Information Sciences 676 (2024) 120818

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Explainable feature selection and ensemble classification via feature polarity

Peng Zhou [a], Ji Liang [a], Yuanting Yan [a,*], Shu Zhao [a], Xindong Wu [b]

[a] Key Laboratory of Intelligent Computing and Signal Processing, Ministry of Education, School of Computer Science and Technology, Anhui University, China

[b] Key Laboratory of Knowledge Engineering with Big Data, Ministry of Education, Hefei University of Technology, China

ARTICLE INFO

*Keywords:*
Feature selection
Explainable machine learning
Feature polarity
Ensemble learning

ABSTRACT

Feature selection aims to choose the most relevant features from the dataset that can enhance the performance and efficiency of machine learning models. Although feature selection has been studied for many years, most existing methods focus on accuracy and efficiency while neglecting the interpretability of selected features. Therefore, inspired by the "Yin-Yang" philosophy, we introduce the concept of feature polarity for the first time and divide the features into positive and negative features. For example, by analyzing a patient's symptoms (features), we can obtain two sets of features to explain whether the patient has the flu. Positive features help us determine if the patient has the flu, while negative features can help us rule out the possibility of the flu. We introduce the PN (Positive and Negative) coefficient to measure the polarity of candidate features and develop a novel and explainable feature selection method based on feature polarity. Furthermore, we propose an ensemble classification framework that leverages both positive and negative features for each class to improve classification performance. Extensive experiments demonstrate the effectiveness of the PN coefficient compared to other information measurements. Moreover, our proposed classification framework performs excellently compared to some state-of-the-art feature selection methods.

## 1. Introduction

Feature selection is a critical task in data mining which aids in selecting the most relevant features from a dataset to enhance the performance and efficiency of machine learning models [28]. Feature selection lies in reducing data dimensionality, lowering model complexity, avoiding overfitting, and improving model generalization ability, thereby achieving better performance, less running time, and better interpretability [38]. For example, feature selection can improve classifier accuracy in classification tasks, reduce misjudgment and omission rates, and improve classification effectiveness and efficiency [4].

Generally, feature selection methods include filter, wrapper, and embedded [28]. Filter-based methods independently evaluate and select features before model training, with commonly used evaluation metrics including information gain, mutual information [43], etc. Wrapper-based methods combine feature selection and model training, evaluating the importance of features through

* Corresponding author.
*E-mail addresses:* doodzhou@ahu.edu.cn (P. Zhou), liangji@stu.ahu.edu.cn (J. Liang), ytyan@ahu.edu.cn (Y. Yan), zhaoshuzs@ahu.edu.cn (S. Zhao), xwu@hfut.edu.cn (X. Wu).

https://doi.org/10.1016/j.ins.2024.120818
Received 27 August 2023; Received in revised form 25 March 2024; Accepted 29 May 2024

Available online 3 June 2024

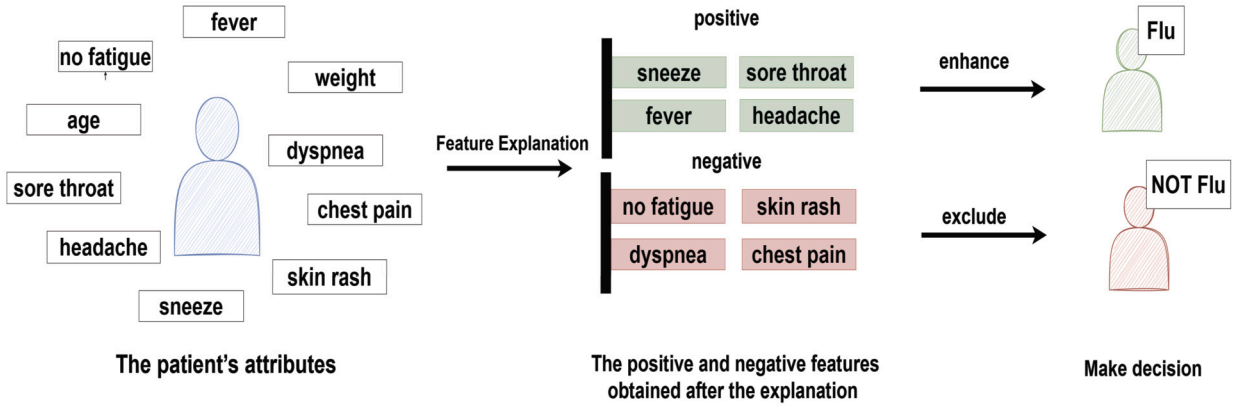



**Fig. 1.** An example to illustrate the feature polarity. By analyzing and interpreting the symptoms or features of a patient, we obtain two sets of features to explain whether the patient has the flu. The green features (positive features) help determine the patient has the flu, while the red features (negative features) can help us rule out the possibility of the patient having the flu.

different strategies [30]. Embedded methods integrate feature selection and model training, such as constraining model parameters through regularization methods while selecting the optimal feature subset [29]. In recent years, ensemble feature selection methods have emerged as a new trend [8]. Their advantage lies in combining different feature selection algorithms and strategies, leveraging the strengths of different algorithms, and reducing uncertainty in feature selection. In addition, ensemble methods can improve stability to a certain extent, but the specific effect may depend on the choice of algorithms and the characteristics of the domain. This has practical implications for researchers and practitioners, as it can help them better understand and apply ensemble feature selection methods [36]. Ensemble feature selection methods also have played an essential role in robust biomarker identification for cancer diagnosis, especially in high-dimensional data analysis and microarray data classification [9]. Employing ensemble feature selection techniques can significantly enhance the robustness of classifiers in biomarker discovery and improve classification performance simultaneously [2]. The study of feature selection has spanned several decades, and numerous methods have been proposed. However, existing feature selection methods primarily focus on issues such as accuracy, efficiency, scalability, while neglecting the interpretability of features.

Due to the extensive development and application of machine learning technology, explainable machine learning has received increasing attention [31]. An explainable system is a model that enables a user to observe and study the mathematical transformation of input data into output data to understand how the system works [41]. Entrusting essential decisions to a system that cannot explain itself will bring unpredictable risks [34]. In feature selection tasks, the interpretability of features holds immense importance. Firstly, it allows us to understand the model's prediction results, enabling us to evaluate its credibility. Moreover, feature interpretability is essential for comprehending the working mechanism of the model. In domains like healthcare, it enables doctors to grasp the logic behind the model's diagnosis, thereby enhancing guidance in clinical practice. Therefore, the comprehensibility of features plays a pivotal role in model credibility. It enables us to gain a deeper insight into the model and utilize it effectively to enhance both its efficiency and performance [7].

Inspired by the "Yin-Yang" philosophy in China and the work of Ribeiro et al. [37], we believe that features can also be divided into "Yin" and "Yang" categories. Specifically, a positive feature (polarity "Yang") indicates that the feature contributed to a specific model prediction result for a specific label. In contrast, the negative feature (polarity "Yin") indicates that the feature denies a specific model prediction result. Feature polarity helps us understand the degree and direction of the impact of features on model prediction results, thereby better carrying out feature selection and optimization. Fig. 1 is an example to illustrate the feature polarity. A series of symptoms can be identified through examination for a patient, including sneezing, fever, headache, fatigue, and dyspnea. By analyzing and interpreting the features of a patient, we can obtain two sets of features. The green features (positive features) help determine the patient has the flu, while the red features (negative features) can help us rule out the possibility of the patient having the flu (We do not know what disease he had, but it was not the flu). Feature selection can be more interpretable by analyzing both positive and negative features. However, existing feature selection methods rarely consider the polarity of features, resulting in feature subsets that may have strong classification effects but lack interpretability.

The motivation and significance of this paper are in three aspects. (1) Although feature selection methods have been studied for many years, only few works focus on the interpretability of selected features. Besides accuracy and efficiency, understanding the underlying reasons behind selecting specific features can significantly enhance the models' trustworthiness, transparency, and applicability in real-world scenarios. Therefore, this paper aims to propose an explainable feature selection method. (2) This paper first gives the formal definition of feature polarity (Positive and Negative features) that is inspired by the Chinese philosophy of "Yin-Yang." By analyzing and selecting positive and negative features, a better understanding of their impact on model prediction results can be achieved, thereby enhancing the interpretability of feature selection. (3) Since we select both positive and negative features for each class from each dataset, existing classical classifiers can not be applied to obtain the classification performance directly. Therefore, inspired by the superiority of Ensemble learning, we propose an ensemble classification framework that can

enhance the performance by training multiple classifiers. By combining ensemble learning with both positive and negative selected features, we can improve classification performance and provide better interpretability.

Motivated by this, we propose the PN coefficient that utilizes the Kendall coefficient [10], which is a non-parametric method that is unaffected by data distribution and outliers. To assess the polarity of candidate features in a particular classification, we calculate the PN coefficient for these features. This allows us to identify positive and negative features without having to go through complex calculations. Furthermore, we propose an ensemble classification framework that employs separate subsets of positive and negative features. By training multiple classifiers using different feature subsets, we combine their classification results using ensemble strategies. This paper makes the following main contributions:

- Inspired by the "Yin-Yang" philosophy, we first propose the concept of feature polarity and present the formal definition of positive and negative features. The feature polarity can enhance human understanding and increase the interpretability and credibility of the model based on positive and negative features.
- To measure the polarity of features, we propose the PN coefficient, which can distingue the positive and negative features. Then, We present a new method for feature selection, founded on the polarity of features. Meanwhile, we design a new strategy to determine the proportion of positive and negative features automatically. Besides the performance of the selected features, our proposed method can explain them from the polarity perspective.
- Based on the selected positive and negative features, we propose a new ensemble classification framework combining multiple classifiers from both positive and negative perspectives to improve classification performance.
- Extensive experiments have been conducted between the PN coefficient and other information measurements to validate the effectiveness of both positive and negative features. Meanwhile, our new proposed classification framework indicates good performance contrasted to eight state-of-the-art feature selection methods, making it widely applicable in practical applications with interpretability.

The rest of this article is organized as follows. Section 2 describes related work. Section 3 presents the proposed method. Section 4 gives the experimental analysis, and Section 5 gives a brief conclusion.

## 2. Related work

### 2.1. Feature selection method

Feature selection has been a subject of study for numerous years and has yielded a multitude of impressive algorithms. These algorithms are primarily categorized into three groups: Filter, Wrapper, and Embedded methods [28].

#### 2.1.1. Filter methods

Filter feature selection methods are employed to choose highly correlated features. Usually, a specific metric is employed to gauge the correlation between features and target variables. For instance, Mahdieh et al. [27] presented a feature selection approach for text classification that encompasses the principles of minimum redundancy and maximum relevance. Emrah et al. [19] have introduced a novel filtering criterion that departs from employing mutual redundancy and instead focuses on selecting the top-ranked features based on ReliefF and Fisher scores. Meanwhile, Zuo et al. [48] have proposed a feature selection technique based on curvature, which assesses the significance of each feature by measuring the curvature of the data manifold. By evaluating the curvature of the features to determine their importance, the most discriminative features are chosen for classification. Zhang et al. [46] have developed a swift feature selection algorithm that employs orthogonal least squares for linear classification. The goal of this algorithm is to improve the classification model's performance by choosing the most relevant features from a provided dataset. He et al. [24] introduced a label enhancement algorithm designed to incorporate latent sample correlations into the label enhancement process. Conventional discrete labels are transformed into label distributions, enhancing their ability to capture the intricate relationship between samples and labels. Furthermore, You et al. [45] proposed an algorithm specialized in learning local causal structures for streaming features that aims to handle the dynamic alterations within the feature space. Hashemi et al. [20] proposed a method that models the feature selection problem as a multi-criteria decision-making process and solves it using information fusion.

#### 2.1.2. Wrapper methods

Wrapper feature selection methods adopt the feature selection model's classification accuracy as the machine learning benchmark. The fundamental concept is to treat the feature selection issue as an optimization problem for the classifier. This involves assessing various feature subsets through iterative training and testing. Specifically, Babak et al. [33] introduced a wrapper feature selection algorithm based on the forest optimization approach. This algorithm integrates region selection concepts and reduces classification errors in most scenarios. Vasilii et al. [4] developed an innovative semi-supervised wrapper feature selection framework using a self-learning algorithm for pseudo-labeling unlabeled samples. This algorithm considers feature weights and progressively eliminates irrelevant features. Tarkhane et al. [39] leverage the wrapper technique, incorporating an improved binary differential evolution algorithm for feature selection. This method can be implemented during the data preprocessing phase to enhance the performance and efficiency of machine learning algorithms. Meanwhile, Liu et al. [30] proposed an accelerated wrapper-based feature selection approach that leverages recursive elimination and distributed computing strategies. Thakkar et al. [40] investigated the impact of various feature selection techniques, such as Chi-Square, Information Gain, and Recursive Feature Elimination, in conjunction with

different classifiers on the performance of intrusion detection systems. Guney et al. [16] proposed an algorithm that integrates feature selection and classifier optimization for constructing an accurate and efficient network intrusion detection system. This algorithm enhances the optimization effectiveness and resolves the interdependency issue between feature selection and classifier optimization by incorporating feature selection during the classifier optimization process. Guyon et al. [18] utilized gene expression data obtained through DNA microarray technology and proposed a Support Vector Machine (SVM) method based on Recursive Feature Elimination (RFE) for selecting a gene subset associated with cancer. They experimentally demonstrated that their method outperformed in cancer classification performance and showed biological relevance to cancer.

#### *2.1.3. Embedded methods*

Embedded methods in feature selection involve integrating the feature selection process directly into the model, enabling simultaneous feature selection and model training. This approach utilizes models with built-in feature selection functions, such as Lasso, Ridge, Elastic Net, etc. These models penalize features to achieve the desired feature selection effect. More specifically, R Xu et al. [44] have proposed a dynamic feature selection algorithm that utilizes Q-learning to assess the importance of each individual feature, enabling the identification and selection of the most pivotal one. In a different study, Hashemi et al. [21] have introduced a highly efficient feature selection algorithm, employing a Pareto-based approach, ultimately improving the accuracy of multi-label classification models. Nie et al. [32] proposed an unsupervised feature selection framework called Fast Sparse Discriminative K-Means (FSDK) to address the issue of continuous pseudo-label matrix deviating from reality in the embedded feature selection process. Guney et al. [17] proposed a robust ensemble feature selection (EFS) technique that utilizes a support vector classifier to assign weights to features and employs the minimum weight threshold method to handle outliers in the ranked feature lists. This method significantly improves gene selection stability while maintaining classification performance and reducing computational complexity.

Ensemble learning is a prolific field in machine learning and has been commonly employed for classification. Ensemble learning can also be applied to feature selection. Specifically, [8] provided the reader with the basic concepts necessary to build an ensemble for feature selection, reviewing the up-to-date advances and commenting on the future trends still to be faced. Bania et al. [6] proposed an ensemble feature selection algorithm named R-GEFS based on graph theory and feature rank aggregation to address the issue of feature redundancy in the feature selection process. Hashemi et al. [22] proposed an ant colony optimization algorithm, Ant-MCDM, based on the multi-criteria decision-making method for solving complex combinatorial optimization problems. The algorithm considers multiple heuristic methods as criteria and employs a multi-criteria decision-making approach to select the best node. Hashemi et al. [23] proposed an online feature selection algorithm called NSOFS, which models the online feature selection process as a multi-objective optimization problem. In this algorithm, the importance of features is measured by a set of multiple feature evaluation metrics. By utilizing multi-objective optimization and Pareto dominance, efficient and accurate feature selection is achieved.

Most existing feature selection methods aim to select features that are “as good as possible” according to some measurements or strategies. However, most existing feature selection algorithms do not consider interpretability. Besides good performance, the interpretability of features is also very important for users to trust the constructed model. Therefore, this paper focuses on the issue of interpretable feature selection.

### *2.2. Explainable machine learning*

Explainable machine learning refers to the process and techniques used to interpret and comprehend the predictions made by a model. The objective is to enhance the trustworthiness and credibility of the model, enabling humans to gain a better understanding of its predictions [15]. In certain domains, such as medicine, interpreting the features is vital for accurate diagnosis and effective treatment of diseases. Explainable machine learning methods can be classified differently from different perspectives. In this paper, the research on explainable machine learning primarily concentrates on the following aspects:

#### *2.2.1. Local interpretability*

Local interpretability involves explaining individual predictions and focuses on understanding how the model generates predictions for specific inputs. By gaining insights into model’s decision-making process, the behavior of the model can be better comprehended. For example, Ribeiro et al. [37] present LIME as a method for local interpretability, which can explain the process behind predicting outcomes for a single sample.

#### *2.2.2. Global interpretability*

Global interpretability seeks to elucidate the entire machine learning model, encompassing more than just individual predictions. It focuses on comprehending the structure, parameters, decision-making mechanism of the entire model, as well as the transformation and prediction of inputs. Commonly employed techniques for global interpretability encompass feature importance analysis and decision tree visualization. For example, Lundberg et al. [42] introduced SHAP as a global interpretability method, which can provide insights into how the model obtains predictions for all samples.

#### *2.2.3. Visualization*

Visualizing the model’s decision-making process is an effective way for humans to understand its prediction results better. Two commonly used methods for this purpose are t-SNE [25] and UMAP [5]. t-SNE is a nonlinear method that enables mapping high-dimensional data to a lower-dimensional space while preserving the local data structure. Similarly, UMAP utilizes manifold learning

and stochastic gradient descent to map high-dimensional data points to coordinates in a lower-dimensional space by optimizing an objective function.

In general, explainable machine learning is essential to get insight into their internal mechanism. Nevertheless, only a limited number of studies emphasize the interpretability of features. Considering the Interpretable feature selection method can help us better understand the model's prediction results, thereby improving the reliability and credibility of feature selection. Therefore, this paper focuses on the interpretability of features from the perspective of feature polarity and validates its effectiveness in classification tasks. In general, explainable machine learning is essential to get insight into their internal mechanism. Nevertheless, few works focus on the interpretability of features by the feature selection methods. Considering the interpretability of features in feature selection can help us better understand the model's prediction results, thereby improving the reliability and credibility of feature selection. Therefore, this paper focuses on the interpretability of features from the perspective of feature polarity and validates the effectiveness of both positive and negative features.

## 3. The proposed method

In this part, we first present the formal definition of the PN coefficient and, the positive and negative features. Then, we present a case study to illustrate the feature polarity calculation and the features' interpretability. Based on this, we designed a new feature selection method to select positive and negative features while removing redundant features. Besides, we propose a novel ensemble classification framework combining multiple classifiers from both positive and negative perspectives to improve classification performance. Finally, the analysis of time complexity of our method is provided.

### 3.1. The definition of feature polarity

Inspired by the "Yin-Yang" philosophy, we believe that features can also be divided into "Yin" and "Yang" categories. Specifically, a positive feature (polarity "Yang") indicates that the feature contributed to a specific model prediction result for a specific label. In contrast, the negative feature (polarity "Yin") indicates that the feature denies a specific model prediction result. In order to determine and measure the polarity of features, we need to define a specific computable measurement.

