Spectrum of Engineering Sciences
ISSN (e) 3007-3138 (p) 3007-312X Volume 3, Issue 12, 2025

ROBUST ESTIMATION METHODS FOR k-NEAREST NEIGHBOURS ENSEMBLE REGRESSION MODEL

Nosheen Faiz[*1], Kalsoom Bibi[2], Muhammad Hamraz[3], Soofia Iftikhar[4]

[*1,2,3]*Department of Statistics, Abdul Wali Khan University, Mardan, Pakistan, 23200.*
[4]*Department of Statistics, Shaheed Benazir Bhutto University, Peshawar, Pakistan*

[*1]nosheenfaiz@awkum.edu.pk

DOI: https://doi.org/10.5281/zenodo.18052842

Keywords
k-nearest neighbours, robust regression, bootstrap ensemble, median estimator, outliers.

Article History
Received: 27 October 2025
Accepted: 12 December 2025
Published: 25 December 2025

Corresponding Author: *
Nosheen Faiz

Abstract
The k-nearest neighbor (k-NN) regression method is widely used due to its simplicity and flexibility; however, its reliance on mean-based neighborhood aggregation makes it highly sensitive to outliers, noise, and skewed data distributions. To address these limitations, this study proposes four robust extensions of the k-NN regression framework: Median k-NN ensemble (MKNNE), Winsorized k-NN ensemble (WKNNE), Trimean k-NN ensemble (TriKNNE), and Trimmed Mean k-NN ensemble (TKNNE). By replacing the conventional sample mean with robust measures of central tendency, the proposed methods enhance robustness without sacrificing computational efficiency. Extensive experiments conducted on ten benchmark datasets with diverse statistical properties demonstrate that the proposed models consistently outperform standard k-NN, random k-NN (RKNN), and optimal k-NN ensemble (OKNNE) methods across multiple evaluation metrics, including R^2, MSE, MAE, and MAPE. The results show that robust neighborhood aggregation is an effective strategy for improving k-NN regression, especially in real-world scenarios involving noisy and heterogeneous data. This work provides a robust and extensible framework for neighborhood-based learning.

INTRODUCTION

Machine learning has become an indispensable tool for extracting meaningful patterns from complex and large-scale data. Since the seminal definition of ML by Samuel [1] as the field that enables computers to learn without being explicitly programmed, researchers have developed a wide range of techniques for data modeling, prediction, and analysis across diverse application domains. Broadly, ML methods are categorized into supervised learning (SL) and unsupervised learning (UL). Supervised learning relies on labeled data to learn predictive relationships, whereas unsupervised learning aims to uncover hidden structures and patterns from unlabeled data [2-4].

Regression analysis constitutes a fundamental component of supervised learning, providing statistical frameworks for modeling relationships between dependent and independent variables. It plays a critical role in prediction, forecasting, and, in certain contexts, causal inference [5, 6]. Classical regression techniques, such as linear regression, Ridge regression, and the least absolute shrinkage and selection operator (LASSO), are widely used due to their simplicity and interpretability. However, these methods are inherently sensitive to outliers, noise, and multicollinearity, which can significantly degrade their predictive performance, particularly in real-world datasets characterized by heterogeneity and measurement errors [7-9]. These limitations have motivated the development of robust, nonparametric learning methods capable of handling noisy and

irregular data. The k-nearest neighbor (k-NN) algorithm is a prominent example of such approaches. As a nonparametric method, k-NN estimates the target value of a query instance by aggregating the responses of its nearest neighbors in the feature space [10]. Several variants, including weighted k-NN (wk-NN) [11], random k-NN, and bootstrap-based k-NN ensembles, have been proposed to enhance predictive performance and reduce sensitivity to irrelevant features and sampling variability [12, 13].

Despite these advancements, traditional k-NN regression typically relies on the arithmetic mean for neighborhood aggregation, making it highly sensitive to outliers and extreme observations within the local neighborhood. To address this issue, robust statistical estimators such as the Median, Trimmed mean, Trimean, and Winsorized Mean have been explored. These estimators reduce the influence of anomalous values and are therefore well suited for constructing robust ensemble regression models.

Motivated by these observations, the present study proposes a robust ensemble k-NN regression framework that integrates robust estimators into the neighborhood aggregation process and introduces a novel robust method to further enhance prediction accuracy under noisy conditions. The proposed approaches are systematically evaluated against classical k-NN, random k-NN, and the optimal k-NN ensemble (OKNNE) using multiple performance metrics, including R^2, MSE, MAE, and MAPE, across a range of benchmark datasets.

1. Literature review

Regression analysis is a fundamental statistical technique that has long been employed to model and estimate relationships between input and output variables [14]. Classical linear regression, formally developed by Gauss and Legendre [15], is computationally efficient and interpretable; however, it is highly sensitive to outliers, which can exert a disproportionate influence on parameter estimates and significantly degrade model performance. To address the limitations of parametric regression models, nonparametric learning approaches have been extensively explored. Among these, the nearest neighbor (NN) method, originally developed in the 1950s [16, 17], later evolved into the k-nearest neighbor (k-NN) algorithm. k-NN is an intuitive, instance-based, nonparametric learning procedure that has demonstrated strong empirical performance and asymptotic properties comparable to Bayes classifiers [18-22]. Owing to its flexibility and minimal distributional assumptions, k-NN has been widely applied in both classification and regression tasks. Several extensions of the k-NN algorithm have been proposed to enhance both predictive performance and computational efficiency. Weighted k-NN assigns distance-based weights to neighboring observations, allowing closer instances to exert greater influence on the prediction [23, 24]. Bootstrap-enhanced nearest neighbor procedures increase robustness by artificially enlarging the training set through resampling techniques [25]. Instance-reduction methods, such as condensed nearest neighbor (C-NN) [26-28], reduced nearest neighbor (R-NN) [29], and class-conditional instance selection for regression (CCISR) [30], aim to reduce dataset size while preserving predictive accuracy, although this is often achieved at the expense of increased computational overhead. Model-based k-NN approaches further improve prediction accuracy by identifying and ignoring irrelevant regions of the feature space [31]. Additionally, for large-scale datasets, fast k-NN search frameworks have been developed to accelerate neighbor discovery while maintaining prediction quality, thereby addressing the computational challenges associated with traditional k-NN methods [32].

Ensemble learning has gained considerable attention due to its ability to improve predictive performance and stability by aggregating multiple weak learners [33]. Prominent ensemble techniques include boosting [34] and bagging [35], both of which rely on independent bootstrap samples to approximate model expectations and reduce variance [35-38]. Several refinements of the bagging framework, including exact bagging, have also been proposed to further enhance performance [13]. Random Forests extend the bagging paradigm by introducing feature-level randomness during node splitting, which reduces correlation among trees and helps mitigate overfitting [39]. More general

ensemble architectures manipulate data or models through subsampling, sub-spacing, sub-classing, or model variation to promote diversity among base learners [40].

A substantial body of research has incorporated k-NN as the base learner within ensemble frameworks. These approaches often employ random feature subsets for each bootstrap sample [12, 41, 42] and optimize the neighborhood size parameter k independently for each base model [43, 44]. Locally linear ensemble methods [45, 46] and hybrid k-NN ensembles combined with forward feature selection [47] have further improved predictive performance, particularly in high-dimensional settings [12]. Recent advances emphasize model selection and adaptive weighting strategies. Notably, the optimal k-NN ensemble (OKNNE) [48] integrates bootstrap sampling, random feature subspaces, and stepwise regression to reduce the influence of irrelevant predictors. Variable-k ensemble classifiers, which combine multiple k-NN models using weighted sum rules, have also demonstrated superior performance compared to traditional fixed-k approaches [49]. Additional developments include multimodal perturbation-based ensembles with heterogeneous distance measures and reduced random subspace bagging (RRSB), which increase classifier diversity without compromising accuracy [50], as well as k-NN–based outlier detection frameworks such as two-phase clustering methods [51]. To further alleviate the adverse effects of outliers on neighborhood-based prediction, recent k-NN ensemble models have progressively incorporated robust aggregation measures, including the Median, Trimmed Mean, Trimean, and Winsorized Mean. These robust estimators provide stable central tendency estimates, thereby enhancing the robustness and accuracy of ensemble k-regression models when applied to noisy and heterogeneous datasets.

