

# A Random Projection $k$ Nearest Neighbours Ensemble for Classification via Extended Neighbourhood Rule

Amjad Ali<sup>1</sup>, Muhammad Hamraz<sup>1</sup>, Dost Muhammad Khan<sup>1</sup>, Wajdan Deebani<sup>2</sup> and Zardad Khan<sup>1\*</sup>

<sup>1\*</sup>Department of Statistics, Abdul Wali Khan University, Mardan, 23200, KPK, Pakistan.

<sup>2</sup>Department of Mathematics, College of Science and Arts, King Abdul Aziz University, Rabigh, Saudi Arabia.

\*Corresponding author(s). E-mail(s): [zkhan.essex@gmail.com](mailto:zkhan.essex@gmail.com);  
Contributing authors: [Amjad.ali@awkum.edu.pk](mailto:Amjad.ali@awkum.edu.pk);  
[mhamraz@awkum.edu.pk](mailto:mhamraz@awkum.edu.pk); [Dostmuhammad@awkum.edu.pk](mailto:Dostmuhammad@awkum.edu.pk);  
[wdeebani@kau.edu.sa](mailto:wdeebani@kau.edu.sa);

## Abstract

Ensembles based on  $k$  nearest neighbours ( $k$ NN) combine a large number of base learners, each constructed on a sample taken from a given training data. Typical  $k$ NN based ensembles determine the  $k$  closest observations in the training data bounded to a test sample point by a spherical region to predict its class. In this paper, a novel random projection extended neighbourhood rule (RPEXNRule) ensemble is proposed where bootstrap samples from the given training data are randomly projected into lower dimensions for additional randomness in the base models and to preserve features information. It uses the extended neighbourhood rule (ExNRule) to fit  $k$ NN as base learners on randomly projected bootstrap samples. Several benchmark datasets are used for assessing the performance of the proposed method in comparison with other state-of-the-art procedures. Classification accuracy, Brier score and Cohen's kappa are used as comparison metrics. The proposed RPEXNRule ensemble and other classical procedures are also compared via simulated data. Based on benchmarking and simulation study, the results given in the paper reveal that the proposed method performs better than the others in the majority of the cases.

**Keywords:** Random Projection, Dimension Reduction, Classification, Extended Neighbourhood Rule,  $k$ NN Ensemble

## 1 Introduction

Classification is one of the most important topics used in supervised learning problems. It classifies unseen observations to their respective classes based on a group of related features in the training data.  $k$  nearest neighbours ( $k$ NN) [1–3] is one of the top-ranked classification procedures [4]. This method was developed by Fix and Hodges in [5] and then used for pattern recognition by Cover and Hart in [6]. It is one of the most simple classification procedures which gives promising results in the case of big datasets [7–10]. It should be the first choice to use when there is a small amount or no prior information available about the data distribution. The main idea behind the  $k$ NN method is to classify an unseen observation into the most frequent class by using majority voting in the labels of the identified  $k$  nearest sample points in the feature space. As the  $k$ NN procedure is one of the top ten fundamental and non-parametric data mining procedures [4] that gives efficient results on several benchmark datasets. However, it may be affected by several data-related problems, like the existence of noise and non-informative features in the data. The classical  $k$ NN have been modified for further improvement, such as weighted  $k$  nearest neighbours (W $k$ NN) [11] and condensed nearest neighbours (CNN) [12–14].

Ensembles of base  $k$ NN learners in conjunction with randomized sampling procedure(s) bring diversity to the classifier and make the models more stable and flexible. Randomization is injected into these procedures by taking random bootstrap samples from the data points and/or random subsets of features from the feature space involved in the data for constructing the base  $k$ NN models. This makes the model more diverse and reduces the chance of the same errors occurring repeatedly [15–17]. Therefore, these methods are more stable and reliable as compared to the single learner classifiers. Moreover, random subsets of the features are also used for dimensionality reduction which minimizes the execution time. Examples of the  $k$ NN based ensembles are bootstrap aggregated  $k$ NN [18], random  $k$ NN [19], ensemble of subset of  $k$ NN [20], weighted heterogeneous distance Metric [21], ensemble of random subspace  $k$ NN [22], optimal  $k$ NN ensemble [23], etc. Almost all the  $k$ NN ensembles use majority voting in the class labels of the nearest instances of the test observation to estimate its class in each base model. The final prediction is obtained by using the second round of the majority voting in the results given by all the base learners. As the  $k$ NN based ensembles combine a large number of base learners, where each model is constructed on a bootstrap sample in conjunction with a subset of features. Therefore, many of these learners are fitted on irrelevant features and several influential features are ignored. Moreover, such ensembles

do not perform well where the test observations follow a pattern of the nearest training observations that lie on a certain path beyond the sphere of the neighbourhood. Due to these problems, the conventional ensembles become weak and hence, give poor prediction accuracy.

A novel random projection extended neighbourhood rule (RPExNRule) for the  $k$ NN ensemble is proposed in this paper. The extended neighbourhood rule (ExNRule) [24] is used for  $k$ NN as base learners. To construct each base model a bootstrap sample is drawn from the training data with the full feature space and then randomly projected into a reduced dimension. The predicted class of an unseen observation is obtained using majority voting in the results given by the base learners. For assessing the performance of the novel ensemble, 15 benchmark problems are used. The classification accuracy, Brier score and Cohen's kappa are used as metrics of performance. The  $k$  nearest neighbour ( $k$ NN), weighted  $k$  nearest neighbours ( $Wk$ NN), random  $k$  nearest neighbour ( $Rk$ NN), random forest (RF), optimal trees ensemble (OTE) and support vector machine (SVM) are used as the competitors. For further evaluation, boxplots of the results are also created to show the precision of the performance.

The remaining article is organized as follows. Section 2 gives detailed literature related to the proposed work. A detailed description of the proposed method, its mathematical background and its algorithm are provided in Section 3. Section 4 gives details of the datasets, experiments and discussion on the results. The paper is concluded in Section 5.

## 2 Related works

The  $k$  nearest neighbours ( $k$ NN) classifier is one of the top ten data mining procedures [4] that gives efficient results in many real life problems. The classical  $k$ NN procedure gives equal weights to the observations determined in the neighbourhood of an unseen sample point. A weighted  $k$  nearest neighbours ( $Wk$ NN) algorithm is proposed by [11]. The weights are assigned based on the distance of the nearest observations from the unseen sample point. The  $Wk$ NN procedure gives promising prediction results on several benchmark problems as compared to the classical  $k$ NN method. However,  $Wk$ NN procedure uses all sample points in the training data which makes it a global method and time consuming. Authors have proposed several methods for the purpose to reduce the training time, such as condensed nearest neighbour (CNN) techniques [12, 14]. The CNN procedures remove identical observations from the training data as these sample points have no extra information to improve the results of the model. However, these methods might ignore the sample points which lie on the boundary and depend on data order. [25] proposed the reduced nearest neighbor (RNN) rule, which is a modified version of the CNN rule. This method removes instances and minimizes the training dataset. However, RNN procedure is computationally complex and laborious. The fast condensed nearest neighbor (FCNN) rule proposed by [26]. This method allows selecting

observations that are much close to the decision boundary. The FCNN has a very small amount of quadratic complexity and is independent of the observations order. Another improvement have been made by [27], using a model based  $k$ NN method. A function is fitted on the training data to estimate the test sample points. This method also reduces the training data size like CNN and RNN methods and provides promising prediction accuracy. However, this method avoids the marginal data points which lies farther along in the space. A clustered  $k$ NN (CkNN) is introduced by [28] which is one of the more robust methods and reducing the problem of unequal distribution of training sample points. The CkNN method determines the nearest instances in the clusters. To determine the threshold used in this technique is difficult task, to compute distances among the observations within different clusters. The selection of the optimal  $k$  value is also unknown. Another method called modified  $k$  nearest neighbour (MkNN) is introduced by [29] which uses weights and validity of the training data points to estimate the response value of a test observation. [30] introduced a  $k$  dimensional tree nearest neighbour ( $k$ DTNN) technique, which splits the training data into sub-planes to orchestrate multidimensional samples. The  $k$ DTNN technique produces a tree that is balance, simple and easy to interpret. However, this method requires intensive search to split the instances blindly into sub-planes. Moreover,  $k$ DTNN involves computational complexity and might miss the data pattern. A sample reduction procedure for regression and classification proposed in [31], inspired by man social behavior. Other improved ensembles can be found in [32–38].