#### 3.1.1. The Kendall coefficient

Kendall coefficient, also known as Kendall's tau coefficient, is a statistical measure used to determine the association between two variables [1]. It measures the ordinal association between two variables, which means it measures the degree of similarity between the rankings of two variables. The Kendall coefficient ranges from -1 to 1, where -1 indicates a perfect negative association, 0 indicates no association, and 1 indicates a perfect positive association.

Let $(x_1, y_1), \ldots, (x_n, y_n)$ be a set of random variables $X$ and $Y$. For each pair of $(x_i, y_i)$ and $(x_j, y_j)$, where $i < j$ are said to be concordant if the sort of $(x_i, y_i)$ and $(x_j, y_j)$ agrees: that is, if either both $x_i > x_j$ and $y_i > y_j$ holds or both $x_i < x_j$ and $y_i < y_j$, $(x_i, y_i)$ and $(x_j, y_j)$ are said to be concordant. Otherwise, they are said to be discordant. A pair $(x_i, y_i), (x_j, y_j)$ is said to be tied if and only if $x_i = x_j$ or $y_i = y_j$. A tied pair is neither concordant nor discordant.

The Kendall $\tau$ coefficient is defined as:

$$\tau = \frac{N_c - N_d}{N_p} = 1 - \frac{2N_p}{\binom{n}{2}}, \tag{1}$$

where $N_c$ is the number of concordant pairs, $N_d$ is number of discordant pairs, and $N_P$ is number of pairs.

Actually, there are three forms of Kendall's coefficient, which include tau-a, tau-b and tau-c;

- **Kendall's tau-a:** it is used when there are no ties in the data. It measures the strength of association between two variables by comparing the number of concordant and discordant pairs, defined as equation (1).
- **Kendall's tau-b:**: it is used when there are ties in the data. It takes into account the number of tied pairs and adjusts the calculation accordingly, defined as equation (2).

$$\tau_B = \frac{N_c - N_d}{\sqrt{\left(N_0 - N_1\right)\left(N_0 - N_2\right)}} \tag{2}$$

  where $N_0 = n(n-1)/2$, $N_1 = \sum_i t_i(t_i - 1)/2$, $N_2 = \sum_j u_i(u_i - 1)/2$, $t_i$ is the number of tied values in the $i^{th}$ group of ties for the first quantity group, and $u_j$ is the number of tied values in the $j^{th}$ group of ties for the second quantity.
- **Kendall's tau-c:** this form is a modification of tau-b that adjusts for ties in both variables. It is used when both variables have ties in the data, defined as equation (3).

$$\tau_C = \frac{2\left(N_c - N_d\right)}{N^2 \frac{(M-1)}{M}} \tag{3}$$

  where $M$ is $min(r, c)$ in which $r$ is the number of rows and $c$ is the number of columns.

A toy example of calculating the Kendall coefficient is as follows. Suppose there are two continuous value vectors $A = [1,3,2,4]^T$ and $B = [3,2,4,1]^T$.

**Step 1:** Compute the number of concordant pairs $N_c$ and discordant pairs $N_d$. This example shows four elements in each vector, with six ranking pairs. We represent these ranking pairs as $(A_i, B_j)$, where $A_i$ is an element from list $A$, and $B_j$ is an element from list $B$. In this example, the ranking pairs are as follows:

$$1.\{(A_1,B_1),(A_2,B_2)\} \rightarrow \{(1,3),(3,2)\}$$
$$2.\{(A_1,B_1),(A_3,B_3)\} \rightarrow \{(1,3),(2,4)\}$$
$$3.\{(A_1,B_1),(A_4,B_4)\} \rightarrow \{(1,3),(4,1)\}$$
$$4.\{(A_2,B_2),(A_3,B_3)\} \rightarrow \{(3,2),(2,4)\}$$
$$5.\{(A_2,B_2),(A_4,B_4)\} \rightarrow \{(3,2),(4,1)\}$$
$$6.\{(A_3,B_3),(A_4,B_4)\} \rightarrow \{(1,3),(4,1)\}$$

In ranking pair 1, we see $A_1 < A_2$ and $B_1 > B_2$. Therefore, the pair is said to be discordant. In ranking pair 2, we see $A_1 < A_3$ and $B_1 < B_3$. Thus, the pair is said to be concordant. Finally, we can get $N_c = 1,\ N_d = 5$.

**Step 2:** Compute the number of ties in ranking pairs. We need to calculate the number of pairing elements with the same values (referred to as ties) in vector $A$ and vector $B$. Because there are no duplicate elements in both A and B, the number of ties in A and B is 0. Thus, $N_1 = 0,\ N_2 = 0$.

**Step 3:** Based on the above results, in this toy example, the Kendall coefficient can be calculated by forms of tau-a or tau-b as follows:

$$\tau = \frac{N_c - N_d}{N_p} = \frac{1-5}{6} = -0.66$$

If $A$ and $B$ are two discrete variables, assuming $A = [d,a,c,b]^T$ and $B = [x,y,z,w]^T$. First, for the values of $A$ and $B$, we can assign a rank to each discrete value. For example, we can map $\{a,b,c,d\}$ to ranks $\{1,2,3,4\}$, and $\{w,x,y,z\}$ to ranks $\{1,2,3,4\}$. This way, we obtain the rank sequences of $A$ and $B$ as $A' = [4,1,3,2]^T$ and $B' = [2,3,4,1]^T$. Then, we can use the steps above to calculate the Kendall coefficient between $A'$ and $B'$.

These three types of Kendall correlation coefficients can be used to measure the ordinal relationship between variables and the consistency of rankings and assist in analyzing the correlation between features and the ranking of variables. The Kendall coefficient is a non-parametric statistic that does not make assumptions about the data distribution. Therefore, the Kendall coefficient is highly adaptable when dealing with various data types in real-world applications. In feature selection tasks, the ordinal consistency of features is crucial in determining their importance. By calculating the Kendall coefficient, we can obtain relative order information between features, thereby determining their correlations and polarities.

### *3.1.2. The PN coefficient*

Inspired by the Kendall coefficient, the PN coefficient is defined as follows:

**Definition 1. PN coefficient** Let $(x_1,y_1),\ldots,(x_n,y_n)$ be a set of random variables $X$ and $Y$.

$$PN(X,Y) = \begin{cases} 1 + \frac{2}{n(n-1)} \sum sgn(x_i - x_j) sgn(y_i - y_j), & i < j < n \\ 1, & \text{Otherwise} \end{cases} \tag{4}$$

where $sgn(\lambda)$ equals 1 if $\lambda > 0$, $sgn(\lambda)$ equals 1 if $\lambda > 0$, and $sgn(\lambda)$ equals 0 if $\lambda = 0$. In other words,

$$sgn(x_i - x_j) sgn(y_i - y_j) = \begin{cases} 1, & (x_i,y_i) \text{ and } (x_j,y_j) \text{ are concordant} \\ -1, & (x_i,y_i) \text{ and } (x_j,y_j) \text{ are disconcordant} \\ 0, & \text{Otherwise} \end{cases}$$

If the pairs of random variables $X$ and $Y$ exist tied pairs. The PN coefficient is calculated as follows:

$$PN(X,Y) = \begin{cases} 1 + \dfrac{\sum sgn(x_i - x_j) sgn(y_i - y_j)}{\sqrt{(N_0 - N_x)(N_0 - N_y)}}, & i < j < n \\ 1, & \text{Otherwise} \end{cases} \tag{5}$$

where $N_0$ is the number of pairs, $N_x$ is the tied number in $X$, and $N_y$ is the tied number in $y$.

The PN coefficients were transformed based on the Kendall coefficient, and the calculation method can be referred to the example of Kendall coefficient calculation in the previous section.

**Theorem 1.** $0 \le PN(X,Y) \le 2$

**Proof.** Consider the case where $i < j < n$. In terms of Definition 1, it can be deduced as:

$$sgn(x_i - x_j)sgn(y_i - y_j) = \begin{cases} 1, & x_i < x_j, y_i < y_j \text{ or } x_i > x_j, y_i > y_j \\ -1, & x_i < x_j, y_i > y_j \text{ or } x_i > x_j, y_i < y_j \\ 0, & \text{Otherwise} \end{cases} \tag{6}$$

There are a total of $\binom{n}{2}$ possible point pairs, where $\binom{n}{2} = \frac{n(n-1)}{2}$ is the binomial coefficient for the number of ways to choose two items from n items. When all the pair combinations are said to be concordant the sort order of $(x_i, x_j)$ and $(y_i, y_j)$, we can obtain the maximum value of the formula $\frac{n(n-1)}{2}$. According to Definition 1 it has the max value 2. Similarly, the formula can achieve a minimum value of 0. Thus, $0 \le PN(X, Y) \le 2$. □

The PN coefficient reflects the relationship between two random variables. If two features are irrelevant or completely random, the value of the PN coefficient is 1. A larger PN coefficient (>1) indicates a stronger positive polarity, while a smaller coefficient (<1) indicates a negative polarity. For example, in the case of a cold (Fig. 2), we calculated the PN coefficients of different features for influenza. The PN coefficient for fever and influenza is 1.6325, while the PN coefficient for skin allergy and influenza is 0.2929. When we use the PN coefficient to measure the relationship between a conditional feature and a specific class label, it reflects the polarity of the feature.

In terms of the PN coefficient, we give the definitions of positive feature and negative feature as follows:

**Definition 2. Positive Feature**

Given a condition feature $f = [x_1, x_2, ..., x_n]^T$ and the decision attribute $d = [y_1, y_2, ..., y_n]^T$. We consider $f$ is a positive feature to $d$, if

$$PN(f, d) > 1 \tag{7}$$

**Definition 3. Negative Feature**

Given a condition feature $f = [x_1, x_2, ..., x_n]^T$ and the decision attribute $d = [y_1, y_2, ..., y_n]^T$. We consider $f$ is a negative feature to $d$, if

$$PN(f, d) < 1 \tag{8}$$

Specifically, positive features contribute to the positive correlation ($y_i = 1$) between condition feature $f$ and decision attribute $d$. In contrast, negative features contribute to the negative correlation ($y_i \neq 1$). For example, if a patient has symptoms such as fever, cough, and sneezing, they are more likely to have a cold. These symptoms can be seen as positive features of a cold, and they play a promoting role in the diagnosis of a cold. Meanwhile, if a patient does not have feelings of fatigue, skin allergies, or chest pain, it can help us to exclude the possibility of the patient having a cold. These symptoms can be seen as negative features of a cold, and they play a role in excluding the diagnosis of a cold.

### *3.2. The feature selection algorithm*

In this segment, a new feature selection algorithm is proposed, that considering the polarity of features and selects positive and negative features. Both positive and negative features are helpful to classification task. For example, when judging whether a patient has a cold. If the patient has symptoms such as fever and cough, which are positive features, the occurrence of these features can effectively help us predict the classification result "cold". If the patient does not feel tired, which is a negative feature, we can exclude the possibility of a cold (although we do not know what disease he had).

For our new method, it selects both positive and negative features simultaneously. Therefore, to determine the ratio of positive and negative features automatically, we utilize PN value of features to define the PN Ratio as follows:

**Definition 4. PN Ratio** The PN Ratio is the ratio of the number of selected positive features to the total number of selected features.

$$PN\,Ratio = \frac{\left|\sum_{i=1}^{n} P_i - n\right| \times n}{\left|\sum_{i=1}^{n} P_i - n\right| \times n + \left|\sum_{j=1}^{m} N_j - m\right| \times m} \tag{9}$$

where $n$ is the number of positive features, $m$ is the number of negative features, $P_i$ is the PN value of the $i^{th}$ positive feature, and $N_j$ is the PN value of $j^{th}$ negative feature.

By utilizing Equation (9), the PN Ratio can be automatically calculated through the employment of PN values and the feature count. This formula can dynamically compute the proportion of positive to negative features based on varying datasets. As a result, it enables the determination of the quantity of positive and negative features while maintaining a fixed feature number.

Suppose the number of selected features is set to $K$. Here we define the calculation formula for the number of positive and negative features as follows:

$$NUM_p = \lfloor PNRatio \times K \rfloor \tag{10}$$

$$NUM_N = K - NUM_p \tag{11}$$

where $NUM_p$ is the number of selected positive features and $NUM_n$ is the number of selected negative features.

Furthermore, considering the issue of feature redundancy, we have incorporated feature interactions and mutual information to determine redundancy among features into our feature selection approach [47]. Suppose $f_1$ and $f_2$ are two condition features, $d$ is the decision feature (class attribute), the interaction value between $f_1$ and $f_2$ on $d$ as:

$$Int_D(f_1, f_2) = I(d; \{f_1, f_2\}) - I(f_1; d) - I(f_2; d) \tag{12}$$

where $I(\cdot, \cdot)$ indicates the mutual information between two features.

If $Int_D(f_1, f_2) < 0$, there is a redundancy between $f_1$ and $f_2$ on $d$. Then, we pick the one with the greater PN coefficient from $f_1$ and $f_2$. This method of determining feature redundancy has the advantage of not requiring parameter settings or thresholds.

In conclusion, we propose an interpretable feature selection method named PNFS(Algorithm 1), considering the polarity of features, and divide the positive and negative features by PN value. Positive features reflect the correlation of features to labels, while negative features, unlike irrelevant features, play an exclusionary role in the classification process.

**Algorithm 1** Positive and Negative Feature Selection.

**Input:**
$F$: the condition feature set;
$d$: the decision feature;
$K$: the number of selected features;
**Output:**
$S$: the subset of selected feature;

1: **Initialization:** Set $P = \{\}$, $N = \{\}$;
2: **For** each feature $f_i$ in $F$
3: **If** $PN(f_i, d) > 1$
4: **For** each feature $f_p$ in $P$
5: **If** $Int_D(f_i, f_p) < 0$ and $PN(f_i, d) < PN(f_p, d)$
6: Discard $f_i$;
7: **Else** $P = P \cup \{f_i\}$;
8: **End If**
9: **End For**
10: **End If**
11: **If** $PN(f_i, d) < 1$
12: **For** each feature $f_n$ in $N$
13: **If** $Int_D(f_i, f_n) < 0$ and $PN(f_n, d) < PN(f_i, d)$
14: Discard $f_i$;
15: **Else** $N = N \cup \{f_i\}$;
16: **End If**
17: **End For**
18: **End If**
19: **End For**
20: $Ratio_{PN}$: calculate the PN Ratio;
21: $S$: select the features with the largest PN coefficient from P and N in terms of $Ratio_{PN}$ and $K$;
22: **Output** $S$.

Specifically, for each feature $f_i$ in $F$, we first calculate the PN coefficient of $f_i$. In steps 3 to 9, if $PN(f_i, d) > 1$, we consider $f_i$ as a positive feature and check whether $f_i$ is redundant to the selected feature $f_p$ in the positive feature subset $P$. Similarly, in steps 10 to 17, $f_i$ is considered a negative feature, and the algorithm will check if feature redundancy exists in negative feature subset $N$. Step 20 calculates the PN Ratio in terms of the selected positive and negative feature subsets. In Step 21, we first rank the features in the positive candidate feature subset ($P$) and the negative candidate feature subset ($N$). We select features with the largest PN coefficient from these two subsets in terms of $Ratio_{PN}$ and $K$. Finally, we output the final selected feature subset $S$.

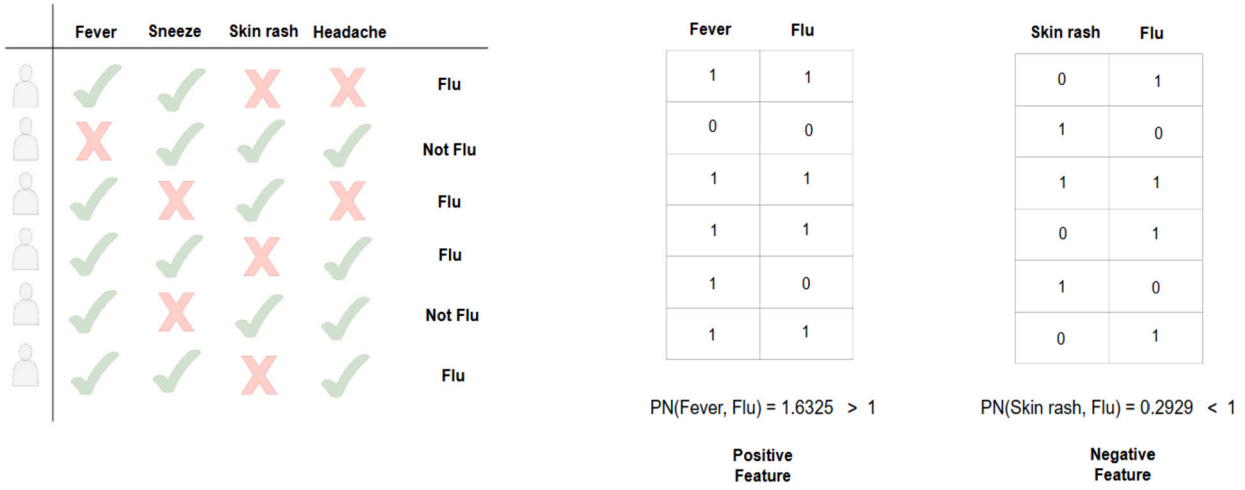



**Fig. 2.** An example of the PN coefficient and PN features. We use the flu example to explain the PN coefficient and our proposed positive and negative features. By calculating the PN coefficient, we determine the positive feature "fever" ($PN(fever, Flu) > 1$) and the negative feature "skin rash" ($PN(Skinrash, Flu) < 1$) to the decision class "flu". In other words, fever can help us diagnose the flu, while skin rash can help us exclude influenza, which is consistent with the logic of disease diagnosis.

### 3.3. A case study of real-world dataset

In this section, we first use a toy example (as shown in Fig. 2) to illustrate the calculation of the PN coefficient. Then, we apply the PN coefficient and PNFS feature selection method on a real-world dataset.