Methodology

This section presents the methodological framework adopted in this study. The methodology begins with a brief overview of the baseline linear regression models, including simple and multiple linear regression, which serve as reference predictors for comparative analysis. Subsequently, methods for identifying and handling outliers are discussed, followed by a review of robust statistical estimators employed in this work, namely the Median, Trimmed Mean, Winsorized Mean, and Trimean.

The methodology then introduces neighborhood-based ensemble learning techniques, including the classical k-nearest neighbor (k-NN) regression, random k-NN (RKNN), and the optimal k-NN ensemble (OKNNE), along with the bagging strategy used to enhance model stability. Building on these foundations, the proposed robust k-NN regression frameworks are formulated by integrating robust estimators into the neighborhood aggregation step.

1.1. Linear Regression

One of the simplest tools of predictive modeling in supervised learning is the use of linear regression models. Simple linear regression model is a linear relationship between a scalar response variable $\mathbf{y}$ and one predictor variable "x". The model can be written as;

$$\hat{y} = \beta_0 + \beta_1 x,$$

Here, the intercept is denoted by β_0 and the slope coefficient by β_1. The parameters β_0 and β_1 are usually estimated by minimizing the sum of squared errors (differences between observed responses y_i and predicted responses $\hat{y}_i$).). The closed form solution of the least squares estimates are given by

$$\beta_1 = \frac{\sum_i (x_i - \bar{x})\,(y_i - \bar{y})}{\sum_i (x_i - \bar{x})^2}, \quad \beta_0 = \bar{y} - \beta_1 \bar{x},$$

where, $\bar{x}$ and $\bar{y}$ are the sample means of the predictor and response, respectively.

The simple model can be extended to the multiple linear regression when more than one input variable is used. In the multivariate case, the response $\hat{y}$ is modelled as

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p,$$

where, $x_1, \ldots, x_p$ are the predictors and their coefficients are respectively $\beta_1, \ldots, \beta_p$. In matrix notation, this can be written as $\hat{\mathbf{y}} = \mathbf{X}\boldsymbol{\beta}$, where $\mathbf{X}$ is the $n \times (p+1)$ design matrix (including a column of ones for the intercept) and $\boldsymbol{\beta}$ is the coefficient vector. The ordinary least squares

(OLS) solution for the value of $\boldsymbol{\beta}$ that minimizes the sum of squared residues is

$$\boldsymbol{\beta} = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{y},$$

Provided $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is invertible. This estimator has the best linear unbiased predictions under Gauss-Markov assumptions.

For linear regression, an interpretable model is obtained, and a baseline for comparison. However, it assumes that the relationship between predictors and response is linear and that the residuals are homoscedastic and normally distributed. The existence of outliers or non-normal error distribution may lead to a significant bias in the OLS estimates. The following section covers methods of detecting and reducing the effect of outliers.

1.2. Outliers in Regression

OLS is extremely sensitive to outliers and to high leverage points which may lead to biased coefficient estimates, to an increase in the value of the coefficient of determination (R^2), and to a distorted inference. Outliers typically show themselves in the form of large residual in diagnostic plots, or influential points. To reduce their effect, in this work robust measures are implemented in ensemble prediction framework based on nearest neighbour models.

1.3. Robust Estimation Techniques

Robust statistical estimators give alternative measures of central tendency or location which are less influenced by outliers than the arithmetic mean. In this study, some good estimators are considered:

1.3.1. Median

The median is defined as

- If n is odd

$$\text{Median} = \left[\frac{(n+1)}{2}\right]^{\text{th}} \text{term},$$

- If n is even

$$\text{Median} = \frac{\left[\left(\frac{n}{2}\right)\text{th term} + [1]\text{th}\right]}{2},$$

The median is very stable to outliers.

1.3.2. Winsorized Mean

Suppose $x_1 \le x_2 \le x_3 \le \cdots \le x_n$ denote the ordered sample of size n. For a winsorization proportion $\alpha \in [0.0.5]$, define

$$k = [n\alpha]$$

The α-winsorized sample is constructed by replacing the lowest k observation with x_{k+1} and the highest k observation with $x_{(n-k)}$:

$$x_i^W = \begin{cases} x_{k+1}, & i \le k \\ x_i, & k \le i \le n-k \\ x_{n-k}, & i > n-k \end{cases}$$

The Winsorized Mean is then defines as:

$\bar{x}_W = \frac{1}{n}\sum_{i=1}^{n} x_i^W$.

This estimator minimizes the effect of extreme values while including all observations, thus being a robust and efficient measure of central tendency.

1.3.3. Trimean

The Trimean (TM), also referred to as the Trimean of Tukey, is a measure of the location of a probability distribution which is the weighted mean of the median of a distribution as well as the two quartiles:

$$TM = \frac{Q_1 + 2Q_2 + Q_3}{4},$$

Here, Q_1 and Q_3 represent the upper and lower quartile, respectively and Q_2 is the median.

1.3.4. Trimmed mean

Suppose, $x_1 \le x_2 \le x_3 \le \cdots \le x_n$ denote the ordered sample of size n. For a trimming proportion $\alpha \in [0.0.5]$, define

$$g = [n\alpha]$$

The number of observation removed from each tail. The α-trimmed sample is obtained by discarding the lowest g order statistics. The α-trimmed mean is then defined as;

$$\bar{x}_T = \frac{1}{n-2g}\sum_{i=g+1}^{n-g} x_i$$

The estimator reduces the influence of extreme values by excluding a specified proportion of observation from both tails, making it a robust alternative to the classical arithmetic mean.

1.4. Nearest-Neighbour Approaches

Nearest-neighbor methods are non-parametric methods which are based on predicting the response to a query point using the responses to the training points that are nearest to the query point. This section includes discussions on the standard k-nearest neighbor model and

two types of ensemble-based classifiers, namely random k-NN and optimal k-NN ensemble.

1.4.1. k-Nearest Neighbour

The k-Nearest Neighbour algorithm makes predictions by taking the average of the results of the k training examples, which are closest (in the feature space) to the query point.For regression, given a query point x, one identifies the k observations $\{(x_{(1)}, y_{(1)}), \ldots, (x_{(k)}, y_{(k)})\}$ with smallest distance to x. The predicted response is then

$$\hat{y} = \frac{1}{k}\sum_{j=1}^{k} y_{(j)}.$$

Euclidean distance is the most commonly used distance but other distance measures (Manhattan, Mahalanobis, etc.) can be used depending on the problem.

The parameter k determines the level of locality of the predictions if k is small, it will result in capturing fine details (which is overfitting), and if large, it will smooth the function (which is under fitting). Euclidean distance is commonly used, but other metrics (Manhattan, Mahalanobis, etc.) can be applied depending on the problem. The parameter k determines the locality of the prediction: small k yields a model that can capture fine detail (at the risk of overfitting), whereas large k produces a smoother function (at the risk of underfitting). k -NN is a very easy method, and doesn't take extreme response value into consideration very well, because only the nearest points are used in making the prediction. Yet, it is vulnerable to the effect of relevant or irrelevant features or noisy features, and it struggles with the curse of dimensionality if the data in hand contains numerous numbers of dimensions. Finding an appropriate k is crucial, and this characteristically is done by cross validation.

