A Weighted $k$-Nearest Neighbours Ensemble With Added Accuracy and Diversity

NAZ GUL$^1$, MUHAMMAD AAMIR$^1$, (Senior Member, IEEE), SAEED ALDAHMANI$^2$, AND ZARDAD KHAN$^2$

$^1$Department of Statistics, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan
$^2$Department of Analytics in the Digital Era, United Arab Emirates University, Al Ain, United Arab Emirates

Corresponding authors: Zardad Khan (zaar@uaeu.ac.ae) and Saeed Aldahmani (saldahmani@uaeu.ac.ae)

ABSTRACT Ensembles based on $k$NN models are considered effective in reducing the adverse effect of outliers, primarily, by identifying the closest observations to a test point in a given training data. Class label of the test point is estimated by taking a majority-vote of the nearest observations' class labels. While identifying the closest observations, certain training patterns might possess high regulatory power than the others. Therefore, assigning weights to observations and then calculating weighted distances are deemed important in addressing this scenario. This paper proposes a $k$NN ensemble that identifies nearest observations based on their weighted distance in relation to the response variable via support vectors. This is done by building a large number of $k$NN models each on a bootstrap sample from the training data along with a randomly selected subset of features from the given feature space. The estimated class of the test observation is decided via majority voting based on the estimates given by all the base $k$NN models. The ensemble is assessed on 14 benchmark and simulated datasets against other classical methods, including $k$NN based models using Brier score, classification accuracy and Kappa as performance measures. On both the benchmark and simulated datasets, the proposed ensemble outperformed the other competing methods in majority of the cases. It gave better overall classification performance than the other methods on 8 datasets. The analyses on simulated datasets reveal that the proposed method is effective in classification problems that involve noisy features in the data. Furthermore, feature weighting and randomization also make the method robust to the choice of $k$, i.e., the number of nearest observations in a base model.

INDEX TERMS Classification, feature weighting, $k$-nearest neighbor ensemble, support vectors.

I. INTRODUCTION

A wide range of supervised learning techniques have been introduced to deal with classification problems. Among these, the nearest neighbour (NN) model is one of the top ranked methods that classifies an unseen observation on the basis of its neighbourhood in a given feature space. Although it is an efficient method but it suffers from the issue of over-fitting. To overcome this disadvantage, one of the most fundamental, simple and appealing approach is $k$-Nearest Neighbors ($k$NN) method [1]. This technique can be used for dealing with both classification and regression problems in machine learning.

In the context of classification, the $k$NN approach estimates a class value for a new/unseen instance by finding its $k$ nearest neighbors whose classes are known [2], [3], [4]. Initially, it was developed to perform discriminant analysis in situations where reliable parametric estimates of probability densities are not known. Now-a-days, the $k$NN method is a preferred method in classifying data when there is little or no prior knowledge about the distribution of the data [5]. It is a widely used classifier because of its simplicity. It shows robustness to noisy training data and effectiveness in case of large training data [5].

In spite of the above advantages this technique has some disadvantages such as: high computation cost due to computing distance of each query point to all the training samples, requirement of large memory for implementation of data
The choice of the parameter $k$ affects the performance of the $k$NN. When $k$ is too small, the method would be more sensitive to the data points that lie outside the general pattern of the data. Similarly, when $k$ is too large, too many points from other classes may be in the neighbourhood [1]. There is no appropriate way to find the value of $k$, however, the square root of the total number of observations in the given dataset is considered as an optimal value of $k$. Generally, $k$ value is taken in odd so as to avoid ties during decision making.

An error or accuracy plot is usually used to find the most suitable value of $k$. Thus $k$ in $k$NN is the core deciding factor.

$k$NN uses different types of distance metrics for calculating distances. The most commonly used method is the Euclidean distance, the straight line distance between the specified training sample and a test sample [9]. Manhattan Distance—the absolute differences of the distance between two points i.e Cartesian coordinates and Hamming Distance—that compare two binary data strings by looking into the entire dataset to discover when data points are similar or dissimilar, are the other choices as distance metrics used in literature. Different distances would yield different accuracy results for a given problem. Other modifications in terms of the distance metric can be found in [10], [11], and [12].

Theoretically, the $k$NN classification performance is determined by the estimate of the conditional class probabilities of the query point in a local region of the data space, which is found by the distance of the $k$th nearest neighbor to the query point. Different values of $k$ yields different conditional class probabilities which make the selection of the neighborhood size $k$ more sensitive and thus affect the estimate. For very small value of $k$, the estimate tends to be very poor owing to the data noise, sparseness, ambiguous and mislabeled points. The increase in the $k$ value smooths the estimate by taking into account a large region around the query point. On the other hand, a very large value of $k$ makes the estimate over-smooth with the introduction of outliers from other classes which degrade classification performance [13]. This reduces the generalizability of the method.

