IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 April 2024, accepted 19 April 2024, date of publication 23 April 2024, date of current version 6 May 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3392729

RESEARCH ARTICLE

An Optimal Random Projection k Nearest Neighbors Ensemble via Extended Neighborhood Rule for Binary Classification

AMJAD ALI[1,2], ZARDAD KHAN[2], DOST MUHAMMAD KHAN[1], AND SAEED ALDAHMANI[2]
[1]Department of Statistics, Abdul Wali Khan University Mardan, Mardan 23200, Pakistan
[2]Department of Statistics and Business Analytics, 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 This paper presents an ensemble method for binary classification, where each base model is based on an extended neighbourhood rule (ExNRule). The ExNRule identifies the neighbours of an unseen observation in a stepwise manner. This rule first selects the sample point closest to the experimental observation and then selects the second observation nearest to the previously chosen one. To find the required data points in the neighbourhood, this search is repeated up to k steps. The test sample point is predicted using majority voting in the class labels of the k chosen neighbours. In the proposed method, a large number of ExNRule based models are constructed on randomly projected bootstrap samples. The error rates of these models are computed using out-of-bag data points. The models are then ranked according to their out-of-bag errors, and a proportion of the most accurate models are selected. The final ensemble is constructed by combining the selected models. The proposed method is compared with other classical procedures on 15 benchmark datasets in terms of classification accuracy, Kohen's kappa and Brier score (BS) as performance metrics. Boxplots of the results are also constructed. The proposed ensemble is outperforming the existing methods on almost all the benchmark datasets. For further evaluation, the proposed method is compared with other kNN based classifiers on 3 datasets using different k values. Furthermore, the performance of the proposed method is also evaluated using simulated data under different scenarios.

INDEX TERMS Classification, kNN, extended neighborhood rule, ensemble learning, bootstrapping, random projection.

I. INTRODUCTION

Supervised learning tasks involve the use of functions that map inputs to outputs based on samples in pairs form, i.e., inputs and outputs. The k-nearest neighbours (kNN) technique is recognized as one of the top ten algorithms in machine learning used for supervised learning (classification and regression) [1], [2], [3], [4], [5], [6]. It predicts the response value of a new point by identifying a set of k observations in the neighbourhood, aiming to minimize the impact of outliers in the training data. This method is known for its simplicity, ease of understanding, robustness to outliers and effectiveness in the case of large training data [7], [8], [9]. Despite its simplicity, kNN produces comparable results and in some situations outperforms most of the complex methods. However, kNN faces challenges related to data, such as non-informative features and/or noise in the dataset.

The ensembles based on kNN, coupled with the implementation of randomization techniques, further enhance the prediction performance. Randomization is integrated by selecting random bootstrap samples from the observations and/or sub-samples from the available features to construct base models. This introduces diversity among the base models, reducing the likelihood of repeating the same error [10], [11], [12], [13]. The majority of the ensembles have been proposed using this approach. Examples of these techniques are bootstrap-aggregated kNN [14], random kNN [15], and ensemble of a subset of kNN [16]. The base kNN uses majority voting in the labels of the observations in the neighbourhood of the new sample point to predict

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

its class, while the final estimate is obtained by a second round majority vote of the results given by the base kNN classifiers. Although these ensembles construct diverse and randomised base models, however, in many situations, they ignore many of the informative features and select irrelevant features to construct the base models, which deprives the ensembles of their prediction accuracy. Considering this issue, it is required to develop techniques that produce randomised and diverse base classifiers while maintaining feature information. Random projection is one of the methods that reduce the dimension from p to p' without ignoring any of the features [17]. Many ensemble methods have been proposed that randomly project the original feature space to a lower dimension several times and then fit the base models. Further details on the applications of this method can be seen in [18].

This paper proposes an optimal random projection extended neighbourhood rule (ORPExNRule) ensemble. This method takes a large number of bootstrap samples from the training data that are randomly projected into lower dimensions. A kNN model is constructed using ExNRule on each of the random projections [19]. Out-of-bag errors are computed for all the base models. These models are then ranked in ascending order and a proportion of the most accurate models are combined into the final ensemble. For assessing the performance of the proposed ensemble, 15 benchmarks and 3 simulated datasets are used. The classical procedures, extended neighbourhood rule (ExNRule) ensemble, kNN, Weighted k nearest neighbours (WkNN), random k nearest neighbours (RkNN), random forest (RF), optimal trees ensemble (OTE) and support vector machine (SVM) are used for comparison purpose via performance metrics; accuracy, kappa and BS. For further assessment, boxplots have also been constructed to demonstrate the precision of the methods.

The remaining paper is organized as below: Related work is summarized in Section II. The proposed ORPExNRule ensemble is discussed in Section III. Experimental setup and results are given in Section IV. The conclusion based on the analyses given in the paper is presented in Section V.

II. RELATED WORK

The k nearest neighbours (kNN) classifier is a well-known machine learning algorithm known for its effectiveness and simplicity [20]. In the standard kNN method, equal weights are assigned to all instances in the neighbourhood of a new observation. A weighted k nearest neighbours (WkNN) method is also proposed [21], that assigns weights based on the distance of the instances in the neighbourhood from the test observation. WkNN presents promising results on several benchmark datasets as compared to the standard kNN. Despite of the effectiveness, WkNN is a global technique and time-consuming since it is based on all observations in the data. To address this issue, condensed nearest neighbour (CNN) procedures have been proposed [22], [23]. CNN removes similar sample points from the training data, but it depends on data order and may ignore points on the boundary. Another modification has been introduced known as the reduced nearest neighbour (RNN) rule [24]. It reduces the training data but suffers from computational complexity. For further improvement, a fast condensed nearest neighbour (FCNN) rule has also been proposed in [25]. It chooses data points close to the decision boundary, with minimal complexity. In [26], a model-based kNN method is proposed that fits a function on the training dataset to predict unseen observations, reducing the training data size while maintaining prediction performance. The clustered kNN (CkNN) method in [27], handles the problem of unevenly distributed training observation by identifying the closest instance in clusters.