1.4.2. Random k-NN Ensemble

The random k-NN method is an inspiration of random forests. Instead of using all features in every base k-NN model, random k- NN creates an ensemble of k-NN models, each of which is trained on a random subset of features and/or training examples. Each base learner chooses a random subsets of original features and applies the standard k-NN algorithm on it. When the predictions of many such random models are averaged, the resulting ensemble has reduced variance and decorrelation of the base models. This has often had the effect of improving generalization, and it also increases the sample robustness when some features are not very relevant or noisy. This is capable of enhancing both generalization and robustness, particularly in cases where certain features are so noisy or insignificant. The hyperparameters to be tuned are the number of features to be sampled and the overall number of base models in the ensemble.

1.4.3. Optimal k-NN Ensemble

The optimal k-NN ensemble (OkNNE) is a more sophisticated ensemble technique, which attempts to integrate a number of k-NN designs in the manner that reduces the effects of non-informative variables. In a single application, a stepwise regression analysis of the training data to identify the subset of features that is most relevant is performed along with each k-NN model in the ensemble.

In other words, given a bootstrap sample or a subset of data, stepwise feature selection is applied to select those features, which predict response best, and a k-NN model is constructed using these selected features only. Various such models are constructed, each possibly based on a dissimilar subset of features. The ensemble prediction of every model is then averaged to produce the final prediction of a query point. It has been demonstrated that this approach yields the correct result by disregarding irrelevant features and emphasizing informative ones. To conclude, OKNNE is an ensemble averaging approach used to perform feature selection to enhance predictive accuracy when there are noisy or redundant features.

1.5. Proposed Robust Ensemble Aggregation

Assume that the training data Ƒ= (X, Y) consists of the response variable Y and a matrix of features Ṕ with "n" rows (observations). It is required to predict the desired value "y" for X_0 assuming that X_0 is Ṕ-dimensional test observation. The number of bootstrap samples denoted by the Ḃ taken from the training dataset Ƒ = (X, Y). The samples are selected in

such a way that each one takes into account a random subset of features Ṕ of size ḿ. Each bootstrap sample selects the k closest neighbours for X_0 using a distance metric, such as Euclidean distance. As a result, each bootstrap sample has become a data matrix Ṙk (ḿ+1).

Let Ẁj(.) represent the value of the objective function that was used to evaluate the model, and let ĿLj(.) represent a linear regression model with j = 0, 1, 2,..., ḿ variables. Ṙk(ḿ+1) final model should be referred to as ĿLj(.). Let $\mathring{y}\dot{r}$, (ṙ = 1, 2, ..., B) be the value predicted by the regression model for X_0. In this manner, estimates of the same test point (X_0), which are B expected values, are obtained: $\mathring{y}_1, \mathring{y}_2, \mathring{y}_3, \ldots., \mathring{y}_B$.

To determine the overall estimate of the test point, the more robust estimation methods i.e., Median, Winsorized Mean, Trimean and Trimmed Mean of each of these estimated values is calculated. Based on these estimators, the following metric is proposed, in order to robustly estimate response value in the neighbourhood of a test point. It is defined as;

k-NN= $\frac{1}{B}\sum_{i=1}^{B} \hat{y}_i$,

Median:

For even number of observation

$$MKNNE = Size\ of\ \left\{\frac{B}{2}\right\} th\ observation.$$

For odd number of observation

$$MKNNE = Size\ of\ \left\{\frac{B+1}{2}\right\} th\ observation.$$

Winsorized Mean:

$$WKNNE = \frac{\hat{y}_{in} \ldots . \hat{y}_{n+1} + \hat{y}_{n+2} \ldots \hat{y}_{in}}{B},$$

Tri Mean:

$$TriKNNE = \frac{Q_1 + 2Q_2 + Q_3}{4},$$

Trimmed Mean:

$$TKNNE = \frac{\sum_{i=p+1}^{B+p} \hat{y}_i}{B-2P},$$

The algorithm for the proposed Robust Estimation Methods For k-Nearest Neighbours Ensemble Model is as follows;

1. Take B bootstrap samples from the training data and, for each sample, randomly select a subset of d < p features.
2. For each bootstrap sample, apply k-NN to identify the k nearest neighbour of the test observation x'.
3. For each bootstrap sample, form a data matrix $M_{k\times(d+1)}$ consisting of the k neighbour observation (with their responses) and the corresponding d features.
4. For each bootstrap sample $b = 1,2,,\ldots,B,$ compute a robust estimate of the response for x' from its neighbourhood:

$$\hat{y}^{(b)}(x') = T\ (y_1^b\ ,\ y_2^b\ ,\ldots\ldots\ldots\ ,\ y_2^b\ ),$$

where, T(.) is robust location measure(e.g., Median, Trimmed Mean or Winsorized Mean) applied to the responses of the k neighbours.

5. The final prediction for x' is obtained by taking the arithmetic mean or average of the B robust neighborhood prediction

$$\hat{y}(x') = \frac{1}{B}\sum_{b=1}^{B} \hat{y}^{(b)}(x').$$

Pseudo code of the proposed method is given in Algorithm 1 and flow chart in Figure 1.

Algorithm 1. Pseudo Code for the Proposed Method

Input:

p = number of features
B = number of bootstrap learners
k = number of neighbours
T(.) = robust location functional
x' = test point

For b = 1 to B do

- Draw bootstrap sample $D^{(b)}$
- Build k-NN model using all p features
- Compute Euclidean distances between x' and all points
- Identify k nearest neighbours
- Form neighbourhood matrix $M^{(b)}$ of dimension k × (p + 1)
- Extract neighbour responses $\{y_1^{(b)}, y_2^{(b)}, \ldots, y_k^{(b)}\}$
- Compute robust prediction:

$$\hat{y}_{(x')}^{(b)} = T(y(b))$$

- Store $\hat{y}_{(x')}^{(b)} = T(y(b))$

End For

Pool all B estimates to get final result i.e., $\hat{y}(x') = \frac{1}{B}\sum_{b=1}^{B} \hat{y}^{(b)}(x')$.

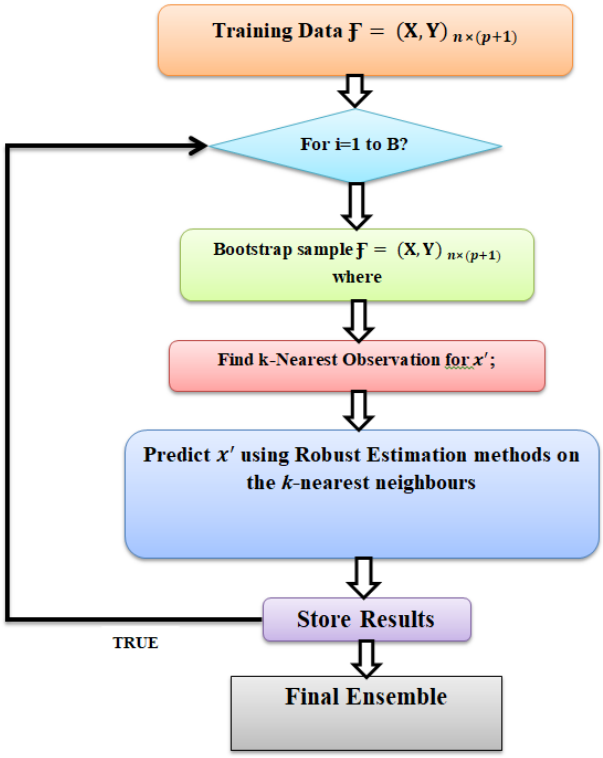

Figure 1: Flow chart of the proposed method.

1.6. Benchmark datasets

A total of 10 datasets are used in order to compare the proposed method with the other state-of-the art methods. These data sets are in a variety of publicly available sources. The datasets have been briefly described in Table 1. The table shows the number of observations, number of variables and source.

Table 1: List of benchmark datasets.

