The Journal of Supercomputing (2020) 76:5745–5762
https://doi.org/10.1007/s11227-019-02975-7

A new improved filter-based feature selection model for high-dimensional data

Deepak Raj Munirathinam[1] · **Mohanasundaram Ranganadhan**[1]

Published online: 26 August 2019

Abstract

Preprocessing of data is ubiquitous, and choosing significant attributes has been one of the important steps in the prior processing of data. Feature selection is used to create a subset of relevant feature for effective classification of data. In a classification of high-dimensional data, the classifier usually depends on the feature subset that has been used for classification. The Relief algorithm is a popular heuristic approach to select significant feature subsets. The Relief algorithm estimates feature individually and selects top-scored feature for subset generation. Many extensions of the Relief algorithm have been developed. However, an important defect in the Relief-based algorithms has been ignored for years. Because of the uncertainty and noise of the instances used for measuring the feature score in the Relief algorithm, the outcome results will vacillate with the instances, which lead to poor classification accuracy. To fix this problem, a novel feature selection algorithm based on Chebyshev distance-outlier detection model is proposed called noisy feature removal-Relief, NFR-ReliefF in short. To demonstrate the performance of NFR-ReliefF algorithm, an extensive experiment, including classification tests, has been carried out on nine benchmarking high-dimensional datasets by uniting the proposed model with standard classifiers, including the naïve Bayes, C4.5 and KNN. The results prove that NFR-ReliefF outperforms the other models on most tested datasets.

Keywords Classification · Data mining · Feature selection · Relief · Bioinformatics · Noisy feature

✉ Mohanasundaram Ranganadhan
mohan.sundhar@gmail.com

Deepak Raj Munirathinam
deepakr416@gmail.com; deepakraj.dm@vit.ac.in

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, Tamil Nadu 632014, India

1 Introduction

Preprocessing of data is an important process in data mining and machine learning models. Feature weighting or feature scoring is an essential step in data preprocessing, specifically in gene-based selection for cancer classification. In the era of big data, healthcare generates vast amounts of data every day. Inherently, healthcare data contain complex, high-dimensional data or microarray data, etc. The increasing abundance of gene-sequence data made possible by high-throughput techniques has triggered far-flung interest in associate sequence information to biological phenotypes. However, the gene expression data merely consist of voluminous gene samples, but with a small sample size. Therefore, feature selection is essential for resolving such problems. Reducing the dimension of the large feature space and selecting the most informative genes for effective classification process with new or existing classifier are general and are commonly adopted techniques in empirical studies. In the context of feature selection, feature weight can be achieved by assigning a continuous relevance value of each feature through the learning algorithms. The feature scoring method is specifically used for instance-based learning models, in which the distance of a metric is typically build up using all features. Moreover, measure or weight of the feature can help to remove noisy features and reduce the overfitting problem, thereby improving the predictive accuracy [1, 2]. In general, the feature selection process includes four important steps: subset generation, subset evaluation, stopping criteria and validation as shown in Fig. 1. Subset generation is considered an important step in the feature selection process, it generates a subset of optimal features using by the certain search strategy, and it can be divided into

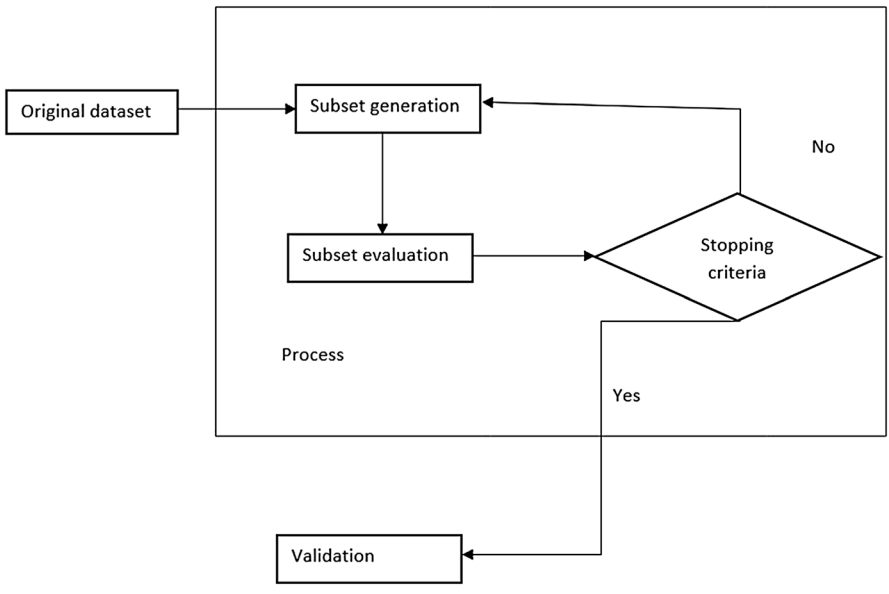

Fig. 1 General feature selection procedure

search direction and search strategy [3, 4]. The search direction is used to direct the starting point of instance search, and it can be classified into backward, forward and bidirectional. In contrast, search strategy starts with the null set and appends the feature simultaneously, and it can be distinguished as a complete search, sequential search and randomized search. Subset evaluation uses certain evaluation criterion to evaluate the generated subset, and it can be divided into dependent, independent and hybrid criterion [5–7]. Different evaluation techniques have been used for feature selection over years, and it can be broadly categorized into four types based on information gain, distance, dependency, consistency, and classifier accuracy, and then, stopping criteria are used to stop the feature selection process. Finally, the validation process is used to measure the classifier model and resultant subset. Validation is performed by classifier error rate test to rate the selected subset, i.e., to find the goodness of the feature subsets [8]. Various researches and development have been carried out till date, and all along the research is kept continue to claim that there is no universal best model available to fit all such problems.