To calculate the $PN(fever, Flu)$, we first obtain two vectors respectively in Fig. 2:

$$Fever = [1,0,1,1,1,1]^T,\ Flu = [1,0,1,1,0,1]^T$$

Each vector has six elements, so there are 15 ranking pairs. We represent these ranking pairs as $(x_i, y_j)$, where $x_i$ is an element from vector $Fever$, and $y_j$ is an element from vector $Flu$. The ranking pairs are shown as follows:

$1.\{(x_1,y_1),(x_2,y_2)\} \to \{(1,1),(0,0)\}; 2.\{(x_1,y_1),(x_3,y_3)\} \to \{(1,1),(1,1)\}$
$3.\{(x_1,y_1),(x_4,y_4)\} \to \{(1,1),(1,1)\}; 4.\{(x_1,y_1),(x_5,y_5)\} \to \{(1,1),(1,0)\}$
$5.\{(x_1,y_1),(x_6,y_6)\} \to \{(1,1),(1,1)\}; 6.\{(x_2,y_2),(x_3,y_3)\} \to \{(0,0),(1,1)\}$
$7.\{(x_2,y_2),(x_4,y_4)\} \to \{(0,0),(1,1)\}; 8.\{(x_2,y_2),(x_5,y_5)\} \to \{(0,0),(1,0)\}$
$9.\{(x_2,y_2),(x_6,y_6)\} \to \{(0,0),(1,1)\}; 10.\{(x_3,y_3),(x_4,y_4)\} \to \{(1,1),(1,1)\}$
$11.\{(x_3,y_3),(x_5,y_5)\} \to \{(1,1),(1,0)\}; 12.\{(x_3,y_3),(x_6,y_6)\} \to \{(1,1),(1,1)\}$
$13.\{(x_4,y_4),(x_5,y_5)\} \to \{(1,1),(1,0)\}; 14.\{(x_4,y_4),(x_6,y_6)\} \to \{(1,1),(1,1)\}$
$15.\{(x_5,y_5),(x_6,y_6)\} \to \{(1,0),(1,1)\}$

Then, we can get the number of concordant pairs $N_c = 4$, the number of discordant pairs $N_d = 0$, the number of tied pairs $N_x = 10$, and the number of tied pairs $N_y = 7$. Therefore,

$$\sum sgn(x_i - x_j)sgn(y_i - y_j) = 1+0+0+0+0+1+1+0 \\ +1+0+0+0+0+0+0 = 4$$

Thus, according to the definition of PN coefficient, the $PN(Fever, Flu)$ can be calculated as follows:

$$PN(Fever, Flu) = 1 + \frac{4}{\sqrt{(15-10)(15-7)}} = 1.6325$$

Similarly, the $PN(Skinrash, Flu) = 0.2929$.

In practical applications, the interpretability of feature selection is very important. On the one hand, selected features can reflect the essential characteristics of the data, thereby improving the accuracy and generalization ability of the model. On the other hand, interpretable features can help humans understand the decision-making process of the model, thereby enhancing the credibility and acceptability of the model.

To demonstrate the interpretability of the PN coefficient, we apply it to the German Credit dataset from UCI, a low-dimensional dataset with 20 features, as an example. This dataset categorizes individuals based on attribute descriptions into good or bad credit risks. After calculating the PN coefficient for each feature, we obtained the value of the PN coefficient for each feature, as shown in Table 1. The positive features (PN coefficient $>$ 1, such as property and status of existing checking account) correlate with creditworthiness, and individuals with these features tend to have lower credit risks. In contrast, the negative features (PN coefficient $<$ 1, such as foreign worker) tend to have high credit risks. Additionally, the irrelevant features (PN coefficient around

**Table 1**
An example of PN coefficient in real-world dataset.

| Description | | PN coefficient |
|---|---|---|
| Attribute12 | Property | 1.3316 |
| Attribute1 | Status of existing checking account | 1.1636 |
| Attribute7 | Present employment since | 1.1084 |
| Attribute3 | Credit history | 1.0672 |
| Attribute11 | Present residence since | 1.0170 |
| Attribute6 | Savings account/bonds | 0.9788 |
| Attribute8 | Installment rate in percentage of disposable income | 0.9534 |
| Attribute9 | Personal status and sex | 0.9256 |
| Attribute20 | Foreign worker | 0.8258 |

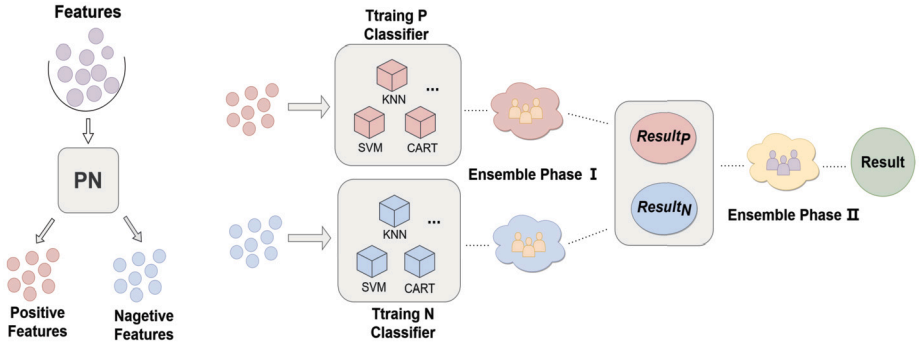



**Fig. 3.** The ensemble positive and negative classification framework.

1) are unrelated to creditworthiness, such as savings account, installment rate, and present residence since. The selection of these features aligns with people's intuitive judgments in daily life, highlighting the interpretability of PN coefficient.

The detailed steps of the PNFS algorithm applied to the above dataset are as follows:

Assuming that PNFS needs to select six features, and the features in dataset German Credit are denoted as $f_1, f_2, ..., f_{20}$.

For feature $f_1$, PNFS calculates the PN coefficient as $PN(f_1, d) = 1.1636 > 1$. Thus, $f_1$ will be added to the positive feature subset, $P = \{f_1\}$. For feature $f_2$, $PN(f_2, d) = 0.9726 < 1$. Therefore, $f_2$ will be added to the negative feature subset set, $N = \{f_2\}$. For feature $f_3$, $PN(f_3, d) = 1.0672 > 1$. Since the set $P$ is not empty, we need to calculate the redundancy between $f_3$ and $f_1$ using equation (12) and we obtain $Int_d(f_3, f_1) = 0.7805 > 0$. As a result, $f_3$ will be added to the subset, and $P = \{f_1, f_3\}$. Other features are treated similarly.

After traversing all the features, PNFS calculates the PNRatio using equation (9), and we get $PNRatio = 0.8698$. Then, using equations (10) and (11), the number of selected positive features is 4, and the number of selected negative features is 2.

Finally, PNFS selects the top four features with the highest PN coefficients from the subset $P$ and the top two features with the lowest PN coefficients from the subset $N$.

Positive and negative features can help us to understand the data. In practical applications, in terms of domain knowledge and expert experience, we can manually analyze the essential features in the data. However, this is very inefficient and expensive. Therefore, we can consider the features' interpretability and credibility by calculating the PN coefficient to determine positive and negative features.

### *3.4. The ensemble PN classification framework*

We introduce a new ensemble classification framework that efficiently incorporates both positive and negative features in classification tasks. Inspired by ensemble learning, we train multiple classifiers tailored to positive and negative features.

In this framework, the PNFS algorithm is employed to select positive and negative features from the target dataset. Following this, positive classifiers are trained using the positive feature subset, while the selected negative feature subset is used to train negative classifiers. Ultimately, we combine the classification results from the positive and negative classifiers to obtain the final outputs. The main steps of this proposed ensemble positive and negative classification framework are illustrated in Fig. 3. This new classification framework, named EPNC, considers feature polarity at two levels and utilizes a diverse range of classifiers for ensemble learning. By combining their results, this classification method enhances the accuracy of classification and provides a more interpretable model.

Assuming that the selected positive and negative feature subsets are denoted as $P$ and $N$, and there are $m$ classifiers in our ensemble strategy framework. We define classifiers set $C = \{C_P, C_N\}$, which includes the positive classifier set $C_P = \{C_{p_1}, C_{p_2}, C_{p_3}, ..., C_{p_m}\}$ and the negative classifier set $C_N = \{C_{n_1}, C_{n_2}, C_{n_3}, ..., C_{n_m}\}$. The main challenge of our proposed ensemble framework is how to integrate the results of positive and negative classifiers. Suppose the target dataset has $k$ instances and $n$ different class labels. After obtaining the classification results, for $m$ positive classifiers, the result set is $PResult = \{RP_1, RP_2, RP_3, ..., RP_m\}$, and for $m$ negative

classifiers, the result set is $NResult = \{RN_1, RN_2, RN_3, ..., RN_m\}$. Let $(RP^{k\times n})_i$ matrix represent the prediction result of $i$-th positive classifiers and $(RN^{k\times n})_i$ matrix represent the prediction result of $i$-th negative classifiers.

For a specific instance $x$ and class label $y$:

$$RP(x,y) = \begin{cases} 1, & x \text{ is predicted to be true by the positive classifier} \\ 0, & \text{otherwise} \end{cases} \tag{13}$$

$$RN(x,y) = \begin{cases} 1, & x \text{ is predicted to be true by the negative classifier} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

In the ensemble phase I, multiple classifiers are trained using positive and negative features for each class. Then, we can get positive classification result $RP$ and negative classification result $RN$ for each instance. We combine the classification result of each instance into ensemble positive matrix $E_P$ and ensemble negative matrix $E_N$. In ensemble phase II, the ensemble positive results $E_P$ and ensemble negative results $E_N$ are merged to obtain the final classification result matrix $E_{PN}$.

In order to derive the ensemble matrix $E_P$ and $E_N$, each $RP$ and $RN$ matrix is merged by counting the predicted outcomes for each category. Hence according to the result of $PResult$ and $NResult$ we can formulate $E_P$ matrix and $E_N$ matrix as:

$$E_P = \begin{bmatrix} \sum_{i=1}^{m} RP_i(1,1), & \sum_{i=1}^{m} RP_i(1,2), & \dots & \sum_{i=1}^{m} RP_i(1,n) \\ \sum_{i=1}^{m} RP_i(2,1), & \sum_{i=1}^{m} RP_i(2,2), & \dots & \sum_{i=1}^{m} RP_i(2,n) \\ \sum_{i=1}^{m} RP_i(3,1), & \sum_{i=1}^{m} RP_i(3,2), & \dots & \sum_{i=1}^{m} RP_i(3,n) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m} RP_i(m,1), & \sum_{i=1}^{m} RP_i(m,2), & \dots & \sum_{i=1}^{m} RP_i(m,n) \end{bmatrix}$$

$$E_N = \begin{bmatrix} \sum_{i=1}^{m} RN_i(1,1), & \sum_{i=1}^{m} RN_i(1,2), & \dots & \sum_{i=1}^{m} RN_i(1,n) \\ \sum_{i=1}^{m} RN_i(2,1), & \sum_{i=1}^{m} RN_i(2,2), & \dots & \sum_{i=1}^{m} RN_i(2,n) \\ \sum_{i=1}^{m} RN_i(3,1), & \sum_{i=1}^{m} RN_i(3,2), & \dots & \sum_{i=1}^{m} RN_i(3,n) \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{m} RN_i(m,1), & \sum_{i=1}^{m} RN_i(m,2), & \dots & \sum_{i=1}^{m} RN_i(m,n) \end{bmatrix}$$

Finally, in ensemble phase 2, the $E_{PN}$ matrix is obtained by merging the $E_P$ and $E_N$ matrices by equation (15).

$$E_{PN}(x,y) = \sum_{i=1}^{m} RP_i(x,y) + \sum_{i=1}^{m} RN_i(x,y) \tag{15}$$

In ${E_{PN}}^{k\times n}$, for each instance $o_i$, we can get a vector $v_i = [w_1, w_2, ..., w_n]$, where $w_i$ is the weight of $o_i$ belongs to the class label $c_i$. The highest weight in each row of $E_{PN}$ is the final classification result of EPNC, as shown in Fig. 4.

For multi-class classification tasks, assume there are $n$ different class labels. PNFS selects positive and negative features corresponding to each class label $l_i$. Subsequently, positive and negative classifiers are constructed within the framework for ensemble classification, resulting in classification results $E_{PN}$ for each class. After comparison, the class label with the highest weight is identified as the final classification result. By incorporating positive and negative features into feature selection and ensemble learning, the EPNC framework enhances the interpretability of selected features and the classification performance in multi-class classification tasks.

By integrating different classifiers and considering the polarity of features, our classification method provides accurate prediction results and considers the interpretability of features.

The detail of the proposed ensemble positive and negative classification framework is shown in Algorithm 2.

Specifically, in step 3, multiple classifiers are trained based on the selected positive and negative feature subsets. Steps 5 to 9 are executed for each instance $x$ in $D$ to acquire the ensemble result matrices from both the positive and negative classifiers. In steps 10 to 12, the ensemble results from the positive and negative classifiers are merged to obtain the weight matrix. Finally, in step 14, the classification result is outputted.

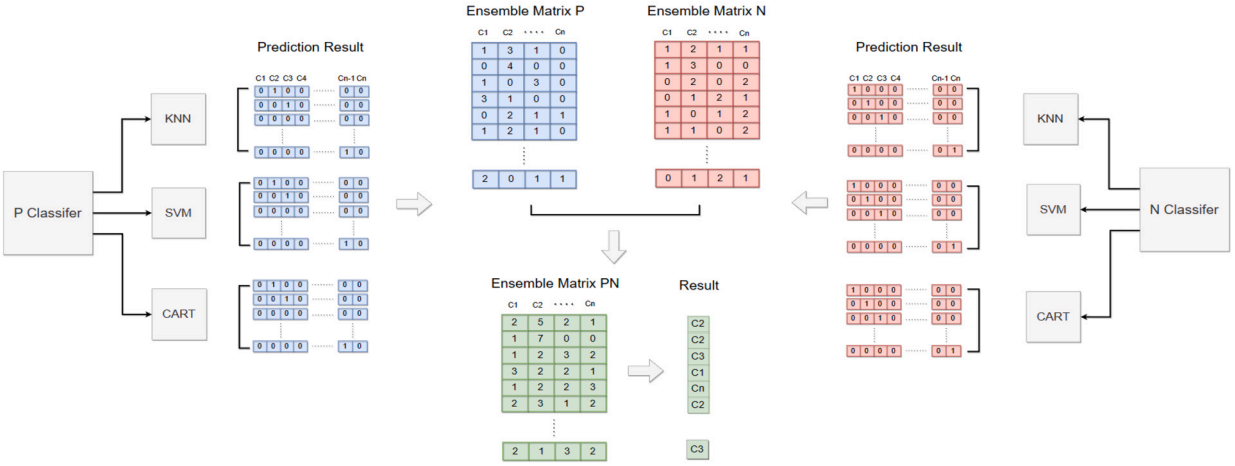



**Fig. 4.** The ensemble strategy of EPNC.

**Algorithm 2** Ensemble Positive and Negative Classification Framework.

**Input:**

$D$: the target dataset;

$K$: the number of selected features;

**Output:**

$R$: the classification results;

Set $R = \{\}$;
Get positive feature subset $f_P$ and negative feature subset $f_N$ by Algorithm 1.
C: training positive and negative classifiers with $f_P$ and $f_N$;
**For** each instance $x$ in dataset $D$:
    **For** each classifier in $C$:
        Obtain predicted results $RP(x, y)$ and $RN(x, y)$;
        $E_P(x, y) = \sum_{i=1}^{n} RP_i(x, y)$;
        $E_N(x, y) = \sum_{i=1}^{n} RN_i(x, y)$;
    **End For**
    $E_{PN}(x, y) = E_P(x, y) + E_N(x, y)$;
    Find the predicted result $C$ with maximum weight in $E_{PN}$;
    $C \rightarrow R$;
**End For**
**Output** $R$.

### 3.5. Time complexity

Here, we present a time complexity estimation for the PNFS and PNEC algorithms. Assuming $n$ represents the number of instances in the target dataset and $m$ represents the number of features.

For the PNFS algorithm, the key steps of this algorithm involve two loops on each feature $f_i$, where the internal loop performs $O(|P|)$ or $O(|N|)$ operations, and the external loop performs $O(|F|)$ operations. The time complexity of the PN coefficient calculation is $O(n^2)$. Hence, the overall time complexity of PNFS is $O((m * n)^2)$.

For the EPNC, step 2 executes the PNFS algorithm and the time complexity is $O(|F|^2)$. Step 3 trains the positive and negative classifiers in this algorithm depends on the number of classifiers $|C|$, which we assume is $O(|C| * \Omega)$. For each instance, it is necessary to iterate through a set of classifiers, calculate the prediction results, and perform ensemble accumulation operations to compute $E_P$ and $E_N$. In the worst case, the time complexity of this step is $O(n * |C| * \Omega)$. Therefore, the overall time complexity of EPNC is $O(n * |C| * \Omega + (m * n)^2)$.

## 4. Experiments

In this part, extensive experiments are conducted to validate the effectiveness of the proposed feature selection method (PNFS) and ensemble classification framework (EPNC). We first present the experimental setup, involving datasets, evaluation metrics, competing algorithms and computational device. Then experiments are conducted between PNFS and four competing information measurements to validate the effectiveness of both positive and negative features. Besides, EPNC is compared to eight state-of-the-art feature selection algorithms to illustrate the superiority of our proposed ensemble classification framework. In the end, we evaluated the performance of EPNC and these competing algorithms varying with different numbers of features.

**Table 2**
Real-world Datasets.

| Data Set | Instances | Features | Classes | Feature Type |
|---|---|---|---|---|
| Arcene | 200 | 10000 | 2 | Integer |
| CNS | 60 | 7129 | 2 | Integer |
| Gisette | 7000 | 5000 | 2 | Integer |
| Advertisements | 2539 | 1558 | 2 | Mixed |
| Leukemia | 72 | 7129 | 2 | Real |
| Colon | 62 | 2000 | 2 | Real |
| Prostate-std | 102 | 6033 | 2 | Real |
| Prostate | 102 | 5966 | 2 | Real |
| Dlbcl | 77 | 6258 | 2 | Real |
| Leukemia-4c | 72 | 7129 | 4 | Real |
| Srbct | 83 | 2308 | 4 | Real |
| Lymphoma-std | 62 | 4026 | 3 | Real |
| Lung2 | 203 | 3312 | 5 | Real |
| MLL | 72 | 12582 | 3 | Integer |
| UNSW-NB15 | 11082 | 31 | 3 | Mixed |
| CICIDS2017 | 11451 | 78 | 4 | Mixed |

### 4.1. Experimental setup

#### 4.1.1. Datasets

Table 2 lists 16 real-world datasets, where datasets Advertisements and CNS are from UCI,[1] datasets Arcene and Gisette are from NIPS 2003, datasets Colon, Srbct, Lung2, Lymphoma-std, Prostate, Prostate-std, Dlbcl, Leukemia, Leukemia-4c, MLL are from.[2] CICIDS2017[3] and UNSW-NB15[4] are network intrusion detection datasets with millions of instances. Because some competing algorithms cannot handle these two massive datasets, we selected a subset of these two datasets for experimentation.

#### 4.1.2. Evaluation metrics

We apply three evaluation metrics to validate the predictive performance and stability of these competing algorithms as follows:

- **Accuracy (ACC)** is a commonly used evaluation metric to measure the performance of a classification model. It represents the proportion of correctly predicted samples to the total number of samples. The formula to calculate accuracy is as follows:

$$ACC = \frac{TP + TN}{TP + FN + FP + FN},$$

  where TP represents True Positive, FP represents False Positive, TN represents True Negative, and FN represents False Negative.
- **F-Measure(F-Score)** is a widely used metric for evaluating the performance of classification models. It combines precision and recall, where a value closer to 1 indicates better model performance. The formula for calculating F-measure is as follows:

$$F - Measure = \frac{2 * Precision * Recall}{Precision + Recall},$$

  where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$.
- **Stability**: Similarity-based measurement introduced by Dunne et al. [13] is used to measure the stability of feature selection algorithms by comparing the similarity of feature selection results across multiple as follows:

$$\Phi(Z) = \frac{1}{M(M-1)} \sum_{i=1}^{M} \sum_{j=1}^{M} \phi(s_i, s_j),$$

  where $Z$ is a set that contains the features selected in each of the $M$ iterations, and $\phi(s_i, s_j)$ computes the similarity of selected features $s_i$ and $s_j$.