Datasets	n	P	Sources of the datasets
Concrete (Con)	103	10	http://archive.ics.uci.edu/ml/datasets/concrete+slump+test
Boston (Bos)	506	14	https://www.openml.org/search?type=data&sort=runs&id=531&status=active
RealE state (R_est)	414	7	https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set
Andro (Andro)	49	36	https://www.openml.org/search?type=data&sort=runs&id=41392&status=active
Stock (Stock)	950	10	https://www.openml.org/search?type=data&sort=runs&id=223%2F&status=active
ATP (ATP)	337	417	https://www.openml.org/search?type=data&sort=runs&id=41475&status=active
Yacht (Yat)	308	7	https://archive.ics.uci.edu/ml/datasets/yacht+hydrodynamics

Datasets	n	P	Sources of the datasets
Wine (Wine)	4898	12	https://archive.ics.uci.edu/ml/datasets/Wine+Quality
CPU (CPU)	209	9	https://www.openml.org/search?type=data&sort=runs&id=561&status=active
Chatfield (Chat)	235	13	https://www.openml.org/search?type=data&sort=runs&id=695&status=active

2. Experimental Setup

Experiments on all datasets were carried out based on a unified evaluation procedure. Each dataset had 70% training and 30% testing and this was kept the same for all of the experiments in order to be comparable. For the proposed method, hyper-parameter values were chosen fixed as it is simple. A total of B = 100 bootstrap samples were created from the training data. In contrast to the ensemble variants that are based on the subsampling of features, the proposed approach makes use of the entire set of features in each bootstrap model. For each of the bootstrap samples, a k-NN model was built where the neighbourhood size k = 0.1 x n where n is number of observations in the training set. For each query point, the k nearest neighbours were found using the Euclidean distance. Instead of using the traditional arithmetic mean of the neighbour responses the proposed method computes the prediction within each bootstrap model using robust statistical estimators (which reduce the sensitivity to noise and extreme values). In more detail, the prediction for each bootstrap model is then derived using one of the following robust neighbourhood measures i.e. Media, Trimean, Trimmed Mean and and Winsorized Mean of the neighbour responses. These robust estimators is used in the neighbourhood level regression and have the benefit of increased stability against local outliers or heavy tailed distributions. After all 100 bootstrap level predictions are collected, then a final ensemble prediction for each test observation is computed by the simple arithmetic mean:

$$\hat{y} = \frac{1}{B}\sum_{I=1}^{B} \hat{y}_{(b)}.$$

where, $\hat{y}_{(b)}$ is the robust neighbourhood prediction of the b-th bootstrap model. Using an ensemble level simple average helps to avoid over shrinkage and retains the traditional ensemble interpretation together with neighbourhood stage robustness.

3. Results and Discussion

Tables 2-5 summarize the predictive performance of 10 benchmark datasets. Overall, there are significant improvements in prediction accuracy in the robust estimators over the classical mean-based k-NN variants. The following sections give a detailed discussion for each performance metric.

Table 2: R^2 for all datasets using k-NN, RKNN, OKNNE, MKNNE, WKNNE, TriKNNE and TKNNE.

Datasets											
Metrics	Methods	Con	Bost	R_est	Andro	Stock	ATP	Yat	Wine	CPU	Chat
	k-NN	0.403	0.267	0.472	0.383	0.980	0.860	0.099	0.011	0.662	0.670
	RKNN	0.466	0.679	0.589	0.650	0.985	0.876	0.315	**0.431**	0.796	0.811
	OKNNE	0.470	0.685	0.642	0.511	0.977	0.193	0.472	0.254	0.814	0.820
R^2	MKNNE	0.770	0.761	0.538	0.367	0.976	0.900	0.941	0.027	0.841	0.836
	WKNNE	0.775	0.767	0.560	0.364	0.977	0.827	0.943	0.067	**0.865**	0.838

	TriKNNE	**0.781**	0.767	**0.651**	**0.925**	**0.987**	**0.980**	**0.943**	0.071	0.848	0.837
	TKNNE	0.778	**0.774**	0.547	0.377	0.977	0.802	0.939	0.066	0.859	**0.839**

Table 3: MSE of all datasets using k-NN, RKNN, OKNNE, MKNNE, WKNNE, TriKNNE and TKNNE.

Datasets											
Metrics	Methods	Con	Bost	R_est	Andro	Stock	ATP	Yat	Wine	CPU	Chat
	k-NN	34.13	67.64	95.08	1.23	0.868	4179.02	196.30	0.595	8989.16	15.827
	RKNN	31.82	**0.26**	75.24	0.69	0.653	**51.22**	157.00	**0.447**	5258.56	9.285
	OKNNE	31.52	27.60	66.42	**0.14**	**0.006**	229.32	118.79	0.589	4536.45	**7.659**
MSE	MKNNE	13.55	19.67	87.42	1.35	1.005	2983.96	14.402	0.762	53.014	325.40
	WKNNE	13.33	19.30	81.02	1.36	0.972	5044.95	**13.317**	0.736	**45.103**	321.64
	TriKNNE	**12.46**	19.18	**64.40**	0.19	0.551	599.99	13.510	0.725	50.662	323.68
	TKNNE	13.14	18.64	85.41	1.34	0.982	5950.98	14.505	0.730	47.116	319.55

Table 4: MAE of all datasets using k-NN, RKNN, OKNNE, MKNNE, WKNNE, TriKNNE and TKNNE.

Datasets											
Metrics	Methods	Con	Bost	R_est	Andro	Stock	ATP	Yat	Wine	CPU	Chat
	k-NN	4.27	4.77	6.08	0.793	0.601	28.581	7.943	0.531	39.006	16.473
	RKNN	4.27	3.47	6.09	0.604	0.607	32.911	6.933	**0.445**	33.411	14.116
	OKNNE	4.24	3.46	5.47	0.667	0.745	53.569	8.115	0.592	31.718	13.781
MAE	MKNNE	2.89	3.07	6.14	0.885	0.713	24.922	1.630	0.640	5.106	12.833
	WKNNE	2.91	3.10	6.07	0.913	0.714	52.719	1.602	0.652	**4.823**	**12.682**
	TriKNNE	**2.80**	**3.05**	**5.33**	**0.289**	**0.551**	**10.225**	**1.580**	0.652	5.033	12.791
	TKNNE	2.88	3.04	6.14	0.895	0.718	57.122	1.628	0.651	4.956	12.732

Table 5: MAPE of all datasets using k-NN, RKNN, OKNNE, MKNNE, WKNNE, TriKNNE and TKNNE.

Datasets											
Metrics	Methods	Con	Bost	R_est	Andro	Stock	ATP	Yat	Wine	CPU	Chat
	k-NN	12.60	22.49	17.61	15.204	1.301	5.785	276.400	9.455	47.269	170.4
	RKNN	13.21	17.62	19.12	11.403	1.320	7.005	77.871	**8.035**	49.333	139.9
	OKNNE	12.93	17.17	16.34	11.664	1.615	11.464	1271.660	10.565	**46.245**	127.8
MAPE	MKNNE	8.09	15.21	18.43	18.944	1.542	5.315	28.339	11.293	150.23	100.9
	WKNNE	8.41	15.42	18.90	19.965	1.546	11.842	28.347	11.543	190.1	**96.11**

TriKNNE	**8.06**	**15.13**	**15.41**	**5.730**	**1.194**	**2.297**	**27.286**	11.535	171.2	98.11
TKNNE	8.21	15.18	18.80	19.121	1.554	12.742	28.992	11.492	200.1	100.9

3.1. R^2 Analysis

Table 2 indicates that TriKNNE consistently achieves the highest R^2 across most datasets, reflecting superior goodness of fit. For Yacht dataset R^2 increases drastically from k-NN = 0.099 and RKNN = 0.315 to TriKNNE = 0.943, showing the great advantage of trimean based smoothing in the extreme outlier environment. In the case of Andro dataset, TriKNNE achieves an extremely high R^2, (0.925) indicating high stability in spite of the noise and irregular distributions in the dataset. Similarly for CPU, Boston, and Stock, all robust estimators out-perform traditional methods confirming that the trimmed, winsorized or median based estimators effectively reduce variance inflation from outliers. These improvements support the fact that adding robust measures to the neighbourhood aggregation step provides much better model generalization.