FS can be segregated into three main categories such as filter method, wrapper method and embedded method, abundant of definitions and explanation about FS and its process has been given in [9–11]. Recent trends, techniques, reviews and limitations have been explained and are detailed in [12, 13]. Wrapper method is a familiar and a classifier-dependent method for predictive model analysis [14]. The wrapper methods measure the quality of selected subsets by using the classifier model. The wrapper method can provide better accuracy than the filter method, but is computationally expensive when the dataset is larger in size. Filter method is a classifier-independent model, reliable, simple and efficient in result execution. Filter method is one of the oldest methods for selecting a significant feature, the filter process can estimate the feature subset by scoring method, and the top feature is considered for feature weights. Several methods have been developed for filter-based feature measure such as Relief [15, 16], Fisher score ratio [17], entropy-based measurements [18] and information gain [19–21].

The rapid growth of bioinformatics creates a larger number of feature sets such as microarray datasets that are typically known as high-dimensional data, which generally contains greater noise, higher redundant and irrelevant data and can lead to degrading the performance of classification. Calculating the significant score of features in microarray dataset is a crucial task. Filter method is a simple and effective model for significant feature weighting which can help to create a subset of the optimal feature set, especially on gene expression data analysis, whereas Relief [6] is considered as one of the best models of filter methods and it is limited to two-class problem [22, 23]. The extension of Relief called ReliefF virtuous is used to handle the multi-class problem [9]. Uniting Relief and ReliefF, further development has been carried out over the years such as R-Relief, SURF, TURF, SURF*, TWRF, MultiSurf, MultiSurf* and RG-Relief. RG-Relief performs well to handle redundant feature, but not that much effect to handle noisy feature. In Sect. 3, we will discuss how the proposed algorithms perform to handle noisy features by incorporating a distance-based outlier detection routine method [24]. Various researches have been carried out to handle redundant and irrelevant feature. However, no reliable and effective model has been developed to handle noisy features. To overcome the stated

limitation in this paper, we have developed a Relief-based model that can handle the noisy feature effectively [25].

2 Related work

2.1 Background

Typically, feature selection model has also been broadly characterized into subset evaluation or individual evaluation. Individual evaluation measures each feature and then assigns scores/weight according to the degree of relevance. In contrast, subset evaluation creates a set of some features called a candidate or subset features, and measures the subsets based on a certain search strategy (discussed in Sect. 1). All along, filter, wrapper or embedded methods fall either individual evaluation or subset evaluation methods. Relief and ReliefF are a well-known model for individual evaluation filter algorithm proficient in detecting feature dependencies based on the concept of nearest neighbors, and it helps to derive the feature statistics that indirectly interpret feature interactions. ReliefF is the extended version of the authentic Relief algorithm, and several extensions have been developed such as R-Relief, SURF, TURF, SURF*, TWRF, MultiSurf, MultiSurf* and STIR and grouped all together named as RBA family (Relief-based algorithm family) [26–28].

2.1.1 Relief

Relief is an individual feature ranking algorithm. Relief is a univariate and filter-based feature selection algorithm developed by Kira and Rendell by inspiring the method of instance-based learning [6, 7]. Relief is a very simple, reliable and effective algorithm to select significant features. Relief can distinguish by three steps: (1) data preprocessing, (2) pre-computing pairwise distance array and (3) selecting nearest neighbors and calculating feature weights. Relief is designed to handle the only binary class problem with the discrete and numerical feature. Relief estimates the feature individually and ranks the feature, according to the degree of relevance. After ranking, the feature Relief selects the top scores feature for setting off their nearest neighbors instances and estimates the nearest hit value and nearest miss value. The feature score is used to weight the instance that has been selected based on nearest neighbors from the feature space called feature value difference. Assume if the feature value difference is observed within the same target class with respect to the neighbor distance, called nearest hit 'h,' in case if the feature value difference is observed in different class with respect to the neighbor distance called near miss 'm,' then the feature score can be updated, as shown in Algorithm 1.

$$W_i = W_i - \left(x_i - H_i\right)^2 + \left(x_i - M_i\right)^2 \tag{1}$$

From Eq. 1, W_i can represent a feature weight measure corresponding to the instance i, $x_i \rightarrow$randomly selected instance. H_i is used for assigning nearest hit instance, and M_i assigns nearest miss instance. According to the reference of [6, 7],

let assume that given dataset D contains the largest number of n instances, and larger number of features p, and all are belonging to known class C. Therefore, each feature in the given datasets is scaled between two intervals [0,1]; hence, if interval lies <0, it means worst, and if it lies $>0 \leq 1$, it means best. Relief often repeats the same iteration at m times; initially, feature weight vector starts with zero value, i.e., $W_i=0$, and calculates most nearby instance value x, concerning the larger features p. In general, Relief estimates and weights the feature if the relevant value of each instance is greater than the fixed default threshold value θ. Kira and Rendell's experiments result exhibits an optimal divergence between irrelevant and relevant feature corresponding to the threshold θ. Typically, the threshold value can be estimated by Chebyshev's inequality measure with a given confidence level represented as (θ) that at θ of 1/sqrt (θ*m) is maximum to good enough to define the probability of type I error $<\theta$, i.e., a relevant threshold value must be $0 \leq \theta \leq 1$] [8].

Algorithm:1 Pseudo code of basic Relief

```
Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes
set all weights W[A]:=0
for i:=1 to m do begin
   randomly select an instance R_i ;
   find nearest hit H and miss M ;
   for A: =1 to a do
      W[A]:=W[A]-diff(A, R_i,H)/m + W[A]-diff(A,R_i,M)/m
   end for
end
```

2.1.2 Diff function in Relief (nearest neighbors)

The key objective of *Diff* function is to compute the distance between two instances for finding nearest difference value, i.e., hits 'H' and miss 'M' neighbors to the corresponding attribute 'A,' and target class C often depends on the randomly selected instance R_i, and nearest hit H and miss M (ex. *Diff* (A (R_i (H, M)), as shown in Eq. 2. Let us assume if the observed value of attribute 'A' differs concerning the instance R_i and H, therefore each attribute of 'A' splits two instances within the same target class which is not feasible, so the quality estimation of $W[A]$ will be reduced. The iteration starts again with the instance R_i and H which have different values for the corresponding attribute 'A'; then again, the attribute 'A' splits two instances with the different target class, so the quality measure of $W[A]$ will increase. The entire process is kept repeated m times, where m can be considered as a user-defined parameter.