It is well known that kNN is a time-consuming procedure because of the proportionality of the execution time to the number of observations and the number of features in the data [28], [29]. An attempt to address this problem led to the construction of k Nearest Neighbors on Feature Projections (kNNFP) [29]. This method sorts training observations as their projections on each feature dimension individually to obtain a fast classification of a test point as compared to the classical kNN procedure. The kNNFP method makes predictions for each feature, builds a kNN on the projections, and then classifies an unseen observation using majority voting on individual classifications by each feature. In kNNFP, all features are considered equally relevant, assuming all features have the same power in voting, which helps in reducing the impact of contrived features. However, if there are several irrelevant features in the data, voting alone may not be sufficient. To handle this issue, weighted kNNFP is constructed in [30], which explores the impact of incorporating feature weights during voting by multiplying the vote of each feature with its weight. This method stores all projections of training observations on linear features in memory as sorted values. Then, it calculates the vote of the feature and its distance from the new observation to identify the final classification. Model-based kNN, models the input data and uses the fitted model for data classification [7]. This method not only improves prediction performance but also demonstrates greater efficiency in terms of execution time. It also automatically selects an appropriate k value (number of observations in the neighbourhood). Other related studies can be seen in [31], [32], and [33].

There is a rich literature available on ensemble procedures where a large number of base kNN models via multiple feature subsets are constructed. To make the final prediction, the results from all these individual models are combined [34]. Another modification is a rank nearest neighbour (RNN), which tries to improve accuracy and reduce execution time by assigning ranks to training data for each category [35]. Similarly, random kNN is also proposed for classifying high-dimensional datasets. It ranks the features based on their discrimination power and relevancy to the response and the top-ranked features are used for the model construction [36]. Another method involves employing the term frequency-

inverse document frequency (TF-IDF) as a weighting scheme to give importance to permission features [37]. Similarly, the work done by [38] first ranks the permission attributes using information gain (IG) as a feature selection procedure and then the obtained ranked feature weights are assigned by implementing an ensemble of extra trees on the ranked features to produce feature subsets that represent the attribute properties. The subsets of features are used to enhance sets of instances with the top 5, 10, and 20 ranked features. Then, weights are computed using ensemble extra trees on the updated datasets to produce a permission feature model. Several ensemble techniques based on the kNN classifier have been proposed in the literature to enhance the performance and diversity of models. Bootstrap aggregation (bagging) is a fundamental technique used for ensemble methods construction [39]. Bagging generates a large number of base models and then finds the final estimate by combining the results of these models [40], [41]. It serves as a foundational method for various state-of-the-art ensemble procedures, wherein several bootstrap samples from the training observations are used to construct base models/classifiers. Each classifier is used to predict unseen data, and the final prediction is obtained by majority voting on the results provided by the base classifiers [39]. In studies, many ensemble techniques combine bagging with random subsets of features to construct base kNN models, for example, [16] and [42]. In [43], a kNN ensemble identifies observations in the neighbourhood based on their weighted distances about the targeted variable through support vectors. This method constructs a large number of base kNN learners using the proposed distance formula each on a bootstrap sample in conjunction with a random subset of features drawn from the total number of features. Furthermore, several studies have also been proposed that optimize the k value in base kNN classifiers to form the ensemble [44], [45]. These procedures try to fine-tune the value of k to achieve accurate predictions. Other ensemble procedures are explored in [46], [47], and [48].

This paper has also proposed an optimal random projection extended neighbourhood rule (ORPExNRule) for binary classification. The ORPExNRule constructs a large number of bass kNN models using ExNRule, each on a randomly projected bootstrap sample. Out-of-bag data is used to calculate their errors for model assessment. The models are then ranked according to their individual out-of-bag errors, and a proportion of the top-ranked models are chosen. The selected models are combined in the final ensemble. The proposed ensemble improves the performance in the following steps:

1) The base models are more randomised and diverse as they are constructed on random projections.
2) The base models ensure accuracy, as they are selected from a large pool of base models via out-of-bag errors.
3) The proposed method does not ignore any of the features as it randomly projects the total feature space into a lower dimension.

III. THE OPTIMAL RANDOM PROJECTION EXTENDED NEIGHBOURHOOD RULE FOR KNN ENSEMBLE

Consider a training dataset $\psi = (X, Y)_{n\times(p+1)}$, where $X_{n\times p}$ represents the feature space while Y indicates the corresponding binary target, i.e., $Y \in \{0, 1\}$. Draw B bootstrap samples from the training data ψ and randomly project to a lower dimension p', i.e., $\mathcal{S}^b_{n\times(p'+1)}$, where, $p' \leq p$ and $b = 1, 2, 3, \ldots, B$. Then apply ExNRule on each random projection to fit base kNN classifiers and compute errors on out-of-bag observations, i.e., E^b, where, $b = 1, 2, 3, \ldots, B$. Rank these models with respect to the out-of-bag errors in ascending order and select a proportion of the top-ranked B' learners. Based on the chosen models, compute the predictions of a new data point, i.e., $\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \ldots, \hat{Y}^{B'}$. The final estimated class of the new observation is the majority vote of the results given by the selected models.

A. MATHEMATICAL DESCRIPTION

Any of the distance formulas can be applied on all randomly projected bootstrap samples, i.e., $\mathcal{S}^b_{n\times(p'+1)}$, where, $b = 1, 2, 3, \ldots, B$, to identify a set of k closest data points to a new sample $X^0_{1\times p'}$ in a step-wise pattern. The distance formula used in this paper is described as follows:

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

where, $i = 1, 2, \ldots, k$.

In each base classifier, the distance formula in Equation 1 is used to identify a sequence of distances as given below:

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

This sequence shows that, $X^i_{1\times p'}$ is the nearest data point to $X^{i-1}_{1\times p'}$, where, $i = 1, 2, 3, \ldots, k$. The corresponding labels of $X^1_{1\times p'}$, $X^2_{1\times p'}$, $X^3_{1\times p'}$, $\ldots$, $X^k_{1\times p'}$ are noted, i.e., $y^1, y^2, y^3, \ldots, y^k$, respectively, and the predicted class of test point $X^0_{1\times p}$ for the b^{th} base model is $\hat{Y}^b$ = majority vote of (y^1, y^2, y^3, $\ldots$, y^k), where, $b = 1, 2, 3, \ldots, B$. The same rules will be used for out-of-bag data points to compute the model performance (out-of-bag error), i.e., E^b. According to this error, all models are ranked in ascending order and select the most accurate B' models to form the final ensemble. The final predicted class of the test data point $X^0_{1\times p}$ is computed from the results of the top-ranked models based on out-of-bag observations, i.e., $\hat{Y}$ = majority vote of ($\hat{Y}^1, \hat{Y}^2, \hat{Y}^3, \ldots, \hat{Y}^{B'}$).