3.2. MSE Analysis

Results in Table 3 show that the TriKNNE estimator yields the lowest MSE across nearly all datasets, further supporting its strong R^2 performance. For Concrete dataset, MSE reduces from 34.13 (*k*-NN) to 12.46 (TriKNNE). Yacht dataset shows performance jump from 196.30 (*k*-NN) to 13.51 (TriKNNE) showing reduction in error. Similarly, MSE decreases from 4179.02 (k-NN) to 599.99 (TriKNNE), again highlighting robustness to heavy-tailed distributions for ATP dataset. This shows that the proposed estimators consistently reduce squared errors because trimming or winsorising effectively suppresses the influence of extreme deviations within the neighbourhood.

3.3. MAE Analysis

As shown in Table 4 (MAE), robust estimators consistently outperform the classical k-NN approach. In particular, TriKNNE yields the lowest MAE for the Yacht dataset (1.580 compared to 7.943 for *k*-NN). Similar improvements are observed for the Stock and CPU datasets, where robust methods maintain lower absolute deviations in the presence of noise. The Wine dataset demonstrates enhanced stability, although MAPE values vary. Among all methods, TriKNNE exhibits the most stable and consistent performance, closely followed by WKNNE and TKNNE, while MKNNE remains competitive but slightly affected by symmetric trimming.

3.4. MAPE Analysis

MAPE is often challenging to interpret due to its sensitivity to small denominators; nevertheless, Table 5 demonstrates substantial improvements achieved by robust methods. For the Yacht dataset, TriKNNE markedly outperforms all competing algorithms, attaining a MAPE of 2.297 compared with 276.400 for k-NN. Similarly, for the Andro dataset, MAPE is reduced from 15.204 (k-NN) to 5.730 (TriKNNE). For the Stock dataset, TriKNNE again achieves the lowest MAPE (1.194), indicating strong stability in percentage-based error measures. Although datasets such as CPU and Chatfield inherently exhibit higher MAPE values due to scale effects, the robust estimators perform at least comparably to, and in some cases better than, k-NN and its variants.

3.5. Comparative Discussion

Overall results across R^2, MSE, MAE, and MAPE demonstrate that robust k-NN estimators consistently outperform the standard mean-based k-NN across most datasets, confirming the sensitivity of classical neighborhood averaging to outliers. Among the proposed methods, TriKNNE shows the most stable and accurate performance, as its joint weighting of quartiles and the median provides robustness and efficiency under skewed and noisy distributions. WKNNE and TKNNE also yield meaningful improvements, particularly for heavy-tailed data, while MKNNE remains competitive when strong trimming is appropriate but may be less effective otherwise. Robust estimation proves especially beneficial

for high-noise datasets such as ATP and Yacht, leading to consistent reductions across all error measures. These findings confirm that replacing the conventional mean with robust statistical estimators substantially enhances k-NN regression performance, offering a reliable framework for noisy and non-Gaussian data and motivating future research in robust and hybrid k-NN models.

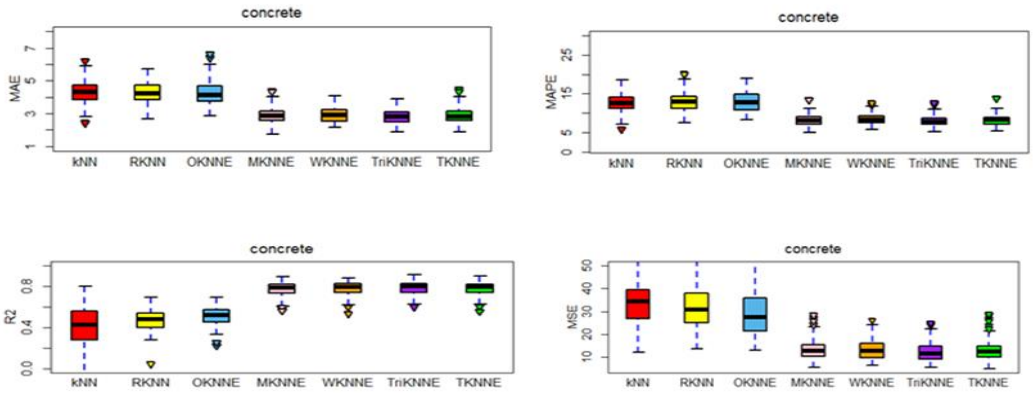

Figure 2 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Concrete dataset.

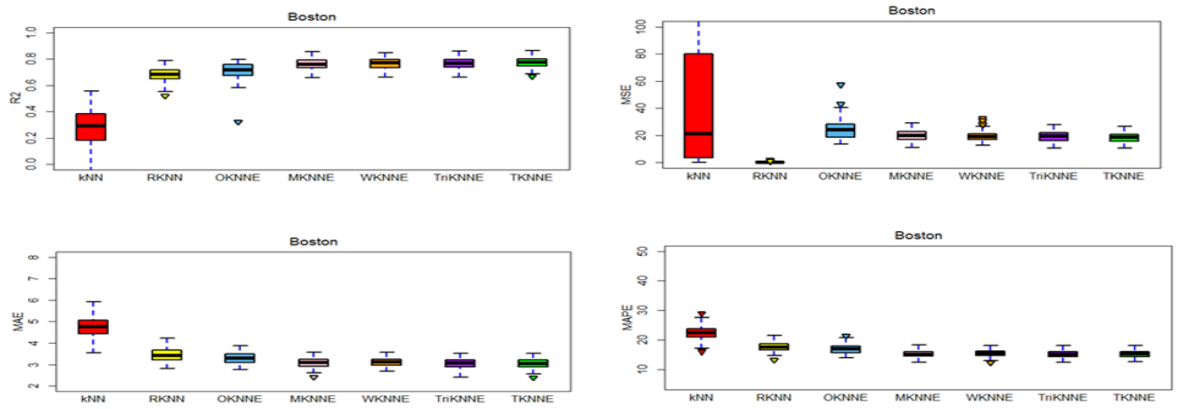

Figure 3 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Boston dataset.

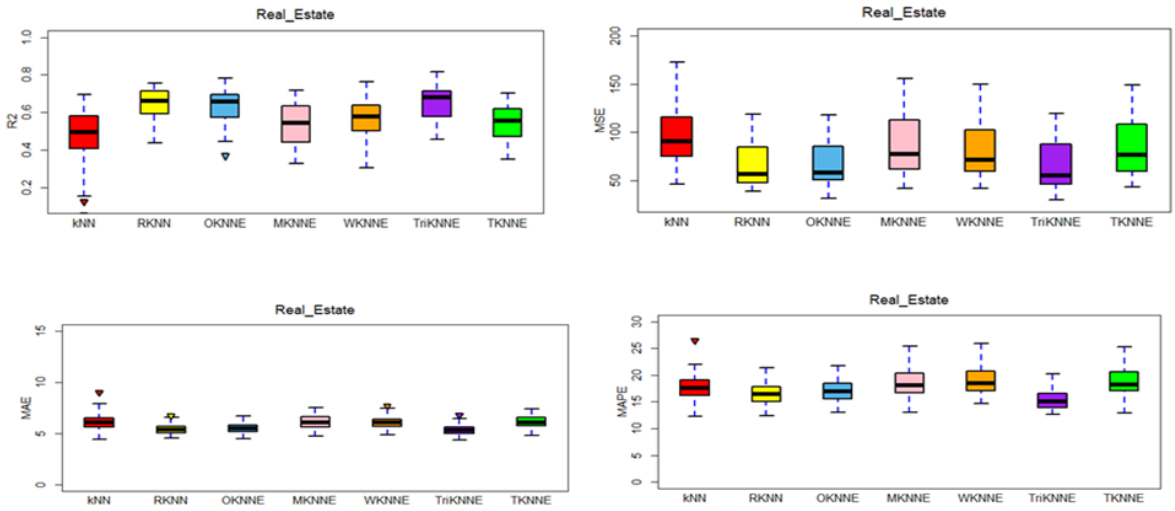

Figure 4 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Real_Estate dataset.