Several ensembles based on  $k$ NN methods are also proposed in literature, that make the classifiers more efficient and diverse. A bootstrap aggregation (bagging) is one of the basic ensemble techniques proposed by [39]. It constructs a new model to predict the exact bootstrap average of several base learners [40–42]. Bagging provides a basic platform for several state-of-the-art ensemble methods. Ensemble techniques based on bagging are constructed by using a large number of bootstrap samples drawn from training data. A base classifier is fitted on each sample and used to predict test data. The final prediction is obtained by using majority voting on the results given by the base learners [39]. The standard bagging procedure is further improved by [18]. Steel extended the exact bagging approach to sub-sampling aggregation with and without replacement techniques. Several ensembles methods have been introduced in literature which use both bagging and subsets of features to fit base  $k$ NN classifiers [19, 20, 43, 44]. Several authors have also introduced  $k$  value optimization based ensemble methods [45, 46]. These methods try to optimize  $k$  value in base  $k$ NN models used in the ensemble. Other ensemble procedures can be found in [47–51]. Despite of efficient results given by these ensembles on many benchmark problems, they might be affected by the mixed class patterns in the data. Moreover, base learners in these methods are constructed on sub-samples which may ignore discriminative feature(s). Therefore, these procedures need further improvement to overcome such problems.

This paper introduces a novel ensemble method called random projection extended neighbourhood rule (RPEXNRule) for  $k$ NN. This procedure constructs a large number of the extended neighbourhood rule (ExNRule) based classifiers [24], each on a randomly projected bootstrap sample and a new data point is predicted by all these models. The final prediction class is obtained using majority voting among the results given by the underlying models. It improves the prediction accuracy in the following steps.

1. The  $k$  nearest observations to a test sample point in the training data are determined in a step-wise pattern to recognize the true pattern of the data.
2. The base learners are more random and diverse as they are constructed on randomly projected bootstrap samples.

### 3 The RPEXNRule for $k$ NN ensemble

Let  $\mathcal{L} = (X, Y)_{n \times (p+1)}$  be a training dataset, where  $X_{n \times p}$  represents the feature space with dimension  $n \times p$  and  $Y$  be a categorical response with two classes i.e.  $Y \in (0, 1)$ . Draw  $B$  bootstrap samples from the training data  $\mathcal{L}$  and randomly project the feature space of these samples to dimension  $p' \leq p$  i.e.  $\mathcal{S}_{n \times (p'+1)}^b$ , where  $b = 1, 2, 3, \dots, B$ . All these projected samples contain the reduced feature space  $\chi_{n \times p'}$  and their corresponding response labels  $Y$ .

Suppose  $X_{1 \times p}^0$  is a test sample point and needs to be predicted. In this respect, it will be randomly projected according to the  $b^{th}$  random projection i.e.  $\chi_{1 \times p'}^0$ . Then apply the extended neighbourhood rule (ExNRule) [24] to each random projection and estimate the test data point. The predictions given by all  $B$  models of the unseen observation  $X_{1 \times p}$  are  $\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \dots, \hat{Y}^B$ . The final predicted class label of the test sample point is a majority vote of the estimates given by all underlying learners.

#### 3.1 Mathematical Description

Mathematical description of the proposed method is given in the following sub-sections.

##### 3.1.1 Random Projection

The basic idea of random projection method is provided by Johnson-Lindenstrauss lemma [52], which is used for dimension reduction [53, 54]. This method projects a  $p$ -dimensional data matrix to a  $p'$ -dimensional space where  $p' \leq p$ . Mathematically it can be expressed as follows.

Let  $\mathcal{R}_{p \times p'}$  be a random matrix generated from any of the well known distribution such as Gaussian, etc. The elements of  $\mathcal{R}_{p \times p'}$  are independently and identically distributed. The given matrix of feature space  $X_{n \times p}$  can be projected using matrix multiplication as:

$$\chi_{n \times p'} = X_{n \times p} \mathcal{R}_{p \times p'}$$

The proposed method takes a large number of bootstrap samples and the feature space of each sample is randomly projected. A model based on ExNRule [24] is fitted to each of these projections.

### 3.1.2 The Extended Neighbourhood Rule

A distance formula to be used in  $b^{th}$  randomly projected bootstrap sample  $S_{n \times (p'+1)}^b$ , where  $b = 1, 2, 3, \dots, B$ , to identify the  $k$  nearest observations in a step-wise pattern to a test data point  $X_{1 \times p'}^0$ . To make the dimension of the unseen observation according to  $b^{th}$  bootstrap sample, it is also randomly projected i.e.  $\chi_{1 \times p'}^0$ . The general formula used for the calculation of distances between data points is given as:

$$\delta_b(\chi_{1 \times p'}^{i-1}, \chi_{1 \times p'}^i)_{min} = \left[ \sum_{j=1}^{p'} \|x_j^{i-1} - x_j^i\|^q \right]^{1/q}, \quad (1)$$

where,  $i = 1, 2, \dots, k$  and  $b = 1, 2, \dots, B$ .

In  $b^{th}$  randomly projected sample, Equation 1 gives the following sequence of minimum distances in a pattern which determines the neighbourhood of the unseen observation.

$$\delta_b(\chi_{1 \times p'}^0, \chi_{1 \times p'}^1)_{min}, \delta_b(\chi_{1 \times p'}^1, \chi_{1 \times p'}^2)_{min}, \\ \delta_b(\chi_{1 \times p'}^2, \chi_{1 \times p'}^3)_{min}, \dots, \delta_b(\chi_{1 \times p'}^{k-1}, \chi_{1 \times p'}^k)_{min}.$$

The above sequence determines  $\chi_{1 \times p'}^i$  is the closest data point to  $\chi_{1 \times p'}^{i-1}$ , where,  $i = 1, 2, 3, \dots, k$ . The selected observations are  $\chi_{1 \times p'}^1, \chi_{1 \times p'}^2, \chi_{1 \times p'}^3, \dots, \chi_{1 \times p'}^k$  and their corresponding classes are  $y^1, y^2, y^3, \dots, y^k$ , respectively. Majority voting is used to predict the unseen observation  $X_{1 \times p}^0$  by the  $b^{th}$  base learner, i.e.,  $\hat{Y}^b = \text{majority vote of } (y^1, y^2, y^3, \dots, y^k)$ , where,  $b = 1, 2, 3, \dots, B$ . The final estimation of the class label of that observation is  $\hat{Y} = \text{majority vote of } (\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \dots, \hat{Y}^B)$ .

## 3.2 Algorithm

The RPEXNRule takes the following steps.

1. Take  $B$  bootstrap samples from training data and reduce the original dimension from  $p$  to  $p'$  using the random projection technique.
2. Fit extended neighbourhood rule (ExNRule) on each projected sample generated in Step 1.
3. Predict an unseen observation via all the models fitted in Step 2.
4. Using majority voting in the results given by all base learners to obtain the final prediction of the unseen observation.

The pseudo code of the proposed RPEXNRule method is given in Algorithm 1 and Figure 1 illustrates the flowchart of this procedure.