B. ALGORITHM

The following steps are considered in the formulation of the proposed ORPExNRule.

1) Draw B bootstrap samples from the given training dataset and randomly project them into lower dimensions.

2) Build base models by applying ExNRule to each random projection formulated in Step 1.
3) Calculate the errors on the out-of-bag observations (i.e., E^b, where, $b = 1, 2, \ldots, B$) for each of the base models created in Step 2.
4) Sort the models in ascending order based on their out-of-bag performance.
5) Using individual out-of-bag errors to select the most accurate models for the final ensemble.
6) The final ensemble determined in Step 5 will be used for prediction.

The pseudo-code of the ORPExNRule ensemble is given in Algorithm 1 and Figure 1 illustrates the flowchart of the proposed method.

Algorithm 1 Pseudo-Code of the Proposed ORPExNRule Classifier

$\psi = (X, Y)_{n\times(p+1)} \leftarrow$ Training data;
$X_{n\times p} \leftarrow$ Feature space having p attributes;
$Y \leftarrow$ Binary response;
$p \leftarrow$ Number of features;
$p' \leftarrow$ Dimension of the projected bootstrap samples;
$n \leftarrow$ Number of data points;
$B \leftarrow$ Number of total bootstrap samples drawn from ψ;
$B' \leftarrow$ Number of the optimal models to construct the final ensemble;
$k \leftarrow$ Number of nearest neighbours identified by the ExNRule;
for $b \leftarrow 1 : B$ **do**
  $S^b_{n\times(p+1)} \leftarrow$ Bootstrap sample taken from ψ;
  $\mathcal{S}^b_{n\times(p'+1)} \leftarrow$ Randomly project the bootstrap sample into a lower dimension;
  $H^b(.) \leftarrow$ Construct a kNN model using the ExNRule on $\mathcal{S}^b_{n\times(p'+1)}$;
  $E^b \leftarrow$ Compute out-of-bag error of $H^b(.)$;
end for
$OH^b(.) \leftarrow$ Rank the constructed models in ascending order via out-of-bag errors.
The selected models will be merged in the final ensemble.

The proposed ensemble method distinguishes itself from other approaches by combining a unique neighbourhood rule selection with the integration of randomization through random projection. Additionally, our method ensures the accuracy of base models by employing model selection based on their performance using out-of-bag samples. None of the existing methods exhibit these characteristics, collectively contributing to an overall enhancement of kNN based ensembles.

IV. EXPERIMENTS AND RESULTS

A. BENCHMARK DATASETS

To compare the proposed optimal random projection extended neighbourhood rule (ORPExNRule) ensemble with the other classical methods, 15 benchmark problems have been used. These datasets are freely available from various repositories. Table 1 summarises the data set giving their names, number of observations n, number of features p, class-wise distribution, and sources.

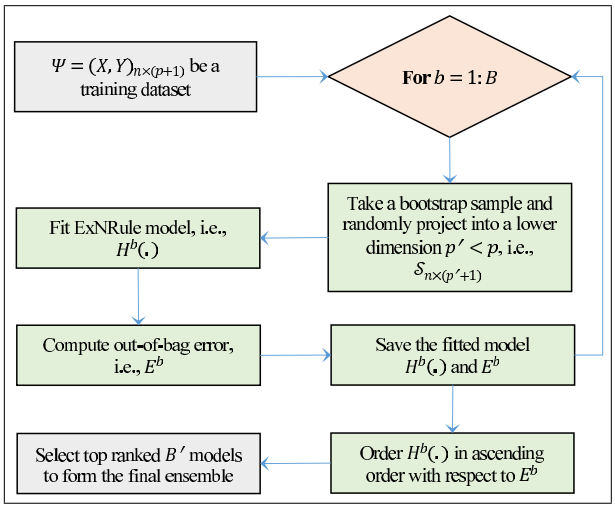

FIGURE 1. Flowchart of the proposed ORPExNRule classifier.

TABLE 1. A summary of the datasets used for model assessment.

Data	p	n	Class Distribution	Sources
KCBIN	86	145	(85, 60)	https://www.openml.org/d/1066
TSVMS	80	156	(54, 102)	https://www.openml.org/d/41976
JEDIT	8	369	(165, 204)	https://www.openml.org/d/1048
CVINE	9	52	(28, 24)	https://www.openml.org/d/815
WISCO	32	194	(104, 90)	https://www.openml.org/d/753
AR	29	36	(28, 8)	https://www.openml.org/d/1062
ILPD	10	583	(415, 167)	https://www.openml.org/d/1480
PLASR	12	182	(130, 52)	https://www.openml.org/d/1490
PHARY	10	193	(74, 119)	https://www.openml.org/d/738
SLEEP	7	62	(33, 29)	https://www.openml.org/d/739
EMONT	9	130	(66, 64)	https://www.openml.org/d/944
MC	39	161	(109, 52)	https://www.openml.org/d/1054
HEART	13	303	(99 , 204)	[49]
PLANR	13	315	(133, 182)	https://www.openml.org/d/915
GRDAM	8	155	(106, 49)	https://www.openml.org/d/1026

TABLE 2. Description of the simulated datasets.

Scenario ID	Distribution for class 0	Distribution for class 1
S_1	$N(\mu = 0, \sigma = 10)$	$N(\mu = 3, \sigma = 10)$
S_2	$N(\mu = 0, \sigma = 7)$	$N(\mu = 3, \sigma = 7)$
S_3	$N(\mu = 0, \sigma = 4)$	$N(\mu = 3, \sigma = 4)$

B. SIMULATED DATASETS

To evaluate the performance of the proposed method across diverse scenarios, 3 datasets with binary target variables are generated. Each scenario comprises 10 features and 100 observations. Among the 100 instances, 50 are generated from a normal distribution with fixed parameter values assigned to class 0. The remaining 50 observations, generated from the same distribution but with different parameter values, are designated for class 1. Table 2 provides an overview of the generated datasets, where the first column denotes the dataset ID, while the second and third columns

TABLE 3. Performance of the proposed ORPExNRule and other standard procedures on the given benchmark datasets.

