)	17.73(∼0)	18.75(∼0)
1	MLP	-20.86(∼0)	NaN	-0.54(0.59)	-0.79(0.44)	-6.22(∼0)	0.36(0.72)	-55.27(∼0)	0.19(0.85)	-21.00(∼0)	-2.72(0.01)	-3.59(∼0)
1	MCFS	-20.62(∼0)	0.54(0.59)	NaN	-0.20(0.84)	-5.77(∼0)	0.91(0.37)	-55.72(∼0)	0.75(0.46)	-20.78(∼0)	-2.28(0.03)	-3.09(∼0)
1	TSFS	-21.07(∼0)	0.79(0.44)	0.20(0.84)	NaN	-6.02(∼0)	1.18(0.24)	-66.93(∼0)	1.03(0.31)	-21.16(∼0)	-2.26(0.03)	-3.16(∼0)
3	boruta	-16.55(∼0)	6.22(∼0)	5.77(∼0)	6.02(∼0)	NaN	6.57(∼0)	-40.70(∼0)	6.59(∼0)	-17.05(∼0)	2.74(0.01)	2.86(0.01)
1	enet	-21.08(∼0)	-0.36(0.72)	-0.91(0.37)	-1.18(0.24)	-6.57(∼0)	NaN	-55.98(∼0)	-0.19(0.85)	-21.21(∼0)	-3.03(∼0)	-3.96(∼0)
5	hsic	2.45(0.02)	55.27(∼0)	55.72(∼0)	66.93(∼0)	40.70(∼0)	55.98(∼0)	NaN	59.73(∼0)	0.32(0.75)	37.65(∼0)	50.38(∼0)
1	lasso	-21.21(∼0)	-0.19(0.85)	-0.75(0.46)	-1.03(0.31)	-6.59(∼0)	0.19(0.85)	-59.73(∼0)	NaN	-21.32(∼0)	-2.94(0.01)	-3.91(∼0)
4	mRMR	1.41(0.17)	21.00(∼0)	20.78(∼0)	21.16(∼0)	17.05(∼0)	21.21(∼0)	-0.32(0.75)	21.32(∼0)	NaN	18.18(∼0)	19.06(∼0)
2	relieff	-17.73(∼0)	2.72(0.01)	2.28(0.03)	2.26(0.03)	-2.74(0.01)	3.03(∼0)	-37.65(∼0)	2.94(0.01)	-18.18(∼0)	NaN	-0.30(0.76)
2	svm_rfe	-18.75(∼0)	3.59(∼0)	3.09(∼0)	3.16(∼0)	-2.86(0.01)	3.96(∼0)	-50.38(∼0)	3.91(∼0)	-19.06(∼0)	0.30(0.76)	NaN

Table 7. Welch's test for reconstruction error between experimented FS methods for Chin dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
7	CAE	NaN	-27.26(∼0)	-17.77(∼0)	-16.87(∼0)	-34.53(∼0)	-27.42(∼0)	-34.05(∼0)	-24.80(∼0)	-3.76(∼0)	-27.21(∼0)	-17.12(∼0)
3	MLP	27.26(∼0)	NaN	8.31(∼0)	8.67(∼0)	-6.01(∼0)	-0.33(0.75)	-3.22(∼0)	3.96(∼0)	20.88(∼0)	0.84(0.41)	3.56(∼0)
5	MCFS	17.77(∼0)	-8.31(∼0)	NaN	0.56(0.58)	-14.23(∼0)	-8.55(∼0)	-12.62(∼0)	-5.22(∼0)	12.70(∼0)	-7.84(∼0)	-2.56(0.01)
5	TSFS	16.87(∼0)	-8.67(∼0)	-0.56(0.58)	NaN	-14.35(∼0)	-8.91(∼0)	-12.78(∼0)	-5.69(∼0)	11.97(∼0)	-8.22(∼0)	-2.98(0.01)
1	boruta	34.53(∼0)	6.01(∼0)	14.23(∼0)	14.35(∼0)	NaN	5.55(∼0)	6.88(∼0)	11.37(∼0)	26.63(∼0)	7.46(∼0)	7.15(∼0)
3	enet	27.42(∼0)	0.33(0.75)	8.55(∼0)	8.91(∼0)	-5.55(∼0)	NaN	-2.72(0.01)	4.27(∼0)	21.05(∼0)	1.17(0.25)	3.76(∼0)
2	hsic	34.05(∼0)	3.22(∼0)	12.62(∼0)	12.78(∼0)	-6.88(∼0)	2.72(0.01)	NaN	9.20(∼0)	25.73(∼0)	4.71(∼0)	5.60(∼0)
4	lasso	24.80(∼0)	-3.96(∼0)	5.22(∼0)	5.69(∼0)	-11.37(∼0)	-4.27(∼0)	-9.20(∼0)	NaN	18.48(∼0)	-3.26(∼0)	1.13(0.27)
6	mRMR	3.76(∼0)	-20.88(∼0)	-12.70(∼0)	-11.97(∼0)	-26.63(∼0)	-21.05(∼0)	-25.73(∼0)	-18.48(∼0)	NaN	-20.68(∼0)	-13.02(∼0)
3	relieff	27.21(∼0)	-0.84(0.41)	7.84(∼0)	8.22(∼0)	-7.46(∼0)	-1.17(0.25)	-4.71(∼0)	3.26(∼0)	20.68(∼0)	NaN	3.09(∼0)
4	svm_rfe	17.12(∼0)	-3.56(∼0)	2.56(0.01)	2.98(0.01)	-7.15(∼0)	-3.76(∼0)	-5.60(∼0)	-1.13(0.27)	13.02(∼0)	-3.09(∼0)	NaN