**Algorithm 1** Pseudo code of the proposed RPEXNRule.

---

```

1:  $\mathcal{L} = (X, Y)_{n \times (p+1)} \leftarrow$  Training data with dimension  $n \times (p+1)$ 
2:  $X_{n \times p} \leftarrow$  Training feature space with  $n$  observations and  $p$  features;
3:  $Y \leftarrow$  Response vector of  $n$  values in the training data;
4:  $X_{1 \times p}^0 \leftarrow$  A test observation to be classified;
5:  $p \leftarrow$  Total number of features in the training data;
6:  $n \leftarrow$  Total number of observations in training data;
7:  $B \leftarrow$  Number of randomly projected bootstrap samples each with a
   reduced dimension  $n \times (p' + 1)$ ;
8:  $p' \leftarrow$  Reduced number of features in each projected sample;
9:  $k \leftarrow$  Number of nearest observations to a test point;
10: for  $b \leftarrow 1 : B$  do
11:    $\mathcal{S}_{n \times (p'+1)}^b \leftarrow$  Randomly projected bootstrap sample with reduced
     feature space  $p'$ ;
12:    $\chi_{1 \times p'}^0 \leftarrow$  Randomly projected test point  $X_{1 \times p}^0$  according to training
     data;
13:   for  $i \leftarrow 1 : k$  do
14:      $\chi_{1 \times p'}^i \leftarrow$  Nearest training sample point to  $\chi_{1 \times p'}^{i-1}$  in  $\mathcal{S}_{n \times (p'+1)}^b$ 
15:      $y^i \leftarrow$  The corresponding class label
16:   end for
17:    $\hat{Y}^b =$  majority vote of  $(y^1, y^2, y^3, \dots, y^k)$ 
18: end for
19:  $\hat{Y} =$  majority vote of  $(\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \dots, \hat{Y}^B)$ 

```

---

## 4 Experiments and results

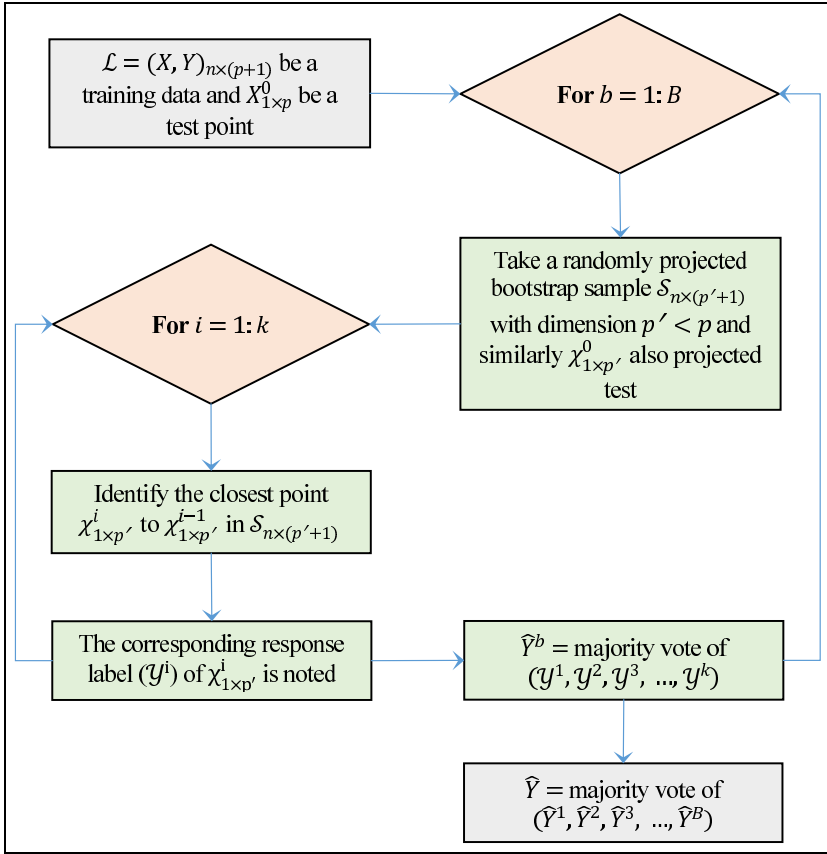
This section provides a detailed discussion on experimental setup, benchmark datasets used for assessment and results given by the proposed method and other classical procedures.

### 4.1 Datasets

This section gives a detailed discussion on the benchmark and simulation datasets used to evaluate the performance of the proposed method against other classical methods.

#### 4.1.1 Benchmark datasets

To compare the proposed random projection extended neighbourhood rule (RPEXNRule) ensemble with other state-of-the-art classifiers, 15 benchmark problems have been used. These datasets are openly available on various repositories. A brief description of these datasets is given in Table 1, showing the data name, number of observations  $n$ , number of features  $p$ , class distribution and its source.



**Fig. 1** Flow chart of the random projection extended neighbourhood rule ensemble.

### 4.1.2 Synthetic datasets

For assessing the performance of the proposed RPEXNRule ensemble, synthetic datasets are generated in three scenarios. Each dataset has 30 features and 200 sample points and a binary response. Half of 200 observations have class 0 and the remaining half have class 1. In the feature space 20 out of 30 features are generated from normal distribution with different parameters for each class, while the remaining 10 features are considered as contrived/irrelevant. These features are also generated from the normal distribution with same parameters i.e.  $N(\mu = 0, \sigma = 1)$  for both classes. Table 2 gives a detailed description of the simulated datasets. In that table the first column gives data IDs, the second and third columns provide the distribution of feature space for class 0 and class 1 respectively. The final column gives the distribution of the contrived features in the data.



**Table 1** Brief description of the datasets along with their corresponding number of features, observations, class-wise distributions and sources.

Data	p	n	Class Distribution	Sources
KCBI	86	145	(60, 85)	<a href="https://www.openml.org/d/1066">https://www.openml.org/d/1066</a>
TSVM	80	156	(102, 54)	<a href="https://www.openml.org/d/41976">https://www.openml.org/d/41976</a>
JEDI	8	369	(204, 165)	<a href="https://www.openml.org/d/1048">https://www.openml.org/d/1048</a>
WISC	32	194	(90, 104)	<a href="https://www.openml.org/d/753">https://www.openml.org/d/753</a>
AR5	29	36	(8, 28)	<a href="https://www.openml.org/d/1062">https://www.openml.org/d/1062</a>
ILPD	10	583	(167, 415)	<a href="https://www.openml.org/d/1480">https://www.openml.org/d/1480</a>
PLRE	12	182	(52, 130)	<a href="https://www.openml.org/d/1490">https://www.openml.org/d/1490</a>
SLEE	7	62	(29, 33)	<a href="https://www.openml.org/d/739">https://www.openml.org/d/739</a>
ECMO	9	130	(64, 66)	<a href="https://www.openml.org/d/944">https://www.openml.org/d/944</a>
MC2	39	161	(52, 109)	<a href="https://www.openml.org/d/1054">https://www.openml.org/d/1054</a>
PREL	13	315	(182, 133)	<a href="https://www.openml.org/d/915">https://www.openml.org/d/915</a>
GRDA	8	155	(49, 106)	<a href="https://www.openml.org/d/1026">https://www.openml.org/d/1026</a>
SLEX	9	60	(30, 30)	[55]
ECOL	7	336	(193, 143)	[56]
SPEC	22	267	(55, 212)	[57]

**Table 2** Brief description of synthetic datasets.

ID	Class 0	Class 1	Contrived features
$S_1$	$N(\mu = 1, \sigma = 5)$	$N(\mu = 0, \sigma = 5)$	$N(\mu = 0, \sigma = 1)$
$S_2$	$N(\mu = 5, \sigma = 5)$	$N(\mu = 0, \sigma = 5)$	$N(\mu = 0, \sigma = 1)$
$S_3$	$N(\mu = 5, \sigma = 5)$	$N(\mu = 0, \sigma = 3)$	$N(\mu = 0, \sigma = 1)$

## 4.2 Experimental setup

Experimental setup based on 15 benchmark problems provided in Table 1 and synthetic datasets given in Table 2 is formed as follows. Each of these datasets is randomly divided into two non-overlapping groups i.e. 70% training and 30% testing parts of the total number of sample points in the data. The said splitting criteria is repeated 500 times for all datasets. The novel random projection extended neighbourhood rule (RPEXNRule) ensemble and other standard classifiers i.e.  $k$  nearest neighbours ( $k$ NN), random  $k$  nearest neighbours ( $Rk$ NN), weighted  $k$  nearest neighbours ( $Wk$ NN), random forest (RF), optimal trees ensemble (OTE) and support vector machine (SVM) are constructed on the training part and tested on the testing part of each dataset. The final results are provided in Tables 3-5. These results are the arithmetic averages of all the results given by 500 splittings.