Fundamental classifiers were employed in Matlab, including KNN (k = 3), SVM (with the linear kernel), and CART, to evaluate the effectiveness of different selected feature subsets in experiments. For each dataset, we use a 5-fold cross-validation method, dividing the dataset into five equal parts, with four parts for training and one part for testing each time. We utilize the same training and testing set division for each competing algorithm. In addition, we conducted ten iterations on each dataset and assessed the results based on their average performance, thus ensuring the reliability of the experimental outcomes.

[1] https://archive.ics.uci.edu/.
[2] https://www.cs.binghamton.edu/~lyu/KDD08/data/.
[3] https://www.unb.ca/cic/datasets/ids-2017.html.
[4] https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Furthermore, we perform a Friedman test at a significance level of 95% under the null hypothesis to verify whether our proposed method and other comparative algorithms have significant differences [12]. If the null hypothesis of the Friedman test was rejected, a post-hoc Nemenyi test was conducted.

#### 4.1.3. Competing algorithms

We compare EPNC with eight state-of-the-art feature selection methods, including CFS [48], OLSFS [46], QLFS [44], PMFS [21], FSDK [32], MRMR [35], MFS-MCDM [20] and SVM-RFE [3]. The comparison algorithms' introduction and parameter settings in the experiment are as follows:

- CFS (Curvature-based Feature Selection) ranks the importance of features based on the concept of Menger curvature. The main idea of the CFS algorithm is to evaluate the significance of features by calculating the Menger curvature on each feature dimension. Menger curvature is a mathematical concept used to measure the curvature of a curve. In CFS, the Menger curvature is employed to quantify the variation in data distribution along feature dimensions. Given a dataset, CFS first calculates the Menger curvature on each feature dimension. Rank the features based on their Menger curvature values to determine their order of importance. Then, iteratively select features starting from the highest-ranked feature and evaluate the classification performance after adding each feature. Finally, the decision to include the feature is based on the classification performance. If the performance improves, the feature is retained, and the next selection round continues. Otherwise, the feature is discarded. CFS does not require setting parameters in experiments.
- OLSFS (Orthogonal Least Squares based Feature Selection) is a fast feature selection algorithm based on Orthogonal Least Squares (OLS). OLSFS uses a novel criterion called Squared Orthogonal Correlation Coefficient (SOCC) for feature ranking. The algorithm employs a greedy search strategy to select a subset of features iteratively. For each candidate feature, it computes the correlation with the class labels using the OLS method. The OLS method estimates the linear relationship between the feature and the class by minimizing the sum of squared residuals. This algorithm demonstrates high computational efficiency when dealing with large-scale datasets and high-dimensional feature spaces, and it exhibits good feature selection performance in many practical applications. In experiments, OLSFS does not require setting parameters.
- QLFS (Q-Learning based Feature Selection) is a dynamic feature selection algorithm based on the Q-Learning mechanism. This algorithm integrates feature selection and Q-Learning into a unified framework. Q-learning is utilized to construct discriminant functions for each class of the data. These discriminant functions aid in distinguishing between different classes of data samples and facilitate the ranking and selection of features. QLFS uses the Q-Learning algorithm to construct discriminant functions for each category. These discriminant functions are used to predict the category of each sample and obtain reward signals based on the prediction results. Subsequently, the feature importance of each category is considered collectively for ranking. During updating the discriminant function vector, feature sorting is performed, effectively selecting discriminative features and improving classification performance. We set the learning rate $\alpha = 30\%$ for QLFS in experiments.
- PMFS (Pareto Dominance based Feature Selection) is a multi-label feature selection algorithm based on Pareto dominance, aiming to address the feature selection problem in high-dimensional multi-label datasets. The algorithm transforms the multi-label feature selection problem into a bi-objective optimization problem concerning the relevance and redundancy of features. It utilizes Pareto dominance to handle this bi-objective space. PMFS models the relevance and redundancy of features in the same space and evaluates them using the concept of Pareto dominance. Specifically, relevance measures the degree of association between a feature and all labels, while redundancy measures the similarity between a feature and other features. The PMFS algorithm can identify a subset of features with high relevance and low redundancy by combining relevance and redundancy. We set $\lambda = 10$ for PMFS in experiments, and this parameter is a weighting factor used to balance the relevance and redundancy of features to find the optimal feature subset.
- FSDK (Fast Sparse Discriminative K-Means) is an efficient feature selection algorithm that combines the classical Least Squares Regression (LSR) and Discriminative K-means (DisK-means) methods. The FSDK algorithm designs an efficient feature selection framework. Firstly, it introduces a weighted pseudo-label matrix with discrete characteristics to avoid trivial solutions in unsupervised LSR methods. In this case, any constraints imposed on the pseudo-label matrix and selection matrix are optional, which is beneficial for simplifying the combinatorial optimization problem. Secondly, it introduces an $l_2,\ p-norm$ regularization term to enforce row sparsity in the selection matrix, where $p$ is a flexible parameter. Therefore, the FSDK model can be seen as a feature selection framework combining the DisK-means algorithm and $l_2,\ p-norm$ regularization to optimize sparse regression problems. Additionally, the model scales linearly with the number of samples, enabling fast large-scale data processing. Extensive testing on various datasets has provided strong evidence of the effectiveness and efficiency of the FSDK algorithm. We set $\gamma = 10,\ p = 1,\ NITR_w = 30,\ NITR_y = 20$ for FSDK in experiments.
- MRMR (Minimum Redundancy Maximum Relevance) is a classic feature selection algorithm that aims to select a subset of features with maximum information and complementarity by maximizing the relevance between features and the target variable while minimizing the redundancy among features. The advantage of the MRMR method is its ability to simultaneously consider the relevance between features and the target variable and the redundancy among features. This allows for selecting less redundant and highly relevant features, thereby improving the effectiveness of feature selection. However, the MRMR method may face high computational complexity when dealing with high-dimensional data, as it requires the computation of a large number of feature correlations. In experiments, MRMR does not require setting parameters.
- MFS-MCDM (Multi-Criteria Decision Making) is a multi-label feature selection algorithm that models the feature selection process as a multi-criteria decision-making process. This method was first applied to multi-label data. It used the well-known

multi-criteria decision-making method, TOPSIS (Technique of Order Preference by Similarity to Ideal Solution), to evaluate the relationships between features and multiple labels as different criteria. The MFS-MCDM method considers multi-label feature selection as an information fusion process. Firstly, it utilizes the ridge regression algorithm to obtain a decision matrix and then calculates the weights of each column of this matrix based on the entropy of each label. Next, the TOPSIS method assigns a score to each feature based on the weighted decision matrix. Finally, a ranking vector of features is generated as the output to improve classification accuracy and reduce computational costs. Users can also choose the desired number of features. We set $\lambda = 10$ for MFS-MCDM in experiments.

- SVM-RFE (Support Vector Machine Recursive Feature Elimination) is a feature selection method based on Support Vector Machines, which aims to find the optimal subset of features by recursively selecting and excluding features. SVM-RFE takes advantage of the SVM's ability to model the importance of features and reduces the dimensionality of the feature space by gradually removing unimportant features. As a result, SVM-RFE can improve the performance of classifiers and obtain more interpretable and efficient feature subsets. Besides, SVM-RFE can be combined with other machine learning models and feature selection methods. Compared to other feature selection methods, SVM-RFE has certain advantages in handling high-dimensional and small sample problems, and it can provide better classification performance and feature selection stability. To improve computational efficiency, SVM-RFE employs the elimination percentage parameter, denoted as E, instead of eliminating features individually. Once the $StopValue$ is reached, the algorithm eliminates one feature at a time until all features are ranked. In this experiment, SVM uses a linear kernel, and the elimination percentage parameter of SVM-RFE, $E$, is set to 20%, the $StopValue$ and the cost parameter $C$ were set to 5% and 1.

#### 4.1.4. Computational device

All experimental results are conducted on a PC with AMD Ryzen 7 5800X 8-Core Processor 3.80 GHz CPU, and 16 GB memory.

### 4.2. PN coefficient vs. other information measurements

This section compares the PN coefficient with other feature selection coefficients. We conduct experiments on fourteen datasets with four classic feature selection coefficients, including Fisher [14], PCC [11], MI [43], and S2N [26].

- Fisher: (Fisher Score) is commonly used to identify features with significant discriminative power for classification tasks. The algorithm is based on the Fisher criterion, which evaluates the importance of features by calculating the ratio of inter-class scatter to intra-class scatter. The main advantages of the Fisher feature selection algorithm are its simplicity and ease of implementation.
- PCC: (Pearson Correlation Coefficient) measures the degree of linear correlation between two variables and ranges from -1 to 1. The advantages of PCC as a feature selection method are its simplicity and fast computation speed. It can help identify features highly correlated with the target variable, thereby reducing the feature space's dimensionality and improving the model's efficiency and performance.
- MI: (Mutual Information) is based on the concept of information theory and is used to measure the correlation between two variables. The MI feature selection method can help identify features highly correlated with the target variable from a given feature set, thereby improving the performance and interpretability of machine learning models. The advantages of the MI feature selection method are its simplicity, understandability, and high computational efficiency. It can be applied not only to supervised learning problems but also to unsupervised learning problems.
- S2N: (Signal-to-Noise) is a statistical measure based on the signal-to-noise ratio, aiming to identify features with significant signals and low noise from high-dimensional data. It utilizes two key concepts to measure the importance of each feature: signal and noise. The signal represents the differences in features between different classes, while the noise represents the differences within features. By calculating the signal-to-noise ratio, one can assess the separability of features between classes and determine their importance. One of the advantages of the S2N algorithm is its ability to capture differences in features between different classes while unaffected by within-feature variations.

We use three basic classifiers (KNN, SVM, and CART) to measure the classification results of features selected by different information measurements. We rank the features in each dataset and select the top ten features. The prediction accuracy, F-Measure, and stability performance of the PN coefficient and other feature selection coefficients are shown in Tables 3 to 5. The best results are highlighted in boldface on the table. The p-values obtained from the Friedman test for accuracy in KNN, SVM, and CART are 0.0005, 4.20E-05, and 0.0007. For F-Measure, the p-values in KNN, SVM, and CART are 5.77e-04, 2.42e-05, and 1.14e-05, and the p-values of stability are 0.1150. Therefore, there is a significant difference between these competing algorithms regarding KNN, SVM, and CART. The CD (critical difference) value is 1.6311.

From Tables 3 to 5, the experimental result revealed that PNFS achieves the highest average predictive accuracy and the lowest average ranks among all these competing coefficients. PNFS gets the highest predictive accuracy on nine of fourteen datasets. In other words, the PN coefficient and PNFS have apparent advantages in the classification results in KNN, CART, and SVM classifiers. We also compared PNFS with other feature selection measurements regarding F-measure and stability. From the experimental results, it can be observed that PNFS demonstrates superior overall performance and stability in feature selection tasks.

Most traditional feature coefficients use statistical methods to measure the relationship between variables during feature selection. The PN coefficient combines features' positive and negative polarities, especially the exclusion effect of negative features. Therefore, our proposed feature selection method has better accuracy and interpretability than traditional feature coefficients. PNFS can produce

**Table 3**
Comparison of PN coefficient and other information measurements on Accuracy.

| Data Set | KNN | | | | | SVM | | | | | CART | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fisher | PCC | MI | S2N | PNFS | Fisher | PCC | MI | S2N | PNFS | Fisher | PCC | MI | S2N | PNFS |
| Arcene | 0.6813 | 0.5886 | 0.6761 | 0.7008 | **0.7283** | 0.6750 | 0.5906 | 0.6583 | 0.6607 | **0.7283** | 0.6493 | 0.5886 | 0.6735 | 0.6794 | **0.7283** |
| CNS | 0.5548 | 0.5833 | 0.5976 | 0.4167 | **0.6667** | 0.5833 | 0.4167 | 0.6083 | 0.5167 | **0.6667** | 0.4167 | 0.6056 | **0.6833** | 0.5542 | 0.6667 |
| Gisette | 0.8629 | 0.5086 | 0.8579 | **0.8714** | 0.8471 | 0.8050 | 0.4936 | 0.8171 | 0.8034 | **0.8471** | 0.8529 | 0.4950 | 0.8093 | **0.8734** | 0.8471 |
| Advertisements | 0.9428 | 0.8157 | 0.9492 | **0.9597** | 0.9322 | 0.9386 | 0.8369 | **0.9534** | 0.9415 | 0.9322 | **0.9682** | 0.8220 | 0.9449 | 0.9563 | 0.9322 |
| Leukemia | 0.9286 | 0.9308 | 0.9357 | 0.8571 | **0.9429** | 0.7367 | 0.8333 | 0.8024 | 0.9028 | **0.9429** | 0.7857 | 0.8571 | 0.8857 | 0.8310 | **0.9429** |
| Colon | 0.6667 | 0.6881 | 0.7476 | 0.7167 | **0.8217** | **0.9125** | 0.7500 | 0.8411 | 0.8717 | 0.8217 | 0.7792 | 0.7409 | 0.6833 | 0.7167 | **0.8217** |
| Prostate-std | 0.9192 | 0.9264 | 0.9071 | **0.9322** | 0.9188 | 0.8825 | 0.6875 | 0.8271 | 0.8644 | **0.9188** | 0.8991 | 0.8569 | 0.8662 | 0.8500 | **0.9188** |
| Prostate | 0.8856 | 0.5829 | 0.8731 | **0.8929** | 0.8917 | **0.8964** | 0.7333 | 0.8667 | 0.8556 | 0.8917 | 0.8643 | 0.8650 | 0.8392 | 0.8617 | **0.8917** |
| Dlbcl | 0.8929 | 0.7253 | 0.9091 | 0.8611 | **0.9167** | 0.9067 | 0.8533 | 0.7467 | 0.8925 | **0.9167** | 0.8133 | 0.8400 | 0.8667 | 0.8267 | **0.9167** |
| Leukemia-4c | 0.8571 | 0.7857 | 0.6429 | 0.7143 | **0.9048** | 0.8143 | 0.7286 | 0.8571 | 0.6429 | **0.9048** | 0.8971 | 0.8714 | 0.6714 | 0.7143 | **0.9048** |
| Srbct | 0.8235 | 0.8659 | 0.9012 | 0.6471 | **0.9412** | 0.7059 | 0.8541 | 0.8878 | 0.7647 | **0.9412** | 0.8429 | 0.7714 | 0.6429 | 0.5714 | **0.9412** |
| Lymphoma-std | 0.8925 | 0.9063 | **0.9375** | 0.8917 | 0.9167 | 0.9042 | **0.9227** | 0.8333 | 0.8985 | 0.9167 | 0.7082 | 0.8235 | 0.7059 | 0.5294 | **0.9167** |
| Lung2 | 0.8293 | 0.8049 | 0.8537 | 0.7949 | **0.9024** | 0.8293 | 0.8049 | 0.8537 | 0.8370 | **0.9024** | **0.9146** | 0.8750 | 0.9019 | 0.8667 | 0.9024 |
| MLL | 0.8762 | 0.7619 | 0.8286 | 0.7714 | **0.9286** | 0.8621 | 0.7286 | 0.9021 | 0.8286 | **0.9286** | 0.6429 | 0.7857 | 0.8714 | 0.6571 | **0.9286** |
| Ave | 0.8295 | 0.7482 | 0.8298 | 0.7877 | **0.8757** | 0.8180 | 0.7310 | 0.8182 | 0.8058 | **0.8757** | 0.7882 | 0.7713 | 0.7890 | 0.7492 | **0.8757** |
| Ave Rank | 3.21 | 3.93 | 2.79 | 3.29 | **1.79** | 2.86 | 4.29 | 2.93 | 3.36 | **1.57** | 3.00 | 3.43 | 3.29 | 3.79 | **1.50** |

**Table 4**
Comparison of PN coefficient and other information measurements on F-Measure.

| Data Set | KNN | | | | | SVM | | | | | CART | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Fisher | PCC | MI | S2N | PNFS | Fisher | PCC | MI | S2N | PNFS | Fisher | PCC | MI | S2N | PNFS |
| Arcene | 0.6190 | 0.5965 | 0.6111 | 0.7059 | **0.7283** | 0.6111 | 0.4848 | 0.6316 | 0.6486 | **0.7283** | 0.6939 | 0.6047 | 0.7059 | 0.6316 | **0.7283** |
| CNS | 0.6154 | 0.4615 | **0.7368** | 0.6667 | 0.6667 | 0.4286 | 0.5455 | **0.7368** | 0.4615 | 0.6667 | 0.6250 | 0.5714 | **0.7368** | 0.5882 | 0.6667 |
| Gisette | **0.8507** | 0.7934 | 0.8320 | 0.7846 | 0.8471 | 0.7080 | 0.7414 | 0.7903 | 0.7692 | **0.8471** | 0.8148 | 0.7742 | 0.8308 | **0.8921** | 0.8471 |
| Advertisements | 0.9231 | 0.7500 | 0.9091 | 0.8889 | **0.9322** | 0.9013 | 0.8889 | 0.8571 | 0.8957 | **0.9322** | 0.8333 | 0.8363 | 0.8577 | 0.9091 | **0.9322** |
| Leukemia | 0.9412 | 0.8889 | 0.8235 | **0.9474** | 0.9429 | 0.7273 | 0.8235 | 0.8192 | 0.8669 | **0.9429** | 0.8924 | 0.9032 | 0.8826 | 0.9003 | **0.9429** |
| Colon | **0.8421** | 0.8235 | 0.7778 | 0.8235 | 0.8217 | 0.8361 | 0.7778 | **0.8721** | 0.8156 | 0.8217 | 0.8000 | 0.7692 | 0.7778 | **0.8750** | 0.8217 |
| Prostate-std | 0.8691 | 0.8378 | 0.8571 | 0.9055 | **0.9188** | 0.7368 | 0.8421 | 0.8696 | 0.7778 | **0.9188** | 0.7826 | 0.8421 | 0.9167 | **0.9204** | 0.9188 |
| Prostate | 0.7827 | 0.7273 | 0.7590 | 0.7847 | **0.8917** | 0.6886 | 0.8036 | 0.7533 | 0.8314 | **0.8917** | 0.6858 | 0.7262 | 0.6643 | 0.6927 | **0.8917** |
| Dlbcl | 0.7083 | 0.6524 | 0.6222 | 0.5046 | **0.9167** | 0.6889 | 0.8102 | 0.6852 | 0.6127 | **0.9167** | 0.6413 | 0.6889 | 0.6111 | 0.8056 | **0.9167** |
| Leukemia-4c | 0.8141 | 0.9018 | 0.8974 | 0.7479 | **0.9048** | 0.8726 | 0.8750 | 0.8166 | 0.6792 | **0.9048** | 0.8056 | **0.9221** | 0.8662 | 0.7952 | 0.9048 |
| Srbct | 0.8992 | 0.8737 | 0.9111 | 0.9064 | **0.9412** | 0.8737 | 0.9064 | 0.8992 | 0.9111 | **0.9412** | 0.8125 | 0.8737 | 0.9221 | 0.8796 | **0.9412** |
| Lymphoma-std | 0.8442 | 0.8674 | 0.8357 | 0.7491 | **0.9167** | 0.7159 | 0.8724 | 0.8637 | 0.8801 | **0.9167** | 0.6835 | 0.8170 | 0.7190 | 0.8663 | **0.9167** |
| Lung2 | 0.7677 | 0.6236 | 0.7283 | 0.7802 | **0.9024** | 0.8078 | 0.8345 | **0.9028** | 0.7512 | 0.9024 | 0.7810 | 0.6436 | 0.7488 | 0.8475 | **0.9024** |
| MLL | 0.7857 | **0.9302** | 0.9048 | 0.6944 | 0.9286 | 0.8056 | 0.7905 | 0.8182 | 0.7488 | **0.9286** | 0.8190 | 0.7905 | 0.8182 | 0.7088 | **0.9286** |
| Ave | 0.8045 | 0.7663 | 0.8004 | 0.7778 | **0.8757** | 0.7430 | 0.7855 | 0.8083 | 0.7607 | **0.8757** | 0.7622 | 0.7688 | 0.7898 | 0.8080 | **0.8757** |
| Ave Rank | 2.86 | 3.79 | 3.50 | 3.21 | **1.50** | 4.00 | 3.36 | 2.93 | 3.43 | **1.29** | 3.86 | 3.71 | 3.36 | 2.71 | **1.36** |