Table 8. Welch's test for classification performance between experimented FS methods for Cll-Sub-111 dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
3	CAE	NaN	10.79(∼0)	9.56(∼0)	9.43(∼0)	-4.82(∼0)	9.08(∼0)	-7.41(∼0)	9.83(∼0)	-6.80(∼0)	10.84(∼0)	8.11(∼0)
1	MLP	-10.79(∼0)	NaN	-1.55(0.13)	-1.16(0.26)	-23.77(∼0)	-0.64(0.53)	-34.21(∼0)	-0.73(0.47)	-14.95(∼0)	0.16(0.88)	-3.09(∼0)
1	MCFS	-9.56(∼0)	1.55(0.13)	NaN	0.28(0.78)	-21.43(∼0)	0.63(0.53)	-30.56(∼0)	0.72(0.48)	-14.10(∼0)	1.68(0.10)	-1.58(0.12)
1	TSFS	-9.43(∼0)	1.16(0.26)	-0.28(0.78)	NaN	-20.12(∼0)	0.36(0.72)	-27.46(∼0)	0.41(0.69)	-14.01(∼0)	1.29(0.21)	-1.77(0.09)
4	boruta	4.82(∼0)	23.77(∼0)	21.43(∼0)	20.12(∼0)	NaN	17.87(∼0)	-4.40(∼0)	20.93(∼0)	-4.48(∼0)	23.63(∼0)	18.44(∼0)
1	enet	-9.08(∼0)	0.64(0.53)	-0.63(0.53)	-0.36(0.72)	-17.87(∼0)	NaN	-23.02(∼0)	∼0(1.00)	-13.72(∼0)	0.76(0.45)	-1.95(0.06)
5	hsic	7.41(∼0)	34.21(∼0)	30.56(∼0)	27.46(∼0)	4.40(∼0)	23.02(∼0)	NaN	28.72(∼0)	-3.37(∼0)	33.69(∼0)	25.72(∼0)
1	lasso	-9.83(∼0)	0.73(0.47)	-0.72(0.48)	-0.41(0.69)	-20.93(∼0)	∼0(1.00)	-28.72(∼0)	NaN	-14.29(∼0)	0.87(0.39)	-2.20(0.03)
6	mRMR	6.80(∼0)	14.95(∼0)	14.10(∼0)	14.01(∼0)	4.48(∼0)	13.72(∼0)	3.37(∼0)	14.29(∼0)	NaN	14.99(∼0)	13.09(∼0)
1	relieff	-10.84(∼0)	-0.16(0.88)	-1.68(0.10)	-1.29(0.21)	-23.63(∼0)	-0.76(0.45)	-33.69(∼0)	-0.87(0.39)	-14.99(∼0)	NaN	-3.20(∼0)
2	svm_rfe	-8.11(∼0)	3.09(∼0)	1.58(0.12)	1.77(0.09)	-18.44(∼0)	1.95(0.06)	-25.72(∼0)	2.20(0.03)	-13.09(∼0)	3.20(∼0)	NaN

Table 9. Welch's test for reconstruction error between experimented FS methods for Cll-Sub-111 dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
9	CAE	NaN	-5.10(∼0)	-6.58(∼0)	-5.31(∼0)	1.32(0.20)	-9.64(∼0)	0.71(0.48)	-9.02(∼0)	-1.16(0.25)	-9.22(∼0)	-7.05(∼0)
4	MLP	5.10(∼0)	NaN	0.39(0.70)	1.08(0.29)	6.94(∼0)	-1.78(0.08)	6.59(∼0)	-0.45(0.66)	4.94(∼0)	-0.48(0.63)	0.66(0.51)
5	MCFS	6.58(∼0)	-0.39(0.70)	NaN	0.99(0.33)	11.58(∼0)	-3.25(∼0)	11.19(∼0)	-1.46(0.15)	7.32(∼0)	-1.55(0.13)	0.39(0.70)
7	TSFS	5.31(∼0)	-1.08(0.29)	-0.99(0.33)	NaN	8.94(∼0)	-4.03(∼0)	8.49(∼0)	-2.50(0.02)	5.50(∼0)	-2.60(0.01)	-0.77(0.45)
11	boruta	-1.32(0.20)	-6.94(∼0)	-11.58(∼0)	-8.94(∼0)	NaN	-16.72(∼0)	-3.01(∼0)	-22.18(∼0)	-4.63(∼0)	-24.29(∼0)	-15.34(∼0)
1	enet	9.64(∼0)	1.78(0.08)	3.25(∼0)	4.03(∼0)	16.72(∼0)	NaN	16.51(∼0)	2.54(0.02)	11.49(∼0)	2.55(0.02)	4.19(∼0)
10	hsic	-0.71(0.48)	-6.59(∼0)	-11.19(∼0)	-8.49(∼0)	3.01(∼0)	-16.51(∼0)	NaN	-23.00(∼0)	-3.79(∼0)	-25.71(∼0)	-15.23(∼0)
3	lasso	9.02(∼0)	0.45(0.66)	1.46(0.15)	2.50(0.02)	22.18(∼0)	-2.54(0.02)	23.00(∼0)	NaN	12.15(∼0)	-0.08(0.94)	2.39(0.02)
8	mRMR	1.16(0.25)	-4.94(∼0)	-7.32(∼0)	-5.50(∼0)	4.63(∼0)	-11.49(∼0)	3.79(∼0)	-12.15(∼0)	NaN	-12.75(∼0)	-8.63(∼0)
2	relieff	9.22(∼0)	0.48(0.63)	1.55(0.13)	2.60(0.01)	24.29(∼0)	-2.55(0.02)	25.71(∼0)	0.08(0.94)	12.75(∼0)	NaN	2.55(0.02)
6	svm_rfe	7.05(∼0)	-0.66(0.51)	-0.39(0.70)	0.77(0.45)	15.34(∼0)	-4.19(∼0)	15.23(∼0)	-2.39(0.02)	8.63(∼0)	-2.55(0.02)	NaN