$$w[A] = W[A] - \frac{Diff\left(A, R_i, H\right)}{m} + \frac{Diff\left(A, R_i, M\right)}{m} \tag{2}$$

where A is the *attribute,* R_i *is the randomly selected instances* ($i=1, 2, 3\ldots n$), H *is the nearest hit,* M is the *nearest miss and* m is the *user-defined parameter.*

Feature weighting can be initialized as $W[A]$ which can divide the output of the function diff by m. Typically, all feature weights can be normalized between the intervals of $[-1, 1]$ by slightly adjusting the target class. Therefore, the function of *Diff* can also be applied to pre-compute the distance instance array. To pre-compute, the distance array can be measured through Manhattan distance or Euclidean distance measure with the corresponding metric values. Therefore, function *Diff* (A, I_1, I_2) calculates the observed difference values between two instances I_1 and I_2; for nominal attributes, the *Diff* function is defined as:

$$Diff\left(A, I_1, I_2\right) = \begin{cases} 0;\ value(A, I_2 = value\left(A, I_1\right) \\ 1;\ otherwise \end{cases} \tag{2}$$

For numerical attributes, *Diff* function is defined as:

$$Diff\left(A, I_1, I_2\right) = \frac{\left|value\left(A, I_1\right) - value\left(A, I_2\right)\right|}{max(A) - max(A)} \tag{3}$$

From Eq. 3, function *Diff* measures features score to update attribute that lies between 0 and 1 with the corresponding instance, which often depends on attribute *'A'* for both numerical and discrete attributes. Therefore, the *Diff* function precisely defined to perform on both the continuous and numeric features. However, the function *Diff* is an essential process that measures Relief importance score to estimate the optimal feature weight $W[A]$ [8]. Relief does not deal with the multi-class problem, it is limited with an only two-class problem, and it also limited to handle noisy features problem, i.e., original Relief designed to handle irrelevant and redundant feature corresponding to numerical and nominal attributes. Relief is also limited to deal with missing data or incomplete data [29, 30].

2.2 ReliefF

ReliefF is an updated version of the original Relief algorithm, specially designed for handling multi-class problems. ReliefF simplifies most of the Relief ideas. The main objective of ReliefF is to improve classification accuracy, particularly on high-dimensional data. ReliefF can deal with multi-class problems as well hand in hand, and it can deal to handle redundant and irrelevant features [9]. ReliefF also deals with incomplete features and noisy features. Algorithm 2 describes how ReliefF can effectively select instances R_i randomly like Relief does, after it searches k for the nearest neighbors within the same target class defined as nearest hits H_j, and again, the iteration continues to search 'k' of nearest neighbor from the different target classes called nearest misses $M_j(C)$. ReliefF also measures the feature weight called feature weight update $W[A]$ corresponding to the attribute 'A,' where 'A' often depends on their values for R_i, corresponding to hits H_j and misses $M_j(C)$ as shown in Eq. 4.

The modified formula is slightly reflecting original Relief. However, ReliefF excludes to measure the average performance of both hits and misses neighbor. The performance of each class of misses and hits can be weighted with corresponding

to the probability of target class $P(c)$, i.e., the input instance of hits and misses can be defined as (0, 1). Therefore, the target class can be considered as the asymmetric value which means we should have to confirm that the missed probability score can help to weight as a sum to 1. According to the target class of hits and miss, the probability of feature weight can be split with the factor 1-P ($class$ (R_i)) shown in Eq. 4. Therefore, this iteration process is kept repeating till m times, and the only difference of ReliefF compared to Relief is a selection of 'k' ($k \rightarrow$ user-defined parameter controls local estimation) corresponding to hits and miss. ReliefF depends on several neighbors and user-defined parameter, ReliefF can be extended with numerous updates, such as TURF, E-Relief, Relief-MMS, SURF and SURF*, and all these variants are grouped in the name of the RBA (Relief-based algorithm) family [1, 2]. ReliefF is widely considered an algorithm to handle irrelevant and relevant data effectively, but no reliable or effective model can handle the noisy feature. Biomedical data contain a larger feature, missing values, noisy feature, incomplete feature and irrelevant feature, particularly high on gene expression data which lead to the poor classification of data causing poor in disease diagnosis especially crucial in cancer research [31]. ReliefF outperforms handling irrelevant feature. However, ReliefF failed to handle a noisy feature effectively [1, 2, 29].

Algorithm:2 Pseudo code of ReliefF

Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes
set all weights W[A] = 0
for *i =1 to m do begin*
  randomly select an instance R_i
  find k nearest hits Hj
  for *each class* $C \neq class(R_i)$ *do*
    from class C find k nearest misses $M_j(C)$*;*
    for *A: = 1 to a do*

$$w[A] = W[A] - \sum_{j=1}^{k} Diff\frac{(A,R_i,H_j)}{m.k} + \sum_{c\neq class(R_i)}\left[\frac{P(c)}{1-P(class(R_i))}\sum_{j=1}^{k} Diff\frac{(A,R_i,M_j(C))}{m.k}\right]$$

    end for
  end for
end
return *vector W of feature scores*

$$w[A] = W[A] - \sum_{j=1}^{k} Diff\frac{(A,R_i,H_j)}{m.k} + \sum_{c\neq class(R_i)}\left[\frac{P(c)}{1-P(class(R_i))}\sum_{j=1}^{k} Diff\frac{(A,R_i,M_j(C))}{m.k}\right] \quad (4)$$

2.3 Problem description

In the era of big data, modeling microarray data containing thousands of attributes is particularly important of genome-expression profiling experiments; nonetheless, it remains a challenging task. One of the notable challenges is to design and implement an effective model for selecting a small set of informative or relevant genes, buried in microarray irrelevant noises. Relief and its variants are a popular and

widespread method for feature selection owing to its high accuracy and low computational cost. However, Relief-based methods usually suffer from instability, specifically in the presence of noise or outlier features.