The proposed RPEXNRule ensemble is used with  $B = 500$  base learners each on a randomly projected bootstrap sample with reduced number of feature from  $p$  to  $p'$ . The Euclidean distance formula is used to determine the  $k = 3$  nearest neighbours in a step-wise pattern. These models are used to predict a test observation using majority voting. The final predicted class of the unseen observation is determined by the second round majority voting of the results given by the base learners. These parameters are used for assessing

the performance of the proposed method and all other standard procedures on the benchmark problems. This method is further assessed using  $k = 3, 5$ , and 7 with only  $k$ NN based classifiers. The proposed method is also compared with  $k$ NN based techniques on the synthetic datasets using  $k = 3$ .

The package `caret` [58] implemented in R is used to construct  $k$ NN classifier, while R package `knn` [59] is used for the fitting of  $Wk$ NN model. The R package `rknn` [60] is used to implement  $Rk$ NN procedure. For the construction of RF and OTE, the R packages `randomForest` [61] and `OTE` [62] are utilized respectively. To construct SVM model the R package `kernlab` [63] is used. Moreover, for tuning the hyper-parameter  $k$ , the R function `tune.knn` is used which is implemented in the R package `e1071` [64]. For  $Rk$ NN, the hyper-parameters i.e. number of nearest neighbours  $k$ , number of base learners  $r$  etc. are tuned manually in the package `rknn` [60] which is also implemented in R. The R function `tune.randomForest` is used to tune `ntree`, `nodesize` and `mtry` in R package `e1071` [64]. The same tuned parameters are used for OTE procedure in R package `OTE` [62]. The R package `kernlab` [63] is used for SVM classifier with linear kernel and other default parameter.

### 4.3 Results

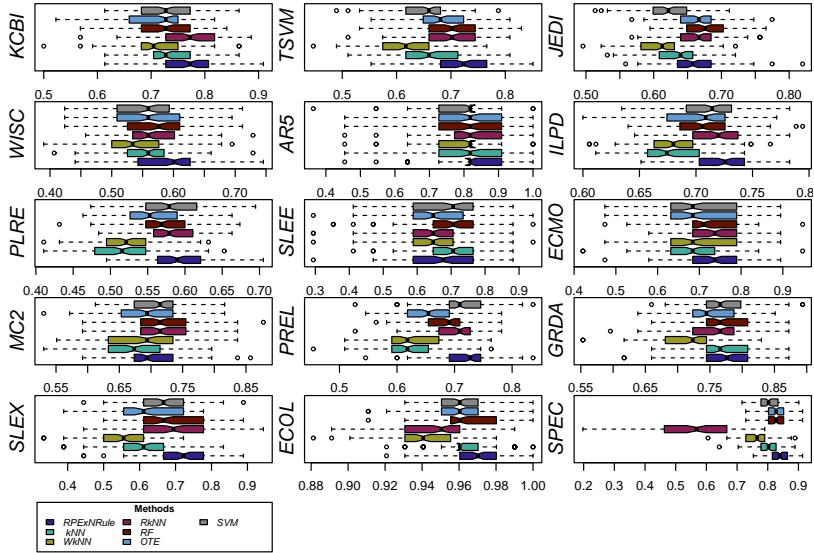
Table 3 gives the results for all considered datasets which shows that the proposed RPEXNRule has outperformed the other standard methods in majority of the cases. The novel RPEXNRule is providing maximum accuracy compared to the other state-of-the-art methods on 12 datasets. The standard  $Rk$ NN outperformed on 1 dataset, while RF gives high accuracy on 2 datasets. The classical  $k$ NN,  $Wk$ NN, OTE and SVM procedures do not give satisfactory results on any of the datasets.

In case of Cohen's kappa, the novel RPEXNRule provides optimal results on 8 datasets, while  $k$ NN method gives maximum kappa values on 2 out of the total datasets. The  $Wk$ NN procedure outperformed on 2 datasets, while  $Rk$ NN and OTE did not give optimal result on any of the datasets. The RF and SVM also give optimal results each on 2 datasets. Similarly, the proposed RPEXNRule ensemble outperformed all the other standard methods in terms of Brier score (BS). Furthermore, boxplots of the results are also constructed and provided in the Figures 2, 3 and 4.

Moreover, performance of the proposed RPEXNRule and other standard  $k$ NN based procedures (i.e.  $k$ NN,  $Wk$ NN and  $Rk$ NN) is also assessed by using various values of  $k$  (i.e.  $k = 3, 5, 7$ ). The results computed for 5 datasets are given in Table 4. It shows that the novel method has outperformed the other models in majority of the cases. The results also reveal that the proposed method is more stable for different  $k$  values as compared to other  $k$ NN base procedures. Furthermore, boxplots of these results are given in Figures 5, 6 and 7.

The performance of the proposed method is also assessed on synthetic datasets in comparison with the other classical  $k$ NN based classifiers. The results are given in Table 5, which shows optimal performance in majority of

the given scenarios. Table 5 reveals that the proposed RPEXNRule ensemble has optimal results when observations of both classes are mixed up and have high variation. Moreover, Figure 8 gives the boxplots of the results computed via RPEXNRule and other classical methods on the synthetic datasets.



**Fig. 2** This figure shows boxplots of accuracy of the proposed RPEXNRule and other classifiers for all datasets.

**Table 3** This table shows accuracy, Cohen’s kappa and Brier score (BS) of the proposed RPExNRule and other classifiers on all benchmark datasets.