**Table 5**
Comparison of PN coefficient and other information measurements on Stability.

| Data Set | Fisher | PCC | MI | S2N | PNFS |
|---|---|---|---|---|---|
| Arcene | 0.6697 | 0.4494 | 0.4595 | **0.7197** | 0.5874 |
| CNS | 0.1388 | 0.1488 | 0.2690 | 0.1588 | **0.3568** |
| Gisette | 0.9597 | 0.8692 | **0.9597** | 0.7797 | 0.7984 |
| Advertisements | 0.8097 | 0.7096 | 0.7163 | 0.7496 | **0.8955** |
| Leukemia | 0.5678 | 0.5678 | 0.3869 | 0.6194 | **0.8566** |
| Colon | **0.7095** | 0.6194 | 0.5472 | 0.6294 | 0.6839 |
| Prostate-std | **0.7696** | 0.5893 | 0.6895 | 0.5994 | 0.7069 |
| Prostate | 0.5660 | 0.2889 | 0.4792 | 0.4091 | **0.8144** |
| Dlbcl | 0.4793 | 0.5894 | 0.3393 | 0.5593 | **0.5952** |
| Leukemia-4c | 0.6686 | 0.7389 | 0.2389 | 0.1588 | **0.8420** |
| Srbct | 0.6291 | 0.7394 | **0.9597** | 0.8994 | 0.7822 |
| Lymphoma-std | 0.4985 | 0.6690 | **0.6695** | 0.5711 | 0.1742 |
| Lung2 | 0.5195 | 0.7097 | 0.3166 | 0.6884 | **0.7350** |
| MLL | 0.4290 | 0.5192 | 0.4992 | **0.7995** | 0.5996 |
| Ave | 0.6011 | 0.5863 | 0.5379 | 0.5958 | **0.6734** |
| Ave Rank | 3.00 | 3.50 | 3.36 | 3.00 | **2.00** |

consistent and reliable feature selection results across different datasets and classification tasks while accurately capturing important features related to the target variable.

Particularly in multi-class datasets, PNFS exhibits significantly higher F-Measure values than other criteria. Traditional feature selection criteria often overlook the complex relationships between different classes in multi-class tasks. They can provide overall information about the relationship between features and the target variable, but they fail to capture subtle differences between different classes, resulting in information loss. On the other hand, PNFS can select positive and negative features corresponding to different classes, providing more accurate and reliable feature selection results, thereby significantly improving the accuracy and performance in multi-class tasks.

In addition, we compared the impact of different numbers of selected features on prediction accuracy across four datasets, ranging from 20 to 100, as shown in Fig. 5. It can be observed that PNFS achieves higher accuracy than other benchmark algorithms. Moreover, PNFS performs well even with the lowest number of features and tends to stabilize as the number of features increases. Performance improvement can generally be observed as the features increase because more relevant features are added, which helps better differentiate between different classes. However, for some competing measurements, further increasing the number of features

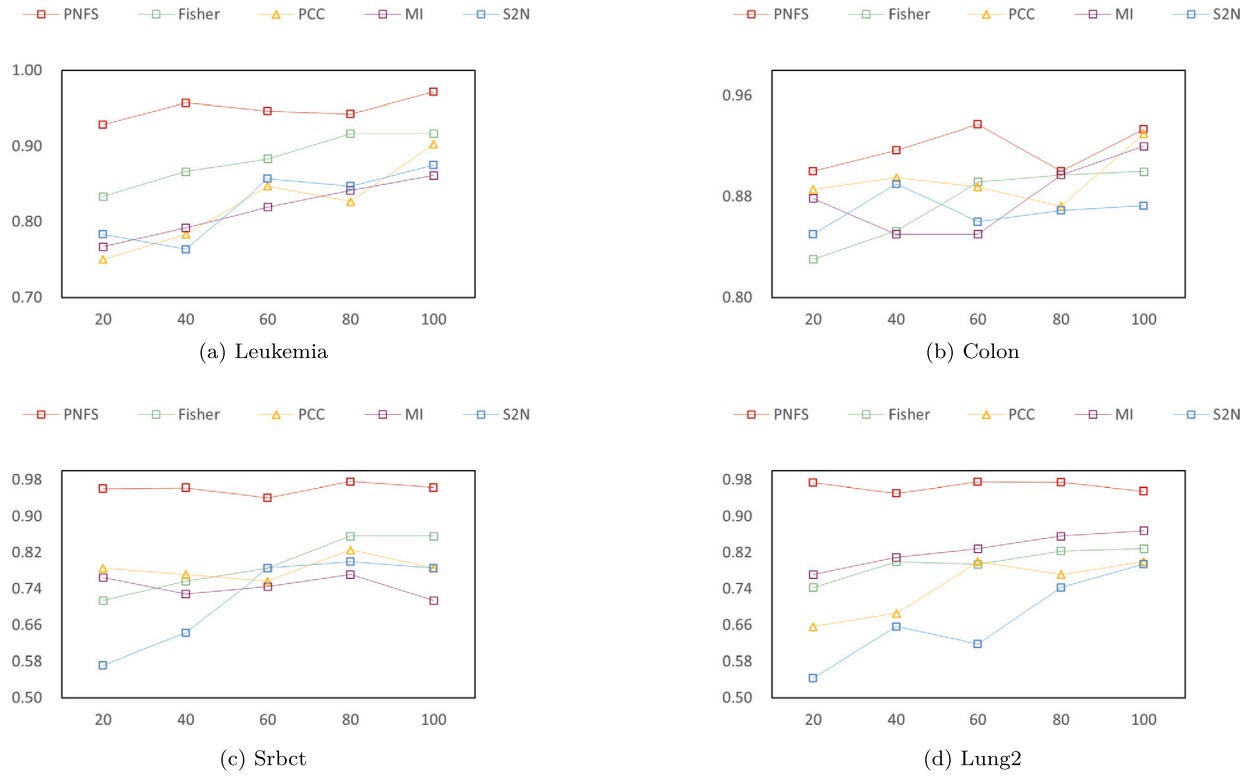



**Fig. 5.** Accuracy of these competing information measurements varying with different numbers of selected features.

may sometimes lead to a decline in performance. This is because the added features may contain noise or redundant information, causing the model to overfit the training data. However, PNFS demonstrates stable performance by utilizing the polarity of features for feature selection, enabling the selected features to effectively differentiate between different classes while maintaining the highest classification performance with both the minimum and maximum number of feature selections.

### *4.3. EPNC vs. state-of-the-art feature selection methods*

This section compares EPNC with eight state-of-the-art feature selection methods, including CFS [48], OLSFS [46], QLFS [44], PMFS [21], FSDK [32], MRMR [35], MFS-MCDM [20] (abbreviated as MCDM in tables) and SVM-RFE [3]. All competing algorithms are implemented in Matlab. Due to the length limit of the paper, the number of selected features was consistently maintained at 10.

Tables 6 to 13 summarize the predictive accuracy, F-Measure, stability, and running time of these competing algorithms. The best results are highlighted in bold face in the tables. According to the Friedman test, the p-values of Accuracy in cases of KNN, SVM, and CART are 5.74e-11, 5.60e-09, and 1.89e-13, respectively. The p-values of F-Measure in cases of KNN, SVM, and CART are 5.92e-09, 2.43e-06, and 1.34e-04, respectively. Therefore, there is a significant difference between these competing algorithms in predictive accuracy and F-Measure. Besides, the p-values of stability and running time of these competing algorithms are 3.98e-15 and 1.77e-21. Thus, there is also a significant difference between these competing algorithms in terms of stability and running time. The CD (critical difference) value is 3.0056. Figs. 6–8 display the results of statistical tests among these competing algorithms.

From Tables 6 to 13 and Figs. 6–8, we can indicate that:

- EPNC *vs*. CFS: CFS ranks and weights features based on the curvature values of each dimension in the dataset, selecting features with higher curvature values, which are often more correlated with the classification decision and can improve classification performance. In some cases, CFS may select too many features, leading to overfitting and reducing the model's generalization ability. According to the statistical test, EPNC performs significantly better than CFS in accuracy and F-measure in cases of KNN, SVM, and CART. Besides, there is also a significant difference in stability between EPNC and CFS. Therefore, EPNC is significantly better than CFS in predictive accuracy and stability. In terms of running time, CFS is faster than EPNC. CFS is a feature selection method that evaluates the relevance of features based on feature curvature. In scenarios with high feature dimensions, the evaluation of feature subsets may be disrupted due to the dependence of the CFS method on data distribution and feature correlation. This ultimately leads CFS to poor classification performance in high-dimensional data.
- EPNC *vs*. OLSFS: The OLSFS method exhibits fast computation speed during the greedy search process. However, when there is a non-linear relationship between the features and the response, the OLSFS method may not obtain the optimal subset of features. There is no significant difference between EPNC and OLSFS in predictive accuracy and F-measure. However, EPNC achieves

**Table 6**
Comparison of EPNC and Competing Methods on Accuracy Using KNN.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.6438 | 0.7056 | 0.6583 | 0.6167 | 0.6714 | 0.6563 | **0.7636** | 0.6813 | 0.7283 |
| CNS | 0.5238 | 0.5267 | 0.5167 | 0.5683 | 0.4167 | 0.5833 | 0.5257 | 0.6167 | **0.6667** |
| Gisette | 0.6707 | **0.8707** | 0.6458 | 0.7167 | 0.6167 | 0.8664 | 0.7083 | 0.8164 | 0.8471 |
| Advertisements | 0.8898 | 0.9089 | 0.8347 | 0.9068 | 0.8323 | 0.9174 | 0.8877 | 0.8157 | **0.9322** |
| Leukemia | 0.6786 | 0.8800 | 0.6829 | 0.9043 | 0.7857 | 0.8571 | 0.8929 | 0.9286 | **0.9429** |
| Colon | 0.6217 | 0.7383 | 0.6875 | 0.7292 | **0.8333** | 0.8125 | 0.7667 | 0.7333 | 0.8217 |
| Prostate-std | 0.7375 | 0.8571 | 0.7167 | 0.8667 | 0.7271 | 0.8375 | 0.8188 | 0.8781 | **0.9188** |
| Prostate | 0.7167 | 0.8357 | 0.7118 | 0.8786 | 0.7364 | **0.9250** | 0.8692 | 0.9167 | 0.8917 |
| Dlbcl | 0.7333 | 0.8587 | 0.7360 | 0.8667 | 0.8334 | 0.8083 | 0.7333 | 0.8133 | **0.9167** |
| Leukemia-4c | 0.7286 | 0.8214 | 0.7143 | 0.8661 | 0.8367 | 0.8214 | 0.7959 | 0.8571 | **0.9048** |
| Srbct | 0.7194 | 0.8824 | 0.6952 | 0.9048 | 0.8429 | 0.8667 | 0.7647 | 0.9333 | **0.9412** |
| Lymphoma-std | 0.8667 | 0.8796 | **0.9833** | 0.9524 | 0.8833 | 0.9667 | 0.8333 | 0.8889 | 0.9167 |
| Lung2 | 0.7436 | 0.9024 | 0.7805 | 0.8049 | 0.7949 | **0.9268** | 0.8293 | 0.7561 | 0.9024 |
| MLL | 0.7143 | 0.8571 | 0.7857 | 0.8714 | 0.7143 | 0.9000 | 0.8286 | 0.8929 | **0.9286** |
| UNSWNB15 | 0.9106 | 0.8777 | 0.9201 | **0.9251** | 0.9084 | 0.9057 | 0.9206 | 0.9016 | 0.9197 |
| CICIDS2017 | 0.9079 | 0.8620 | 0.8594 | 0.8563 | 0.8659 | 0.9157 | 0.8712 | 0.9245 | **0.9393** |
| Ave | 0.7379 | 0.8290 | 0.7456 | 0.8272 | 0.7687 | 0.8479 | 0.8006 | 0.8347 | **0.8824** |
| Ave Rank | 7.38 | 4.63 | 7.13 | 4.25 | 6.25 | 3.81 | 5.38 | 4.19 | **1.81** |

**Table 7**
Comparison of EPNC and Competing Methods on Accuracy Using SVM.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.6219 | 0.6361 | 0.6583 | **0.7313** | 0.6438 | 0.7063 | 0.7194 | 0.6917 | 0.7283 |
| CNS | 0.5667 | 0.5833 | 0.5972 | 0.5714 | 0.5278 | 0.5417 | 0.5833 | 0.6389 | **0.6667** |
| Gisette | 0.6136 | **0.8743** | 0.5636 | 0.5014 | 0.5214 | 0.8621 | 0.5021 | 0.8286 | 0.8471 |
| Advertisements | 0.8432 | 0.9343 | 0.8538 | 0.9237 | 0.8875 | **0.9513** | 0.9089 | 0.8301 | 0.9322 |
| Leukemia | 0.7286 | 0.9029 | 0.7357 | 0.9314 | 0.9286 | 0.8571 | 0.7857 | 0.9286 | **0.9429** |
| Colon | 0.6817 | 0.8167 | 0.8125 | 0.7983 | **0.8333** | 0.8205 | 0.7639 | 0.7833 | 0.8217 |
| Prostate-std | 0.7625 | 0.8667 | 0.7214 | 0.8452 | 0.8071 | 0.8563 | 0.8875 | 0.8167 | **0.9188** |
| Prostate | 0.8125 | 0.8857 | 0.7107 | 0.8667 | 0.7094 | 0.9167 | 0.8571 | **0.9357** | 0.8917 |
| Dlbcl | 0.7867 | 0.9067 | 0.8533 | 0.8800 | 0.7667 | 0.8334 | 0.8667 | 0.8833 | **0.9167** |
| Leukemia-4c | 0.7381 | 0.7571 | 0.7273 | 0.8117 | 0.7143 | 0.8429 | 0.6905 | 0.7041 | **0.9048** |
| Srbct | 0.7245 | 0.8690 | 0.7083 | 0.9043 | 0.7114 | 0.9278 | 0.8204 | 0.8824 | **0.9412** |
| Lymphoma-std | 0.9038 | 0.9643 | 0.9306 | **0.9667** | 0.8667 | 0.9063 | 0.8333 | 0.8958 | 0.9167 |
| Lung2 | 0.7729 | **0.9268** | 0.7561 | 0.7726 | 0.8049 | 0.8522 | 0.7729 | 0.8205 | 0.9024 |
| MLL | 0.6429 | 0.8143 | 0.6714 | 0.9000 | 0.7143 | 0.8714 | 0.8571 | 0.7857 | **0.9286** |
| UNSWNB15 | 0.7116 | 0.6900 | 0.8475 | 0.8858 | 0.8294 | 0.5293 | 0.9052 | 0.7053 | **0.9197** |
| CICIDS2017 | 0.7332 | 0.6961 | 0.7485 | 0.7707 | 0.5707 | 0.5362 | 0.5511 | 0.4817 | **0.9393** |
| Ave | 0.7278 | 0.8203 | 0.7435 | 0.8163 | 0.7398 | 0.8007 | 0.7691 | 0.7883 | **0.8824** |
| Ave Rank | 7.13 | 3.94 | 6.25 | 4.00 | 6.50 | 4.31 | 5.69 | 5.38 | **1.75** |

**Table 8**
Comparison of EPNC and Competing Methods on Accuracy Using CART.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.6438 | **0.7828** | 0.6792 | 0.7063 | 0.6125 | 0.7107 | 0.7136 | 0.6984 | 0.7283 |
| CNS | 0.5200 | 0.5267 | 0.5150 | 0.5683 | 0.5417 | 0.5278 | 0.6205 | 0.5833 | **0.6667** |
| Gisette | 0.6521 | 0.8764 | 0.6028 | 0.5014 | 0.5144 | 0.8786 | **0.8791** | 0.8204 | 0.8471 |
| Advertisements | 0.6438 | 0.9513 | 0.8496 | 0.9089 | 0.9492 | **0.9619** | 0.8981 | 0.9131 | 0.9322 |
| Leukemia | 0.7143 | 0.8857 | 0.7381 | 0.9043 | 0.7619 | 0.8571 | 0.9214 | 0.9286 | **0.9429** |
| Colon | 0.8056 | 0.7708 | 0.7917 | **0.8542** | 0.7167 | 0.8333 | 0.7813 | 0.8125 | 0.8217 |
| Prostate-std | 0.8125 | 0.8610 | 0.7156 | 0.8929 | 0.7865 | 0.8962 | 0.8857 | 0.8083 | **0.9188** |
| Prostate | 0.7042 | 0.8340 | 0.6833 | 0.8875 | 0.7079 | 0.8525 | 0.8167 | **0.9083** | 0.8917 |
| Dlbcl | 0.8104 | 0.8256 | 0.8476 | 0.8667 | 0.8056 | 0.7333 | 0.8741 | 0.8133 | **0.9167** |
| Leukemia-4c | 0.7245 | 0.7857 | 0.6429 | 0.8571 | 0.7143 | 0.8143 | 0.8714 | 0.7143 | **0.9048** |
| Srbct | 0.6471 | 0.7851 | 0.7286 | 0.8290 | 0.8367 | 0.7616 | 0.8235 | 0.8824 | **0.9412** |
| Lymphoma-std | 0.8167 | 0.8333 | 0.8667 | 0.8833 | 0.7667 | 0.9000 | 0.8796 | 0.8958 | **0.9167** |
| Lung2 | 0.6892 | 0.7533 | 0.8073 | 0.7780 | 0.7829 | 0.7870 | 0.7575 | 0.8780 | **0.9024** |
| MLL | 0.6714 | 0.7714 | 0.6571 | 0.7429 | 0.7857 | 0.8143 | 0.7429 | 0.8571 | **0.9286** |
| UNSWNB15 | 0.9156 | 0.9237 | 0.9174 | 0.9215 | 0.9170 | 0.9228 | 0.9323 | **0.9504** | 0.9197 |
| CICIDS2017 | 0.9074 | 0.9192 | 0.9205 | 0.9236 | 0.9227 | 0.9314 | 0.9210 | 0.9301 | **0.9393** |
| Ave | 0.7299 | 0.8179 | 0.7477 | 0.8141 | 0.7576 | 0.8239 | 0.8324 | 0.8372 | **0.8824** |
| Ave Rank | 7.75 | 5.19 | 7.25 | 4.38 | 6.56 | 3.88 | 4.25 | 3.75 | **1.94** |