Metrics	Methods	Datasets															Mean
		KCBIN	TSVMS	JEDIT	CVINE	WISCO	AR	ILPD	PLASR	PHARY	SLEEP	EMONT	MC	HEART	PLANR	GRDAM	
Accuracy	ORPExNRule	**0.772**	**0.724**	0.666	**0.798**	**0.595**	**0.845**	0.711	0.586	**0.812**	0.673	**0.741**	0.703	0.803	**0.718**	**0.785**	**0.729**
	ExNRule	0.768	0.716	**0.683**	0.769	0.574	0.836	**0.719**	**0.591**	0.768	0.671	0.733	0.716	**0.828**	0.709	0.778	0.724
	*k*NN	0.741	0.666	0.632	0.782	0.556	0.821	0.671	0.515	0.778	0.680	0.677	0.675	0.798	0.625	0.770	0.692
	W*k*NN	0.709	0.620	0.608	0.756	0.536	0.789	0.678	0.521	0.736	0.661	0.707	0.686	0.750	0.623	0.720	0.673
	R*k*NN	0.762	0.700	0.667	0.748	0.567	0.834	0.714	0.585	0.745	0.641	0.732	**0.717**	0.825	0.702	0.754	0.713
	RF	0.723	0.698	0.677	0.778	0.568	0.825	0.709	0.570	0.804	**0.684**	0.710	0.711	0.821	0.679	0.768	0.715
	OTE	0.711	0.682	0.663	0.744	0.562	0.793	0.700	0.559	0.795	0.650	0.702	0.694	0.809	0.653	0.747	0.698
	SVM	0.734	0.641	0.623	0.786	0.551	0.782	0.715	0.580	0.808	0.679	0.698	0.706	**0.828**	0.713	0.772	0.708
Kappa	ORPExNRule	**0.535**	**0.350**	0.320	**0.591**	**0.191**	**0.558**	0.145	0.081	**0.567**	0.354	**0.484**	0.197	0.602	0.031	**0.479**	**0.366**
	ExNRule	0.527	0.316	**0.360**	0.535	0.144	0.543	0.092	**0.090**	0.439	0.349	0.467	0.222	**0.651**	0.008	0.449	0.346
	*k*NN	0.465	0.283	0.254	0.559	0.114	0.516	0.182	0.001	0.519	0.363	0.355	0.199	0.591	-0.027	0.467	0.323
	W*k*NN	0.406	0.201	0.211	0.504	0.073	0.437	**0.215**	0.033	0.435	0.322	0.412	0.241	0.494	**0.063**	0.374	0.295
	R*k*NN	0.516	0.286	0.325	0.495	0.130	0.525	0.112	0.068	0.384	0.292	0.467	0.237	0.646	-0.010	0.357	0.322
	RF	0.441	0.299	0.347	0.553	0.132	0.503	0.200	0.081	0.559	**0.378**	0.422	0.254	0.638	0.003	0.447	0.350
	OTE	0.418	0.265	0.316	0.486	0.117	0.430	0.197	0.064	0.540	0.309	0.402	0.235	0.613	-0.013	0.410	0.319
	SVM	0.447	0.212	0.241	0.564	0.101	0.415	0.018	0.083	0.560	0.363	0.397	**0.280**	0.650	0.001	0.447	0.319
BS	ORPExNRule	**0.167**	**0.196**	0.217	**0.151**	**0.246**	**0.115**	0.182	0.250	**0.144**	**0.218**	0.180	0.200	0.138	**0.218**	**0.156**	**0.185**
	ExNRule	0.170	0.197	**0.205**	0.162	0.251	0.117	0.176	0.240	0.178	**0.218**	**0.179**	0.196	0.132	0.222	0.157	0.187
	*k*NN	0.186	0.236	0.266	**0.151**	0.308	0.132	0.229	0.324	0.176	0.243	0.227	0.237	0.167	0.278	0.186	0.223
	W*k*NN	0.291	0.380	0.392	0.244	0.464	0.211	0.322	0.479	0.264	0.339	0.293	0.314	0.250	0.377	0.280	0.327
	R*k*NN	0.171	0.198	0.214	0.172	0.252	0.124	**0.179**	**0.239**	0.184	0.229	0.180	**0.194**	0.147	0.225	0.164	0.191
	RF	0.180	0.197	0.211	0.157	0.254	0.149	0.182	0.247	0.149	0.227	0.186	0.196	**0.127**	0.238	0.166	0.191
	OTE	0.191	0.197	0.222	0.183	0.263	0.176	0.188	0.256	0.155	0.260	0.208	0.206	0.133	0.247	0.182	0.204
	SVM	0.193	0.216	0.226	0.166	0.250	0.122	0.201	0.244	0.158	0.238	0.222	0.198	0.128	**0.218**	0.165	0.196

TABLE 4. Performance of the proposed ORPExNRule and existing kNN based methods on 3 datasets for various k values.

Metrics	Methods	Datasets								
		KCBIN			TSVM			JEDIT		
		$k=3$	$k=5$	$k=7$	$k=3$	$k=5$	$k=7$	$k=3$	$k=5$	$k=7$
Accuracy	ORPExNRule	**0.772**	**0.771**	**0.766**	**0.724**	**0.718**	**0.713**	0.666	0.665	0.665
	ExNRule	0.768	0.759	0.745	0.716	0.709	0.694	**0.683**	**0.677**	**0.681**
	kNN	0.742	0.757	0.758	0.666	0.657	0.650	0.632	0.641	0.646
	WkNN	0.709	0.709	0.727	0.620	0.620	0.649	0.608	0.629	0.630
	RkNN	0.764	0.766	0.764	0.700	0.700	0.690	0.667	0.669	0.671
Kappa	ORPExNRule	**0.535**	**0.526**	0.491	**0.350**	**0.303**	**0.264**	0.320	0.319	0.320
	ExNRule	0.527	0.506	0.474	0.316	0.269	0.208	**0.360**	**0.347**	**0.357**
	kNN	0.465	0.498	0.501	0.283	0.245	0.205	0.254	0.273	0.287
	WkNN	0.406	0.406	0.441	0.201	0.201	0.242	0.211	0.252	0.253
	RkNN	0.516	0.525	**0.522**	0.286	0.265	0.219	0.325	0.331	0.336
BS	ORPExNRule	**0.167**	**0.165**	**0.164**	**0.196**	0.201	0.205	0.217	0.219	0.222
	ExNRule	0.170	0.171	0.173	0.197	0.200	0.204	**0.205**	**0.206**	**0.207**
	kNN	0.186	0.171	0.169	0.236	0.221	0.219	0.266	0.247	0.238
	WkNN	0.291	0.291	0.207	0.380	0.380	0.244	0.392	0.292	0.283
	RkNN	0.171	0.169	0.169	0.198	**0.199**	**0.202**	0.214	0.215	0.216