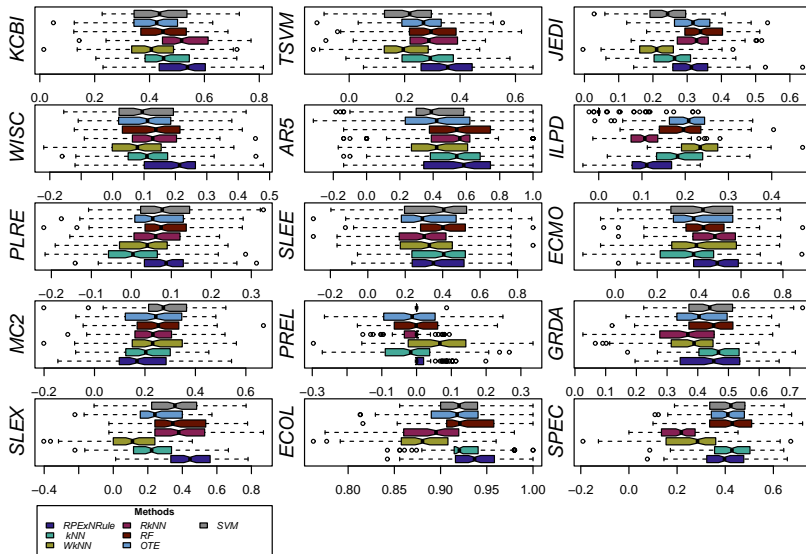
Metrics	Methods	Datasets															Mean
		KCBI	TSVM	JEDI	WISC	AR5	ILPD	PLRE	SLEE	ECMO	MC2	PREL	GRDA	SLEX	ECOL	SPEC	
Accuracy	OExNRule	<b>0.766</b>	<b>0.724</b>	0.663	<b>0.594</b>	<b>0.836</b>	<b>0.722</b>	<b>0.591</b>	0.675	<b>0.737</b>	0.701	<b>0.717</b>	<b>0.774</b>	<b>0.723</b>	<b>0.967</b>	<b>0.838</b>	<b>0.735</b>
	kNN	0.741	0.666	0.632	0.556	0.817	0.678	0.515	0.680	0.677	0.675	0.625	0.770	0.608	0.965	0.803	0.694
	WkNN	0.709	0.620	0.608	0.538	0.785	0.682	0.521	0.661	0.707	0.686	0.623	0.720	0.557	0.943	0.761	0.675
	RkNN	0.762	0.701	0.665	0.568	0.829	0.718	0.582	0.640	0.730	<b>0.715</b>	0.701	0.754	0.691	0.947	0.541	0.703
	RF	0.723	0.698	<b>0.677</b>	0.567	0.824	0.707	0.570	<b>0.684</b>	0.710	0.711	0.679	0.768	0.686	0.965	0.825	0.720
	OTE	0.711	0.682	0.663	0.561	0.790	0.702	0.559	0.650	0.702	0.694	0.653	0.747	0.628	0.959	0.824	0.702
	SVM	0.734	0.641	0.623	0.551	0.785	0.710	0.580	0.679	0.698	0.706	0.713	0.772	0.679	0.963	0.807	0.709
Kappa	OExNRule	<b>0.520</b>	<b>0.352</b>	0.314	<b>0.189</b>	<b>0.527</b>	0.116	<b>0.083</b>	0.358	<b>0.477</b>	0.182	0.023	0.440	<b>0.452</b>	<b>0.931</b>	0.395	<b>0.357</b>
	kNN	0.465	0.283	0.254	0.113	0.511	0.186	0.001	0.363	0.355	0.199	-0.027	<b>0.467</b>	0.221	0.927	<b>0.426</b>	0.316
	WkNN	0.406	0.201	0.211	0.076	0.429	<b>0.238</b>	0.033	0.322	0.412	0.241	<b>0.063</b>	0.374	0.125	0.882	0.262	0.285
	RkNN	0.516	0.289	0.320	0.132	0.519	0.108	0.064	0.288	0.461	0.232	-0.012	0.363	0.384	0.891	0.206	0.317
	RF	0.441	0.299	<b>0.347</b>	0.132	0.504	0.197	0.081	<b>0.378</b>	0.422	0.254	0.003	0.447	0.373	0.927	0.421	0.348
	OTE	0.418	0.265	0.316	0.117	0.426	0.201	0.064	0.309	0.402	0.235	-0.013	0.410	0.258	0.916	0.408	0.315
	SVM	0.447	0.212	0.241	0.101	0.419	0.018	<b>0.083</b>	0.363	0.397	<b>0.280</b>	0.001	0.447	0.363	0.924	0.407	0.314
BS	OExNRule	<b>0.167</b>	0.196	0.217	<b>0.244</b>	<b>0.117</b>	<b>0.177</b>	0.247	<b>0.214</b>	<b>0.180</b>	0.197	0.215	<b>0.157</b>	0.210	0.035	<b>0.124</b>	<b>0.180</b>
	kNN	0.186	0.236	0.266	0.308	0.133	0.218	0.324	0.243	0.227	0.237	0.278	0.186	0.271	0.034	0.140	0.219
	WkNN	0.291	0.380	0.392	0.462	0.215	0.318	0.479	0.339	0.293	0.314	0.377	0.280	0.443	0.057	0.239	0.325
	RkNN	0.176	0.202	0.219	0.267	0.122	0.183	<b>0.242</b>	0.244	0.183	0.209	0.237	0.168	0.234	0.073	0.166	0.195
	RF	0.180	<b>0.189</b>	<b>0.211</b>	0.255	0.126	0.178	0.247	0.227	0.186	<b>0.196</b>	0.238	0.166	0.236	0.034	0.128	0.186
	OTE	0.191	0.197	0.222	0.263	0.178	0.184	0.256	0.260	0.208	0.206	0.247	0.182	0.263	0.036	0.137	0.202
	SVM	0.193	0.216	0.226	0.250	0.148	0.200	0.244	0.238	0.222	0.198	<b>0.208</b>	0.165	<b>0.209</b>	<b>0.032</b>	0.128	0.192

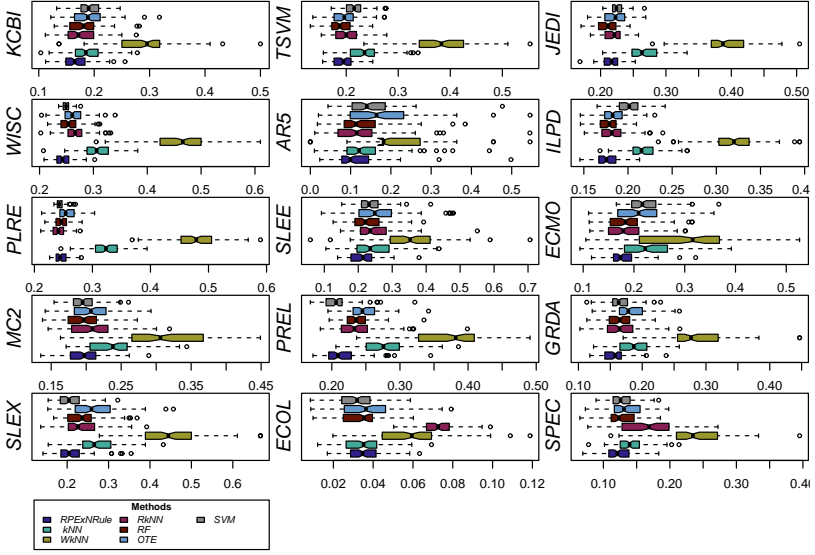
**Table 4** This table shows accuracy, Cohen’s kappa and Brier score (BS) of the proposed RPEXNRule and  $k$ NN based classifiers for different values of  $k$  on five benchmark datasets.

Metrics	Methods	Datasets															Mean
		<i>KCBI</i>			<i>TSVM</i>			<i>JEDI</i>			<i>WISC</i>			<i>AR</i>			
		<i>k</i> = 3	<i>k</i> = 5	<i>k</i> = 7	<i>k</i> = 3	<i>k</i> = 5	<i>k</i> = 7	<i>k</i> = 3	<i>k</i> = 5	<i>k</i> = 7	<i>k</i> = 3	<i>k</i> = 5	<i>k</i> = 7	<i>k</i> = 3	<i>k</i> = 5	<i>k</i> = 7	
Accuracy	RPEXNRule	<b>0.766</b>	0.751	0.736	<b>0.724</b>	<b>0.724</b>	<b>0.712</b>	<b>0.722</b>	<b>0.720</b>	<b>0.719</b>	<b>0.591</b>	<b>0.600</b>	<b>0.605</b>	<b>0.737</b>	<b>0.755</b>	<b>0.761</b>	<b>0.708</b>
	<i>k</i> NN	0.741	0.757	0.758	0.666	0.657	0.650	0.678	0.688	0.699	0.515	0.550	0.576	0.677	0.693	0.702	0.667
	W <i>k</i> NN	0.709	0.709	0.727	0.620	0.620	0.649	0.682	0.683	0.683	0.521	0.519	0.525	0.707	0.702	0.700	0.650
	R <i>k</i> NN	0.762	<b>0.765</b>	<b>0.765</b>	0.701	0.696	0.695	0.718	0.717	0.718	0.582	0.586	0.589	0.730	0.747	0.757	0.702
Kappa	RPEXNRule	<b>0.520</b>	0.486	0.450	<b>0.352</b>	<b>0.327</b>	<b>0.267</b>	0.116	0.077	0.057	<b>0.083</b>	<b>0.094</b>	0.098	<b>0.477</b>	<b>0.513</b>	<b>0.526</b>	<b>0.296</b>
	<i>k</i> NN	0.465	0.498	0.501	0.283	0.245	0.205	0.186	0.184	0.195	0.001	0.063	<b>0.109</b>	0.355	0.387	0.409	0.272
	W <i>k</i> NN	0.406	0.406	0.441	0.201	0.201	0.242	<b>0.238</b>	<b>0.221</b>	<b>0.215</b>	0.033	0.021	0.029	0.412	0.404	0.401	0.258
	R <i>k</i> NN	0.516	<b>0.524</b>	<b>0.525</b>	0.289	0.258	0.234	0.108	0.079	0.071	0.064	0.064	0.064	0.461	0.500	0.520	0.285
BS	RPEXNRule	<b>0.167</b>	<b>0.170</b>	0.173	<b>0.196</b>	<b>0.199</b>	<b>0.203</b>	<b>0.177</b>	<b>0.180</b>	<b>0.183</b>	0.247	0.247	<b>0.247</b>	<b>0.180</b>	0.183	0.186	<b>0.196</b>
	<i>k</i> NN	0.186	0.171	<b>0.169</b>	0.236	0.221	0.219	0.218	0.203	0.193	0.324	0.280	0.262	0.227	0.207	0.193	0.221
	W <i>k</i> NN	0.291	0.291	0.207	0.380	0.380	0.244	0.318	0.258	0.250	0.479	0.377	0.349	0.293	0.277	0.270	0.311
	R <i>k</i> NN	0.176	0.181	0.185	0.202	0.214	0.221	0.183	0.189	0.194	<b>0.242</b>	<b>0.245</b>	0.248	0.183	<b>0.179</b>	<b>0.185</b>	0.202