To deal with the sensitivity issue of different neighborhood size $k$, several weighted voting methods have been developed in the literature.

The basic idea behind weighted $k$NN (W$k$NN), is to give more weight to the nearby points and less weight to the points that are far away. Any function whose value decreases as the distance increases can be used as a weight. The most simple function usually used is the inverse distance function. Thus different weights are assigned to the $k$ neighbors based on their distance to the test point which makes use of training observations [14]. Many related models are proposed based on weighting scheme of $k$NN classifiers, some of them can be found in [13], [15], [16], [17], and [18].

Although W$k$NN shows better performance than other weighting methods for $k$NN through their empirical comparisons in many cases, it still suffers from the issue of outliers in the data. This issue particularly occurs when the sample size is small [19], [20]. Moreover, the irregular class distribution of a data set affects both $k$NN and W$k$NN. The later is also affected by the choice of $k$ [13], [21]. The presence of irrelevant features in the given feature space also affect $k$NN, often in data with large number of features.

Ensemble procedures have been proposed to improve the performance in such situations [22], [23], [24], [25], [26]. Pooling a large number of base models is termed as ensemble learning method. This approach has gained a lot of interest by developing efficient and sophisticated techniques to improve the classification accuracy of base models. Diversity and accuracy are the two desirable characteristics of ensemble methods stating that errors made by classifiers are uncorrelated and weak models do not take part in decision making. The most frequently used procedure is bagging (bootstrap aggregation), that pools the results of classifiers fitted on bootstrap samples, drawn randomly from training data [27], [28], [29]. Similarly, an extended weighted voting ensemble method, ensembles (groups) base classifiers and then select the group with the highest vote based on the single classifiers weights [30]. Motivated by this idea, this work explores the idea of features weighting instead of assigning weights to observation in conjunction with random subsets of feature selection for ensemble construction. The proposed weighting leads to accurate base models while building them on random feature spaces to introduce diversity. Moreover, the proposed method is also robust to the choice of $k$ in that only regulatory features take part in decision making whereas the rest are penalized by assigning them low weights. The key features of the proposed ensemble method, that makes the proposal novel, are:

- Base models in the ensemble are accurate due to feature weighting.
- The ensemble consist of diverse models due to bootstrapping and feature subsets.
- The ensemble is robust to the choice of $k$ due to feature weighting by support vectors which works equally efficient in case of large or small neighbourhood values.

II. RELATED WORK

Several methods are proposed in literature based on the idea of weighting and randomization. The selection of the neighborhood size $k$ still remains a major challenging problem to be solved for $k$NN. Modified $k$ Nearest Neighbor(M$k$NN) is proposed in the literature. It uses weights like weighted $k$NN while looking for the validity of the data point when classifying the nearest neighbour. The validity of all data samples is computed in the training set according to its neighbors, and then a weighted $k$NN is performed on any
test samples [15]. The distance-weighted $k$-nearest neighbor rule (D$k$NN) [13] is another approach to improve the classification performance of $k$NN. D$k$NN uses a dual distance-weighted function based on $W_k$NN to determine the class of the query point by majority weighted voting [13].

It is well known that $k$NN is a time consuming method because of the proportionality of the classification time to the number of features and the number of training instances [15], [31]. To overcome this issue, an attempt called the “$k$ Nearest Neighbors on Feature Projections” ($k$NNFP) [31] was made. This technique sorts training instances as their projections on each feature dimension separately to achieve fast classification of a new instance as compared to $k$NN. $k$NNFP first makes one set of predictions for each feature and then builds a $k$NN on the projections. A new instance is finally classified through a majority voting on individual classifications made by each feature. All the features are equally relevant, that is, each feature has the same power in the voting is the basic assumption of $k$NNFP approach. To some extent, this reduces the hindrance of irrelevant features. However, if the irrelevant features are in large number, then voting alone is not sufficient. To overcome this issue, $W_k$NNFP [17] was introduced that investigates the effect of incorporating feature weights while voting by multiplying the vote of each feature with its weight. This algorithm stores all the projections of training instances on linear features in memory as sorted values. Then the vote of the feature and its distance from the test point is computed to give ultimate classification. Model based $k$ nearest neighbor [16] models the input data and use the fitted model for classification of the data. This approach not only improves the accuracy but also shows more efficiency in terms of execution time. It also chooses appropriate value for $k$ automatically.