**Table 9**
Comparison of EPNC and Competing Methods on F-Measure Using KNN.

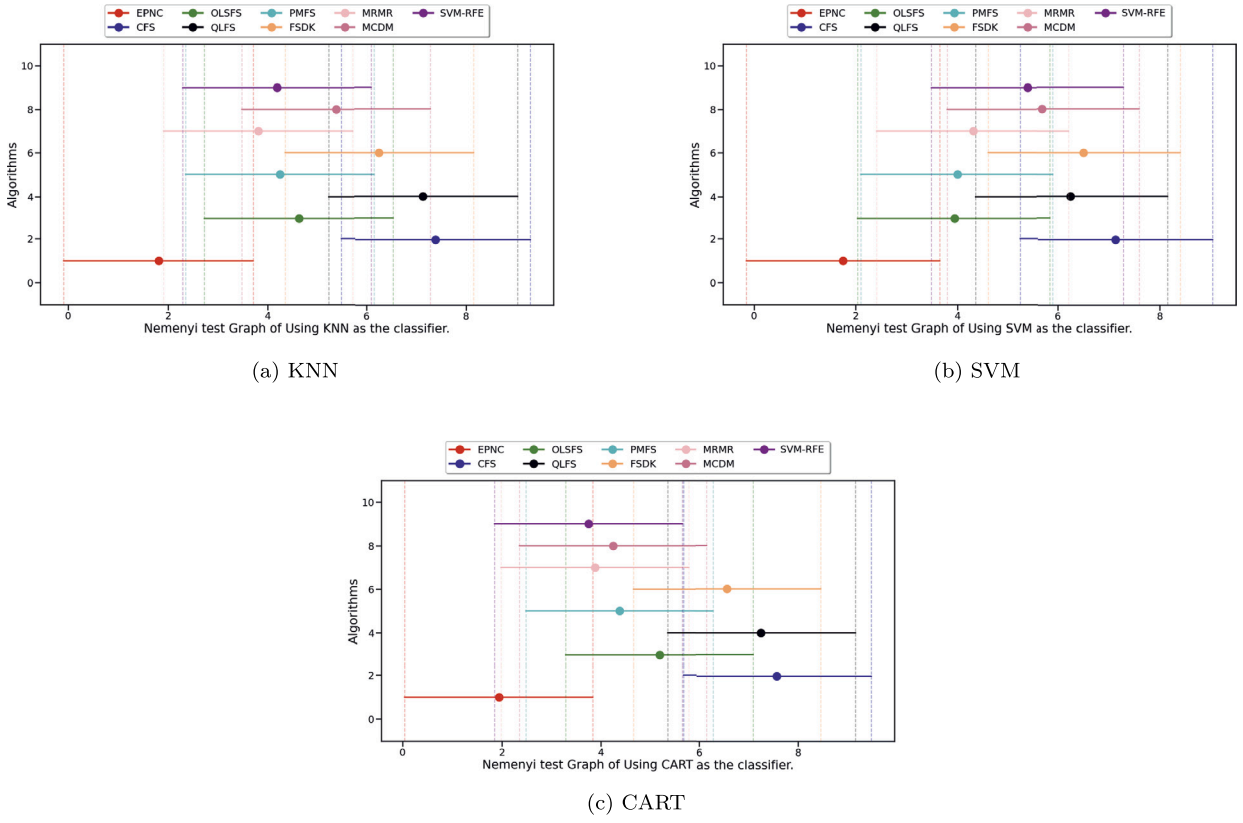
| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.7027 | 0.6857 | 0.6061 | 0.6897 | 0.6842 | 0.6486 | 0.6667 | **0.7568** | 0.7027 |
| CNS | 0.7143 | 0.6667 | 0.7500 | 0.7778 | 0.8421 | 0.7368 | 0.6250 | 0.8235 | **0.8571** |
| Gisette | 0.5824 | **0.8762** | 0.4928 | 0.6695 | 0.6657 | 0.8739 | 0.6571 | 0.8055 | 0.8374 |
| Advertisements | 0.5973 | 0.7650 | 0.6217 | 0.6577 | 0.7417 | **0.7937** | 0.6557 | 0.5899 | 0.7838 |
| Leukemia | 0.8000 | 0.7273 | 0.6667 | 0.9091 | 0.8571 | 0.8966 | 0.8333 | 0.7500 | **0.9231** |
| Colon | 0.8235 | **0.8750** | 0.7778 | 0.7500 | 0.8421 | 0.7692 | 0.8000 | 0.8333 | 0.8571 |
| Prostate-std | 0.7500 | 0.8000 | 0.7368 | 0.8333 | 0.7604 | **0.8421** | 0.8182 | 0.6154 | 0.7692 |
| Prostate | 0.7368 | 0.7500 | 0.6667 | 0.9167 | 0.7692 | 0.7273 | 0.9091 | 0.7778 | **0.9474** |
| Dlbcl | 0.5000 | 0.8571 | 0.5714 | 0.8000 | 0.7500 | 0.6667 | 0.5839 | 0.7273 | **0.8889** |
| Leukemia-4c | 0.6746 | 0.7500 | 0.6523 | **0.9158** | 0.8730 | 0.8857 | 0.7491 | 0.8042 | 0.8926 |
| Srbct | 0.7557 | 0.9286 | 0.6131 | 0.9020 | 0.6224 | 0.9495 | 0.8590 | 0.9273 | **0.9580** |
| Lymphoma-std | 0.9111 | 0.8992 | 0.8333 | **0.9580** | 0.8286 | 0.9161 | 0.8454 | 0.8095 | 0.9137 |
| Lung2 | 0.7010 | 0.8475 | 0.7634 | 0.7218 | 0.7452 | 0.8373 | 0.8603 | 0.9093 | **0.9529** |
| MLL | 0.7833 | 0.9153 | 0.7579 | **0.9394** | 0.7313 | 0.9153 | 0.8593 | 0.8102 | 0.9327 |
| UNSWNB15 | 0.7626 | 0.7550 | 0.7787 | 0.7741 | 0.7860 | 0.7636 | 0.7358 | 0.7585 | **0.8147** |
| CICIDS2017 | 0.6605 | 0.5159 | 0.5850 | 0.6487 | 0.6140 | 0.6918 | 0.5882 | 0.6932 | **0.7149** |
| Ave | 0.7160 | 0.7884 | 0.6796 | 0.8040 | 0.7571 | 0.8071 | 0.7529 | 0.7745 | **0.8591** |
| Ave Rank | 6.50 | 4.75 | 7.56 | 3.81 | 5.25 | 4.13 | 6.00 | 5.06 | **1.81** |

**Table 10**
Comparison of EPNC and Competing Methods on F-Measure Using SVM.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.6667 | **0.8095** | 0.5294 | 0.5946 | 0.5789 | 0.6842 | 0.6111 | 0.5882 | 0.7027 |
| CNS | 0.7368 | 0.7778 | 0.8000 | 0.7059 | 0.8421 | 0.7500 | 0.7143 | 0.8235 | **0.8571** |
| Gisette | 0.7096 | **0.8594** | 0.6957 | 0.6577 | 0.6574 | 0.8539 | 0.6571 | 0.7797 | 0.8374 |
| Advertisements | 0.2887 | 0.7626 | 0.4771 | 0.7246 | 0.5310 | **0.7887** | 0.6557 | 0.3529 | 0.7838 |
| Leukemia | 0.8333 | 0.8571 | 0.7273 | 0.8571 | 0.6000 | 0.9091 | 0.8000 | 0.8889 | **0.9231** |
| Colon | 0.7778 | 0.7143 | 0.8750 | 0.8333 | 0.8235 | 0.7692 | **0.9333** | 0.8421 | 0.8571 |
| Prostate-std | 0.7368 | **0.8889** | 0.5833 | 0.8000 | 0.7826 | 0.8148 | 0.8696 | 0.7778 | 0.7692 |
| Prostate | 0.8182 | 0.8696 | 0.6667 | 0.9167 | 0.6957 | **0.9524** | 0.9000 | 0.9333 | 0.9474 |
| Dlbcl | 0.5455 | 0.6667 | 0.8333 | 0.7692 | 0.8571 | 0.5714 | 0.8000 | 0.7639 | **0.8889** |
| Leukemia-4c | 0.5945 | 0.8857 | 0.6852 | 0.7869 | 0.7270 | **0.9302** | 0.6709 | 0.5789 | 0.8926 |
| Srbct | 0.5833 | 0.9495 | 0.7741 | 0.9167 | 0.8580 | 0.8497 | 0.8951 | 0.9416 | **0.9580** |
| Lymphoma-std | 0.7483 | **0.9161** | 0.8125 | 0.9111 | 0.7483 | 0.8737 | 0.7778 | 0.9161 | 0.9137 |
| Lung2 | 0.8097 | 0.7712 | 0.7569 | 0.5524 | 0.7931 | 0.8578 | 0.7087 | 0.8966 | **0.9529** |
| MLL | 0.7231 | 0.8857 | 0.7852 | 0.8593 | 0.7238 | 0.9153 | 0.8598 | 0.9030 | **0.9327** |
| UNSWNB15 | 0.4998 | 0.3598 | 0.7183 | 0.6348 | 0.6989 | 0.4784 | 0.7066 | 0.5298 | **0.8147** |
| CICIDS2017 | 0.6515 | 0.5673 | 0.5879 | 0.6487 | 0.6230 | **0.7199** | 0.6230 | 0.6977 | 0.7149 |
| Ave | 0.6702 | 0.7838 | 0.7067 | 0.7606 | 0.7213 | 0.7949 | 0.7614 | 0.7634 | **0.8591** |
| Ave Rank | 6.94 | 4.44 | 6.31 | 5.31 | 6.00 | 3.81 | 5.44 | 4.44 | **2.06** |

**Table 11**
Comparison of EPNC and Competing Methods on F-Measure Using CART.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | **0.7429** | 0.6452 | 0.5455 | 0.5854 | 0.5882 | 0.5882 | 0.6875 | 0.6471 | 0.7027 |
| CNS | 0.6250 | 0.5714 | 0.7778 | 0.7059 | 0.7368 | 0.7143 | 0.7500 | 0.8000 | **0.8571** |
| Gisette | 0.6819 | 0.8792 | 0.6749 | 0.6577 | 0.6532 | **0.8832** | 0.6577 | 0.8254 | 0.8374 |
| Advertisements | 0.2979 | 0.7500 | 0.4571 | 0.7246 | 0.7534 | **0.8142** | 0.6557 | 0.6800 | 0.7838 |
| Leukemia | 0.8333 | 0.9091 | 0.7500 | 0.8889 | 0.7692 | 0.8000 | 0.8889 | 0.8571 | **0.9231** |
| Colon | 0.7143 | 0.7778 | 0.8750 | **0.9333** | 0.8000 | 0.8235 | 0.8421 | 0.8889 | 0.8571 |
| Prostate-std | 0.8182 | 0.8462 | 0.8148 | 0.8696 | 0.7273 | **0.9231** | 0.8800 | 0.8868 | 0.7692 |
| Prostate | 0.7826 | 0.8696 | 0.6316 | 0.9412 | 0.8148 | 0.8333 | 0.9231 | 0.9167 | **0.9474** |
| Dlbcl | 0.5714 | 0.8571 | 0.7273 | **0.9091** | 0.6667 | 0.7500 | 0.8000 | 0.6667 | 0.8889 |
| Leukemia-4c | 0.6963 | 0.8750 | 0.7269 | 0.8693 | 0.8519 | 0.8857 | 0.8296 | 0.7661 | **0.8926** |
| Srbct | 0.5262 | 0.8958 | 0.6577 | 0.9020 | 0.6433 | 0.8856 | 0.8643 | 0.8910 | **0.9580** |
| Lymphoma-std | 0.8992 | 0.8125 | 0.9111 | 0.8286 | 0.8095 | **0.9161** | 0.7333 | 0.8125 | 0.9137 |
| Lung2 | 0.6954 | 0.6447 | 0.8726 | 0.6895 | 0.7021 | 0.6212 | 0.8235 | 0.9311 | **0.9529** |
| MLL | 0.7482 | 0.6540 | 0.6996 | 0.8278 | 0.8857 | 0.8750 | 0.7852 | **0.9407** | 0.9327 |
| UNSWNB15 | **0.8208** | 0.8157 | 0.8014 | 0.8137 | 0.8168 | 0.8137 | 0.7973 | 0.8077 | 0.8147 |
| CICIDS2017 | 0.6684 | 0.6037 | 0.6583 | **0.7151** | 0.6394 | 0.6485 | 0.7143 | 0.7038 | 0.7149 |
| Ave | 0.6951 | 0.7754 | 0.7239 | 0.8039 | 0.7411 | 0.7985 | 0.7895 | 0.8139 | **0.8591** |
| Ave Rank | 6.38 | 5.25 | 6.44 | 4.25 | 6.25 | 4.50 | 5.13 | 4.19 | **2.31** |




(a) KNN (b) SVM

(c) CART

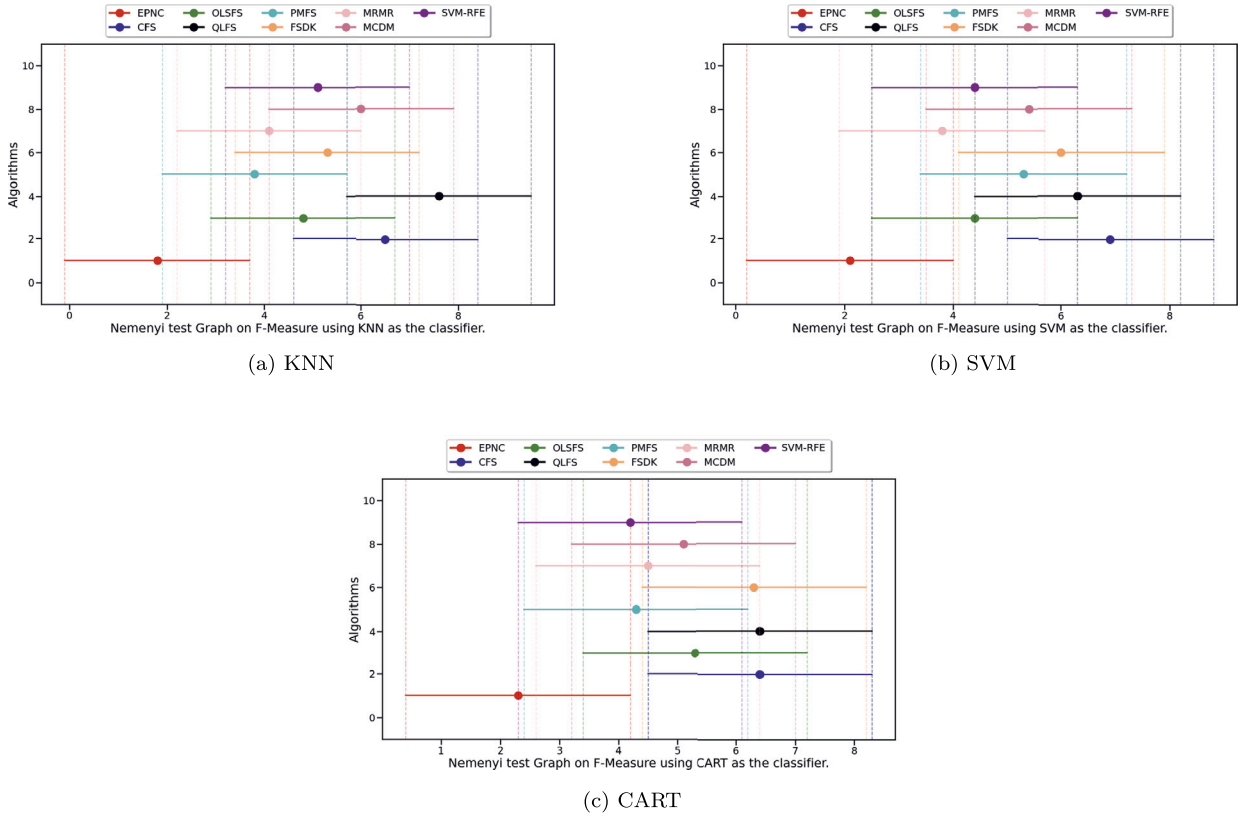
**Fig. 6.** Statistical test for EPNC and competing algorithms on Accuracy.