Table 10. Welch's test for classification performance between experimented FS methods for CNS dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
4	CAE	NaN	11.65(∼0)	6.57(∼0)	8.32(∼0)	-26.40(∼0)	8.31(∼0)	-24.22(∼0)	11.08(∼0)	-9.63(∼0)	7.11(∼0)	7.27(∼0)
1	MLP	-11.65(∼0)	NaN	-1.94(0.06)	-1.75(0.09)	-38.96(∼0)	-1.34(0.19)	-50.54(∼0)	-0.53(0.60)	-14.38(∼0)	-2.81(0.01)	-2.08(0.04)
1	MCFS	-6.57(∼0)	1.94(0.06)	NaN	0.45(0.65)	-28.19(∼0)	0.70(0.49)	-24.83(∼0)	1.58(0.12)	-12.53(∼0)	-0.38(0.70)	0.06(0.95)
1	TSFS	-8.32(∼0)	1.75(0.09)	-0.45(0.65)	NaN	-32.88(∼0)	0.29(0.77)	-33.34(∼0)	1.31(0.20)	-13.27(∼0)	-0.96(0.34)	-0.42(0.67)
7	boruta	26.40(∼0)	38.96(∼0)	28.19(∼0)	32.88(∼0)	NaN	32.13(∼0)	14.33(∼0)	38.30(∼0)	3.04(∼0)	31.32(∼0)	30.51(∼0)
1	enet	-8.31(∼0)	1.34(0.19)	-0.70(0.49)	-0.29(0.77)	-32.13(∼0)	NaN	-31.52(∼0)	0.91(0.37)	-13.30(∼0)	-1.22(0.23)	-0.69(0.50)
5	hsic	24.22(∼0)	50.54(∼0)	24.83(∼0)	33.34(∼0)	-14.33(∼0)	31.52(∼0)	NaN	48.66(∼0)	-2.37(0.02)	30.56(∼0)	28.71(∼0)
1	lasso	-11.08(∼0)	0.53(0.60)	-1.58(0.12)	-1.31(0.20)	-38.30(∼0)	-0.91(0.37)	-48.66(∼0)	NaN	-14.18(∼0)	-2.37(0.02)	-1.67(0.10)
6	mRMR	9.63(∼0)	14.38(∼0)	12.53(∼0)	13.27(∼0)	-3.04(∼0)	13.30(∼0)	2.37(0.02)	14.18(∼0)	NaN	12.74(∼0)	12.85(∼0)
3	relieff	-7.11(∼0)	2.81(0.01)	0.38(0.70)	0.96(0.34)	-31.32(∼0)	1.22(0.23)	-30.56(∼0)	2.37(0.02)	-12.74(∼0)	NaN	0.49(0.63)
2	svm_rfe	-7.27(∼0)	2.08(0.04)	-0.06(0.95)	0.42(0.67)	-30.51(∼0)	0.69(0.50)	-28.71(∼0)	1.67(0.10)	-12.85(∼0)	-0.49(0.63)	NaN

Table 11. Welch's test for reconstruction error between experimented FS methods for CNS dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
6	CAE	NaN	3.56(∼0)	0.88(0.38)	-0.25(0.80)	-3.26(∼0)	-4.76(∼0)	2.96(0.01)	-3.16(∼0)	-2.16(0.04)	0.24(0.82)	1.51(0.14)
11	MLP	-3.56(∼0)	NaN	-3.20(∼0)	-4.22(∼0)	-12.26(∼0)	-15.34(∼0)	-2.22(0.03)	-11.51(∼0)	-6.10(∼0)	-4.60(∼0)	-3.07(∼0)
8	MCFS	-0.88(0.38)	3.20(∼0)	NaN	-1.22(0.23)	-5.29(∼0)	-7.19(∼0)	2.45(0.02)	-5.09(∼0)	-3.25(∼0)	-0.82(0.42)	0.67(0.51)
5	TSFS	0.25(0.80)	4.22(∼0)	1.22(0.23)	NaN	-3.19(∼0)	-4.81(∼0)	3.64(∼0)	-3.07(∼0)	-2.01(0.05)	0.55(0.58)	1.95(0.06)
2	boruta	3.26(∼0)	12.26(∼0)	5.29(∼0)	3.19(∼0)	NaN	-2.50(0.02)	14.06(∼0)	0.07(0.94)	0.35(0.73)	4.77(∼0)	7.32(∼0)
1	enet	4.76(∼0)	15.34(∼0)	7.19(∼0)	4.81(∼0)	2.50(0.02)	NaN	18.35(∼0)	2.46(0.02)	1.74(0.09)	6.85(∼0)	9.72(∼0)
10	hsic	-2.96(0.01)	2.22(0.03)	-2.45(0.02)	-3.64(∼0)	-14.06(∼0)	-18.35(∼0)	NaN	-12.73(∼0)	-5.69(∼0)	-4.02(∼0)	-2.20(0.03)
3	lasso	3.16(∼0)	11.51(∼0)	5.09(∼0)	3.07(∼0)	-0.07(0.94)	-2.46(0.02)	12.73(∼0)	NaN	0.31(0.76)	4.56(∼0)	6.97(∼0)
4	mRMR	2.16(0.04)	6.10(∼0)	3.25(∼0)	2.01(0.05)	-0.35(0.73)	-1.74(0.09)	5.69(∼0)	-0.31(0.76)	NaN	2.73(0.01)	4.04(∼0)
7	relieff	-0.24(0.82)	4.60(∼0)	0.82(0.42)	-0.55(0.58)	-4.77(∼0)	-6.85(∼0)	4.02(∼0)	-4.56(∼0)	-2.73(0.01)	NaN	1.67(0.10)
9	svm_rfe	-1.51(0.14)	3.07(∼0)	-0.67(0.51)	-1.95(0.06)	-7.32(∼0)	-9.72(∼0)	2.20(0.03)	-6.97(∼0)	-4.04(∼0)	-1.67(0.10)	NaN