In the context of ensemble approaches, a large number of base $k$NN models also known as pooling several nearest neighbor models via multiple feature subsets (MFS) is proposed in the literature. A random subset of feature space is used for each base model of the ensemble. The final class is predicted by pooling the results of base models [32]. Rank Nearest Neighbour (RNN) is another modification based on the classic $k$NN, which tries to improve accuracy with less execution time by assigning ranks to training data for each category [18]. A similar technique, random $k$NN, is used to classify high dimensional data sets. The features are ranked on the basis of their discriminative power and the resultant set of top ranked features is obtained for the ultimate model [33]. Another approach uses Term Frequency-Inverse Document Frequency (TF-IDF) as a weighting scheme to assign weight to permission features [34]. Similarly, the work done by [35] first ranks the permission attributes using Information Gain (IG) as a feature selection method and then the obtained ranked features weights are assigned by implementing ensemble extra trees on the ranked features to produce feature subsets that represent the attribute properties. The final subsets of features are used to update the sets of observations with 5, 10 and 20 top ranked features. Then, weightages are computed using ensemble extra trees on the updated datasets to produce a permission feature model.

The BagInRand [36] technique adapted bagging to $k$NN classifiers by inducing randomness to distance metrics. Double-Bagging approach use out-of-bag observations to train a second classifier. To perform a linear discriminate analysis on each bootstrapping step, the out-of-bag sample is used. The discriminate variables of each bootstrap sample are incorporated as supplementary predictors for the classification tree [37]. The work proposed in [38] and [39] reduce data size and boost up the execution time by removing those samples that are identical and do not provide extra information.

Although, a model based $k$NN improve the prediction performance and reduces the size of training data, however, this procedure fails in case of class imbalanced problem. The $k$-d tree nearest neighbour [40] is developed by dividing the total training dataset in half plane and use it for the formation of multi-dimensional observations. This method produces perfectly balanced tree in less time. However, $k$-d tree nearest neighbour is computationally complex and misclassify data pattern. A hybrid method based on SVM and $k$NN, which deals with multi-class problems and gives appreciable performance is proposed in [41]. The authors in [42] have suggested to combine different base $k$NN learners using various distance function weights acquired by genetic algorithm. To introduce diversity in the ensemble, authors in [43] have used different metrics as perturbations parameters, for distance calculation. To select a diverse and accurate optimal base models, the work in [44] tried to improve bagging by applying an optimization process called Selecting Base Classifiers on Bagging(SBCB). This work propose feature weighting for identifying the nearest neighbours to estimate the class label of a test/unseen observation. Features that do not possess high discriminative ability will be assigned low weights while those that actually regulate the response in the training data will be assigned high weights. All the above mentioned methods, fail to provide a frame work of ensemble construction that builds accurate and diverse models and show robustness to the choice of $k$. Therefore, this paper proposes an ensemble methods based on $k$ nearest neighbours classifiers that achieves the two desirable characteristics of accuracy and diversity and is also robust to the choice of the number of nearest neighbours.

### III. PROPOSED METHOD

The proposed technique give weights to $k$-Nearest neighbours like weighted $k$NN by identifying discriminative features. Furthermore, for extra randomization and avoiding overfitting, base $k$NN models are built on bootstrap sample with random feature subsets. This leads to an ensemble of diverse/random and accurate $k$NN (DR$k$NN) models.

Suppose a training dataset $L = (X_{n \times p}, Y)$, where $X_{n \times p}$ is a feature space with $n$ observations and $p$ features and $Y$ is a binary response variable and $X'_{i \times p}$ is a test point. Let $B$ bootstrap samples are taken from $L$, each with a random
sub-sample of \( d \leq p \) features. To determine \( k \) observations in the neighbourhood of a test point, fit a kNN model using a distance formula with feature weights \( w \) computed from support vector machine (SVM). The method uses majority voting to predict the class of the test point in each model. In this way, \( B \) estimates of the unseen observation are obtained i.e., \( \hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \ldots, \hat{Y}^B \). The final predicted class of the unknown observation \( X'_{1 \times p} \) is a second round majority vote of the predictions given by base models.

The ordinary kNN model uses the following Euclidean distance formula to determine the nearest observations:

\[
D(X'_{1 \times p}, X_{1 \times p'}) = \sqrt{\sum_{j=1}^{p'} (x'_j - x_j)^2},
\]

whereas the proposed method uses a Euclidean distance in conjunction with weights \( w \) given by SVM classifier:

\[
D_w(X'_{1 \times p}, X_{1 \times p'}) = \sqrt{\sum_{j=1}^{p'} w(x'_j - x_j)^2}.
\]

**Feature Weights \( w \):**

As the SVM algorithm uses a hyperplane (\( H \)) to classify the data points in their respective classes i.e.,

\[
H : w^T(\psi(x)) + b = 0
\]

The distance between a given point \( \psi(x_0) \) and the hyperplane \( H \) is given by

\[
d_H(\psi(x)) = \frac{|w^T(\psi(x)) + b|}{||w||_2},
\]

where, \( ||w||_2 \) is the Euclidean norm defined as

\[
||w||_2 = \sqrt{w_1^2 + w_2^2 + w_3^2 + \ldots + w_p^2}
\]

The weight vectors are the arguments that maximize the distance given in Equation 3, that is,

\[
w = \arg \max_{w} \left[ \min \{d_H(\psi(x)) \} \right]
\]

Based on the above description, the proposed algorithm takes the following steps for weighted distance calculations in the base kNN models and ensemble formation.