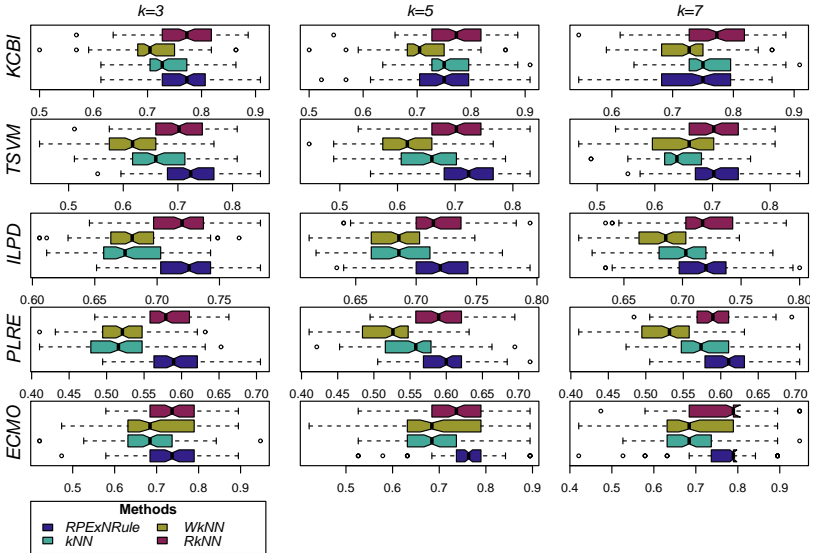
**Table 5** This table shows accuracy, Cohen's kappa and Brier score (BS) of the proposed RPEXNRule and  $k$ NN based classifiers on synthetic datasets.

Metrics	Methods	Scenario		
		$S_1$	$S_2$	$S_3$
Accuracy	RPEXNRule	<b>0.574</b>	<b>0.982</b>	0.987
	$k$ NN	0.555	0.971	0.966
	$Wk$ NN	0.530	0.927	0.948
	$Rk$ NN	0.560	0.978	<b>0.994</b>
Kappa	RPEXNRule	<b>0.184</b>	<b>0.964</b>	0.973
	$k$ NN	0.118	0.941	0.931
	$Wk$ NN	0.064	0.852	0.895
	$Rk$ NN	0.152	0.955	<b>0.987</b>
BS	RPEXNRule	<b>0.242</b>	0.072	0.041
	$k$ NN	0.300	<b>0.029</b>	0.029
	$Wk$ NN	0.470	0.073	0.052
	$Rk$ NN	0.245	0.069	<b>0.032</b>

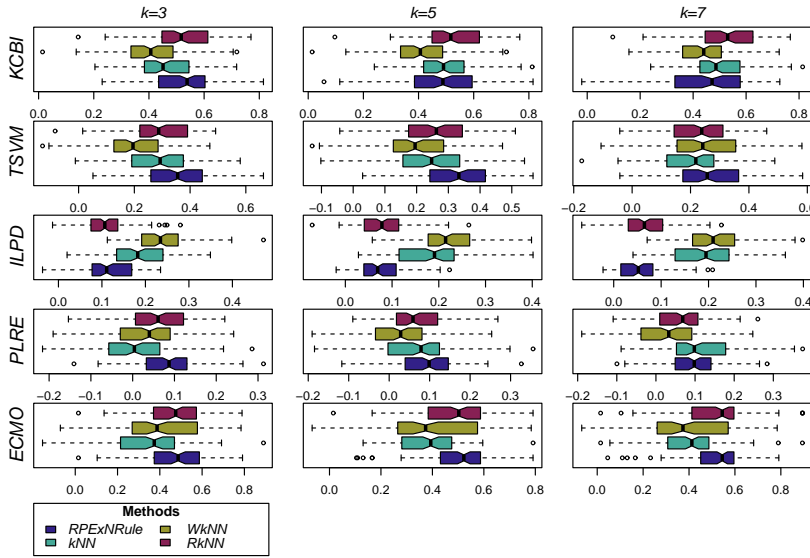
**Fig. 3** This figure shows boxplots of Cohen's kappa of the proposed RPEXNRule and other classifiers for all datasets.



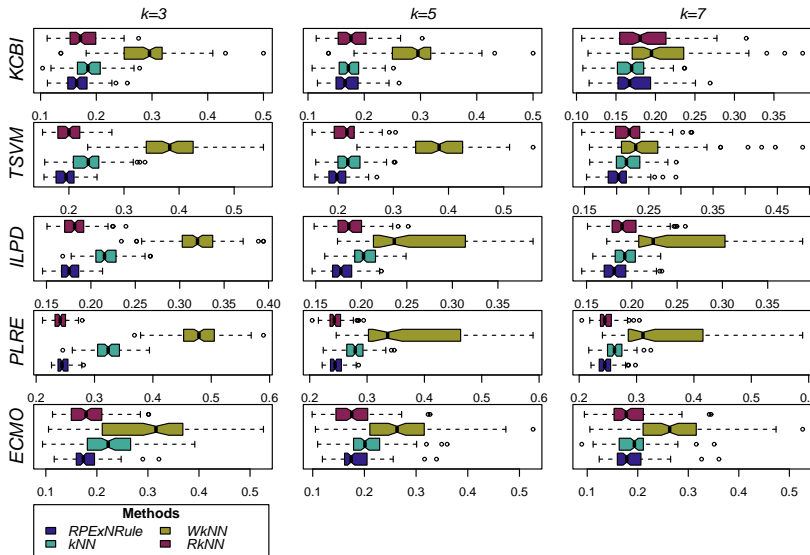
**Fig. 4** This figure shows boxplots of Brier score (BS) of the proposed RPEXNRule and other classifiers for all datasets.



**Fig. 5** This figure shows boxplots of accuracy of the proposed RPEXNRule and  $k$ NN based classifiers for different  $k$  values.

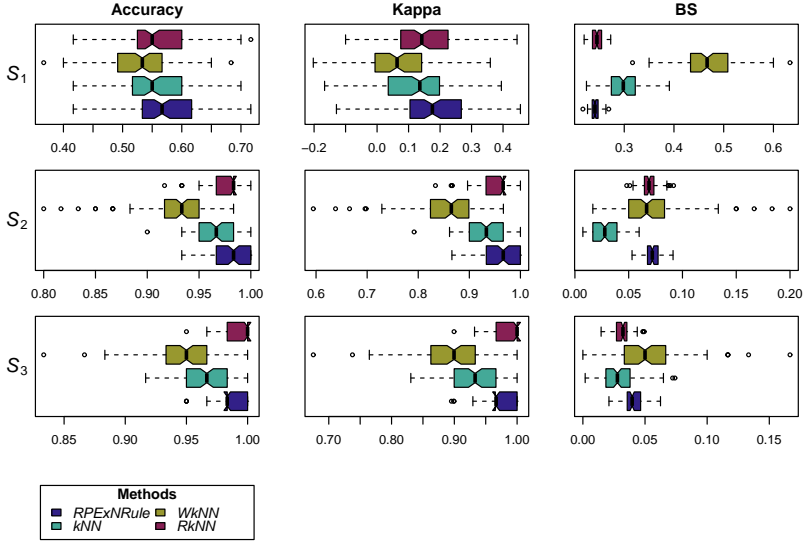


**Fig. 6** This figure shows boxplots of Cohen's kappa of the proposed RPEXNRule and  $k$ NN based classifiers for different  $k$  values.