Table 12. Welch's test for classification performance between experimented FS methods for Colon dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
4	CAE	NaN	6.30(∼0)	5.67(∼0)	3.66(∼0)	-3.29(∼0)	5.76(∼0)	-6.29(∼0)	6.90(∼0)	3.01(∼0)	2.14(0.04)	5.05(∼0)
1	MLP	-6.30(∼0)	NaN	-1.07(0.29)	-1.74(0.09)	-13.03(∼0)	-0.14(0.89)	-18.83(∼0)	0.43(0.67)	-4.33(∼0)	-1.90(0.07)	-0.70(0.49)
1	MCFS	-5.67(∼0)	1.07(0.29)	NaN	-0.98(0.33)	-12.90(∼0)	0.82(0.42)	-19.72(∼0)	1.60(0.12)	-3.48(∼0)	-1.36(0.18)	0.18(0.86)
2	TSFS	-3.66(∼0)	1.74(0.09)	0.98(0.33)	NaN	-7.49(∼0)	1.53(0.13)	-10.46(∼0)	2.12(0.04)	-1.48(0.15)	-0.60(0.56)	1.01(0.32)
5	boruta	3.29(∼0)	13.03(∼0)	12.90(∼0)	7.49(∼0)	NaN	11.30(∼0)	-5.61(∼0)	14.64(∼0)	9.04(∼0)	4.42(∼0)	9.94(∼0)
1	enet	-5.76(∼0)	0.14(0.89)	-0.82(0.42)	-1.53(0.13)	-11.30(∼0)	NaN	-15.73(∼0)	0.53(0.60)	-3.75(∼0)	-1.76(0.09)	-0.53(0.60)
6	hsic	6.29(∼0)	18.83(∼0)	19.72(∼0)	10.46(∼0)	5.61(∼0)	15.73(∼0)	NaN	21.72(∼0)	15.16(∼0)	6.11(∼0)	13.78(∼0)
1	lasso	-6.90(∼0)	-0.43(0.67)	-1.60(0.12)	-2.12(0.04)	-14.64(∼0)	-0.53(0.60)	-21.72(∼0)	NaN	-5.07(∼0)	-2.16(0.04)	-1.10(0.28)
3	mRMR	-3.01(∼0)	4.33(∼0)	3.48(∼0)	1.48(0.15)	-9.04(∼0)	3.75(∼0)	-15.16(∼0)	5.07(∼0)	NaN	0.38(0.71)	2.95(0.01)
2	relieff	-2.14(0.04)	1.90(0.07)	1.36(0.18)	0.60(0.56)	-4.42(∼0)	1.76(0.09)	-6.11(∼0)	2.16(0.04)	-0.38(0.71)	NaN	1.38(0.18)
1	svm_rfe	-5.05(∼0)	0.70(0.49)	-0.18(0.86)	-1.01(0.32)	-9.94(∼0)	0.53(0.60)	-13.78(∼0)	1.10(0.28)	-2.95(0.01)	-1.38(0.18)	NaN

Table 13. Welch's test for reconstruction error between experimented FS methods for Colon dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
11	CAE	NaN	-12.56(∼0)	-6.68(∼0)	-1.14(0.26)	-10.01(∼0)	-16.50(∼0)	-12.51(∼0)	-14.84(∼0)	-3.98(∼0)	-15.21(∼0)	-8.02(∼0)
4	MLP	12.56(∼0)	NaN	3.52(∼0)	9.27(∼0)	1.72(0.09)	-4.79(∼0)	2.12(0.04)	-2.05(0.05)	10.75(∼0)	-3.01(∼0)	2.67(0.01)
8	MCFS	6.68(∼0)	-3.52(∼0)	NaN	4.92(∼0)	-2.00(0.05)	-6.50(∼0)	-2.63(0.01)	-4.92(∼0)	3.95(∼0)	-5.46(∼0)	-0.89(0.38)
10	TSFS	1.14(0.26)	-9.27(∼0)	-4.92(∼0)	NaN	-7.45(∼0)	-12.25(∼0)	-8.83(∼0)	-10.84(∼0)	-2.13(0.04)	-11.25(∼0)	-5.99(∼0)
6	boruta	10.01(∼0)	-1.72(0.09)	2.00(0.05)	7.45(∼0)	NaN	-5.62(∼0)	-0.37(0.72)	-3.50(∼0)	7.55(∼0)	-4.22(∼0)	1.10(0.28)
1	enet	16.50(∼0)	4.79(∼0)	6.50(∼0)	12.25(∼0)	5.62(∼0)	NaN	9.40(∼0)	3.57(∼0)	16.22(∼0)	1.98(0.05)	5.92(∼0)
5	hsic	12.51(∼0)	-2.12(0.04)	2.63(0.01)	8.83(∼0)	0.37(0.72)	-9.40(∼0)	NaN	-5.77(∼0)	10.92(∼0)	-6.63(∼0)	1.64(0.11)
3	lasso	14.84(∼0)	2.05(0.05)	4.92(∼0)	10.84(∼0)	3.50(∼0)	-3.57(∼0)	5.77(∼0)	NaN	14.03(∼0)	-1.34(0.19)	4.17(∼0)
9	mRMR	3.98(∼0)	-10.75(∼0)	-3.95(∼0)	2.13(0.04)	-7.55(∼0)	-16.22(∼0)	-10.92(∼0)	-14.03(∼0)	NaN	-14.40(∼0)	-5.36(∼0)
2	relieff	15.21(∼0)	3.01(∼0)	5.46(∼0)	11.25(∼0)	4.22(∼0)	-1.98(0.05)	6.63(∼0)	1.34(0.19)	14.40(∼0)	NaN	4.78(∼0)
7	svm_rfe	8.02(∼0)	-2.67(0.01)	0.89(0.38)	5.99(∼0)	-1.10(0.28)	-5.92(∼0)	-1.64(0.11)	-4.17(∼0)	5.36(∼0)	-4.78(∼0)	NaN