1. Take \( B \) bootstrap samples from the given training data \( \mathcal{L} = (X_{n \times p}, Y) \) considering a random subset of features with each sample.
2. For each bootstrap sample and the random feature subset, build a base kNN model using Equation 2 to find the weighted distances of observations to a test point.
3. Use majority voting to estimate the class label of the test point based on the nearest neighbours identified in each of the \( B \) models.
4. Allow the \( B \) models collectively to vote for ultimate class label prediction.

The pseudocode of the proposed DR\&NN ensemble is given in Algorithm 1 and its flowchart is shown in Figure 1.

### Algorithm 1 Pseudocode of the Novel Algorithm

1. \( X_{n \times p} \) -- Feature space with \( p \) features and \( n \) instances.
2. \( y_n \) -- Response variable of training observations.
3. \( X'_{1 \times p} \) -- An unseen observation.
4. \( B \) -- Total number of bootstrap samples taken from training data.
5. \( k \) -- User specified number of nearest observations.
6. \( p \) -- Number of features in the data.
7. \( d \) -- Size of feature subset; where \( d \leq p \).
8. for \( b \leftarrow 1 : B \) do
9. \( S_{n \times d} \) -- Take a bootstrap sample with \( d \) features from the training data
10. \( X'_{1 \times d} \) -- Subset of \( d \) values from test point \( X'_{1 \times p} \)
11. \( H(.) \) -- Fit a kNN model using distance formula in Equation 2.
12. \( \hat{Y}^b \) -- Predict the unseen observation
13. end for
14. \( \hat{Y} \) = majority vote of \((\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \ldots, \hat{Y}^B)\)

**IV. EXPERIMENT AND RESULTS**

To assess the performance of the proposed method and to compare the results with the other stat-of-the-art methods a total of 14 benchmark datasets are considered. These datasets are described in the following sub-section.

**A. BENCHMARK DATASETS**

The benchmark datasets used in this paper are taken from openml. A brief description of these datasets is given in Table 1. The table shows the number of features \( p \), number of observations \( n \), class-wise distribution and a hyperlink to the source of each dataset.

**B. EXPERIMENTAL SETUP**

Each dataset is divided into two mutually exclusive parts, i.e., 70% training and the remaining 30% as testing. A total
of 500 individual learners are constructed for every method, each on a random bootstrap sample taken from the training observations. For the proposed method, a subset of features, i.e., $p' = \sqrt{p}$ and $k = 3$ neighbours are selected. The predictions are given by each base model using majority voting. Final prediction is the model value of the results produced by all 500 base learners. The other methods used for comparison are the k-nearest neighbor (kNN), Random kNN (R$k$NN), weighted kNN (W$k$NN), random forest (RF) and support vector machine (SVM).

The R packages caret [45] and kknn [46] are used for kNN and weighted kNN respectively, while for random kNN, the R library rknn [47] is used. Similarly, the R libraries kernlab [48] and randomForest [49] are used for SVM and random forest models, respectively. The underlying parameters of all the methods are fine tuned via 10-fold cross validation.

**C. DISCUSSION ON RESULTS FROM BENCHMARKING**

It is clear from the table 2 that the proposed method shows overall good performance on 8 out of 14 datasets. R$k$NN is showing better results than the others on KCB1 dataset. Similarly, R$k$NN also show higher accuracy and kappa for the data set sonar while Brier score on this dataset is lower for the proposed method. kNN show high accuracy and less prediction error on dataset Vine, while kappa is high for SVM method. The proposed method show high accuracy on dataset MC, while SVM and R$k$NN show good results for the same dataset in terms of kappa and Brier score, respectively. The proposed and other kNN based techniques are also assessed for the different values of parameter k, i.e., $k = 3, 5, 7$. The results are given in Table 3. It is clear from the table that the proposed method outperformed the others on the majority of the datasets in terms of all the performance metrics and thus, show that proposed method is not affected by the parameter $k$ as much as the other kNN based methods. Boxplots are constructed for accuracy, kappa and BS to provide further illustration as given in Figures 2-6.