**Table 12**
Comparison of EPNC and Competing Methods on Stability.

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 0.2891 | 0.2993 | 0.4194 | 0.5295 | 0.2190 | 0.2392 | **0.5996** | 0.3493 | 0.5874 |
| CNS | 0.1787 | 0.0387 | 0.3491 | 0.4392 | 0.1188 | 0.1588 | **0.4793** | 0.2289 | 0.3568 |
| Gisette | 0.6894 | 0.6493 | 0.3287 | 0.3587 | 0.2281 | **0.8196** | 0.5270 | 0.6293 | 0.7984 |
| Advertisements | 0.8188 | 0.7584 | 0.2854 | 0.4968 | 0.5672 | **0.9195** | 0.5270 | 0.5169 | 0.8955 |
| Leukemia | 0.3687 | 0.2887 | 0.3291 | 0.6896 | 0.1086 | 0.3691 | 0.5293 | 0.5293 | **0.8566** |
| Colon | 0.1156 | 0.2060 | 0.3769 | 0.4774 | 0.2050 | 0.4271 | 0.4774 | 0.4673 | **0.6839** |
| Prostate-std | 0.1085 | 0.2187 | 0.3189 | 0.5493 | 0.1084 | 0.3890 | 0.6294 | 0.5493 | **0.7069** |
| Prostate | 0.1185 | 0.2087 | 0.3589 | 0.5893 | 0.1284 | 0.5492 | 0.6895 | 0.5693 | **0.8144** |
| Dlbcl | 0.1086 | 0.2488 | 0.4391 | 0.4992 | 0.1084 | 0.3790 | 0.5393 | 0.5293 | **0.5952** |
| Leukemia-4c | 0.1188 | 0.1388 | 0.5726 | 0.6695 | 0.1488 | 0.5594 | **0.8583** | 0.5293 | 0.8420 |
| Srbct | 0.1463 | 0.3371 | 0.3920 | **0.8594** | 0.4577 | 0.6485 | 0.7087 | 0.7288 | 0.7822 |
| Lymphoma-std | 0.0577 | 0.1178 | 0.4748 | 0.4687 | 0.0075 | 0.1880 | 0.4687 | **0.4787** | 0.1742 |
| Lung2 | 0.0973 | 0.2477 | 0.4835 | **0.8365** | 0.0271 | 0.3179 | 0.3581 | 0.3781 | 0.7350 |
| MLL | 0.0584 | 0.2086 | 0.5595 | 0.6093 | 0.0017 | 0.4891 | **0.6694** | 0.6193 | 0.5996 |
| UNSWNB15 | 0.7362 | 0.7343 | 0.7048 | 0.5276 | 0.3652 | 0.8524 | 0.5837 | 0.4686 | **0.9410** |
| CICIDS2017 | 0.3232 | 0.8853 | 0.2888 | 0.6457 | 0.3948 | **0.9312** | 0.9008 | 0.5871 | 0.8524 |
| Ave | 0.2709 | 0.3491 | 0.4176 | 0.5779 | 0.1997 | 0.5148 | 0.5966 | 0.5099 | **0.7013** |
| Ave Rank | 7.00 | 6.44 | 5.63 | 3.56 | 8.13 | 4.56 | 2.94 | 4.31 | **2.19** |

higher accuracy on average and outperforms OLSFS on most datasets. In terms of stability, EPNC is significantly better than OLSFS. Among these competing algorithms, OLSFS gets the shortest running time. OLSFS is a fast feature selection algorithm based on orthogonal least squares and considers both the correlation between features and global information in the feature selection process. However, OLSFS does not handle well for non-linear models and can not capture the importance of non-linear features that could cause the loss of vital data and information, resulting in reduced predictive accuracy.

- EPNC *vs*. QLFS: QLFS is an incremental, task-oriented, and model-free learning algorithm. It can gradually learn and update the feature selection model to adapt to data changes and new learning tasks. However, due to the need to construct and update discriminant function vectors, QLFS may have higher computational and storage requirements for large-scale datasets, which can lead to a decrease in algorithm efficiency. The experimental data shows that EPNC performs significantly better than QLFS




(a) KNN (b) SVM

(c) CART

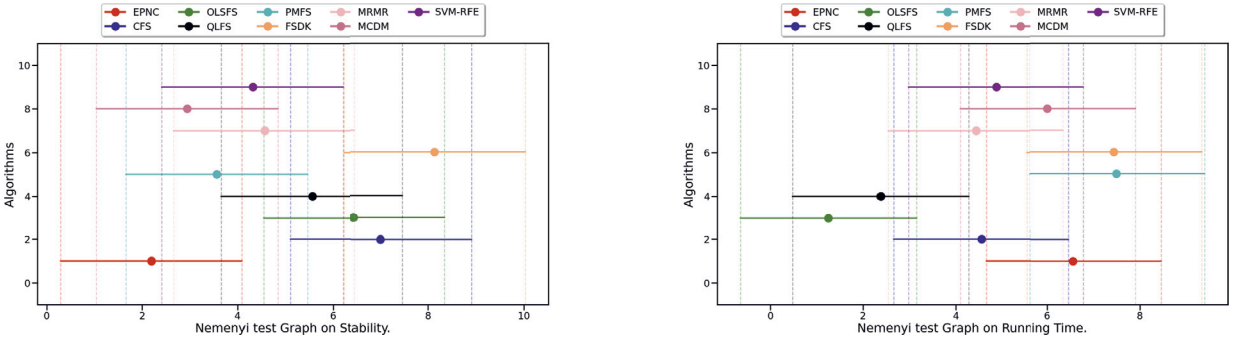
**Fig. 7.** Statistical test for EPNC and competing algorithms on F-Measure.

**Table 13**
Running time(seconds).

| Data Set | CFS | OLSFS | QLFS | PMFS | FSDK | MRMR | MCDM | SVM-RFE | EPNC |
|---|---|---|---|---|---|---|---|---|---|
| Arcene | 3.1991 | **0.0557** | 0.2174 | 380.3341 | 72.5020 | 3.2847 | 259.6121 | 19.6646 | 8.8122 |
| CNS | 0.7270 | **0.0098** | 0.0312 | 133.3603 | 9.1745 | 2.0296 | 90.7762 | 6.2561 | 2.8688 |
| Gisette | 54.7282 | **1.1866** | 2.8295 | 45.8220 | 41.3927 | 3.9780 | 31.9137 | 650.1322 | 756.0843 |
| Advertisements | 5.7140 | **0.0728** | 0.2279 | 0.9334 | 1.5175 | 0.5380 | 0.5100 | 1.4626 | 32.3867 |
| Leukemia | 0.8889 | **0.0121** | 0.0433 | 133.5603 | 63.8723 | 2.1126 | 91.1412 | 0.0906 | 3.1572 |
| Colon | 0.2104 | **0.0038** | 0.0077 | 1.8874 | 3.5180 | 0.5488 | 0.8641 | 0.0531 | 0.8150 |
| Prostate-std | 1.0838 | **0.0135** | 0.0473 | 77.7318 | 65.8804 | 1.7514 | 52.2600 | 0.1097 | 3.3049 |
| Prostate | 1.0564 | **0.0131** | 0.0445 | 75.0917 | 69.9917 | 1.7162 | 50.5239 | 0.1062 | 3.2288 |
| Dlbcl | 0.8314 | **0.0113** | 0.0411 | 89.2834 | 54.2020 | 1.8363 | 59.2470 | 0.0805 | 2.8713 |
| Leukemia-4c | 1.0053 | **0.0143** | 0.0840 | 122.7899 | 11.2786 | 2.5452 | 82.6397 | 22.9050 | 5.6391 |
| Srbct | 0.3751 | **0.0062** | 0.0205 | 2.8647 | 5.4243 | 0.7640 | 1.5120 | 0.2077 | 1.9503 |
| Lymphoma-std | 0.4930 | **0.0078** | 0.0248 | 19.7749 | 18.7764 | 1.3789 | 12.8665 | 0.0910 | 2.2110 |
| Lung2 | 1.3240 | **0.0160** | 0.1176 | 10.2625 | 11.9371 | 1.1414 | 6.1958 | 0.3820 | 7.1372 |
| MLL | 0.8365 | **0.0117** | 0.0748 | 67.5032 | 50.2558 | 2.0564 | 44.6546 | 0.2163 | 3.3937 |
| UNSWNB15 | 0.4824 | 0.0095 | 0.4215 | 0.0092 | 6.0425 | 0.0205 | **0.0038** | 1887.1152 | 19.3163 |
| CICIDS2017 | 1.2502 | 0.0232 | 0.5414 | 0.0041 | 6.1710 | 0.0572 | **0.0108** | 3809.0477 | 72.9354 |
| Ave | 4.6379 | **0.0917** | 0.2984 | 72.5758 | 30.7460 | 1.6100 | 49.0457 | 399.8700 | 57.8820 |
| Ave Rank | 4.56 | **1.25** | 2.38 | 7.50 | 7.44 | 4.44 | 6.00 | 4.88 | 6.56 |

on accuracy and F-measure in KNN, SVM, and CART cases. On average, EPNC is much more stable than QLFS. QLFS is much faster than EPNC in terms of running time. QLFS focuses on preserving local information and comparing features in smaller subsets. In some cases, QLFS may not capture global information effectively.

- EPNC *vs*. PMFS: The PMFS method models the correlation and redundancy of features in the same space and evaluates them separately using the concept of Pareto dominance and clustering methods. This comprehensive consideration helps select feature subsets that have both relevance and low redundancy, improving the effectiveness of feature selection. However, the performance of the Pareto dominance algorithm may decline when the number of objectives increases. Although the PMFS method performs well in handling two objectives, it may have limitations when dealing with problems with a larger number of objectives. The experimental results show no significant difference between EPNC and PMFS on predictive accuracy and F-measure. However,




(a) Stability (b) Running Time

**Fig. 8.** Statistical test for EPNC and competing algorithms on Stability and Running Time.

EPNC gets higher predictive accuracy than PMFS on most datasets in KNN, SVM, and CART cases. Meanwhile, EPNS is more stable than PMFS. In terms of running time, EPNS and PMFS perform similarly. PMFS is a filter-based feature selection method that uses a bi-objective measure to solve multi-objective optimization problems. On most datasets, PMFS spends more running time than EPNC. PMFS involves comparing numerous solutions to obtain results in multi-objective optimization, leading to increased time complexity when dealing with high-dimensional datasets.

- EPNC *vs*. FSDK: The FSDK method exhibits efficiency and flexibility in handling large-scale high-dimensional data for feature selection. However, although the method introduces a weighted pseudo-label matrix to mitigate trivial solutions in unsupervised LSR, the continuous pseudo-label matrix obtained from spectral analysis deviates somewhat from the actual circumstances. According to the statistical test, EPNC performs significantly better than FSDK in predictive Accuracy, F-measure, and stability cases. Regarding running time, FSDK spends more time than EPNC on most datasets. FSDK is an unsupervised feature selection method based on parse discriminative K-means. Since accurate label information is not used, the performance of FSDK is not as good as that of the supervised feature selection algorithms among these competing algorithms.
- EPNC *vs*. MRMR: MRMR method offers the advantages of maximizing relevance and minimizing redundancy, resulting in selecting a subset of highly predictive and independent features. However, it comes with a high sensitivity to data distribution and the assumption of a single target variable. There is no significant difference between EPNC and MRMR in predictive accuracy, F-measure, and stability. EPNC achieves higher average accuracy and lower average ranks than MRMR. Meanwhile, MRMR consumes much less running time than EPNC. MRMR is a classical Mutual Information-based feature selection method that aims to maximize the relevance between features and the target variable while minimizing the redundancy among features. MRMR is an excellent feature selection algorithm but cannot handle continuous datasets directly.
- EPNC *vs*. MFS-MCDM: MFS-MCDM models the feature selection problem as a multi-criteria decision-making process. This approach allows for considering the relationships between multiple labels and provides a more comprehensive evaluation of the correlation between features and labels. It performs well in multi-label feature selection tasks. However, it involves high computational complexity. The algorithm's computational cost can be significant when dealing with large-scale datasets. On predictive accuracy, EPNC outperformed MFS-MCDM significantly in the case of SVM. Regarding F-measure, EPNC significantly performs better than MFS-MCDM in the case of KNN. In other cases, EPNC gets much higher performance on average than MFS-MCDM. MFS-MCDM and EPNC perform similarly in terms of stability and running time. MFS-MCDM is a multi-label feature selection method that utilizes an information fusion approach to integrate evaluations from different labels. Like MFS-MCDM, EPNC can also handle multi-label feature selection tasks by selecting positive and negative features for each class. With the help of the ensemble mechanism, EPNC achieves superior performance.
- EPNC *vs*. SVM-RFE: SVM-RFE utilizes Support Vector Machine models, known for their excellent performance in handling high-dimensional data and nonlinear problems, making them effective for feature selection. By recursively eliminating features, SVM-RFE removes features with minimal impact on the model in each iteration. This recursive process helps identify the most influential subset of features that significantly affect the model's performance and demonstrates good performance on different datasets. According to the statistical test, there is no significant difference between EPNC and SVM-RFE regarding predictive accuracy and F-measure. Regarding average accuracy and F-measure, EPNC is more than 5% higher than SVM-RFE at least. On stability, EPNC is more than 20% higher than SVM-RFE on average. Regarding running time, SVM-RFE spends much more time on average than EPNC. On datasets UNSWNB15 and CICIDS2017, SVM-RFE's execution time far exceeds EPNC's. SVM-RFE requires retraining the support vector machine model in each iteration, significantly impacting the overall running time.

In sum, by using both positive and negative features, EPNC trains multiple classifiers for each dataset and achieves the best performance in predictive accuracy and stability among these competing algorithms. Just like a coin with both sides. EPNC spends much more running time than most of these competing algorithms.

Moreover, Table 14 compares the time complexity of these competing feature selection methods. In the table, the symbol $n$ represents the total number of instances, $m$ represents the number of features in the original feature set, $k$ represents the size of selected features, and $F$ represents the number of iterations required.

**Table 14**
Time complexity of these competing algorithms.

| Method | Type | Computational Complexity |
|---|---|---|
| EPNC | Ensemble, Supervised | $O(n * \|C\| * \Omega + \|n * m\|^2)$ |
| CFS | Filter, Unsupervised | $O(nm + klogk)$ |
| OLSFS | Filter, Supervised | $O(m^3)$ |
| QLFS | Filter, Supervised | $O(n * m)$ |
| PMFS | Filter, Unsupervised | $O(n^2m^2)$ |
| FSDK | Embedded,Unsupervised | $O(nmF)$ |
| MRMR | Filter, Univariate | $O(nm * k + m^2)$ |
| MCDM | Filter, Univariate | $O(n^3 + mn + mn^2)$ |
| SVM-RFE | Wrapper, Supervised | $O(F * n^2m)$ |

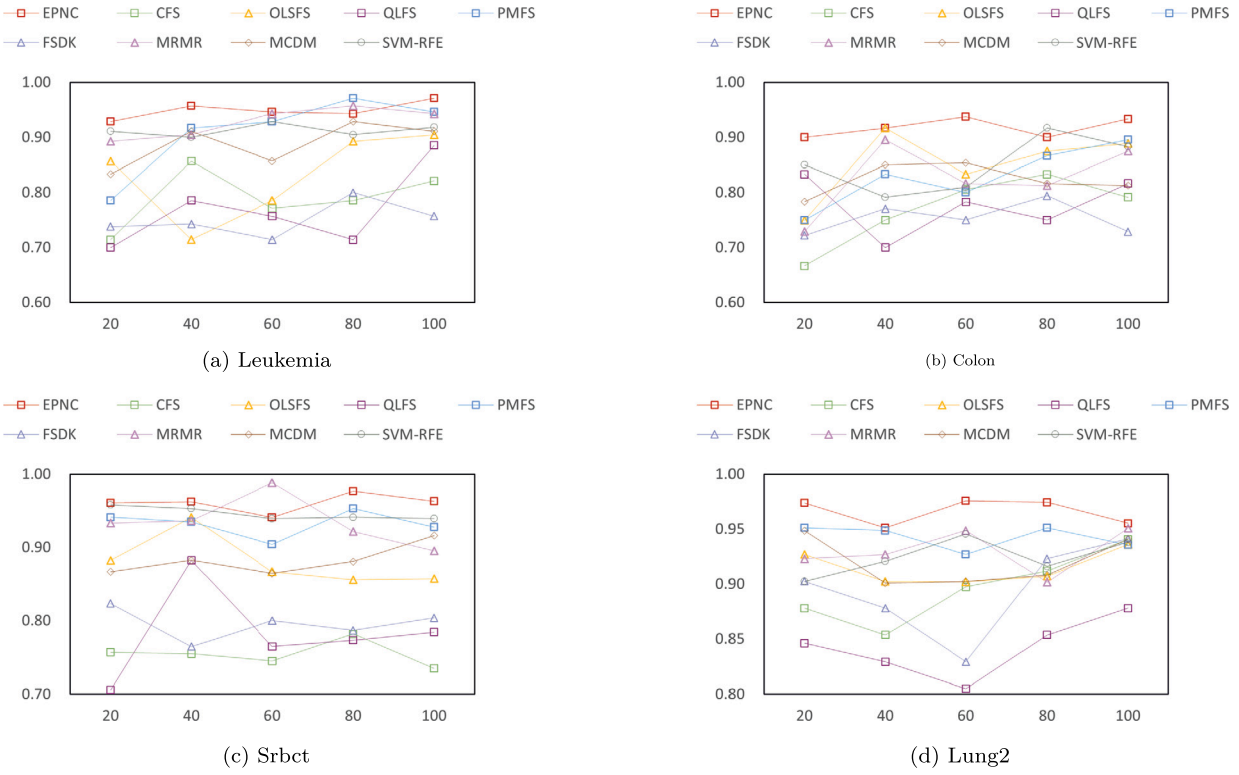



**Fig. 9.** Accuracy of these competing algorithms varying with different numbers of selected features.

### *4.4. Performance with different numbers of selected features*

To test the impact of different numbers of selected features on our new framework, we conducted experiments with competing algorithms on four datasets with different feature numbers (from 20 to 100). Figs. 9 to 11 displays the experimental findings on various datasets with different numbers of features in cases of accuracy, F-measure and stability.