Table 14. Welch's test for classification performance between experimented FS methods for Lsvt dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
1	CAE	NaN	-3.19(∼0)	-0.86(0.40)	0.87(0.39)	-0.44(0.66)	-2.40(0.02)	-17.75(∼0)	-2.25(0.03)	0.74(0.46)	-2.54(0.02)	-6.61(∼0)
3	MLP	3.19(∼0)	NaN	2.78(0.01)	3.41(∼0)	3.07(∼0)	0.96(0.34)	-1.71(0.10)	1.13(0.27)	3.41(∼0)	1.69(0.10)	0.61(0.55)
1	MCFS	0.86(0.40)	-2.78(0.01)	NaN	1.46(0.15)	0.53(0.60)	-1.92(0.06)	-13.70(∼0)	-1.75(0.09)	1.41(0.17)	-1.78(0.08)	-5.16(∼0)
1	TSFS	-0.87(0.39)	-3.41(∼0)	-1.46(0.15)	NaN	-1.20(0.24)	-2.69(0.01)	-11.20(∼0)	-2.55(0.02)	-0.21(0.84)	-2.77(0.01)	-5.54(∼0)
1	boruta	0.44(0.66)	-3.07(∼0)	-0.53(0.60)	1.20(0.24)	NaN	-2.26(0.03)	-20.60(∼0)	-2.09(0.04)	1.13(0.27)	-2.35(0.02)	-6.73(∼0)
2	enet	2.40(0.02)	-0.96(0.34)	1.92(0.06)	2.69(0.01)	2.26(0.03)	NaN	-3.61(∼0)	0.18(0.86)	2.68(0.01)	0.68(0.50)	-0.70(0.49)
5	hsic	17.75(∼0)	1.71(0.10)	13.70(∼0)	11.20(∼0)	20.60(∼0)	3.61(∼0)	NaN	4.01(∼0)	13.43(∼0)	7.35(∼0)	7.84(∼0)
2	lasso	2.25(0.03)	-1.13(0.27)	1.75(0.09)	2.55(0.02)	2.09(0.04)	-0.18(0.86)	-4.01(∼0)	NaN	2.54(0.02)	0.48(0.63)	-0.96(0.34)
1	mRMR	-0.74(0.46)	-3.41(∼0)	-1.41(0.17)	0.21(0.84)	-1.13(0.27)	-2.68(0.01)	-13.43(∼0)	-2.54(0.02)	NaN	-2.84(0.01)	-6.14(∼0)
2	relieff	2.54(0.02)	-1.69(0.10)	1.78(0.08)	2.77(0.01)	2.35(0.02)	-0.68(0.50)	-7.35(∼0)	-0.48(0.63)	2.84(0.01)	NaN	-2.20(0.03)
4	svm_rfe	6.61(∼0)	-0.61(0.55)	5.16(∼0)	5.54(∼0)	6.73(∼0)	0.70(0.49)	-7.84(∼0)	0.96(0.34)	6.14(∼0)	2.20(0.03)	NaN

Table 15. Welch's test for reconstruction error between experimented FS methods for Lsvt dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
11	CAE	NaN	-29.55(∼0)	-25.60(∼0)	-24.18(∼0)	-32.35(∼0)	-20.53(∼0)	-32.72(∼0)	-23.69(∼0)	-20.87(∼0)	-29.45(∼0)	-16.30(∼0)
2	MLP	29.55(∼0)	NaN	4.32(∼0)	8.80(∼0)	-0.59(0.56)	5.39(∼0)	2.04(0.05)	7.01(∼0)	8.44(∼0)	0.89(0.38)	9.45(∼0)
5	MCFS	25.60(∼0)	-4.32(∼0)	NaN	3.76(∼0)	-5.61(∼0)	1.97(0.06)	-3.98(∼0)	2.55(0.02)	4.34(∼0)	-3.63(∼0)	6.09(∼0)
8	TSFS	24.18(∼0)	-8.80(∼0)	-3.76(∼0)	NaN	-11.41(∼0)	-0.79(0.43)	-10.81(∼0)	-0.97(0.34)	1.30(0.20)	-8.23(∼0)	3.69(∼0)
1	boruta	32.35(∼0)	0.59(0.56)	5.61(∼0)	11.41(∼0)	NaN	6.28(∼0)	4.40(∼0)	8.76(∼0)	10.05(∼0)	1.71(0.10)	10.60(∼0)
6	enet	20.53(∼0)	-5.39(∼0)	-1.97(0.06)	0.79(0.43)	-6.28(∼0)	NaN	-5.06(∼0)	0.05(0.96)	1.69(0.10)	-4.85(∼0)	3.65(∼0)
4	hsic	32.72(∼0)	-2.04(0.05)	3.98(∼0)	10.81(∼0)	-4.40(∼0)	5.06(∼0)	NaN	7.58(∼0)	9.03(∼0)	-0.89(0.38)	9.69(∼0)
7	lasso	23.69(∼0)	-7.01(∼0)	-2.55(0.02)	0.97(0.34)	-8.76(∼0)	-0.05(0.96)	-7.58(∼0)	NaN	2.01(0.05)	-6.41(∼0)	4.17(∼0)
9	mRMR	20.87(∼0)	-8.44(∼0)	-4.34(∼0)	-1.30(0.20)	-10.05(∼0)	-1.69(0.10)	-9.03(∼0)	-2.01(0.05)	NaN	-7.91(∼0)	2.38(0.02)
3	relieff	29.45(∼0)	-0.89(0.38)	3.63(∼0)	8.23(∼0)	-1.71(0.10)	4.85(∼0)	0.89(0.38)	6.41(∼0)	7.91(∼0)	NaN	9.01(∼0)
10	svm_rfe	16.30(∼0)	-9.45(∼0)	-6.09(∼0)	-3.69(∼0)	-10.60(∼0)	-3.65(∼0)	-9.69(∼0)	-4.17(∼0)	-2.38(0.02)	-9.01(∼0)	NaN