3 Proposed method

The key objective of the proposed algorithm is to find and remove the noisy features from the feature subsets, and the proposed algorithm is shown in algorithm 3 called NFR-ReliefF. NFR is evolved from the Chebyshev distance-outlier detection method, and it incorporates the basics of ReliefF and slightly adjusts the feature weight measure $W[i]$ corresponding to function *Diff* as shown in Eq. 5. NFR-R intakes both variance and mean of the discrimination among two instances into account as the evaluation criterion of the feature score measure, which produce more stable and accurate results. The proposed model introduces a new routine to handle the noisy feature based on distance-outlier detection routine, and it measures the quality of attribute by selecting instances randomly concerning i, and then, it splits two instances within the same class to increase the quality of subset which contains reliable features. The nearest neighbors of each instance can be measured according to the pre-defined threshold value. Therefore, each of the instances is estimated individually to acquire informative attributes.

$$w[i] = W[i] + diff(X(i), near\,miss(i)X)^2 - diff(X(i), near\,hit(i)X)^2 \tag{5}$$

From Eq. 5, '$W[i]$' represents a feature score or feature weight corresponding to the instance I, $x \rightarrow$ randomly selected instance. *diff(X(i) → represents difference value lies between nearest miss and nearest hits)*, assume in a given dataset D contain the largest number of n instances, and larger number of features p, all are belonging to known class C. Therefore, each feature in the given datasets is scaled between two intervals [0,1]; hence, if interval lies < 0, it means worst, and if it lies $> 0 \leq 1$, it means best. NFR-ReliefF often repeats the same iteration at m times; initially, feature weight vector starts with zero value, i.e., $Wi = 0$, and calculates most nearby instance value x, concerning the features p. In general, Relief estimates and weights the feature if the relevant value of each instance is greater than the fixed default threshold value θ. NFR-ReliefF result exhibits an optimal divergence between irrelevant and relevant feature corresponding to the threshold θ, which help to measure the noise level in the feature. Typically, the threshold value can be estimated by Chebyshev's inequality measure with a given confidence level represented as (θ) that at θ of 1/sqrt (θ*m) is maximum to good enough to define the probability of type I error $< \theta$, i.e., a relevant threshold value must be $0 \leq \theta \leq 1$].

NFR-R is also used to measure the nearest hit 'H' and nearest miss 'M' value of the instance of the Chebyshev distance routine method. The complete working principle of NFR algorithm has been neatly sketched and is shown in Fig. 2. NFR is a six-step process to extract optimal features from large feature space. In step 2, Chebyshev distance-based outlier detection routine has applied to manipulate noisy features from the large feature space. Step 3 selects noisy and outlier feature to remove from the feature space. From step 4, NFR algorithm selects

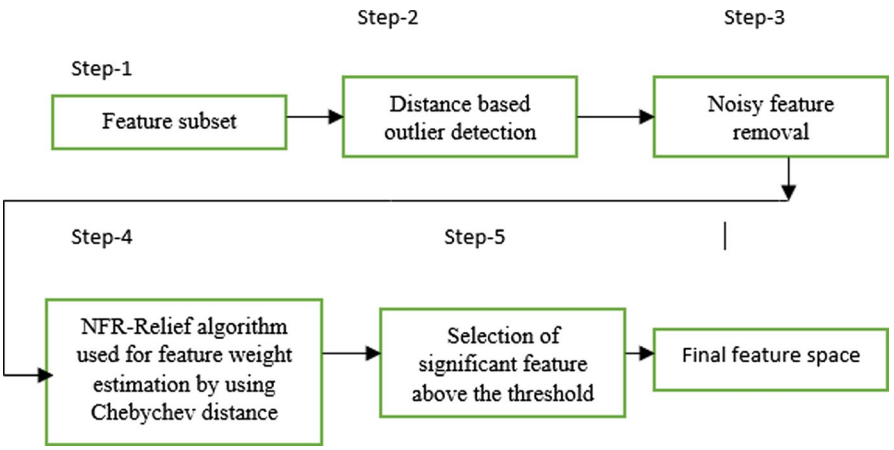

Fig. 2 Work flow diagram of NFR-relief algorithm

mostly relevant feature and estimates the feature weights. In step 5, significant features will be selected that lies above the pre-defined threshold value. Step 6 selects the significant feature and creates a feature subset. Distance-based outlier detection measure is efficient and simple to detect noisy feature effectively from the datasets. The proposed method is effective to handle noisy feature, and this algorithm overcomes the limitation of RG-R algorithm. In the next section, we will illustrate NFR-R in detail as well as discuss the performance of this algorithm in terms of classification accuracy with randomly selected datasets.

Algorithm:3 Pseudo code of the proposed algorithm-NFR-R

Input: for each training instance a vector of attribute values and the class value
Output: the vector W of estimations of the qualities of attributes
set all weights W[A]:=0
for *i:=1 to m do begin*
randomly select an instance R_i *;*
find nearest hit H and miss M ;
Chebyshev = max(|pi-qi|)
Select subset(X,k)
If /X/ = 1 then return sinlge element X
else
if(/X_1*/ + /*X_2*/ >=k) then*
return a
for *A :=1 to a do*

$$w[A] = W[A] + diff(X(i), near\ miss(i)X)^2 - diff(X(i), near\ hit(i)X)^2$$

Return (W[i])
Weight update
W[A]:= median value W[i]
if *W[i]* $\geq\theta$ *put it into selected feature subset*
end

3.1 NFR-Relief and its applications

The proposed work signifies the basic idea of Relief-based algorithms. The key idea of NFR-ReliefF is to remove the noisy features from the feature space to enhance classification accuracy. In the context of feature selection, in this paper, we provided an in-depth introduction and working principle of Relief algorithm and its various extensions. The proposed NFR-ReliefF algorithm can be described into four general of RBA research and reviewed important methods that differ within these branches. This proposed work highlights a number of advancement and application toward RBA-based research, such as (a) NFR-ReliefF is proficient in detecting noise feature on gene–gene interaction analysis of data in the context of bioinformatics, as well as reliable at univariate effects interactions; (b) recently, various heuristic and random search algorithms have been proposed to handle noisy feature; however, the proposed model is proficient in handling noise on hyperspectral data; (c) in general, RBAs are 'anywhere and anytime' algorithm, (d) NFR is one of the best choices of neighbor instance-based learning model, considered critical aspects of RBA triumph, recommended for different types of noisy feature removal, say agricultural data, mixed data types, complex pattern noise analysis, spatial image, images processing, etc., (e) NFR-Relief effectively removes the noise present in the feature space to proficiently evaluate the individual feature (i.e., feature weighting), lead to enhance reliable accuracy of machine learning-based methods such as classification on healthcare data and disease prediction, and (f) the proposed algorithm can easily adapt to RBA family and flexible to handle noise data on other specific application domains.