outline the feature distributions for class 0 and class 1, respectively. It is worthwhile to note that standard deviations decrease from S_1 to S_3, indicating reduced variability within each class. This variation reveals distinct patterns, important for knowing the impact on techniques and evaluating model robustness across various data distributions.

C. EXPERIMENTAL SETUP

The experimental setup for the 15 benchmark datasets given in Table 1 is given as follows. The datasets are divided into two non-overlapping sets, where 70% is randomly selected for the training part while the remaining is used as the testing part. The proposed ORPExNRule and other standard classifiers are fitted on the training part and the remaining 30% is used for their assessment. This criterion is repeated 500 times and the results are averaged in Tables 3 and 4.

The proposed ORPExNRule is used with $B = 500$ base models, each built on a randomly projected bootstrap sample. The dimension is reduced to $p' = \sqrt{p}$, where p is the total number of features in the data and $k = 3$ is used as the number of observations in the neighbourhood. Euclidean distance is used to determine neighbours in a stepwise pattern. After the models are constructed, out-of-bag errors are computed and 25% (i.e., $B' = 125$) top-ranked models are selected to form the final ensemble. These parameters serve as tuning elements for the proposed ensemble but are held fixed for analytical simplicity.

R packages `caret` [50], `kknn` [51] and `rknn` [52] are used to fit kNN, WkNN and RkNN, respectively. R package `randomForest` [53] is used to construct RF while `OTE` [54] for OTE. For support vector machine (SVM), the R package `kernlab` [55] is used. Different packages are used to tune different hyperparameters such as k in kNN, `ntree` and `mtry` in RF and OTE, etc.

D. DISCUSSION ON THE RESULTS

Table 3 presents results for 15 benchmark datasets, comparing the accuracy, kappa, and Brier score (BS) of the proposed ensemble with classical methods. The proposed ORPExNRule excels with the highest accuracy on 9 datasets, notably outperforming on KCBIN, TSVM, and others, showcasing adaptability across diverse datasets. ExNRule achieves maximum accuracy on 4 datasets (JEDIT, ILPD, PLASR, HEART). In contrast, kNN and WkNN do not outperform on any of the datasets, while RkNN shows promising results on MC data. RF outperforms on SLEEP, and OTE falls short in all cases. SVM exhibits promising results on HEART datasets. Overall, the proposed method's superior accuracy suggests its effectiveness across various data domains compared to existing procedures.

In terms of the kappa metric, the proposed ORPExNRule demonstrates promising results on 8 datasets, including KCBIN, TSVM, CVINE, etc. ExNRule outperforms the others on JEDIT, PLASR, and HEART datasets. In contrast, classical kNN and RkNN give poor performance across all datasets, while WkNN excels on ILPD and PLANR datasets.

TABLE 5. Performance of the proposed ORPExNRule and existing kNN based methods on simulated datasets.

Scenarios	Methods	Metrics		
		Accuracy	Kappa	BS
S_1	ORPExNRule	**0.743**	**0.489**	**0.182**
	ExNRule	0.710	0.435	0.200
	kNN	0.680	0.353	0.221
	WkNN	0.660	0.320	0.340
	RkNN	0.690	0.388	0.195
S_1	ORPExNRule	**0.860**	**0.718**	**0.125**
	ExNRule	0.837	0.674	0.158
	kNN	0.797	0.585	0.147
	WkNN	0.793	0.583	0.207
	RkNN	0.817	0.633	0.147
S_1	ORPExNRule	0.973	0.945	0.036
	ExNRule	**0.980**	**0.959**	0.073
	kNN	0.973	0.944	**0.024**
	WkNN	0.963	0.924	0.037
	RkNN	0.973	0.945	0.053

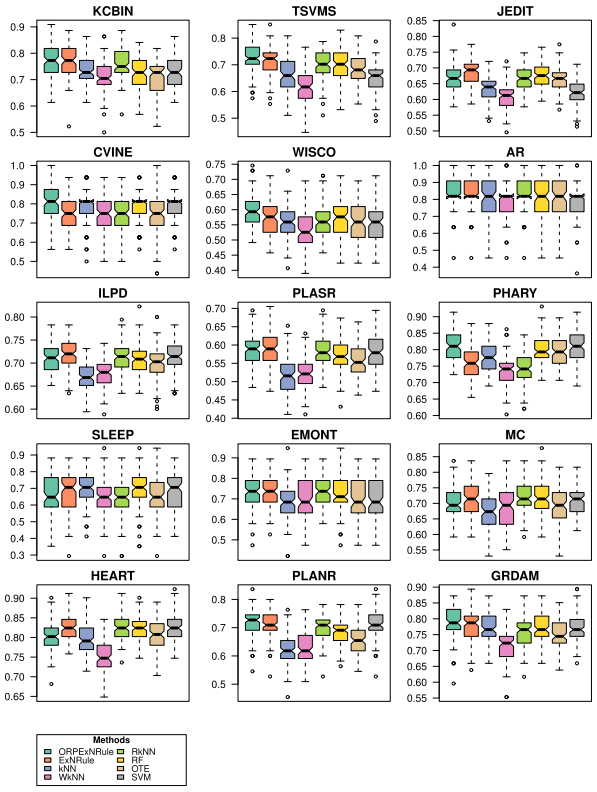

FIGURE 2. Accuracy given by the proposed ORPExNRule and other standard procedures on the given benchmark datasets.