Table 16. Welch's test for classification performance between experimented FS methods for Prostate-GE dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
4	CAE	NaN	11.81(∼0)	10.58(∼0)	3.06(∼0)	9.33(∼0)	5.19(∼0)	-85.44(∼0)	7.06(∼0)	-14.29(∼0)	12.86(∼0)	10.39(∼0)
1	MLP	-11.81(∼0)	NaN	-0.21(0.84)	-2.25(0.03)	-3.49(∼0)	-1.85(0.07)	-50.57(∼0)	-1.32(0.20)	-16.61(∼0)	-0.11(0.91)	-0.52(0.61)
1	MCFS	-10.58(∼0)	0.21(0.84)	NaN	-2.09(0.04)	-3.05(∼0)	-1.65(0.11)	-45.40(∼0)	-1.10(0.28)	-16.48(∼0)	0.11(0.91)	-0.30(0.77)
2	TSFS	-3.06(∼0)	2.25(0.03)	2.09(0.04)	NaN	0.50(0.62)	0.69(0.49)	-19.32(∼0)	1.24(0.22)	-14.31(∼0)	2.22(0.03)	1.92(0.06)
3	boruta	-9.33(∼0)	3.49(∼0)	3.05(∼0)	-0.50(0.62)	NaN	0.41(0.69)	-58.22(∼0)	1.31(0.20)	-15.85(∼0)	3.61(∼0)	2.75(0.01)
1	enet	-5.19(∼0)	1.85(0.07)	1.65(0.11)	-0.69(0.49)	-0.41(0.69)	NaN	-27.22(∼0)	0.62(0.54)	-15.31(∼0)	1.82(0.08)	1.44(0.16)
5	hsic	85.44(∼0)	50.57(∼0)	45.40(∼0)	19.32(∼0)	58.22(∼0)	27.22(∼0)	NaN	33.60(∼0)	-6.75(∼0)	56.72(∼0)	46.03(∼0)
1	lasso	-7.06(∼0)	1.32(0.20)	1.10(0.28)	-1.24(0.22)	-1.31(0.20)	-0.62(0.54)	-33.60(∼0)	NaN	-15.83(∼0)	1.27(0.21)	0.85(0.40)
6	mRMR	14.29(∼0)	16.61(∼0)	16.48(∼0)	14.31(∼0)	15.85(∼0)	15.31(∼0)	6.75(∼0)	15.83(∼0)	NaN	16.65(∼0)	16.41(∼0)
1	relieff	-12.86(∼0)	0.11(0.91)	-0.11(0.91)	-2.22(0.03)	-3.61(∼0)	-1.82(0.08)	-56.72(∼0)	-1.27(0.21)	-16.65(∼0)	NaN	-0.44(0.66)
1	svm_rfe	-10.39(∼0)	0.52(0.61)	0.30(0.77)	-1.92(0.06)	-2.75(0.01)	-1.44(0.16)	-46.03(∼0)	-0.85(0.40)	-16.41(∼0)	0.44(0.66)	NaN

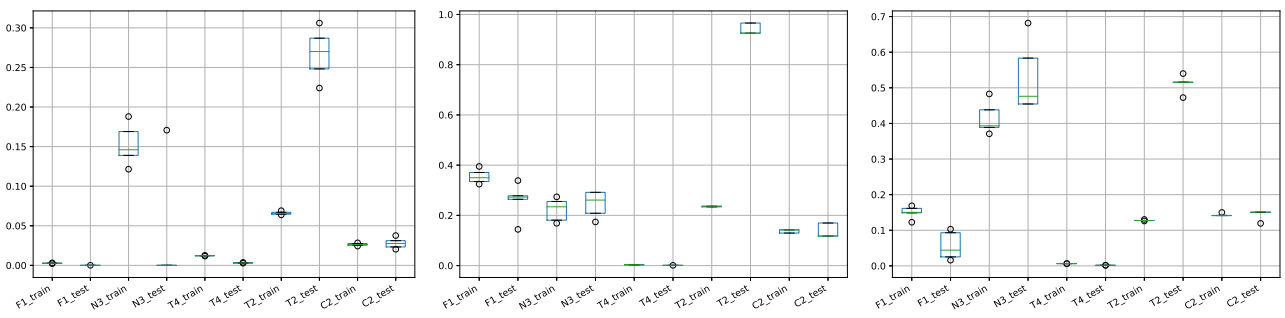
Table 17. Welch's test for reconstruction error between experimented FS methods for Prostate-GE dataset

rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
9	CAE	NaN	6.61(∼0)	4.77(∼0)	7.75(∼0)	7.58(∼0)	4.57(∼0)	3.46(∼0)	5.58(∼0)	-5.32(∼0)	6.41(∼0)	3.82(∼0)
2	MLP	-6.61(∼0)	NaN	-2.66(0.01)	1.67(0.10)	1.56(0.13)	-1.55(0.13)	-6.40(∼0)	-1.21(0.23)	-9.43(∼0)	-0.34(0.74)	-5.37(∼0)
6	MCFS	-4.77(∼0)	2.66(0.01)	NaN	4.40(∼0)	4.16(∼0)	0.46(0.65)	-2.81(0.01)	1.28(0.21)	-8.42(∼0)	2.34(0.02)	-2.03(0.05)
1	TSFS	-7.75(∼0)	-1.67(0.10)	-4.40(∼0)	NaN	∼0(1.0)	-2.75(0.01)	-9.26(∼0)	-2.78(0.01)	-10.01(∼0)	-2.03(0.05)	-7.90(∼0)
1	boruta	-7.58(∼0)	-1.56(0.13)	-4.16(∼0)	∼0(1.0)	NaN	-2.66(0.01)	-8.44(∼0)	-2.63(0.01)	-9.95(∼0)	-1.90(0.07)	-7.27(∼0)
5	enet	-4.57(∼0)	1.55(0.13)	-0.46(0.65)	2.75(0.01)	2.66(0.01)	NaN	-2.55(0.02)	0.56(0.58)	-8.29(∼0)	1.31(0.20)	-2.00(0.05)
8	hsic	-3.46(∼0)	6.40(∼0)	2.81(0.01)	9.26(∼0)	8.44(∼0)	2.55(0.02)	NaN	4.23(∼0)	-7.72(∼0)	6.00(∼0)	0.88(0.38)
4	lasso	-5.58(∼0)	1.21(0.23)	-1.28(0.21)	2.78(0.01)	2.63(0.01)	-0.56(0.58)	-4.23(∼0)	NaN	-8.88(∼0)	0.91(0.37)	-3.43(∼0)
10	mRMR	5.32(∼0)	9.43(∼0)	8.42(∼0)	10.01(∼0)	9.95(∼0)	8.29(∼0)	7.72(∼0)	8.88(∼0)	NaN	9.32(∼0)	7.91(∼0)
3	relieff	-6.41(∼0)	0.34(0.74)	-2.34(0.02)	2.03(0.05)	1.90(0.07)	-1.31(0.20)	-6.00(∼0)	-0.91(0.37)	-9.32(∼0)	NaN	-4.99(∼0)
7	svm_rfe	-3.82(∼0)	5.37(∼0)	2.03(0.05)	7.90(∼0)	7.27(∼0)	2.00(0.05)	-0.88(0.38)	3.43(∼0)	-7.91(∼0)	4.99(∼0)	NaN