4 Result and discussion

4.1 Dataset description

Ten benchmarking microarray datasets have been used to measure the proposed NFR-R model as listed below, and the detailed description of the dataset is shown in Table 1. These datasets are widely accepted and tested on several algorithms, and all these datasets are related to human cancers, prostate tumors, central nervous system, colorectal, lungs, leukemia and large B cell lymphoma. The properties of the datasets are summarized in Table 1. All these datasets are available at http://sunflower.kuicr.kyoto-u.ac.jp/~ruan/LHR/.

4.2 Evaluation method

The performance of the proposed algorithm can be compared with MultiSurf and SURF, to measure the performance of the proposed model, we have used three performance metrics such as *the number of features selected*, *runtime measure and classification accuracy, and* three classifiers have been used to measure the

Table 1 Characteristics of the microarray datasets

Dataset names	Total number of instances	Total number of gene features	Target class	Dataset nature
Leukemia	72	7128	3	Microarray
Colon-1	37	8825	2	Microarray
DLBCL	77	7129	3	Microarray
Lung	181	12,533	2	Microarray
Prostate 1	102	12,600	3	Microarray
Prostate 2	88	12625	3	Microarray
Prostate 3	33	12,626	3	Microarray
GCM	280	16,063	2	Microarray
Tox-171	100	5749	3	Microarray

classification accuracy of NFR-Relief such as C4.5, naïve Bayes and KNN. The results are estimated through EARR (*extended adjusted ratio of ratios*) values deployed in 2013 [13, 20]. EARR is a unified metric, which can use to measure the classification accuracy concerning the runtime and number of features selected [26]. EARR measures the noise level of feature and calculates the linear relation between attributes and the target classes, and EARR can measure the model by taking the ratio of metric values. Assuming that a given dataset D contains the number of features (d_1, d_2, d_3……. d_n) with respect to the attribute value 'A,' i.e., ($A = a_1$, a_2, a_3, a_4, a_5 ……. a_n) up to 'n' attributes, then EARR of Ai and Di can be denoted as shown in Eq. 6. From Eq. 6, α and β can be represented as user-defined parameters to measure importance score value between runtime and number features. acc_i^k represents the accuracy of the NFR-R concerning the instance 'i,' and 'k.' Let us assume if several selected noisy features from datasets k are based on the algorithm *'i,'* this makes to allow the user to analyze noise feature difference between runtime and number of selected features with the help of accuracy measure. Let us assume if the multiple features compare with the selected models, the average arithmetic means of $EARR_{A_iA_j}^{D_k}$ will be increased to 'M,' as defined in Eq. 7.

$$EARR_{A_iA_j}^{D_k} = \frac{acc_i^k / acc_j^k}{1 + \alpha . \log\left(\frac{t_i^k}{t_j^k}\right) + \beta . \log\left(n_i^k / n_j^k\right)} (1 \le i \ne j \le M, 1 \le k \le N \tag{6}$$

$$EARR_{A_i}^{D_k} = \frac{1}{M-1} \sum_{j=1\&i=1}^{M} EARR_{A_iA_j}^{D_k} \tag{7}$$

Therefore, the evaluation metric can be distinguished by three global known classification algorithms such as tree-based classifier 1 C4.5, naïve Bayes and KNN. Each classifier performs to measure the noise level observed in feature subset to measure average classification accuracy, by using tenfold CV (cross-validation) for

training and testing the datasets [19]. Typically, all the experiments implemented on the same machine due to runtime are machine dependent.

4.3 Algorithm comparisons with dimensions of data

This section discusses the performance of the proposed approach NFR-Relief by comparing with the existing algorithm (such as ReliefF [9], MultiSurf [1] and SURF [32]) by noise data removal, all the existing algorithm has been developed by the inspiration of instance-based learn mechanisms. The dimension of datasets is differing from 37 to 13,000 features. However, the proposed algorithm outperforms to handle noisy features on datasets colon-1, leukemia, prostate 1, prostate 2, lung, prostate 3 and GCM. However, datasets tox-171 and DLBCL achieved lower accuracy due to incomplete feature and missing values, as shown in Table 2. Table 3 shows a comparison result of the proposed algorithm concerning the number of noisy feature selections. Similarly, Table 4 shows a comparison result of the proposed algorithm concerning the average runtime accuracy in term of seconds and Tables 5, 6 and 7 exhibit the comparison result of classification accuracy for the classifiers. To measure the runtime accuracy of the proposed model, NFR-Relief has been tested on nine benchmarking microarray datasets, and the average runtime is very much better than the existing algorithm toward to handle noise features. However, dataset prostate 3 and dataset prostate 2 are not achieved the desired result. However, to compare with other datasets NFR-Relief achieved a greater result to remove noisy features from the datasets. However, in the process of iteration NFR-Relief faced an isolated feature problem, where the feature does not have noise, but not related to a relevant feature, and it cannot be said that isolated feature is an irrelevant feature. However, this issue will be given crucial attention and consideration for future assignments.