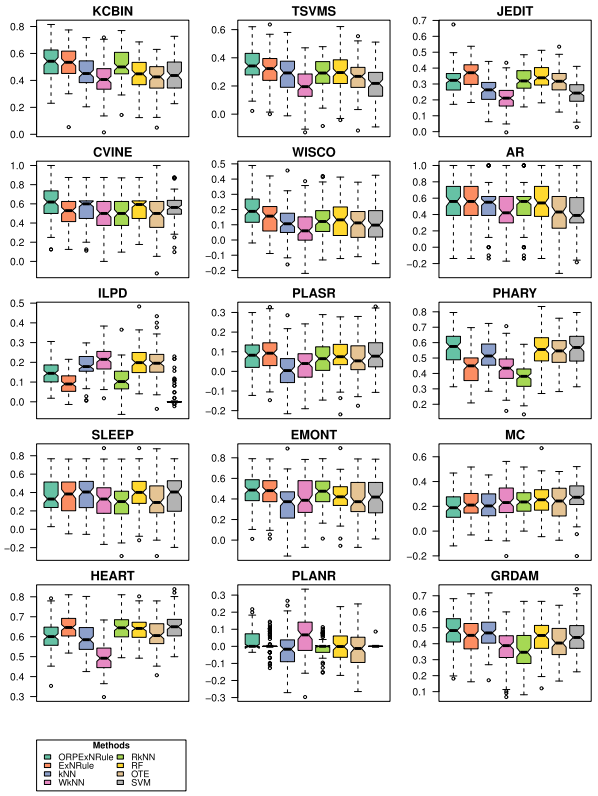

FIGURE 3. Kappa values given by the proposed ORPExNRule and other standard procedures on the given benchmark datasets.

RF attains optimal results on SLEEP data, while OTE falls short on all datasets. SVM provides high performance on the MC dataset in terms of the kappa metric.

In the case of BS, the proposed ensemble gives minimum value on 9 datasets as highlighted in the table. ExNRule outperforms on 3 datasets, i.e., JEDIT, SLEEP and EMONT, while kNN and WkNN do not perform well on any of the considered datasets. RkNN outperformed the others on 3 datasets, i.e., ILPD, PLSR and MC. RF demonstrates promising performance on the HEART dataset, while OTE performed poorly on all the datasets. SVM classifier also outperformed on 1 dataset, i.e., PLANR, as compared to the other methods. The proposed method excels by providing the minimum Brier score on the majority of the cases, showing that its estimated class probabilities align well with the actual outcomes. Moreover, boxplots of the classification accuracy, kappa and BS are also constructed in Figures 2, 3 and 4, respectively. These plots also indicate a precise prediction performance, consistency and stability of the proposed procedure.

The proposed ORPExNRule not only outperforms the rest of the methods on datasets individually but also keeps impressive performance on average. This indicates consistency across different data domains and highlights the reliability and steadiness of the method in different scenarios.

For further assessment, various k values are used to check the robustness of the proposed method against the

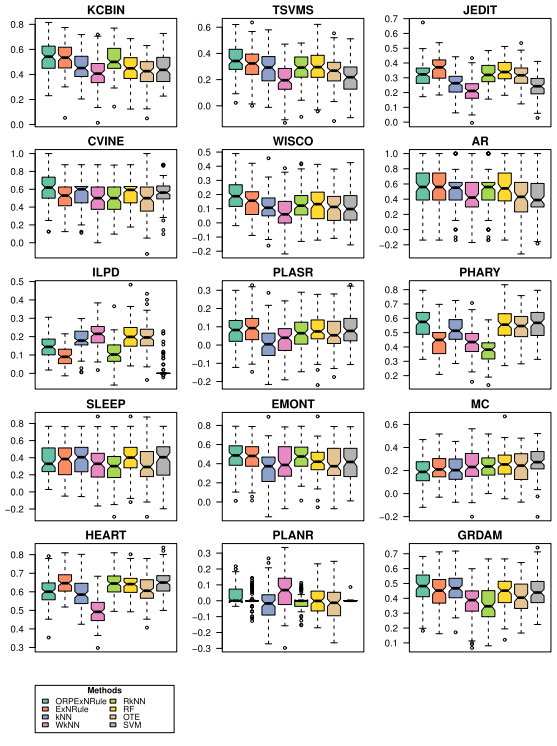

FIGURE 4. BS given by the proposed ORPExNRule and other standard procedures on the given benchmark datasets.

existing kNN procedures on 3 datasets and the results are given in Table 4. This table shows a comparative analysis

of the methods $k = 3, 5$ and 7, using three metrics, i.e., accuracy, kappa, and Brier Score (BS). The proposed ORPExNRule consistently outperforms other methods in terms of accuracy on KCBIN and TSVM datasets, while ExNRule provides competitive performance on JEDIT data. For the kappa metric, ORPExNRule achieves the highest values on KCBIN and TSVM datasets, while ExNRule performs well on JEDIT data. In terms of Brier Score, ORPExNRule consistently demonstrates lower scores on KCBIN and TSVM datasets, indicating superior probabilistic predictions. This comprehensive evaluation provides valuable insights into the methods' performance under different conditions, highlighting ORPExNRule's consistency and reliability across multiple metrics and datasets.

Table 5 presents results for the three different simulation scenarios, i.e., S_1, S_2, and S_3. The proposed ORPExNRule consistently outperforms standard kNN based techniques across all datasets, particularly excelling in scenarios with mixed-class compositions. This shows the robustness of the proposed method in handling intricate class distributions within complex datasets, showcasing its adaptability in situations where instances from different classes are closely intertwined. The observed results demonstrate the effectiveness of the proposed procedure in binary classification problems with mixed-class complexities.

V. CONCLUSION

In this paper, an ensemble classifier has been proposed. This method constructs a large number of base kNN models using extended neighbourhood rule, each constructed on a randomly projected bootstrap sample. The error rates of these models are computed using out-of-bag data. The models are then ranked according to their out-of-bag errors, and a proportion of the most accurate models are selected. The final ensemble is constructed by merging the selected models. Various benchmark and simulated datasets are used to assess the efficacy of the proposed method. Classification accuracy, Cohen's kappa, and Brier score (BS) are used as Performance measures. The new method is compared with standard methods such as ExNRule, kNN, WkNN, RkNN, RF, OTE, and SVM. The results have shown that the proposed ensembles outperformed all the existing methods. The findings have shown that the recommended ensembles delivered the best performance in terms of all the considered metrics. The results are also shown via boxplots to assess the consistency and precision of the techniques.