**D. SIMULATION STUDY**

The proposed method is assessed under two different scenarios, two datasets each with 30 features and 200 observations with binary response are generated. In the first scenario, the first 10 features are important as they regulate the response variable. The remaining 20 features do not regulate the response. Based on the important features, out of total

**TABLE 1.** A short description of the datasets used in this paper.

| Data    | $p$  | $n$  | Class wise distribution | Source                                    |
|---------|------|------|-------------------------|-------------------------------------------|
| KCB1    | 86   | 145  | (85, 60)                | https://www.openml.org/d/1066             |
| Vine    | 8    | 52   | (28, 24)                | https://www.openml.org/d/0815             |
| Sleep   | 7    | 62   | (33, 29)                | https://www.openml.org/d/07359            |
| MC      | 39   | 161  | (109, 52)               | https://www.openml.org/d/1054             |
| Sonar   | 60   | 208  | (111, 97)               | https://www.openml.org/d/0040             |
| Wisc    | 32   | 194  | (104, 90)               | https://www.openml.org/d/0753             |
| TSVM    | 80   | 156  | (54, 102)               | https://www.openml.org/d/41976            |
| PlasmaR | 13   | 315  | (133, 182)              | https://www.openml.org/d/0915             |
| EMon    | 9    | 130  | (66, 64)                | https://www.openml.org/d/0944             |
| grub-dam| 9    | 155  | (106, 49)               | https://www.openml.org/d/0000             |
| ILPD    | 10   | 583  | (415, 167)              | https://www.openml.org/d/1480             |
| Heart   | 13   | 303  | (204, 99)               | https://www.openml.org/d/                 |
| PRelax  | 12   | 182  | (130, 52)               | https://www.openml.org/d/1490             |
| Cleve   | 13   | 303  | (165, 138)              | https://www.openml.org/d/40710            |
| Method | SVM | RF | KNN | WNN | ANN | BRNN |
|--------|-----|-----|-----|-----|-----|------|
| Accuracy | 0.836 | 0.768 | 0.765 | 0.779 | 0.756 | 0.783 |
| Kappa   | 0.0368 | 0.0365 | 0.0356 | 0.0366 | 0.0355 | 0.0363 |

**Table 2:** Results of the proposed DRNN and other state-of-the-art methods.
### TABLE 3. Results of the proposed DR\(k\)NN and other \(k\)NN based methods for different values of the hyper parameter \(k\).

| Metrics | Methods | Datasets |
|---------|---------|----------|
|         |         | TSVMs    | Wisc    | MC       |
|         |         | \(k = 3\) | \(k = 5\) | \(k = 7\) | \(k = 3\) | \(k = 5\) | \(k = 7\) | \(k = 3\) | \(k = 5\) | \(k = 7\) |
| Accuracy| DR\(k\)NN | 0.7106 | 0.6991 | 0.6917 | 0.5718 | 0.5777 | 0.5828 | 0.7208 | 0.7248 | 0.7230 |
|         | \(k\)NN | 0.6657 | 0.6570 | 0.6502 | 0.5561 | 0.5749 | 0.5728 | 0.6769 | 0.6916 | 0.6869 |
|         | Wk\(k\)NN | 0.6202 | 0.6202 | 0.6491 | 0.5379 | 0.5379 | 0.5464 | 0.6769 | 0.6863 | 0.6906 |
|         | Rk\(k\)NN | 0.6980 | 0.6942 | 0.6934 | 0.5649 | 0.5744 | 0.5764 | 0.7153 | 0.7185 | 0.7183 |
| Kappa   | DR\(k\)NN | 0.3084 | 0.2655 | 0.2333 | 0.1399 | 0.1516 | 0.1612 | 0.2461 | 0.2490 | 0.2428 |
|         | \(k\)NN | 0.2827 | 0.2450 | 0.2053 | 0.1132 | 0.1506 | 0.1484 | 0.2018 | 0.2117 | 0.1820 |
|         | Wk\(k\)NN | 0.2013 | 0.2013 | 0.2021 | 0.0763 | 0.0763 | 0.0939 | 0.2412 | 0.2412 | 0.2343 |
|         | Rk\(k\)NN | 0.2793 | 0.2509 | 0.2297 | 0.1246 | 0.1435 | 0.1476 | 0.2341 | 0.2278 | 0.2275 |
| BS      | DR\(k\)NN | 0.1943 | 0.2004 | 0.2061 | 0.2508 | 0.2516 | 0.2517 | 0.1967 | 0.2008 | 0.2058 |
|         | \(k\)NN | 0.2356 | 0.2209 | 0.2188 | 0.3078 | 0.2731 | 0.2558 | 0.2371 | 0.2198 | 0.2157 |
|         | Wk\(k\)NN | 0.3797 | 0.3797 | 0.2439 | 0.4620 | 0.4620 | 0.3586 | 0.3136 | 0.3136 | 0.2481 |
|         | Rk\(k\)NN | 0.1951 | 0.1950 | 0.1953 | 0.2518 | 0.2517 | 0.2519 | 0.1946 | 0.1947 | 0.2447 |

200 samples 100 are generated from a distribution with fix parameters and assigned to one class, i.e., 0, and the remaining 100 generated from the same distribution with different parameter values are reserved for Class 1. The remaining features are randomly generated that do not affect the response. The idea is to have two different sets of features, one that consists of informative variables where feature weighting is expected to work, and the other with non-informative variables that are expected to be ignored by the algorithm while identifying the nearest neighbours for a new/unseen observation.