From Figs. 9 to 11, we can observe that:

- Accuracy: With the increase of selected features, EPNC outperforms other feature selection algorithms in predictive accuracy in most cases. Meanwhile, EPNC is stable under different numbers of features. In contrast, the predictive accuracy of the comparative algorithms is significantly affected by the increase in selected feature numbers. For example, when the feature number increases to 60 on dataset Colon, the performance of these competing algorithms drops while EPNC has some improvements.
- F-Measure: In terms of F-measure, EPNC always gets the highest performance among these competing algorithms, varying with different numbers of features. The experimental results revealed that our method outperformed other feature selection techniques, consistently achieving higher F-measure values. This signifies the robustness and effectiveness of our approach in accurately identifying relevant features and optimizing performance. These findings further validate the superiority of our method in handling feature variations and highlight the importance of our feature selection strategy.
- Stability: Regarding stability, EPNC consistently outperforms competing algorithms with the increase of selected features. For instance, on datasets Leukemia, Colon, and Lung2, when the feature number increases to 40, the stability of all these competing algorithms drops while EPNC is very stable and has slight improvements. The selected feature subsets were consistently similar

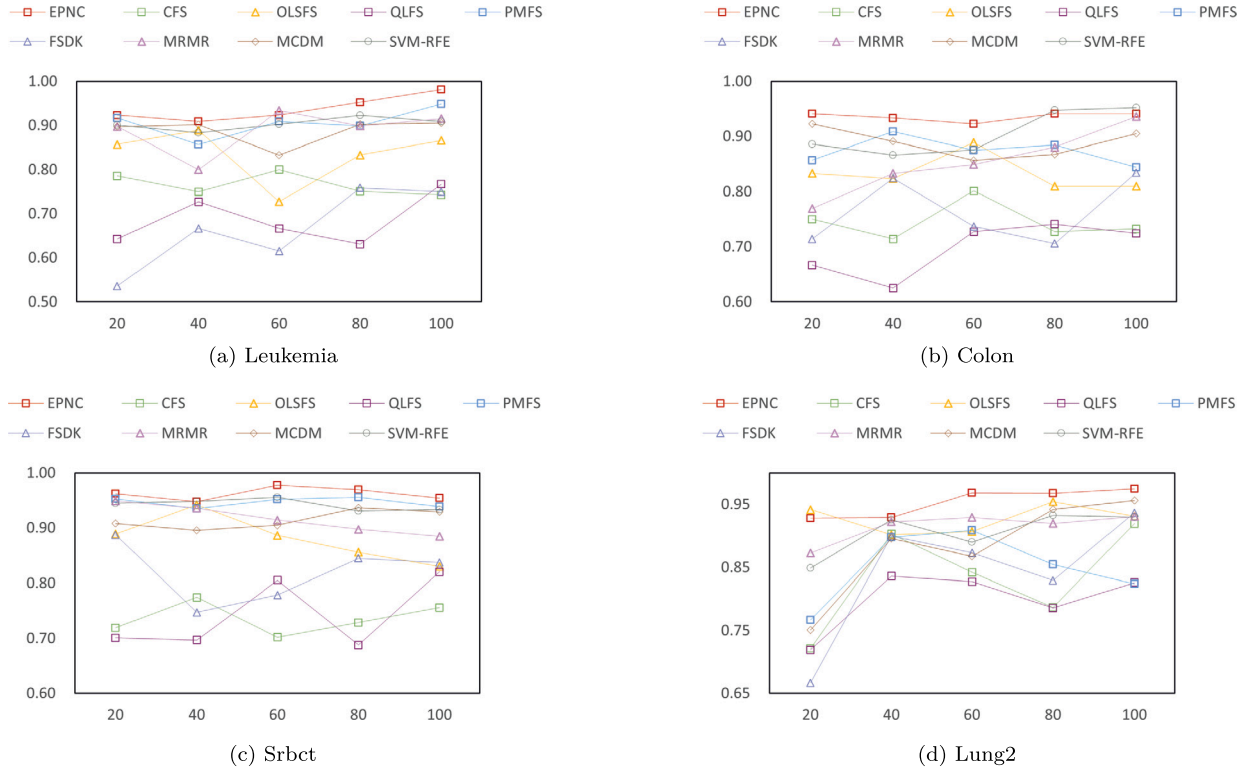



**Fig. 10.** F-Measure of these competing algorithms varying with different numbers of selected features.

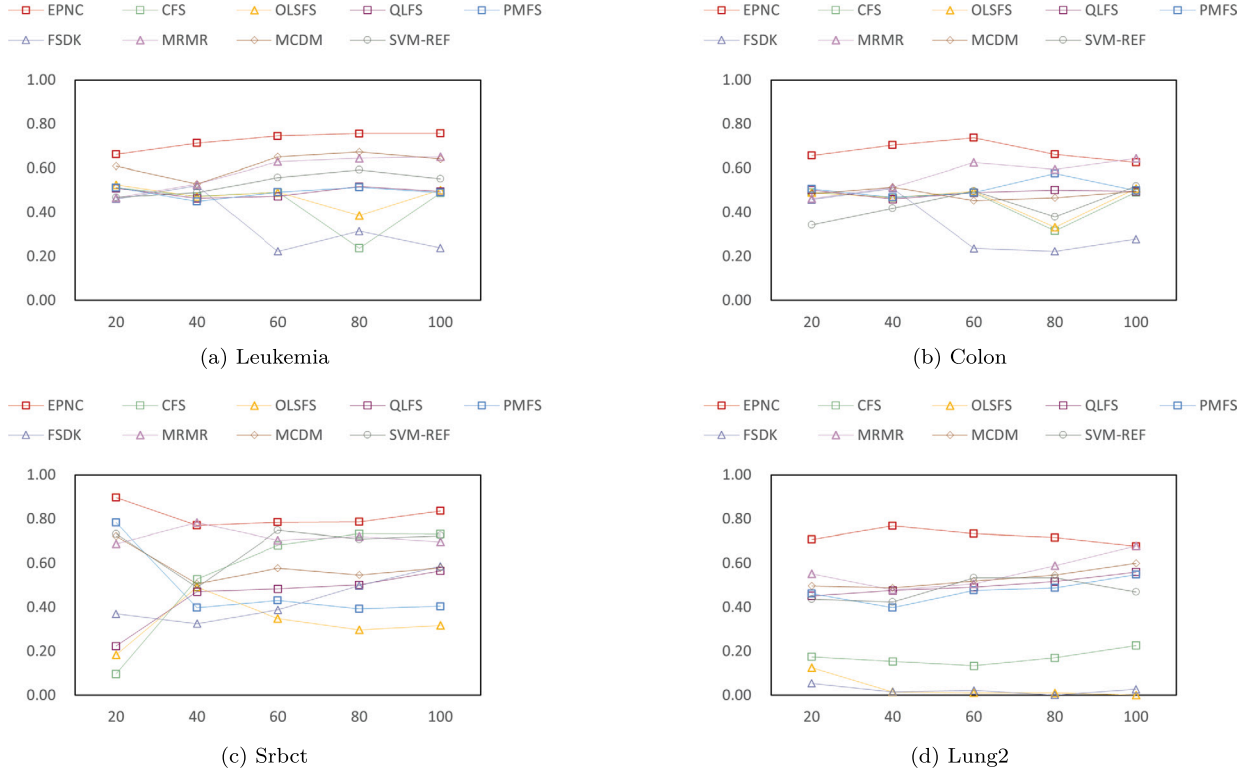



**Fig. 11.** Stability of these competing algorithms varying with different numbers of selected features.

across different subsets of the dataset, indicating that our method was robust to selected feature variations and could adapt to different scenarios. This stability is crucial for ensuring the reliability and reproducibility of the feature selection process.

In sum, ENPC is robust and stable compared to these competing algorithms, varying with different numbers of selected features.

## 5. Discussion

This paper proposes an interpretable feature selection method based on feature polarity and improves classification performance through an ensemble learning framework. Through experimental validation, we demonstrate the effectiveness of the PN coefficient and our classification framework. The following is a summary and discussion of the usefulness of both positive and negative features, the proposed method's generalization ability, advantages, disadvantages, and interpretability.

- Usefulness of both positive and negative features: Positive and negative features have different effects on the performance of the classification task. Positive features contribute positively to the prediction outcome of a specific label, while negative features negate the prediction outcome of a specific label. We train multiple classifiers for the selected positive and negative features for each class in the dataset. A classification model can more accurately identify and classify instances with specific features using positive features. On the other hand, negative features help exclude instances that do not possess specific features, reducing the false positive rate of the classifier. The complementary nature of positive and negative features significantly improves the accuracy and efficiency of classification. In extensive experiments conducted in this study, the proposed method demonstrated excellent performance, validating the effectiveness of positive and negative features.
- Advantages and disadvantages: In the experiments, PNFS and EPNC achieved the lowest average rankings among all these competing algorithms in accuracy, F-measure, and stability, demonstrating excellent performance. Additionally, PNFS is a parameter-free feature selection method that can automatically adjust the positive-negative ratio of selected features, providing good flexibility and adaptability. Meanwhile, using an ensemble learning mechanism, EPNC combines feature polarity and multiple classifiers while improving classification performance and stability. However, PNFS has some limitations in terms of time complexity. PNFS selects positive and negative features for each class, which can be time-consuming for datasets with many class labels. Additionally, EPNC integrates multiple classifiers, resulting in longer classification times.
- Generalization ability: Based on the Kendall coefficient, the PN coefficient can handle both discrete and continuous features, while most existing information measurements are designed to deal with a single feature type. This makes our new approach applicable to the feature selection requirements of different practical application scenarios. Besides, our new ensemble classification framework trains multiple classifiers regarding the selected positive and negative features for each class in the dataset. Based on the superiority of ensemble learning, our new classification framework can achieve stable and superior performance on different datasets. Extensive experiments conducted on real-world datasets validate the generalization ability of our new method.
- Interpretability: The use of positive and negative features enhances the interpretability of the model. By observing and analyzing the impact of positive and negative features, we can understand the basis and decision-making process of the model's predictions. For example, in medical diagnosis, positive features can help doctors determine whether a patient has a specific disease, while negative features can help exclude the possibility of a particular disease. By understanding the importance and polarity of features to the model, we can better comprehend its prediction outcomes and evaluate its credibility and applicability. This is of significant importance in fields such as medical decision-making, financial forecasting, and security detection.

## 6. Conclusion

In this article, we tackle the feature selection problem by proposing an interpretable method from the perspective of feature polarity. We present the formal definition of positive and negative features and define the PN coefficient to calculate the feature polarity. Meanwhile, we can adaptively obtain the proportion of positive and negative features for different datasets. Based on these, we propose a new feature selection method regarding explainability of features. Moreover, we design a novel ensemble classification framework from both positive and negative perspectives. Extensive experiments illustrate that our new feature selection method and ensemble classification framework are effective compared to traditional feature selection coefficients and state-of-the-art feature selection methods.

Indeed, our new method also has some limitations, such as the high time complexity. In future work, we will focus on the theoretical analysis of the proposed method and the relationship between positive and negative feature subsets. Besides, in some real-world applications, the dataset is not static and may exist in stream mode. We will apply feature polarity to online streaming feature selection issues.

## CRediT authorship contribution statement

**Peng Zhou:** Conceptualization, Formal analysis, Funding acquisition, Methodology, Supervision, Writing – review & editing. **Ji Liang:** Writing – original draft, Validation, Software, Methodology, Investigation. **Yuanting Yan:** Funding acquisition, Formal analysis. **Shu Zhao:** Funding acquisition. **Xindong Wu:** Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

Data will be made available on request.

## Acknowledgements

This work is supported in part by the National Natural Science Foundation of China under grants (62376001, 62376002, 62120106008), and the Science Foundation of Anhui Province of China under grants 2308085MF215.

## References

[1] Hervé Abdi, The Kendall rank correlation coefficient, in: Encyclopedia of Measurement and Statistics. Sage, Thousand Oaks, CA, 2007, pp. 508–510.
[2] Thomas Abeel, Thibault Helleputte, Yves Van de Peer, Pierre Dupont, Yvan Saeys, Robust biomarker identification for cancer diagnosis with ensemble feature selection methods, Bioinformatics 26 (3) (2010) 392–398.
[3] Amazona Adorada, Ratih Permatasari, Panji Wisnu Wirawan, Adi Wibowo, Adi Sujiwo, Support vector machine-recursive feature elimination (svm-rfe) for selection of microrna expression features of breast cancer, in: 2018 2nd International Conference on Informatics and Computational Sciences (ICICoS), IEEE, 2018, pp. 1–4.
[4] Khawaja Tehseen Ahmed, Shahida Ummesafi, Amjad Iqbal, Content based image retrieval using image features information fusion, Inf. Fusion 51 (2019) 76–99.
[5] Mebarka Allaoui, Mohammed Lamine Kherfi, Abdelhakim Cheriet, Considerably improving clustering algorithms using umap dimensionality reduction technique: a comparative study, in: International Conference on Image and Signal Processing, Springer, 2020, pp. 317–325.
[6] Rubul Kumar Bania, R-gefs: condorcet rank aggregation with graph theoretic ensemble feature selection algorithm for classification, Int. J. Pattern Recognit. Artif. Intell. 36 (09) (2022) 2250032.
[7] Umang Bhatt, Alice Xiang, Shubham Sharma, Adrian Weller, Ankur Taly, Yunhan Jia, Joydeep Ghosh, Ruchir Puri, José MF Moura, Peter Eckersley, Explainable machine learning in deployment, in: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, 2020, pp. 648–657.
[8] Verónica Bolón-Canedo, Amparo Alonso-Betanzos, Ensembles for feature selection: a review and future trends, Inf. Fusion 52 (2019) 1–12.
[9] Verónica Bolón-Canedo, Noelia Sánchez-Marono, Amparo Alonso-Betanzos, José Manuel Benítez, Francisco Herrera, A review of microarray datasets and applied feature selection methods, Inf. Sci. 282 (2014) 111–135.
[10] Jun Chen, Meng Yang, Wenping Gong, Yang Yu, Multi-neighborhood guided Kendall rank correlation coefficient for feature matching, IEEE Trans. Multimed. (2022) 1–15.
[11] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, Israel Cohen, Pearson correlation coefficient, in: Noise Reduction in Speech Processing, 2009, pp. 1–4.
[12] Janez Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (1) (2006) 1–30.
[13] Kevin Dunne, Padraig Cunningham, Francisco Azuaje, Solutions to instability problems with sequential wrapper-based approaches to feature selection, J. Mach. Learn. Res. 1 (22) (2002).
[14] Quanquan Gu, Zhenhui Li, Jiawei Han, Generalized Fisher score for feature selection, preprint, arXiv:1202.3725, 2012.
[15] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Franco Turini, Fosca Giannotti, Dino Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (5) (2018) 1–42.
[16] Hüseyin Güney, Feature selection-integrated classifier optimisation algorithm for network intrusion detection, Concurr. Comput., Pract. Exp. 35 (23) (2023) e7807.
[17] Huseyin Guney, Huseyin Oztoprak, A robust ensemble feature selection technique for high-dimensional datasets based on minimum weight threshold method, Comput. Intell. 38 (5) (2022) 1616–1658.
[18] Isabelle Guyon, Jason Weston, Stephen Barnhill, Vladimir Vapnik, Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[19] Emrah Hancer, Bing Xue, Mengjie Zhang, Differential evolution for filter feature selection based on information theory and feature ranking, Knowl.-Based Syst. 140 (2018) 103–119.
[20] Amin Hashemi, Mohammad Bagher Dowlatshahi, Hossein Nezamabadi-Pour, Mfs-mcdm: multi-label feature selection using multi-criteria decision making, Knowl.-Based Syst. 206 (2020) 106365.
[21] Amin Hashemi, Mohammad Bagher Dowlatshahi, Hossein Nezamabadi-pour, An efficient Pareto-based feature selection algorithm for multi-label classification, Inf. Sci. 581 (2021) 428–447.
[22] Amin Hashemi, Mehdi Joodaki, Nazanin Zahra Joodaki, Mohammad Bagher Dowlatshahi, Ant colony optimization equipped with an ensemble of heuristics through multi-criteria decision making: a case study in ensemble feature selection, Appl. Soft Comput. 124 (2022) 109046.
[23] Amin Hashemi, Mohammad-Reza Pajoohan, Mohammad Bagher Dowlatshahi, Nsofs: a non-dominated sorting-based online feature selection algorithm, Neural Comput. Appl. 36 (3) (2024) 1181–1197.
[24] Zhuoxin He, Yaojin Lin, Chenxi Wang, Lei Guo, Weiping Ding, Multi-label feature selection based on correlation label enhancement, Inf. Sci. 647 (2023) 119526.
[25] Hanlin Hu, Jiale Guo, Yifan Wu, Wei Nai, Zan Yang, Dan Li, t-sne dimensionality reduction method based on sobol sequence initialized archerfish hunting optimizer algorithm, in: 2022 7th International Conference on Computational Intelligence and Applications (ICCIA), 2022, pp. 43–47.
[26] Don H. Johnson, Signal-to-noise ratio, Scholarpedia 1 (12) (2006) 2088.
[27] Mahdieh Labani, Parham Moradi, Fardin Ahmadizar, Mahdi Jalili, A novel multivariate filter method for feature selection in text classification problems, Eng. Appl. Artif. Intell. 70 (2018) 25–37.
[28] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, Huan Liu, Feature selection: a data perspective, ACM Comput. Surv. 50 (6) (2017) 1–45.
[29] Yaojin Lin, Haoyang Liu, Hong Zhao, Qinghua Hu, Xingquan Zhu, Xindong Wu, Hierarchical feature selection based on label distribution learning, IEEE Trans. Knowl. Data Eng. 35 (6) (2023) 5964–5976.
[30] Wei Liu, Jianyu Wang, Recursive elimination current algorithms and a distributed computing scheme to accelerate wrapper feature selection, Inf. Sci. 589 (2022) 636–654.

[31] Yi Mei, Qi Chen, Andrew Lensen, Bing Xue, Mengjie Zhang, Explainable artificial intelligence by genetic programming: a survey, IEEE Trans. Evol. Comput. 27 (3) (2023) 621–641.
[32] Feiping Nie, Zhenyu Ma, Jingyu Wang, Xuelong Li, Fast sparse discriminative k-means for unsupervised feature selection, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[33] Babak Nouri-Moghaddam, Mehdi Ghazanfari, Mohammad Fathian, A novel multi-objective forest optimization algorithm for wrapper feature selection, Expert Syst. Appl. 175 (2021) 114737.
[34] Zhixin Pan, Jennifer Sheldon, Prabhat Mishra, Hardware-assisted malware detection and localization using explainable machine learning, IEEE Trans. Comput. 71 (12) (2022) 3308–3321.
[35] Hanchuan Peng, Fuhui Long, Chris Ding, Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy, IEEE Trans. Pattern Anal. Mach. Intell. 27 (8) (2005) 1226–1238.
[36] Barbara Pes, Ensemble feature selection for high-dimensional data: a stability analysis across multiple domains, Neural Comput. Appl. 32 (10) (2020) 5951–5973.
[37] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.
[38] Saúl Solorio-Fernández, J. Ariel Carrasco-Ochoa, José Fco Martínez-Trinidad, A review of unsupervised feature selection methods, Artif. Intell. Rev. 53 (2) (2020) 907–948.
[39] Omid Tarkhaneh, Thanh Thi Nguyen, Samaneh Mazaheri, A novel wrapper-based feature subset selection method using modified binary differential evolution algorithm, Inf. Sci. 565 (2021) 278–305.
[40] Ankit Thakkar, Ritika Lohiya, Attack classification using feature selection techniques: a comparative study, J. Ambient Intell. Humaniz. Comput. 12 (2021) 1249–1266.
[41] Erico Tjoa, Cuntai Guan, A survey on explainable artificial intelligence (xai): toward medical xai, IEEE Trans. Neural Netw. Learn. Syst. 32 (11) (2021) 4793–4813.
[42] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, Dan Suciu, On the tractability of shap explanations, J. Artif. Intell. Res. 74 (2022) 851–886.
[43] Jorge R. Vergara, Pablo A. Estévez, A review of feature selection methods based on mutual information, Neural Comput. Appl. 24 (2014) 175–186.
[44] Ruohao Xu, Mengmeng Li, Zhongliang Yang, Lifang Yang, Kangjia Qiao, Zhigang Shang, Dynamic feature selection algorithm based on q-learning mechanism, Appl. Intell. (2021) 1–12.
[45] Dianlong You, Siqi Dong, Shina Niu, Huigui Yan, Zhen Chen, Shunfu Jin, Di Wu, Xindong Wu, Local causal structure learning for streaming features, Inf. Sci. (2023) 119502.
[46] Sikai Zhang, Zi-Qiang Lang, Orthogonal least squares based fast feature selection for linear classification, Pattern Recognit. 123 (2022) 108419.
[47] Peng Zhou, Peipei Li, Shu Zhao, Xindong Wu, Feature interaction for streaming feature selection, IEEE Trans. Neural Netw. Learn. Syst. 32 (10) (2021) 4691–4702.
[48] Zheming Zuo, Jie Li, Han Xu, Noura Al Moubayed, Curvature-based feature selection with application in classifying electronic health records, Technol. Forecast. Soc. Change 173 (2021) 121127.