In comparison, the ExNRule method may unintentionally overlook important features and build models based on irrelevant ones due to its feature subset sampling approach. In contrast, our proposed method aims to overcome these limitations by using bootstrap sampling on all features. This key difference highlights the potential superiority of our approach in capturing the full range of available information in the data and creating more robust and accurate base models for the ensemble classifier. Additionally, our proposed method selects the most accurate models using out-of-bag errors, further improving the accuracy of the final ensemble.

However, it is imperative to acknowledge the potential limitations associated with the proposed ORPExNRule, in terms of training time when dealing with high dimensional or big datasets. The adverse effects of this problem could be mitigated by using parallel computing. Furthermore, using the idea of random projections might not be well suited for datasets with categorical features.

Recognizing these limitations offers valuable insights for refining the proposed ORPExNRule and guiding future research endeavours to enhance its practical applicability and robustness.

REFERENCES

[1] 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.
[2] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers—A tutorial," *ACM Comput. Surveys*, vol. 54, no. 6, pp. 1–25, Jul. 2022.
[3] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *Amer. Statistician*, vol. 46, no. 3, p. 175, Aug. 1992.
[4] T. Hastie and R. Tibshirani, *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Cham, Switzerland: Springer, 2009.
[5] A. S. Tarawneh, E. S. Alamri, N. N. Al-Saedi, M. Alauthman, and A. B. Hassanat, "CTELC: A constant-time ensemble learning classifier based on KNN for big data," *IEEE Access*, vol. 11, pp. 89791–89802, 2023.
[6] T.-J. Su, T.-S. Pan, Y.-L. Chang, S.-S. Lin, and M.-J. Hao, "A hybrid fuzzy and k-nearest neighbor approach for debris flow disaster prevention," *IEEE Access*, vol. 10, pp. 21787–21797, 2022.
[7] N. Bhatia and Vandana, "Survey of nearest neighbor techniques," 2010, *arXiv:1007.0085*.
[8] S. Kulkarni and M. V. Babu, "Introspection of various k-nearest neighbor techniques," *UACEE Int. J. Adv. Comput. Sci. Appl.*, vol. 3, pp. 89–92, Jun. 2013.
[9] S. Li, K. Zhang, Q. Chen, S. Wang, and S. Zhang, "Feature selection for high dimensional data using weighted k-nearest neighbors and genetic algorithm," *IEEE Access*, vol. 8, pp. 139512–139528, 2020.
[10] S. Bay, "Nearest neighbor classification from multiple feature subsets," *Intell. Data Anal.*, vol. 3, no. 3, pp. 191–209, Sep. 1999.
[11] Y. Bao, N. Ishii, and X. Du, "Combining multiple k-nearest neighbor classifiers using different distance functions," in *Proc. Int. Conf. Intell. Data Eng. Automated Learn.*, 2004, pp. 634–641.
[12] C. Domeniconi and B. Yan, "Nearest neighbor ensemble," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 1, 2004, pp. 228–231.
[13] N. García-Pedrajas and D. Ortiz-Boyer, "Boosting k-nearest neighbor classifier by means of input space projection," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10570–10582, Sep. 2009.
[14] B. M. Steele, "Exact bootstrap k-nearest neighbor learners," *Mach. Learn.*, vol. 74, no. 3, pp. 235–255, Mar. 2009.
[15] S. Li, E. J. Harner, and D. A. Adjeroh, "Random KNN," in *Proc. IEEE Int. Conf. Data Mining Workshop*, Dec. 2014, pp. 629–636.
[16] A. Gul, A. Perperoglou, Z. Khan, O. Mahmoud, M. Miftahuddin, W. Adler, and B. Lausen, "Ensemble of a subset of KNN classifiers," *Adv. Data Anal. Classification*, vol. 12, no. 4, pp. 827–840, 2018.
[17] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Applications to image and text data," in *Proc. 7th ACM SIGKDD Int. Conf. Knowl. discovery data mining*, Aug. 2001, pp. 245–250.
[18] T. I. Cannings and R. J. Samworth, "Random-projection ensemble classification," *J. Roy. Stat. Soc. Ser. B, Stat. Methodol.*, vol. 79, no. 4, pp. 959–1035, Sep. 2017.
[19] A. Ali, M. Hamraz, N. Gul, D. M. Khan, S. Aldahmani, and Z. Khan, "A k nearest neighbour ensemble via extended neighbourhood rule and feature subsets," *Pattern Recognit.*, vol. 142, Oct. 2023, Art. no. 109641.
[20] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[21] T. Bailey, "A note on distance-weighted k-nearest neighbor rules," *Trans. Systems, Man, Cybernetics*, vol. 8, no. 4, pp. 311–313, 1978.
[22] E. Alpaydin, "Voting over multiple condensed nearest neighbors," in *Lazy Learning*. Springer, 1997, pp. 115–132.
[23] K. Gowda and G. Krishna, "The condensed nearest neighbor rule using the concept of mutual nearest neighborhood," *IEEE Trans. Inf. Theory*, vols. IT-25, no. 4, pp. 488–490, Jul. 1979.
[24] G. Gates, "The reduced nearest neighbor rule," *IEEE Trans. Inf. Theory*, vols. IT-18, no. 3, pp. 431–433, May 1972.
[25] F. Angiulli, "Fast condensed nearest neighbor rule," in *Proc. 22nd Int. Conf. Mach. Learn. - ICML*, 2005, pp. 25–32.
[26] 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.*, Catania, Italy, Nov. 2003, pp. 986–996.
[27] Y. Zhou, Y. Li, and S. Xia, "An improved KNN text classification algorithm based on clustering," *J. Comput.*, vol. 4, no. 3, pp. 230–237, Mar. 2009.
[28] H. Parvin, H. Alizadeh, and B. Minaei-Bidgoli, "MKNN: Modified k-nearest neighbor," in *Proc. World Congr. Eng. Comput. Sci.*, vol. 1, 2008, pp. 1–12.
[29] T. Yavuz and H. A. Guvenir, "Application of k-nearest neighbor on feature projections classifier to text categorization," in *Proc. 13th Int. Symp. Comput. Inf. Sci.*, vol. 98, 1998, pp. 135–142.