4.4 Comparison based on the EARR values

EARR model has significantly considered measuring algorithm efficacy, and EARR can be described as a unified multi-criterion approach that can generally use to analyze algorithm efficiency [13, 20]. According to the observed result of classifiers concerning the three evaluations, the classification accuracy values of noise removal can be shown in Tables 3, 4 and 5. By using Eq. 7, EARR values can measure to check the algorithm efficiency of NFR-Relief and the result will be compared with ReliefF, MultiSurf and SURF for the feature set [3, 4]. Typically, the gained result of EARR of attribute A_i is greater than (i.e., $A_j < A_j$ or $A_j = A_{j,}$) of EARR A_j. Then, A_i is a maximum of instance attribute A_j [13, 20]. First, as per the result observed from Table 8 and Fig. 3, the EARR values of the proposed approach are gaining greater value than SURF, i.e., EARR (NFR-R_A) is 0.89, whereas EARR of SURF is 0.78. Second, the EARR value of MultiSurf is 0.787 which is smaller than EARR of the proposed NFR-R_A which is 0.825. Third, the EARR of ReliefF is 0.767 and EARR of the proposed NFR-R_A is 0.861. Hence, the proposed model works slightly better to handle noisy features than the classic ReliefF, MultiSurF and SURF. The average runtime is estimated

Table 2 Feature selection accuracy level with noisy features

Dataset	Dataset dimension	Feature selection algorithm	Selected feature	Accuracy C4.5	Accuracy by naïve Bayes	Accuracy KNN
Leukemia	7128	MultiSurf	10	52.36	56.77	56.91
		SURF	15	53.57	64.57	55.00
		ReliefF	11	53.68	58.54	56.89
		Proposed	11	54.13	59.17	57.22
Colon-1	8825	MultiSurf	16	89.02	48.72	82.10
		SURF	15	88.01	73.61	83.12
		ReliefF	15	91.00	77.53	82.55
		Proposed	12	90.18	78.22	85.50
DLBCL	7129	MultiSurf	16	71.51	69.56	70.25
		SURF	18	69.58	70.26	64.52
		ReliefF	17	73.02	69.12	56.56
		Proposed	14	80.02	71.14	58.00
Lung	12,533	MultiSurf	18	81.85	58.45	80.11
		SURF	18	82.26	67.56	80.58
		ReliefF	17	79.21	68.14	81.20
		Proposed	13	78.60	71.01	81.56
Prostate 1	12,600	MultiSurf	20	68.52	75.25	69.85
		SURF	20	74.51	86.00	72.26
		ReliefF	18	75.55	74.45	71.13
		Proposed	17	76.33	75.20	71.02
Prostate 2	12,625	MultiSurf	18	78.36	90.75	69.56
		SURF	15	72.88	88.81	72.26
		ReliefF	16	73.52	91.58	66.12
		Proposed	14	71.90	91.56	72.25
Prostate 3	12,626	MultiSurf	30	67.25	75.51	72.11
		SURF	27	68.36	74.58	72.58
		ReliefF	18	64.23	75.02	74.20
		Proposed	17	65.08	72.12	73.45
GCM	16,063	MultiSurf	19	76.58	71.25	66.25
		SURF	20	74.23	63.12	74.52
		ReliefF	12	73.25	56.56	66.56
		Proposed	14	75.85	62.25	71.56
Tox-171	5749	MultiSurf	62	57.12	55.01	61.78
		SURF	35	68.15	63.02	62.15
		ReliefF	58	54.12	58.15	56.52
		Proposed	40	66.60	62.55	88.11

in terms of seconds, the proposed method outperforms on dataset GCM, prostate-1, Lung, DLBCL, Colon1 etc., as shown in Fig. 4. Figure 5 shows the significant features that has extracted from the original datasets.

Table 3 Comparison with the number of obtained features

Datasets	Proposed	MultiSurf	SURF	ReliefF
Leukemia	11	12	14	12
Colon-1	12	14	15	13
DLBCL	14	13	18	15
Lung	14	16	16	13
Prostate 1	18	18	19	18
Prostate 2	14	16	15	13
Prostate 3	18	31	28	19
GCM	13	18	21	11
Tox-171	20	81	22	24

Table 4 Comparison with average runtime in seconds after noise removal

Datasets	Proposed	MultiSurf	SURF	ReliefF
Leukemia	40.05	43.54	60.21	48.25
Colon-1	35.44	44.48	58.58	59.15
DLBCL	21.45	54.47	45.25	39.48
Lung	34.45	54.15	56.48	58.20
Prostate 1	44.15	56.17	57.41	59.56
Prostate 2	50.01	56.12	52.40	57.89
Prostate 3	55.12	55.25	54.23	52.45
GCM	28.23	48.09	55.16	59.58
Tox-171	1.18	1.59	1.58	1.59

Table 5 Comparison of average accuracy by % with classifier 1

Datasets	Proposed	MultiSurf	SURF	ReliefF
Leukemia	81.15	43.34	68.21	41.25
Colon-1	84.24	78.48	82.58	81.15
DLBCL	78.15	69.55	83.25	70.55
Lung	82.15	72.15	74.48	82.30
Prostate 1	81.25	75.17	78.55	68.26
Prostate 2	92.00	80.12	82.40	86.19
Prostate 3	87.21	94.25	76.23	87.31
GCM	81.14	80.24	68.16	82.35
Tox-171	77.45	81.25	84.29	78.28

5 Conclusion

In this work, we have introduced an improved univariate heuristic model for feature selection task in filter approach. The proposed NFR-ReliefF algorithm evolved from Chebyshev distance-outlier detection method, it incorporates the basics of ReliefF

Table 6 Comparison of average accuracy with KNN classifier in (%)

Datasets	Proposed	MultiSurf	SURF	ReliefF
Leukemia	81.53	71.34	78.21	77.25
Colon-1	84.34	77.18	80.58	79.15
DLBCL	79.55	68.75	79.25	78.55
Lung	81.79	71.15	74.48	82.20
Prostate 1	80.32	74.18	78.55	78.56
Prostate 2	91.25	81.12	80.40	86.49
Prostate 3	84.63	86.25	74.23	87.31
GCM	87.38	81.25	76.16	80.35
Tox-171	87.15	82.25	82.29	78.48

Table 7 Comparison of average accuracy with a) Naive Bayes classifier and, b) C4.5 classifier in (%)

Datasets	Proposed	MultiSurf	SURF	ReliefF
a) Naive Bayes classifier				
Leukemia	82.35	78.34	81.21	79.12
Colon-1	84.34	79.28	82.58	81.15
DLBCL	81.65	72.75	81.25	77.55
Lung	82.89	76.15	77.34	83.25
Prostate 1	80.47	77.18	79.55	79.56
Prostate 2	90.15	87.12	86.40	88.49
Prostate 3	86.36	88.25	79.23	88.41
GCM	86.13	79.12	78.31	82.13
Tox-171	84.29	83.19	82.12	81.14
b) C4.5 classifier				
Leukemia	80.32	78.21	71.34	82.20
Colon-1	91.25	80.58	77.18	78.56
DLBCL	84.63	79.25	68.75	86.49
Lung	87.38	74.48	71.15	87.31
Prostate 1	87.15	78.55	74.18	79.55
Prostate 2	82.20	80.40	81.12	81.79
Prostate 3	78.56	74.23	86.25	80.32
GCM	86.49	76.16	81.25	91.25
Tox-171	87.31	82.29	82.25	79.55