Table 18. Welch's test for classification performance between experimented FS methods for Tox-171 dataset

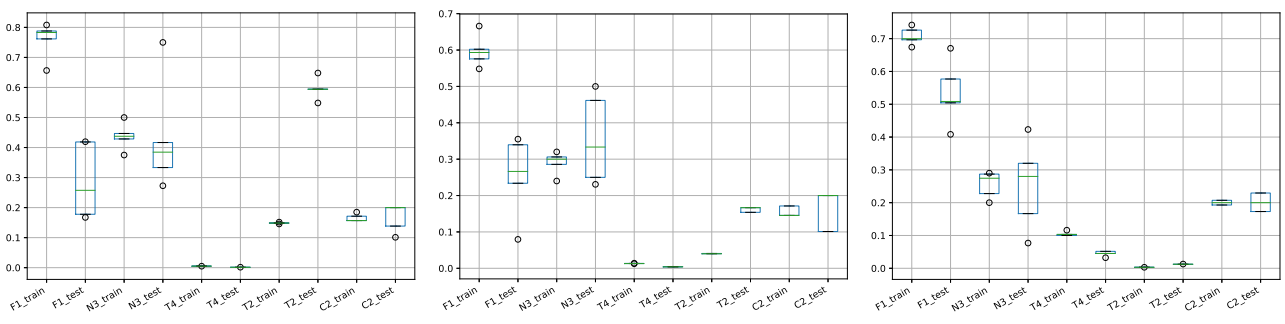
rank	method	CAE	MLP	MCFS	TSFS	boruta	enet	hsic	lasso	mRMR	relieff	svm_rfe
7	CAE	NaN	6.61(∼0)	4.77(∼0)	7.75(∼0)	7.58(∼0)	4.57(∼0)	3.46(∼0)	5.58(∼0)	-5.32(∼0)	6.41(∼0)	3.82(∼0)
1	MLP	-6.61(∼0)	NaN	-2.66(0.01)	1.67(0.10)	1.56(0.13)	-1.55(0.13)	-6.40(∼0)	-1.21(0.23)	-9.43(∼0)	-0.34(0.74)	-5.37(∼0)
4	MCFS	-4.77(∼0)	2.66(0.01)	NaN	4.40(∼0)	4.16(∼0)	0.46(0.65)	-2.81(0.01)	1.28(0.21)	-8.42(∼0)	2.34(0.02)	-2.03(0.05)
1	TSFS	-7.75(∼0)	-1.67(0.10)	-4.40(∼0)	NaN	∼0(1.00)	-2.75(0.01)	-9.26(∼0)	-2.78(0.01)	-10.01(∼0)	-2.03(0.05)	-7.90(∼0)
1	boruta	-7.58(∼0)	-1.56(0.13)	-4.16(∼0)	∼0(1.00)	NaN	-2.66(0.01)	-8.44(∼0)	-2.63(0.01)	-9.95(∼0)	-1.90(0.07)	-7.27(∼0)
3	enet	-4.57(∼0)	1.55(0.13)	-0.46(0.65)	2.75(0.01)	2.66(0.01)	NaN	-2.55(0.02)	0.56(0.58)	-8.29(∼0)	1.31(0.20)	-2.00(0.05)
6	hsic	-3.46(∼0)	6.40(∼0)	2.81(0.01)	9.26(∼0)	8.44(∼0)	2.55(0.02)	NaN	4.23(∼0)	-7.72(∼0)	6.00(∼0)	0.88(0.38)
3	lasso	-5.58(∼0)	1.21(0.23)	-1.28(0.21)	2.78(0.01)	2.63(0.01)	-0.56(0.58)	-4.23(∼0)	NaN	-8.88(∼0)	0.91(0.37)	-3.43(∼0)
8	mRMR	5.32(∼0)	9.43(∼0)	8.42(∼0)	10.01(∼0)	9.95(∼0)	8.29(∼0)	7.72(∼0)	8.88(∼0)	NaN	9.32(∼0)	7.91(∼0)
2	relieff	-6.41(∼0)	0.34(0.74)	-2.34(0.02)	2.03(0.05)	1.90(0.07)	-1.31(0.20)	-6.00(∼0)	-0.91(0.37)	-9.32(∼0)	NaN	-4.99(∼0)
5	svm_rfe	-3.82(∼0)	5.37(∼0)	2.03(0.05)	7.90(∼0)	7.27(∼0)	2.00(0.05)	-0.88(0.38)	3.43(∼0)	-7.91(∼0)	4.99(∼0)	NaN

Table 19. Welch's paired t-test for reconstruction error between experimented FS methods for Tox-171 dataset



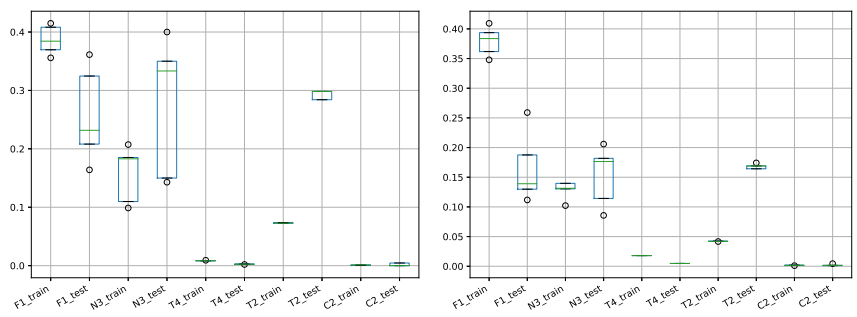
(a) CARCINOM (b) CHIN (c) CLL-SUB-111

Fig. 10: Complexity plots for the three datasets: CARCINOM , CHIN , and CLL-SUB-111