In the second scenario, observations on features are generated in such a manner that corresponding to both the classes, they do not differ significantly between the classes and there is no discrimination between important and non-important features. In this way, features are not allowed to differ significantly between the two classes. Furthermore, in this scenario, all the features carry equal importance.

Table 4 gives a brief description of the synthetic datasets, where the first column shows the ID of the datasets, the second and third columns represent feature’s distributions according to Class 0 and Class 1.
TABLE 4. Description of the synthetic datasets.

| Scenario | Distribution for class 0 | Distribution for class 1 | Non-informative features |
|----------|--------------------------|--------------------------|--------------------------|
| D1       | Norm(μ = 5, σ = 5)       | Norm(μ = 10, σ = 10)     | Norm(μ = 0, σ = 1)       |
| D2       | Norm(μ = 5, σ = 5)       | Norm(μ = 10, σ = 5)      | Norm(μ = 0, σ = 1)       |
| D3       | Norm(μ = 10, σ = 5)      | Norm(μ = 10, σ = 10)     | Nil                      |

TABLE 5. Comparison of the proposed DRkNN method with the other classical kNN and its derivatives based on synthetic datasets.

| Metrics | Methods  | Senarios |
|---------|----------|----------|
|         |          | D1       | D2       | D3       |
| Accuracy| DRkNN    | 0.8653   | 0.8303   | 0.7903   |
|         | kNN      | 0.8271   | 0.8201   | 0.7223   |
|         | WkNN     | 0.8058   | 0.789    | 0.714    |
|         | RkNN     | 0.8625   | 0.8300   | 0.7968   |
| Kappa   | DRkNN    | 0.7795   | 0.86     | 0.5486   |
|         | kNN      | 0.7356   | 0.8196   | 0.7755   |
|         | WkNN     | 0.7295   | 0.6596   | 0.5850   |
|         | RkNN     | 0.6526   | 0.6383   | 0.4443   |
| BS      | DRkNN    | 0.8613   | 0.8151   | 0.7260   |
|         | kNN      | 0.8939   | 0.8653   | 0.8480   |
|         | WkNN     | 0.8004   | 0.8351   | 0.7011   |
|         | RkNN     | 0.7191   | 0.8254   | 0.5369   |

FIGURE 7. Simulation results of the DRkNN and other state-of-the-art methods.

V. CONCLUSION

Inspired by the weighted kNN, the proposed method classifies the given data on the basis of k-nearest neighbor via assigning weights to features obtained by using support vectors. For making the base model diverse, random feature subsets are used for the models. The final class of the test point is assigned by using majority vote in the predicted classes given by all base models. The results of the proposed ensemble are compared with base kNN, weighted kNN, random kNN, random forest and support vector machine on 14 datasets. Different values of k are tried for kNN base models. The performance is checked using accuracy, Cohen’s kappa and Brier score. The results clearly show that the proposed ensemble outperformed the other standard procedures on 8 out of 14 datasets.

The main intuition behind the efficiency of the suggested methods is that each base learner is constructed on a randomly selected bootstrap sample drawn from training observations with a random subset of features, which ensures diversity in the model and also ensures accuracy due to feature weighting. This technique draws bootstrap samples from training data and draws a hyperplane to find support vectors used as a weighting scheme which makes this method more efficient. Our simulation analysis also revealed that the proposed method is effective in that case of datasets with non-informative features. Moreover, the robustness of the proposed method was tested by using different values of k.

The proposed method could further be improved by model selection based on out-of-bag observations or sub-sampling. For additional randomness in the base models, the idea of randomly projecting the given feature space into lower dimensions could also be used. This might be very helpful in high dimensional settings. Model selection as given in [50] could also be incorporated for further improvements. Furthermore, considering the ideas of feature engineering and feature weighting [51], [52], [53], [54], [55], [56], in conjunction with the proposed method, might open further research avenues for improved classification and prediction. Incorporating the above mentioned ideas might increase execution time of the proposed method. This issue can be solved by using parallel computing for parallelizing Step 1 and Step 2 of the proposed algorithm.

REFERENCES

[1] G. Guo, H. Wang, D. Bell, Y. Bi, and K. Greer, “KNN model-based approach in classification,” in Proc. OTM Confederated Int. Conf. Move Meaningful Internet Syst. Cham, Switzerland: Springer, 2003, pp. 966–996.
[2] E. Fix and J. L. Hodges, “Discriminatory analysis. Nonparametric discrimination: Consistency properties,” *Int. Stat. Rev.*, vol. 57, no. 3, pp. 238–247, 1989.