Table 8 Comparison based on EARR values

	Microarray dataset			
	ReliefF	MultiSurf	SURF	Proposed
ReliefF	–	0.778	0.814	0.767
MultiSurf	0.871	–	1.115	0.787
SURF	0.825	0.841	–	0.780
Proposed	0.861	0.848	0.890	–

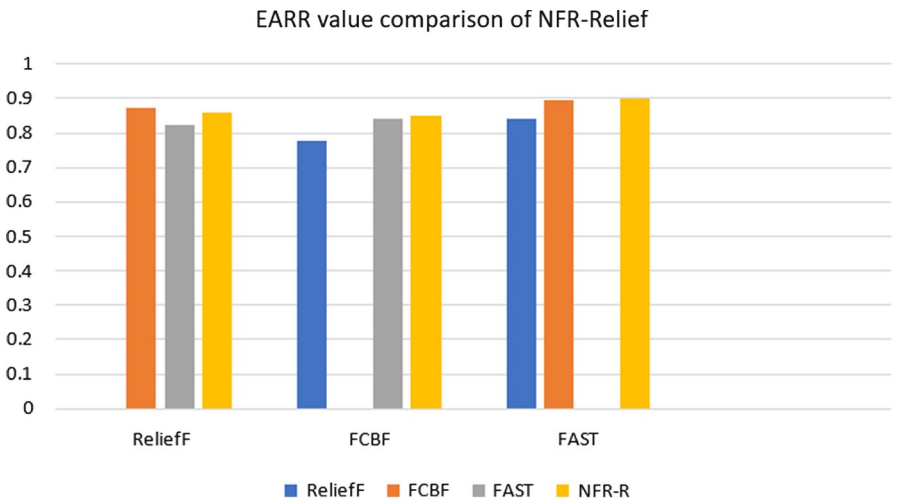

Fig. 3 EARR values comparisons

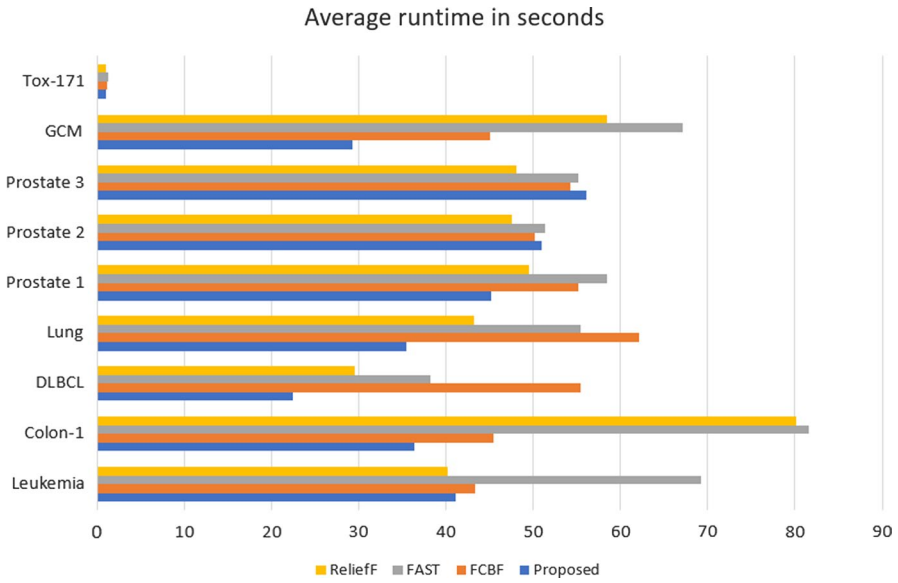

Fig. 4 Average runtime comparisons of each algorithm in term of seconds

and slightly adjusts the feature weight measure $W[i]$ corresponding to function *Diff*, and NFR-R measures the quality of attribute by selecting instances randomly with respect to target class, and then, it splits two instances within the same class to increase the quality of subset which contains reliable features, and it also measures both variance and mean of the discrimination among two instances into account as the evaluation criterion of the feature score measure, which produce more stable

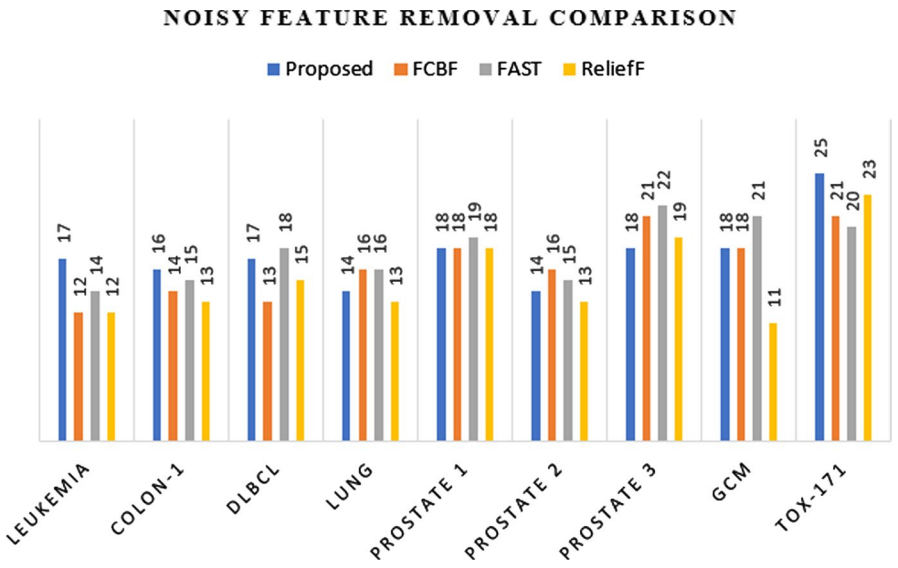

Fig. 5 Algorithm wise noise feature removal in terms of numbers

and accurate results. Therefore, the proposed method has been evaluated in terms of classification accuracy and EARR (extended adjusted ratio of ratios) values, as well as execution time performance using nine benchmark microarray datasets. We compare the classification accuracy for the selected features using three well-known classifiers KNN, naive Bayes and C4.5. The overall performance of the proposed method has been found excellent in terms of both runtime and classification accuracy for all these datasets. However, for future work, we may improve computational speed via low-level programming languages (e.g., C or C++).