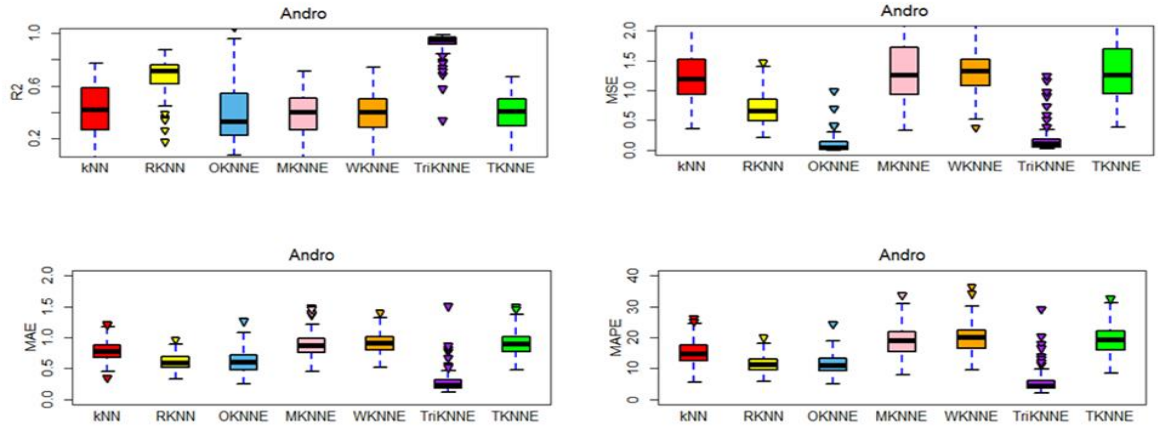

Figure 5 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Andro dataset.

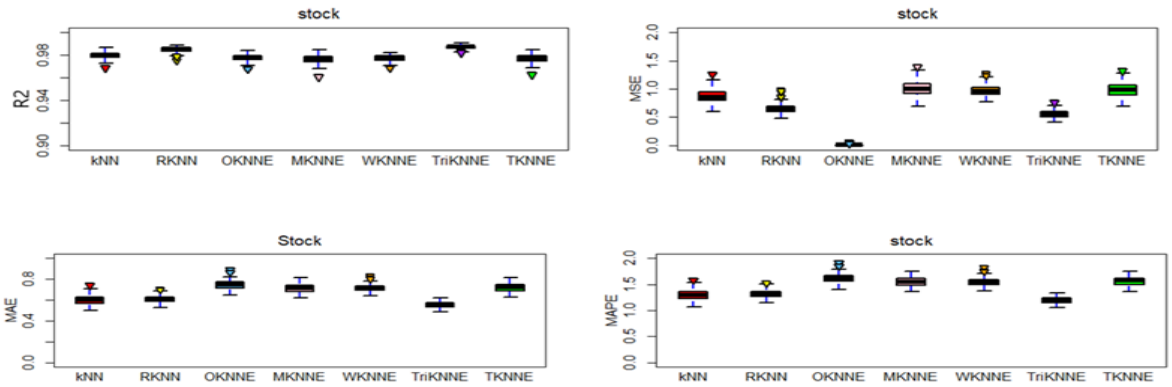

Figure 6 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Stock dataset.

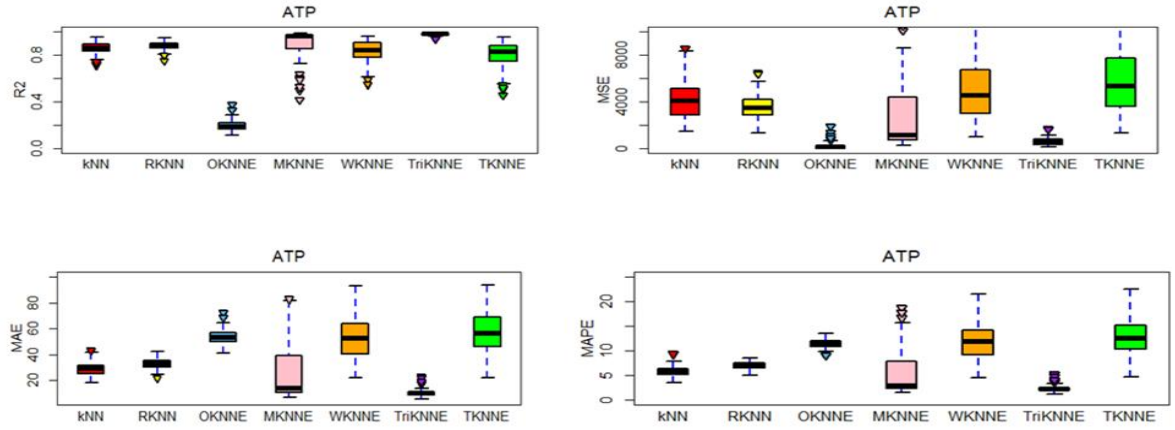

Figure 7 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the ATP dataset.

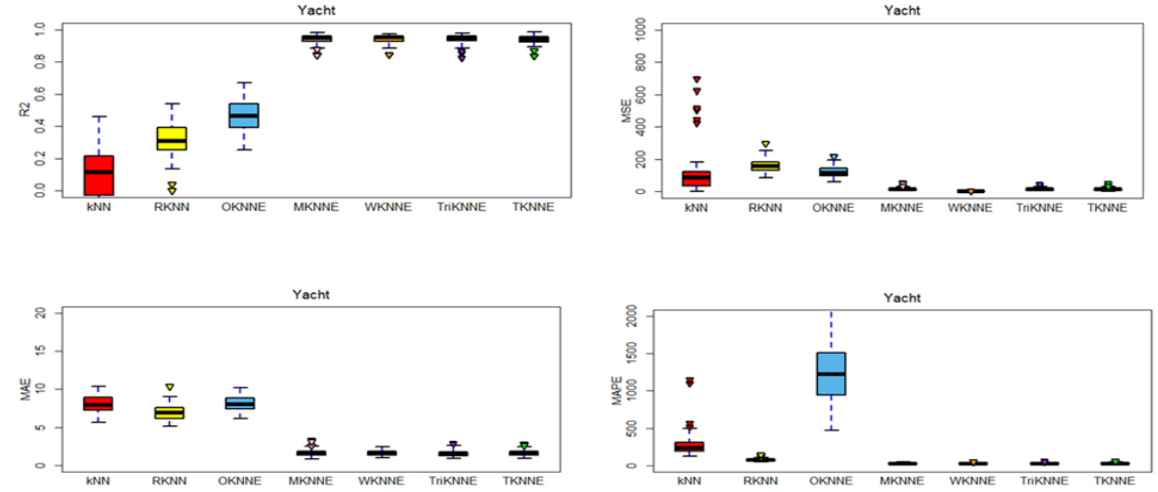

Figure 8 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Yatch dataset.