[3] E. Fix and J. L. Hodges, “Discriminatory analysis-nonparametric discrimination: Small sample performance,” Defence Tech. Inf. Center, California Univ Berkeley, Berkeley, CA, USA, Tech. Rep. ADA300391, 1952.

[4] C. J. Veenman and M. J. T. Reinders, “The nearest subclass classifier: A compromise between the nearest mean and nearest neighbor classifier,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1417–1429, Sep. 2005.

[5] B. Dasarathy, *Nearest Neighbour Norms: Nearest Neighbour Pattern Classification Techniques*. USA: IEEE Computer Society Press, 1991.

[6] P. E. Hart, D. G. Stork, and R. O. Duda, *Pattern Classification*, Hoboken, NJ, USA: Wiley, 2000.

[7] B. S. Kim and S. B. Park, “A fast K nearest neighbor finding algorithm based on the ordered partition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 761–766, Nov. 1986.

[8] A. Djouadi and E. Boutkhatte, “A fast algorithm for the nearest-neighbor classifier,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 277–282, Mar. 1997.

[9] L. P. Peterson, “K-nearest neighbor,” Scholarpedia, vol. 4, no. 2, p. 1883, 2009.

[10] B. Zhang and S. N. Srihari, “Fast K-nearest neighbor classification using cluster-based trees,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 4, pp. 525–528, Apr. 2004.

[11] C. Domeniconi, J. Peng, and D. Gunopulos, “Locally adaptive metric nearest-neighbor classification,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1293, Sep. 2002.

[12] E. Marchiori, “Class conditional nearest neighbor for large margin instance selection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 2, pp. 364–370, Feb. 2010.

[13] G. Jiaping, L. Du, Z. Yuhong, and X. Taisong, “A new distance-weighted K-nearest neighbor classifier,” *J. Inf. Comput. Sci.*, vol. 9, no. 6, pp. 1429–1436, 2012.

[14] S. A. Dudani, “A distance-weighted K-nearest-neighbor rule,” *IEEE Trans. Syst. Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.

[15] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, “MKNN: Modified K-nearest neighbor,” in *Proc. World Cong. Eng. Comput. Sci.*, vol. 1, 2008, pp. 1–4.

[16] N. Bhata, “Survey of nearest neighbor techniques,” 2010, arXiv:1007.0085.

[17] H. A. Guvenir and A. Akkus, “Weighted K-nearest neighbor classification on feature projections,” in *Proc. 12th Int. Conf. Comput. Inf. Sci.*, 1997, pp. 1–8.

[18] S. C. Bagui, S. Bagui, K. Pal, and N. R. Pal, “Breast cancer detection using weighted K-nearest neighbor classifier,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 173–202, Sep. 2002.

[19] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Hoboken, NJ, USA: Wiley, 2000.

[20] J. Zavrel, “An empirical re-examination of weighted voting for K-NN,” in *Proc. 17th Int. Conf. Mach. Learning (ICML)*, 1998, pp. 135–142.

[21] S. D. Bay, “Combining nearest neighbor classifiers through multiple feature subsets,” in *Proc. ICML*, vol. 98, 1998, pp. 37–45.

[22] S. Li, E. J. Harner, and D. A. Adjeroh, “Random KNN feature selection—A fast and stable alternative to random forests,” *BMC Bioinf.*, vol. 12, no. 1, pp. 1–11, Dec. 2011.

[23] D. O. Sahin, O. E. Kural, S. Akleyelek, and E. Kılıç, “New results on permission based static analysis for Android malware,” in *Proc. 6th Int. Symp. Digit. Forensic Secur. (ISDFS)*, Mar. 2018, pp. 1–4.

[24] X.-D. Zeng, S. Chao, and F. Wong, “Optimization of bagging classifiers based on SBCB algorithm,” in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2010, pp. 262–267.

[25] M. Kuhn, “Caret: Classification and regression training,” R Package Version 6.0-93, 2021. [Online]. Available: https://github.com/topepo/caret/

[26] S. Klaus and H. Klaus, “KKKN: Weighted K-nearest neighbors,” R Package Version 1.3.1, 2016. [Online]. Available: https://github.com/KlausVigo/kkknn

[27] Z. Li, “KKNN: Random KNN classification and regression,” R Package Version 1.2-1, 2015. [Online]. Available: https://drizzlecran/kknn/

[28] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, “Kernlab—An S4 package for kernel methods in R,” *J. Stat. Softw.*, vol. 11, no. 9, pp. 1–20, 2004.

[29] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 2, no. 3, pp. 18–22, Nov. 2002.