References

1. Urbanowicz RJ, Olson RS, Schmitt P, Meeker M, Moore JH (2018a) Benchmarking relief-based feature selection methods for bioinformatics data Mining. https://perma.cc/9YND-H5V5, pp 168–188
2. Urbanowicz RJ, Meeker M, LaCava W, Olson RS, Moore JH (2018b) Relief-based feature selection: introduction and review. https://perma.cc/VCG2-8MC6, pp 189–203
3. Xing EP, Jordan MI, Karp RM (2001) Feature selection for high-dimensional genomic microarray data. In: Proceedings of the Eighteenth International Conference on Machine Learning, June 2001. https://perma.cc/VD29-NNEP, pp 601–608
4. Lei Y, Liu H (2004) Redundancy based feature selection for microarray data. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. https://perma.cc/Y8NU-UW72, pp 737–742
5. Le T, Urbanowicz R, Moore J, McKinney B (2018) Statistical inference relief (STIR) feature selection. Bioinformatics, PMID 30239600, https://perma.cc/LA5X-WD9S, pp 1–8
6. Kira K, Rendell L (1992) A practical approach to feature selection. In: ML92 Proceedings of the Ninth International Workshop on Machine Learning: Morgan Kaufmann Publishers Inc, pp 49–256
7. Kira K, Rendell L The feature selection problem: traditional method and a new algorithm. In: AAAI-1992, Proceeding, pp 129–134
8. Robnik-Sikonja M, Kononenko I (2003) Theoretical and empirical analysis ReliefF and RReliefF. Mach Learn 53(1–2):23–69

9. Kononenko I, Šimec E, Robnik-Šikonja M (1997) Overcoming the myopia of inductive learning algorithms with ReliefF, Kluwer Academic Publisher, pp 1–17
10. Wang G, Song Q, Sun H, Zhang X, Xu B, Zhou Y (2013) A feature subset selection algorithm automatic recommendation in method. J Artif Intell Res 47:1–34
11. Lo S (2010) The effects of feature selection and model selection on the correctness of classification. In: Proceedings of the IEEE, International Conference on Industrial Engineering and Engineering Management, 2010, pp 989–993
12. Krawiec K (2010) The evolutionary feature selection and construction. In: Encyclopaedia of machine learning, Springer, Berlin, pp 353–357
13. Ahmed S, Zhang M, Peng L (2013) Enhanced feature selection for in biomarker discovery in LC-MS data using GP. In: IEEE Congress on Evolutionary Computation (CEC), 2013, pp 584–591
14. Liu H, Zhao Z (2009) Manipulating data and dimension reduction methods: feature selection. In: Encyclopaedia of complexity and systems science, Springer, Berlin, pp 5348–5359
15. Sun Y (2007) Iterative relief for feature weighting: algorithms, theories, and applications. IEEE Trans Pattern Anal Mach Intell 29(6):1035–1051
16. Sun Y, Todorovic S, Goodison S (2010) Local-learning-based feature selection for high-dimensional data analysis. IEEE Trans Pattern Anal Mach Intell 32(9):1610–1626
17. Jain AK, Duin RPW, Mao J (2000) Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell 22(1):4–37
18. Koller D, Sahami M (1996) Toward an optimal feature selection. In: Saitta L (ed) the Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann Press, pp 284–292
19. Yu L, Liu H (2003) Feature selection for the high-dimensional data: a fast correlation—based filter solution. In: Proceedings of 20th International Conference on Machine Learning, vol 20, no. 2, pp 856–863
20. Song Q, Ni J, Wang G (2013) A fast clustering-based feature sub-set selection algorithm for the high-dimensional data. IEEE Trans Knowl Data Eng 25:1–14
21. Kwak N, Choi C-H (2002) Input feature selection by mutual information based on Parzen window. IEEE Trans Pattern Anal Mach Intell 24:1667–1671
22. Bollon-Canedo V, Sanchez-Marono N, Alonso-Betanzos A (2013) A review of feature selection methods on synthetic data. Knowl Inf Syst 34(3):483–519
23. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
24. Guyon Isabelle, Elisseeff André (2003) An introduction to variable and feature selection. J Mach Learn Res 3:1157–1182
25. Menze BH, Kelm BM, Masuch R, Himmelreich U, Bachert P, Petrich W, Hamprecht FA (2009) A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data. BMC Bioinform 10(1):213
26. Jović A, Brkić K, Bogunović, N (2015) A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). IEEE, pp 1200–1205
27. Saeys Y, Inza I, Larrañaga P (2007) A review of feature selection techniques in bioinformatics. Bioinformatics 23(19):507–517
28. Ladha L, Deepa T (2011) Feature selection methods and algorithms. Int J Comput Sci Eng 3(5):1787–1797
29. Dash M (2011) RELIEF-C: efficient feature selection for clustering over noisy data tools. ISSN 1082–3409:869–872
30. Robnik Sikonjam M, Kononenko I (2003) Theoretical and empirical analysis of ReliefF and RReliefF. Mach Learn 53(1):23–69
31. Liu H, Motoda H, Setiono R, Zhao Z (2010) Feature selection: an ever-evolving frontier in data mining. In: Feature Selection for Data Mining, vol. 10 of JMLR Proceedings, JMLR.org, pp 4–13
32. Greene CS, Penrod NM, Kiralis J, Moore JH (2009) Spatially uniform relief (surf) for computationally-efficient filtering of gene–gene interactions BioData mining, https://doi.org/10.1186/1756-0381-2-5