[30] H. A. Guvenir and A. Akkus, "Weighted k nearest neighbor classification on feature projections," in *Proc. 12-th Int. Symp. Comput. Inf. Sci.*, 1997, pp. 1–12.
[31] Z. Wang, J. Na, and B. Zheng, "An improved kNN classifier for epilepsy diagnosis," *IEEE Access*, vol. 8, pp. 100022–100030, 2020.
[32] Y. Liang, C. Sun, J. Jiang, X. Liu, H. He, and Y. Xie, "An efficiency-improved clustering algorithm based on KNN under ultra-dense network," *IEEE Access*, vol. 8, pp. 43796–43805, 2020.
[33] F. Seghir, A. Drif, S. Selmani, and H. Cherifi, "Wrapper-based feature selection for medical diagnosis: The BTLBO-KNN algorithm," *IEEE Access*, vol. 11, pp. 61368–61389, 2023.
[34] S. D. Bay, "Combining nearest neighbor classifiers through multiple feature subsets," in *Proc. ICML*, vol. 98, 1998, pp. 37–45.
[35] S. C. Bagui, S. Bagui, K. Pal, and N. R. Pal, "Breast cancer detection using rank nearest neighbor classification rules," *Pattern Recognit.*, vol. 36, no. 1, pp. 25–34, Jan. 2003.
[36] 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.
[37] D. Ö. Sahin, O. E. Kural, S. Akleylek, and E. Kiliç, "New results on permission based static analysis for Android malware," in *Proc. 6th Int. Symp. Digit. Forensic Security (ISDFS)*, Mar. 2018, pp. 1–4.
[38] P. Feng, J. Ma, C. Sun, X. Xu, and Y. Ma, "A novel dynamic Android malware detection system with ensemble learning," *IEEE Access*, vol. 6, pp. 30996–31011, 2018.
[39] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.
[40] Z.-H. Zhou and Y. Yu, "Adapt bagging to nearest neighbor classifiers," *J. Comput. Sci. Technol.*, vol. 20, no. 1, pp. 48–54, Jan. 2005.
[41] Z.-H. Zhou and Y. Yu, "Ensembling local learners through multimodal perturbation," *IEEE Trans. Syst. Man Cybern., Part B (Cybernetics)*, vol. 35, no. 4, pp. 725–735, Aug. 2005.
[42] J. Gu, L. Jiao, F. Liu, S. Yang, R. Wang, P. Chen, Y. Cui, J. Xie, and Y. Zhang, "Random subspace based ensemble sparse representation," *Pattern Recognit.*, vol. 74, pp. 544–555, Feb. 2018.
[43] N. Gul, M. Aamir, S. Aldahmani, and Z. Khan, "A weighted k-Nearest neighbours ensemble with added accuracy and diversity," *IEEE Access*, vol. 10, pp. 125920–125929, 2022.
[44] S. Grabowski, "Voting over multiple K-NN classifiers," in *Modern Problems of Radio Engineering, Telecommunications and Computer Science*. Piscataway, NJ, USA: IEEE Press, 2002, pp. 223–225.
[45] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 5, pp. 1774–1785, May 2018.
[46] A.-J. Gallego, J. Calvo-Zaragoza, J. J. Valero-Mas, and J. R. Rico-Juan, "Clustering-based k-nearest neighbor classification for large-scale data with neural codes representation," *Pattern Recognit.*, vol. 74, pp. 531–543, Feb. 2018.
[47] Y. Zhang, G. Cao, B. Wang, and X. Li, "A novel ensemble method for k-nearest neighbor," *Pattern Recognit.*, vol. 85, pp. 13–25, Jan. 2019.
[48] P. Yuan, B. Wang, and Z. Mao, "Using multiple classifier behavior to develop a dynamic outlier ensemble," *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 2, pp. 501–513, Feb. 2021.
[49] R. Rahman, *Heart Attack Analysis & Prediction Dataset*. San Francisco, CA, USA: Kaggle, 2022.
[50] M. Kuhn. (2021). *CARET: Classification Regression Training R Package Version 6.0–90*. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v028i05
[51] K. Schliep and K. Hechenbichler. (2016). *KKNN: Weighted K-nearest Neighbors R Package Version 1.3.1*. [Online]. Available: https://CRAN.R-project.org/package=kknn
[52] S. Li. (2015). *RKNN: Random KNN Classification Regression R Package Version 1.2–1*. [Online]. Available: https://CRAN.R-project.org/package=rknn
[53] A. Liaw and M. Wiener, "Classification and regression by random forest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[54] Z. Khan, A. Gul, A. Perperoglou, O. Mahmoud, W. Adler, and B. Lausen. (2020). *OTE: Optim. Trees Ensembles for Regression, Classification Class Membership Probab. Estimation*. [Online]. Available: https://CRAN.R-project.org/package=OTE
[55] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, "Kernlab-AnS4Package for kernel methods INR," *J. Stat. Softw.*, vol. 11, no. 9, pp. 1–20, 2004.

AMJAD ALI received the bachelor's and master's degrees in statistics from the University of Peshawar, Pakistan, in 2013 and 2017, respectively, and the M.Phil. and Ph.D. degrees from the Department of Statistics, Abdul Wali Khan University Mardan, Pakistan, in 2020 and 2023, respectively. His research interests include linear models, machine learning, applied statistics, causal inference, and computational statistics.

ZARDAD KHAN received the master's degree 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 was an Associate Professor in statistics with Abdul Wali Khan University Mardan, Pakistan. He has also done a one year postdoctorate from the University of Essex, U.K. Currently, he is an Assistant Professor in statistics with United Arab Emirates University, United Arab Emirates. His research interests include machine learning, applied statistics, computational statistics, biostatistics, and survival analysis.

DOST MUHAMMAD KHAN received the bachelor's, master's, and Ph.D. degrees in statistics from the University of Peshawar, Pakistan, in 2000, 2003, and 2012, respectively. He is an Associate Professor in statistics with Abdul Wali Khan University Mardan, Pakistan. His research interests include robust statistics, applied statistics, survival analysis, statistical inference, and computational statistics.

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 Associate Professor in statistics with United Arab Emirates University. His research interests include graphical models, biostatistics, applied statistics in finance, and computational statistics.