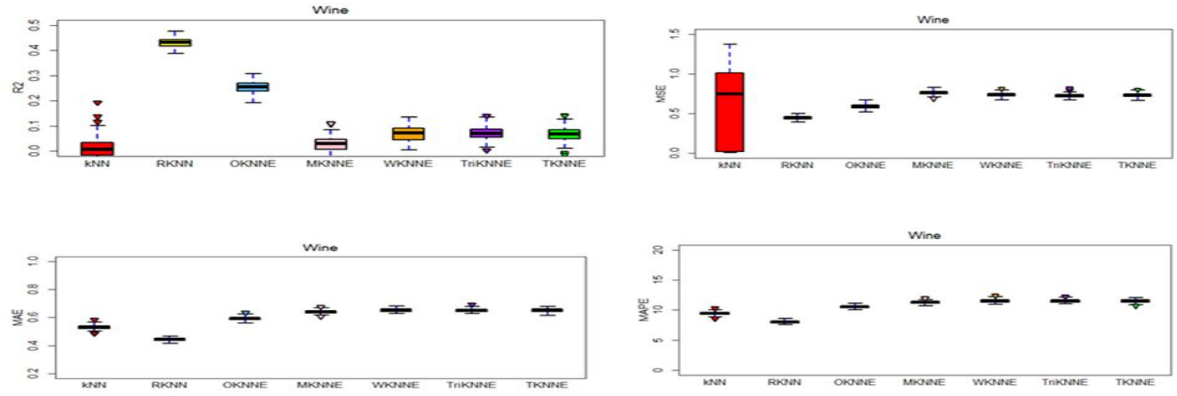

Figure 9 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Wine dataset.

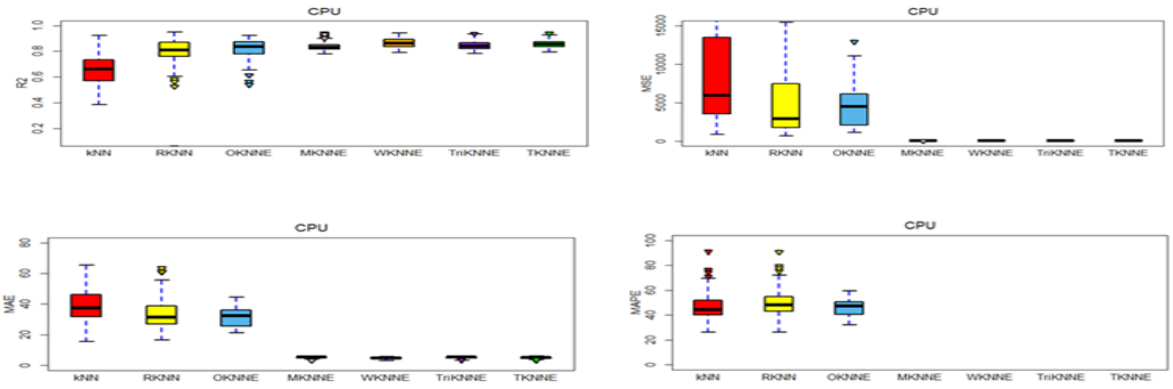

Figure 10 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the CPU dataset.

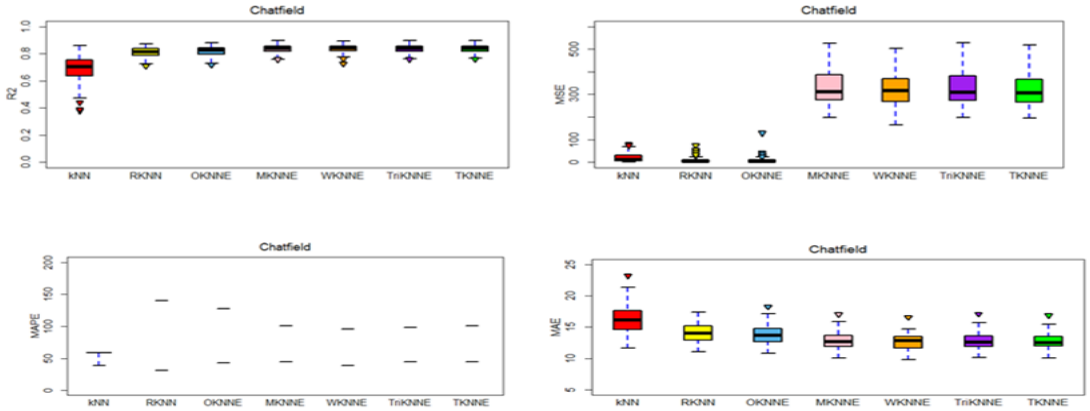

Figure 11 Comparative boxplots of MAE, MAPE, R^2, and MSE for k-NN and its robust variants (RKNN, OKNNE, MKNNE, WKNNE, TriKNNE, and TKNNE) on the Chatfield dataset.

Across the analyzed datasets, TKNNE consistently delivers robust and high-performing predictions. For Concrete and Boston (Figures 1-2), TKNNE produces close predictions with minimal spread, whereas traditional methods show greater sensitivity to heteroscedastic noise. In Real Estate, Andro, and Stock (Figures 3-5), which feature mixed distributions, TKNNE demonstrates stability, while classical approaches exhibit high vertical variance indicative of larger errors. For ATP, Yacht, and Wine (Figures 6-8), including the Yacht dataset where performance gaps are most pronounced, k-NN and RKNN are affected by severe outliers, whereas TKNNE remains insensitive, highlighting its robustness to non-linear and complex data patterns. Similarly, for CPU and Chatfield (Figures 9-10), both irregular datasets, TKNNE boxplots remain compact with the smallest errors, further confirming its reliability and effectiveness across diverse and challenging data conditions. Boxplot analysis of ten datasets shows TKNNE achieves the highest R^2 and lowest errors, with TriKNNE also outperforming classical k-NN variants. Traditional methods exhibit higher variance and more outliers, while the reduced spread in the proposed methods highlights their robustness and generalizability.

4. Conclusion

This study introduced four robust extensions of the k-Nearest Neighbor regression framework MKNNE, WKNNE, TriKNNE, and TKNNE designed to address the sensitivity of the classical k-NN method to outliers, noise, and skewed neighborhood distributions. By incorporating robust central tendency estimators in place of the traditional sample mean, the proposed methods achieved

significant improvements in prediction accuracy across ten benchmark datasets with diverse statistical characteristics. Empirical results demonstrated that the robust estimators consistently outperformed standard *k*-NN, RKNN, and OKNNE models. Improvements were observed across all major evaluation metrics (R^2, MSE, MAE, MAPE), with TriKNNE emerging as the most stable and accurate estimator overall. Its balanced integration of median and quartile information provided increased resistance to extreme values while maintaining the efficiency required for high-quality regression predictions. Similarly, WKNNE and TKNNE showed notable performance gains, particularly in datasets with heavy-tailed distributions or irregular noise patterns. These findings confirm that incorporating robust estimation in the neighborhood aggregation step is an effective strategy for enhancing *k*-NN regression. The practical significance of the proposed methods is highlighted by datasets such as Yacht, ATP, and Concrete, demonstrating their utility in real-world applications where measurement noise, anomalies, or local irregularities are common. Beyond predictive accuracy, this work establishes a foundation for further research on the role of robustness in neighborhood-based learning. Future research could explore: adaptive neighborhood selection techniques, integration of robust estimators with metric learning frameworks, hybrid models that jointly optimize distance metrics and robust aggregation, and applications to high-dimensional and domain-specific datasets, including medical, financial, and environmental data.

Overall, the proposed robust *k*-NN estimators offer a simple yet powerful enhancement to classical *k*-NN regression. Their consistent performance across diverse datasets makes them practical, reliable, and computationally efficient tools for predictive modeling in noisy and heterogeneous data environments.

REFERENCES

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. II—Recent progress. IBM Journal of research and development, 11(6), 601-617.

Mahesh, B. (2020). Machine learning algorithms-a review. International Journal of Science and Research (IJSR).[Internet], 9, 381-386.

Nasteski, V. (2017). An overview of the supervised machine learning methods. Horizons. b, 4, 51-62.