**Fig. 7** This figure shows boxplots of Brier score (BS) of the proposed RPEXNRule and  $k$ NN based classifiers for different  $k$  values.





**Fig. 8** This figure shows boxplots of accuracy, Cohen’s kappa and Brier score (BS) of the proposed RPEXNRule and  $k$ NN based classifiers for synthetic datasets.

## 5 Conclusion

This paper proposed an ensemble method using extended neighbourhood rule (ExNRule) for constructing base  $k$ NN models, each on a randomly projected bootstrap sample. The  $k$  nearest observations to an unseen data point are identified in a  $k$  steps pattern. In the first step, it determines the nearest observation to the test sample point. In the second step, it searches for another observation which is closest to the previously selected sample point. This process is continued until the desired  $k$  observations are obtained in the neighbourhood. Each base learner uses majority voting in the response classes of the  $k$  nearest observations to predict the class of the unseen observation. The final estimated class of the test observation is a second round majority vote of the results obtained from the base learners. The performance of the proposed method is validated on 15 benchmark problems and 3 synthetic datasets. The results are compared with classical  $k$ NN, W $k$ NN, R $k$ NN, RF, OTE and SVM classifiers. The classification accuracy, Cohen’s kappa and Brier score (BS) are used as performance metrics.

The basic intuition behind the performance of the proposed ensemble is the step-wise selection of the  $k$  closest observations and randomly projecting the feature space. The standard  $k$ NN based ensembles ignore influential features in many cases which leads to its poor prediction performance. Moreover, these procedures fail to perform well in a case where unseen data observations follow a pattern of the nearest sample points in the training data that lie on a certain path not enclosed by the given sphere of neighbourhood. To overcome these

issues, the proposed RPEXNRule ensemble uses the extended neighbourhood rule (ExNRule) to fit  $k$ NN as base classifiers on randomly projected bootstrap samples. The results also reveal that the proposed method is less sensitive to the parameter  $k$  as compare to the other  $k$ NN base classifiers. Moreover, base models are constructed on randomly projected bootstrap samples, drawn from training data, which reduces the dimension without any information loss and ensure diversity due to randomization in the base learners.

The proposed ensemble combines a large number of  $B$  ExNRule base learners each constructed on a randomly projected bootstrap sample and the closest observation is selected in a step-wise manner, which makes the procedure time-consuming and laborious. To solve this problem, it is required to use R package `parallel` [65] to parallelize steps 8-16 of Algorithm 1. The performance can further be improved by selecting a suitable distance formula. Feature selection method such as [66–69], could be used for further improvement in the performance of the proposed method.

## References

- [1] Cunningham, P., Delany, S.J.: k-nearest neighbour classifiers-a tutorial. *ACM Computing Surveys (CSUR)* **54**(6), 1–25 (2021)
- [2] Altman, N.S.: An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician* **46**(3), 175–185 (1992)
- [3] Hastie, T., Tibshirani, R.: *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer, ??? (2009)
- [4] Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., *et al.*: Top 10 algorithms in data mining. *Knowledge and information systems* **14**(1), 1–37 (2008)
- [5] Fix, E., Hodges, J.: Discriminatory analysis, nonparametric discrimination: consistency properties. us air force school of aviation medicine. Technical Report 4 (3) (1951)
- [6] Cover, T., Hart, P.: Nearest neighbor pattern classification. *IEEE transactions on information theory* **13**(1), 21–27 (1967)
- [7] Hand, D.J.: Principles of data mining. *Drug safety* **30**(7), 621–622 (2007)
- [8] Amal, M.-A., Riadh, B.-A.A.: Survey of nearest neighbor condensing techniques. *International Journal of Advanced Computer Science and Applications* **2**(11), 59–64 (2011)
- [9] Kulkarni, S., Babu, M.V.: Introspection of various k-nearest neighbor techniques. *UACEE International Journal of Advances in Computer Science and Its Applications* **3**, 89–92 (2013)

- [10] Abbasifard, M.R., Ghahremani, B., Naderi, H.: A survey on nearest neighbor search methods. *International Journal of Computer Applications* **95**(25), 39–52 (2014)
- [11] Bailey, T., AK, J., et al.: A note on distance-weighted k-nearest neighbor rules. **8**(4), 311–313 (1978)
- [12] Alpaydin, E.: Voting over multiple condensed nearest neighbors. In: *Lazy Learning*, pp. 115–132. Springer, ??? (1997)
- [13] Zhai, J.-H., Li, N., Zhai, M.-Y.: The condensed fuzzy k-nearest neighbor rule based on sample fuzzy entropy. In: *International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 282–286 (2011). IEEE
- [14] Gowda, K., Krishna, G.: The condensed nearest neighbor rule using the concept of mutual nearest neighborhood (corresp.). *IEEE Transactions on Information Theory* **25**(4), 488–490 (1979)
- [15] Bay, S.D.: Nearest neighbor classification from multiple feature subsets. *Intelligent data analysis* **3**(3), 191–209 (1999)
- [16] Domeniconi, C., Yan, B.: Nearest neighbor ensemble. In: *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1, pp. 228–231 (2004). IEEE
- [17] García-Pedrajas, N., Ortiz-Boyer, D.: Boosting k-nearest neighbor classifier by means of input space projection. *Expert Systems with Applications* **36**(7), 10570–10582 (2009)
- [18] Steele, B.M.: Exact bootstrap k-nearest neighbor learners. *Machine Learning* **74**(3), 235–255 (2009)
- [19] Li, S., Harner, E.J., Adjero, D.A.: Random knn. In: *2014 IEEE International Conference on Data Mining Workshop*, pp. 629–636 (2014). IEEE
- [20] Gul, A., Perperoglou, A., Khan, Z., Mahmoud, O., Miftahuddin, M., Adler, W., Lausen, B.: Ensemble of a subset of k nn classifiers. *Advances in data analysis and classification* **12**(4), 827–840 (2018)
- [21] Zhang, Y., Cao, G., Wang, B., Li, X.: A novel ensemble method for k-nearest neighbor. *Pattern Recognition* **85**, 13–25 (2019)
- [22] Rashid, M., Mustafa, M., Sulaiman, N., Abdullah, N.R.H., Samad, R.: Random subspace k-nn based ensemble classifier for driver fatigue detection utilizing selected eeg channels. *Traitement du Signal* **38**(5), 1259–1270