[30] A. Ali, M. Hamraz, P. Kumam, D. M. Khan, U. Khalil, M. Sulaiman, and Z. Khan, “A K-nearest neighbours based ensemble via optimal model selection for regression,” *IEEE Access*, vol. 8, pp. 132095–132105, 2020.

[31] M. Hamraz, Z. Khan, D. M. Khan, N. Gul, A. Ali, and S. Aldahmani, “Gene selection in binary classification problems within functional genomics experiments via robust Fisher score,” *IEEE Access*, vol. 10, pp. 51682–51692, 2022.

[32] Z. Khan, M. Naeem, U. Khalil, D. M. Khan, S. Aldahmani, and M. Hamraz, “Feature selection for binary classification within functional genomics experiments via quartile range and clustering,” *IEEE Access*, vol. 7, pp. 78159–78169, 2019.

[33] R. Zhao, Y. Mu, L. Zou, and X. Wen, “A hybrid intrusion detection system based on feature selection and weighted stacking classifier,” *IEEE Access*, vol. 10, pp. 71414–71426, 2022.

[34] E.-S.-M. El-Kenawy, S. Mirjalili, F. Alassery, Y. D. Zhang, M. M. Eid, S. Y. El-Mashad, B. A. Aloyaydi, A. Ibrahim, and A. A. Abdelhamid, “Meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems,” *IEEE Access*, vol. 10, pp. 40536–40555, 2022.

[35] D. S. A. Elminnaam, S. A. Ibrahim, E. H. Houssein, and S. M. Elsayed, “An efficient chaotic gradient-based optimizer for feature selection,” *IEEE Access*, vol. 10, pp. 9271–9286, 2022.
N. Gul et al.: Weighted $k$-Nearest Neighbours Ensemble With Added Accuracy and Diversity

[56] N. Javaid, A. Almogren, M. Adil, M. U. Javed, and M. Zuair, “RFE based feature selection and KNNOR based data balancing for electricity theft detection using BiLSTM-LogitBoost stacking ensemble model,” IEEE Access, vol. 10, pp. 112948–112963, 2022.

[57] E. Esenogho, I. D. Mienye, T. G. Swart, K. Aruleba, and G. Obaido, “A neural network ensemble with feature engineering for improved credit card fraud detection,” IEEE Access, vol. 10, pp. 16400–16407, 2022.

NAZ GUL received the bachelor’s and master’s degrees in statistics from the University of Peshawar, Pakistan, in 2006 and 2009, respectively, and the M.S. degree in statistics from the University of Essex, U.K., in 2018. She is currently pursuing the Ph.D. degree with the Department of Statistics, Abdul Wali Khan University Mardan, Pakistan. She is also working as a Lecturer in statistics with Abdul Wali Khan University Mardan. Her research interests include machine learning, applied statistics, biostatistics, survival analysis, and computational statistics.

MUHAMMAD AAMIR (Senior Member, IEEE) received the bachelor’s degree (Hons.) in statistics from Quaid-i-Azam University, Islamabad, Pakistan, in 2005, the master’s degree in statistics from Quaid-i-Azam University, Islamabad, Pakistan, in 2013, and the Ph.D. degree in mathematics from Universiti Teknologi Malaysia, Malaysia, in 2018. In 2006, he joined the Department of Statistics, Islamia College University Peshawar, Pakistan, as a Lecturer. In 2009, he moved to Abdul Wali Khan University Mardan, Pakistan, where he was a Lecturer and became an Assistant Professor, in 2013. His current research interests include sampling, machine learning for mathematical problem diagnostics, econometrics, and time series. He also serves as an Active Reviewer for various international journals, including IEEE Access, PLOS ONE, Resources Policy, Energy, Pattern Recognition, Journal of Cluster Science, Helion, and many more.

SAEED ALDAHMANI received the bachelor’s degree in statistics from United Arab Emirates University, United Arab Emirates, in 2007, the master’s degree in statistics from Macquarie University, Australia, in 2010, the master’s degree in applied finance from Western Sydney University, Australia, in 2011, and the Ph.D. degree in statistics from the University of Essex, U.K., in 2017. He is currently an Assistant Professor of statistics with United Arab Emirates University. His research interests include graphical models, biostatistics, applied statistics in finance, and computational statistics.

ZARDAD KHAN received the master’s degree (Hons.) in statistics from the University of Peshawar, Pakistan, in 2008, the M.Phil. degree in statistics from Quaid-i-Azam University, Islamabad, Pakistan, in 2011, and the Ph.D. degree in statistics from the University of Essex, U.K., in 2015. He is currently an Assistant Professor in statistics with United Arab Emirates University, United Arab Emirates. He has also done a one year postdoctorate from the University of Essex. He has worked with Abdul Wali Khan University Mardan, Pakistan, as an Associate Professor in statistics. His research interests include machine learning, applied statistics, computational statistics, biostatistics, causal inference, and survival analysis.