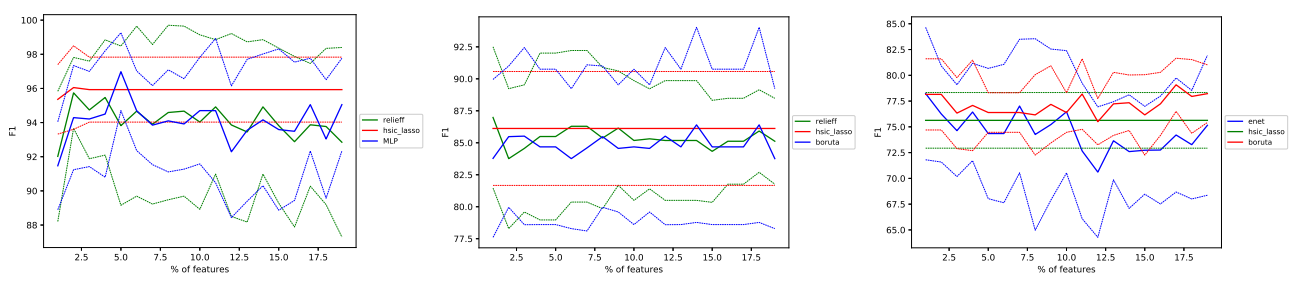
(a) CNS (b) COLON (c) LSVT

Fig. 11: Complexity plots for the three datasets: CNS , COLON , and LSVT



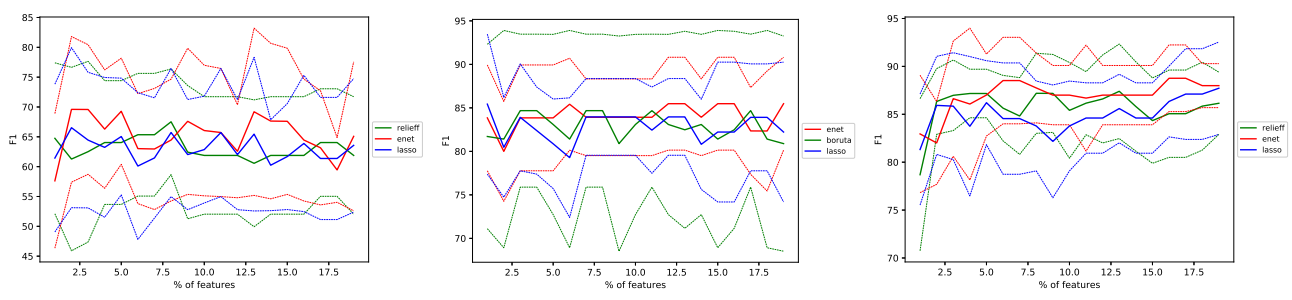
(a) PROSTATE-GE (b) TOX-171

Fig. 12: Complexity plots for the two datasets: PROSTATE-GE and TOX-171



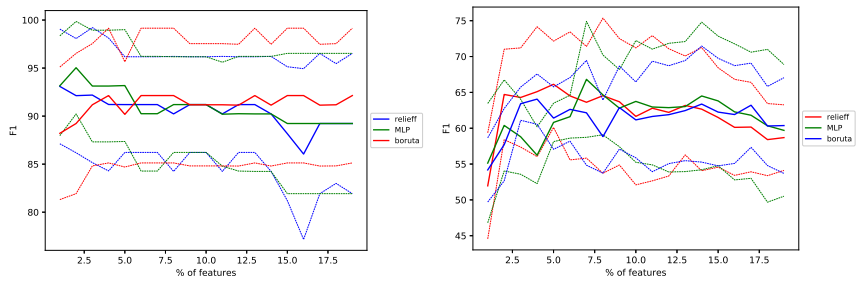
(a) CARCINOM (b) CHIN (c) CLL-SUB-111

Fig. 13: Performance for top 3 performing methods for CARCINOM , CHIN , and CLL-SUB-111



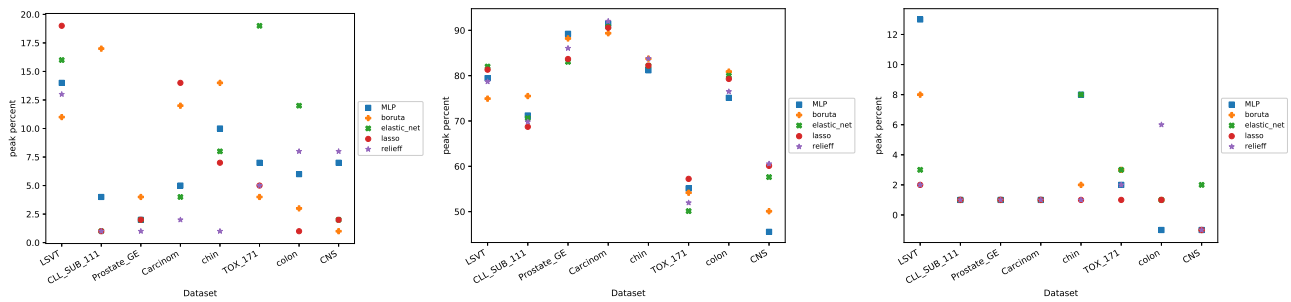
(a) CNS (b) COLON (c) LSVT

Fig. 14: Performance for top 3 performing methods for CNS , COLON , and LSVT



(a) PROSTATE-GE (b) TOX-171

Fig. 15: Performance for top 3 performing methods for PROSTATE-GE and TOX-171



(a) % reach highest F1

(b) Max F1 score

(c) % F1 score above baseline

Fig. 16: Y-axis represents the % of selected features at which the listed methods (a) achieve their best performance for classification (highest F_1 score) (c) when the method first achieves better performance for classification as compared to using all features. Plot (b) shows the maximum Classification performance (F_1 score in percentage) by these methods.