- [23] Ali, A., Hamraz, M., Kumam, P., Khan, D.M., Khalil, U., Sulaiman, M., Khan, Z.: A k-nearest neighbours based ensemble via optimal model selection for regression. *IEEE Access* **8**, 132095–132105 (2020)
- [24] Ali, A., Hamraz, M., Gul, N., Khan, D.M., Khan, Z., Aldahmani, S.: A k nearest neighbours classifiers ensemble based on extended neighbourhood rule and features subsets. *arXiv preprint arXiv:2205.15111* (2022)
- [25] Gates, G.: The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory* **18**(3), 431–433 (1972)
- [26] Angiulli, F.: Fast condensed nearest neighbor rule. In: *Proceedings of the 22nd International Conference on Machine Learning*, pp. 25–32 (2005)
- [27] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: Knn model-based approach in classification. In: *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pp. 986–996 (2003). Springer
- [28] Yong, Z., Youwen, L., Shixiong, X.: An improved knn text classification algorithm based on clustering. *Journal of computers* **4**(3), 230–237 (2009)
- [29] Parvin, H., Alizadeh, H., Minaei-Bidgoli, B.: Mknn: Modified k-nearest neighbor. In: *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1 (2008). Citeseer
- [30] Sproull, R.F.: Refinements to nearest-neighbor searching in k-dimensional trees. *Algorithmica* **6**(1), 579–589 (1991)
- [31] An, S., Hu, Q., Wang, C., Guo, G., Li, P.: Data reduction based on nn-knn measure for nn classification and regression. *International Journal of Machine Learning and Cybernetics* **13**(3), 765–781 (2022)
- [32] Tahir, M.A., Bouridane, A., Kurugollu, F.: Simultaneous feature selection and feature weighting using hybrid tabu search/k-nearest neighbor classifier. *Pattern Recognition Letters* **28**(4), 438–446 (2007)
- [33] Rohban, M.H., Rabiee, H.R.: Supervised neighborhood graph construction for semi-supervised classification. *Pattern Recognition* **45**(4), 1363–1372 (2012)
- [34] Chen, M., Li, L., Wang, B., Cheng, J., Pan, L., Chen, X.: Effectively clustering by finding density backbone based-on knn. *Pattern Recognition* **60**, 486–498 (2016)
- [35] Xu, Y., Zhang, Y., Zhao, J., Yang, Z., Pan, X.: Knn-based maximum margin and minimum volume hyper-sphere machine for imbalanced data classification. *International Journal of Machine Learning and Cybernetics*

**10**(2), 357–368 (2019)

- [36] Yuan, X., Yu, H., Liang, J., Xu, B.: A novel density peaks clustering algorithm based on k nearest neighbors with adaptive merging strategy. *International Journal of Machine Learning and Cybernetics* **12**(10), 2825–2841 (2021)
- [37] Sun, L., Qin, X., Ding, W., Xu, J., Zhang, S.: Density peaks clustering based on k-nearest neighbors and self-recommendation. *International Journal of Machine Learning and Cybernetics* **12**(7), 1913–1938 (2021)
- [38] Zhang, Z.-L., Luo, X.-G., Zhou, Q.: Drcw-frknn-ovo: distance-based related competence weighting based on fixed radius k nearest neighbour for one-vs-one scheme. *International Journal of Machine Learning and Cybernetics* **13**(5), 1441–1459 (2022)
- [39] Breiman, L.: Bagging predictors. *Machine learning* **24**(2), 123–140 (1996)
- [40] Caprile, B., Merler, S., Furlanello, C., Jurman, G.: Exact bagging with k-nearest neighbour classifiers. In: *International Workshop on Multiple Classifier Systems*, pp. 72–81 (2004). Springer
- [41] Zhou, Z.-H., Yu, Y.: Adapt bagging to nearest neighbor classifiers. *Journal of Computer Science and Technology* **20**(1), 48–54 (2005)
- [42] Zhou, Z.-H., Yu, Y.: Ensembling local learners through multimodal perturbation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **35**(4), 725–735 (2005)
- [43] Li, S., Harner, E.J., Adjero, D.A.: Random knn feature selection—a fast and stable alternative to random forests. *BMC bioinformatics* **12**(1), 1–11 (2011)
- [44] Gu, J., Jiao, L., Liu, F., Yang, S., Wang, R., Chen, P., Cui, Y., Xie, J., Zhang, Y.: Random subspace based ensemble sparse representation. *Pattern Recognition* **74**, 544–555 (2018)
- [45] Grabowski, S.: Voting over multiple k-nn classifiers. In: *Modern Problems of Radio Engineering, Telecommunications and Computer Science (IEEE Cat. No. 02EX542)*, pp. 223–225 (2002). IEEE
- [46] Zhang, S., Li, X., Zong, M., Zhu, X., Wang, R.: Efficient knn classification with different numbers of nearest neighbors. *IEEE transactions on neural networks and learning systems* **29**(5), 1774–1785 (2017)
- [47] Aburomman, A.A., Reaz, M.B.I.: A novel svm-knn-pso ensemble method for intrusion detection system. *Applied Soft Computing* **38**, 360–372

(2016)

- [48] Gallego, A.-J., Calvo-Zaragoza, J., Valero-Mas, J.J., Rico-Juan, J.R.: Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation. *Pattern Recognition* **74**, 531–543 (2018)
- [49] Zhang, Y., Cao, G., Wang, B., Li, X.: A novel ensemble method for k-nearest neighbor. *Pattern Recognition* **85**, 13–25 (2019)
- [50] Zhai, J., Zhang, S., Wang, C.: The classification of imbalanced large data sets based on mapreduce and ensemble of elm classifiers. *International Journal of Machine Learning and Cybernetics* **8**(3), 1009–1017 (2017)
- [51] Yuan, P., Wang, B., Mao, Z.: Using multiple classifier behavior to develop a dynamic outlier ensemble. *International Journal of Machine Learning and Cybernetics* **12**(2), 501–513 (2021)
- [52] Johnson, W.B.: Extensions of lipschitz mappings into a hilbert space. *Contemp. Math.* **26**, 189–206 (1984)
- [53] Fradkin, D., Madigan, D.: Experiments with random projections for machine learning. In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 517–522 (2003)
- [54] Sulic, V., Perš, J., Kristan, M., Kovacic, S.: Efficient dimensionality reduction using random projection. In: *15th Computer Vision Winter Workshop*, pp. 29–36 (2010)
- [55] Vanschoren, J.: Sleuth Ex 2015 Dataset (SLEX). Accessed date: 2022-06-13
- [56] Vanschoren, J.: Ecoli Dataset. Accessed date: 2022-06-13
- [57] Krzysztof, J.C., Kurgan, L.A.: SPECT Heart Data Set. Accessed date: 2022-06-13
- [58] Kuhn, M.: *Caret: Classification and Regression Training*. (2021). R package version 6.0-90. <https://CRAN.R-project.org/package=caret>
- [59] Schliep, K., Hechenbichler, K.: *Kknn: Weighted k-Nearest Neighbors*. (2016). R package version 1.3.1. <https://CRAN.R-project.org/package=kknn>
- [60] Li, S.: *Rknn: Random KNN Classification and Regression*. (2015). R package version 1.2-1. <https://CRAN.R-project.org/package=rknn>
- [61] Liaw, A., Wiener, M.: *Classification and regression by randomforest*. R

News **2**(3), 18–22 (2002)

- [62] Khan, Z., Gul, A., Perperoglou, A., Mahmoud, O., Adler, W., Miftahuddin, Lausen, B.: OTE: Optimal Trees Ensembles for Regression, Classification and Class Membership Probability Estimation. (2020). R package version 1.0.1. <https://CRAN.R-project.org/package=OTE>
- [63] Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software* **11**(9), 1–20 (2004)
- [64] Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F.: E1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. (2021). R package version 1.7-9. <https://CRAN.R-project.org/package=e1071>
- [65] R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2021). R Foundation for Statistical Computing. <https://www.R-project.org/>
- [66] Tawhid, M.A., Ibrahim, A.M.: Feature selection based on rough set approach, wrapper approach, and binary whale optimization algorithm. *International journal of machine learning and cybernetics* **11**(3), 573–602 (2020)
- [67] Hamraz, M., Gul, N., Raza, M., Khan, D.M., Khalil, U., Zubair, S., Khan, Z.: Robust proportional overlapping analysis for feature selection in binary classification within functional genomic experiments. *PeerJ Computer Science* **7**, 562 (2021)
- [68] Hashemi, A., Dowlatshahi, M.B., Nezamabadi-pour, H.: Ensemble of feature selection algorithms: a multi-criteria decision-making approach. *International Journal of Machine Learning and Cybernetics* **13**(1), 49–69 (2022)
- [69] Wang, Z., Zhong, J., Chen, Y., Zhong, P.: Sparse robust multiview feature selection via adaptive-weighting strategy. *International Journal of Machine Learning and Cybernetics* **13**(5), 1387–1408 (2022)