Bhavsar, P., Safro, I., Bouaynaya, N., Polikar, R., & Dera, D. (2017). Machine learning in transportation data analytics. In Data analytics for intelligent transportation systems (pp. 283-307). Elsevier.

Freedman, D. A. (2009). Statistical models: theory and practice. cambridge university press.

Cook, R. D., & Weisberg, S. (1982). Criticism and influence analysis in regression. Sociological methodology, 13, 313-361.

Seal, H. L. (1967). Studies in the History of Probability and Statistics. XV The historical development of the Gauss linear model. Biometrika, 54(1-2), 1-24.

Yan, X., & Su, X. (2009). Linear regression analysis: theory and computing. world scientific.

Muthukrishnan, R., & Rohini, R. (2016, October). LASSO: A feature selection technique in predictive modeling for machine learning. In 2016 IEEE international conference on advances in computer applications (ICACA) (pp. 18-20). IEEE.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbour nonparametric regression. The American Statistician, 46(3), 175-185.

Liu, W., & Chawla, S. (2011, May). Class confidence weighted knn Algorithms for imbalanced data sets. In Pacific-Asia conference on knowledge discovery and data mining (pp. 345-356). Springer, Berlin, Heidelberg.

Li, S., Harner, E. J., & Adjeroh, D. A. (2014, December). Random knn. In 2014 IEEE International Conference on Data Mining Workshop (pp. 629-636). IEEE.

Steele, B. M. (2009). Exact bootstrap k-nearest neighbour learners. Machine Learning, 74(3), 235-255.

Draper, N. R., & Smith, H. (1998). Applied regression analysis (Vol. 326). John Wiley & Sons.

Goldberger, A. S. (1970). Econometric Theory, New York, 1964. Goldberger Econometric Theory1964.

Fix, E., & Hodges, J. L. (1989). Discriminatory analysis. Nonparametric discrimination: Consistency properties. International Statistical Review/Revue Internationale de Statistique, 57(3), 238-247.

Fix, E., & Hodges Jr, J. L. (1952). Discriminatory analysis-nonparametric discrimination: Small sample performance. California Univ Berkeley.

Cover, T., & Hart, P. (1967). Nearest neighbour pattern classification. IEEE transactions on information theory, 13(1), 21-27.

Dasarathy, B. V. (1991). Nearest neighbour (NN) norms: NN pattern classification techniques. IEEE Computer Society Tutorial.

Dasarathy, B. V. (2002). Data mining tasks and methods: Classification: Nearest-neighbour approaches. In Handbook of data mining and knowledge discovery (pp. 288-298).

Babu, V. S., & Viswanath, P. (2009). Rough-fuzzy weighted k-nearest leader classifier for large data sets. Pattern Recognition, 42(9), 1719-1731.

Duda, R., Hart, P., & Stork, D. (2000). Pattern classification. The 2nd Edition. Hoboken.

Dudani, S. A. (1976). The distance-weighted k-nearest-neighbour rule. IEEE Transactions on Systems, Man, and Cybernetics, (4), 325-327.

Bailey, T., & AK, J. (1978). A NOTE ON DISTANCE-WEIGHTED K-NEAREST NEIGHBOUR RULES.

Hamamoto, Y., Uchimura, S., & Tomita, S. (1997). A bootstrap technique for nearest neighbour classifier design. IEEE transactions on pattern analysis and Machine intelligence, 19(1), 73-79.

Gowda, K., & Krishna, G. (1979). The condensed nearest neighbour rule using the concept of mutual nearest neighbourhood (corresp.). IEEE Transactions on Information Theory, 25(4), 488-490.

Angiulli, F. (2005, August). Fast condensed nearest neighbour rule. In Proceedings of the 22nd international conference on Machine learning (pp. 25-32).

Alpaydin, E. (1997). Voting over multiple condensed nearest neighbours. In Lazy learning (pp. 115-132). Springer, Dordrecht.

Gates, G. (1972). The reduced nearest neighbour rule (corresp.). IEEE transactions on information theory, 18(3), 431-433.

Rodríguez-Fdez, I., Mucientes, M., & Bugarín, A. (2013, July). An instance selection algorithm for regression and its application in variance reduction. In 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1- 8). IEEE.

Guo, G., Wang, H., Bell, D., Bi, Y., & Greer, K. (2003, November). KNN model-based approach in classification. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg.

Shokrzade, A., Ramezani, M., Tab, F. A., & Mohammad, M. A. (2021). A novel extreme learning machine based kNN classification method for dealing with big data. Expert Systems with Applications, 183, 115293.

Zhou, Z. H. (2012). Ensemble methods: foundations and algorithms. CRC press.

Bartlett, P., Freund, Y., Lee, W. S., & Schapire, R. E. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. The annals of statistics, 26(5), 1651-1686.

Breiman, L. (1996). Bagging predictors. Machine learning, 24(2), 123-140.

Caprile, B., Merler, S., Furlanello, C., & Jurman, G. (2004, June). Exact bagging with k-nearest neighbour classifiers. In International Workshop on Multiple Classifier Systems (pp. 72-81). Springer, Berlin, Heidelberg.

Zhou, Z. H., & Yu, Y. (2005). Adapt bagging to nearest neighbour classifiers. Journal of Computer Science and Technology, 20(1), 48-54.

Zhou, Z. H., & Yu, Y. (2005). Ensembling local learners throughmultimodal perturbation. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 35(4), 725-735.

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Bagheri, M. A., Gao, Q., & Escalera, S. (2013, December). A framework towards the unification of ensemble classification methods. In 2013 12th International Conference on Machine Learning and Applications (Vol. 2, pp. 351-355). IEEE.

Gul, A., Perperoglou, A., Khan, Z., Mahmoud, O., Miftahuddin, M., Adler, W., & Lausen, B. (2018). Ensemble of a subset of kNN classifiers. Advances in data analysis and classification, 12(4), 827-840.

Gu, J., Jiao, L., Liu, F., Yang, S., Wang, R., Chen, P., & Zhang, Y. (2018). Random subspace based ensemble sparse representation. Pattern Recognition, 74, 544-555.

Grabowski, S. (2002, February). Voting over multiple k-nn classifiers. In Modern problems of radio engineering, telecommunications and computer science (IEEE Cat. No. 02EX542) (pp. 223-225). IEEE.

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient kNN classification with different numbers of nearest neighbours. IEEE transactions on neural networks and learning systems, 29(5), 1774-1785.

Bottou, L., & Vapnik, V. (1992). Local learning algorithms. Neural computation, 4(6), 888-900.

Bontempi, G., Bersini, H., & Birattari, M. (2001). The local paradigm for modeling and control: from neuro-fuzzy to lazy learning. Fuzzy sets and systems, 121(1), 59-72.

Kainulainen, L., Miche, Y., Eirola, E., Yu, Q., Frénay, B., Séverin, E., & Lendasse, A. (2011). Ensembles of local linear models for bankruptcy analysis and prediction. Case Studies In Business, Industry And Government Statistics, 4(2), 116-133.

Ali, A., Hamraz, M., Kumam, P., Khan, D. M., Khalil, U., Sulaiman, M., & Khan, Z. (2020). A k-nearest neighbours based ensemble via optimal model selection for regression. IEEE Access, 8, 132095-132105.

Hassanat, A. B., Abbadi, M. A., Altarawneh, G. A., & Alhasanat, A. A. (2014). Solving the problem of the K parameter in the KNN classifier using an ensemble learning approach. arXiv preprint arXiv:1409.0919.

Zhang, Y., Cao, G., Wang, B., & Li, X. (2019). A novel ensemble method for k- nearest neighbour. Pattern Recognition, 85, 13-25.

Yang, P., & Huang, B. (2008, December). KNN based outlier detection algorithm in large dataset. In 2008 International Workshop on Education Technology and Training & 2008 International Workshop on Geoscience and Remote Sensing (Vol. 1, pp. 611-613). IEEE.