Electronic Journal of Applied Statistical Analysis
EJASA, Electron. J. App. Stat. Anal.
http://siba-ese.unisalento.it/index.php/ejasa/index
e-ISSN: 2070-5948
DOI: 10.1285/i20705948v16n3p784

A Hybrid Regression Model for Improving Prediction Accuracy

By Poojari and B.

15 December 2023

Electronic Journal of Applied Statistical Analysis
Vol. 16, Issue 3, 2023, 784-801
DOI: 10.1285/i20705948v16n3p784

A Hybrid Regression Model for Improving Prediction Accuracy

Satyanarayana Poojari*[a] and Ismail B.[b]

[a]*Department of Statistics, Mangalore University, Mangalagangothri, India.*
[b]*Department of Statistics, Yenepoya (Deemed to be University), Mangalore, India.*

15 December 2023

Regression Tree (RT) and K-Nearest Neighbor (KNN) models play significant roles in machine learning. RT facilitates interpretable decision-making, aiding in the comprehension of complex data relationships, while KNN is valued for its simplicity, adaptability to non-linear data, and robustness to noise, making it a versatile tool across various applications. The primary drawback of Regression Tree is its tendency to assign the same predicted value (average value) to all tuples satisfying the same corresponding splitting criterion. K-Nearest Neighbors (KNN) is sensitive to irrelevant or redundant features since all features contribute to similarity. This paper proposes a hybrid regression model based on Regression Tree (RT) and KNN, addressing the aforementioned issues. The model's performance is compared with KNN using 10 types of distance measures and further assessed against RT, K-Nearest Neighbor regression (KNN), and Support Vector Regression (SVR) through a Monte Carlo simulation study. Simulation results indicate that the hybrid model outperforms all other regression models, regardless of sample size, when observations follow normal distributions or t-distributions.The proposed model's effectiveness is demonstrated through a real-life application using data on global warming in Delhi.

keywords: Regression Tree, KNN, Hybrid model, SVR, Simulation.

1 Introduction

There is a need to develop algorithms to address the challenges arising from the large volume of data across various sectors due to the rapid advancement in technology. Anuradha

[*] Corresponding author: sathya1301@gmail.com.

ISSN: 2070-5948
http://siba-ese.unisalento.it/index.php/ejasa/index

and Thambusamy (2015) identified powerful and commonly used machine learning decision tree algorithms such as ID3, C4.5, C5.0, and CART (Classification and Regression Trees). C4.5, an enhanced version of ID3 developed by Ross Quinlan, stands out among these. Machine learning methods are increasingly applied to diverse datasets to uncover hidden patterns and facilitate accurate predictions for future decision-making. The applications of machine learning extend to sectors such as retail, banking, military, and healthcare. In a study by Adhatrao et al. (2013), ID3 and C4.5 classification algorithms were utilized to predict students performance. Muhamad Safiih et al. (2016) demonstrated that the performance of a multiple regression model can be enhanced through a hybrid approach. They proposed an Artificial Neural Network for the MLR, termed as Artificial Neural Network-Multiple Linear Regression (ANN-MLR). Chakraborty et al. (2019) suggested a hybrid regression model based on a regression tree and support vector regression for predicting boiler water quality.

Recent studies have applied meta-heuristic optimization algorithms to feature and parameter selection in SVMs. Al-Thanoon et al. (2019) used a hybrid FA-PSO for penalized SVR in chemometrics. Ismael and Benbouziane (2020) enhanced HHOA for v-SVR. Liu et al. (2021) applied GWO for SVC, and Zhang et al. (2021) used HHOA for v-SVR. Algamal et al. (2021) improved GOA for SVR. In 2023, Liu introduced a hybrid Pelican and Black Hole Algorithm (POABHA) for kernel semi-parametric fusion modeling. These studies collectively represent significant advancements in optimizing SVMs with various meta-heuristic techniques.

CART (Classification and Regression Tree) is the most popular, efficient, and widely used method for constructing decision trees, introduced by Breiman et al. (1984). CART considers binary splits (($X_i \leq$ splitting point and $X_i >$ splitting point) for each variable based on the splitting point, which minimizes the relative sum of squared errors in the two partitions resulting from the splitting point. For continuous variables, all consecutive midpoints are considered to select the final best splitting point. The best variable to start the split in each node is selected using the Gini Index criterion. The variable with the minimum Gini index is chosen as the splitting variable at that node. The two branches of each node are the outcomes of the splitting point. Splitting stops when the reduction in error from the best split falls below the pre-specified threshold parameter (complex parameter), which is usually in the range 0.001-0.005. Loh and Shih (1997),Shih (2004) observed that the above splitting procedure is biased as it searches for all possible splits on all variables and suggested that a proper normalization method will overcome this difficulty.

To prevent overfitting, a sequence of values of threshold parameters is considered, and the final threshold value is selected according to the cross-validation technique based on the minimum prediction error criterion. Alternatively, one can also select the final threshold value using the 1-Standard Error rule, which yields a prediction error one standard deviation larger than the minimum error estimated by the cross-validation method. CART has several advantages over traditional regression models. When there are deviations from a linear form, CART performs better than the traditional regression model if interactions are not included. To address the problem of identifying the queried object from a large volume of datasets, which involves time and computational complexity, various techniques have been proposed. Among these, the KNN algorithm is a simple, highly efficient, and effective technique. The KNN technique has a wide variety of applications in many fields, namely text mining, medicine, agriculture, and finance. Other applications include the prediction of solvent accessibility in protein molecules and estimating the amount of glucose in the blood of a diabetic person, etc.

In the KNN algorithm, the whole data is classified into training data and test data. Distance is evaluated from all training points, and the point with the lowest distance is called the nearest neighbor. The non-structured KNN technique has been improved to meet the increase in dimensionality of the data space. These algorithms increase the speed of the basic KNN algorithm. (Cover and Hart, 1967) proposed that the nearest neighbor can be calculated based on the value of K, which specifies the number of nearest neighbors to define a class of test data. KNN can also be used for regression. It is useful to weight the contributions of the neighbors so that the nearer neighbors contribute more to the average than the more distant ones.

After selecting the value of K, KNN prediction is the average of the K nearest neighbours outcome:

$$\hat{m}_k(x) = \frac{1}{n}\sum_{i=1}^{n} W_{ki}(x)Y_i \tag{1.1}$$

where $\{W_{k1}(x), W_{k2}(x), \ldots, W_{kn}(x)\}$ is a sequence of weights defined through a set of indexes $J_x = \{i : X_i \text{ is one of the } k \text{ nearest observations to } x\}$.
The KNN sequence is constructed as:

$$W_{ki}(x) = \begin{cases} \frac{n}{k}, & \text{if } i \in J_x; \\ 0, & \text{otherwise.} \end{cases}$$

A regression-based KNN algorithm was applied to gene function prediction by Yao and Ruzzo (2006). This algorithm demonstrated effectiveness in predicting gene function according to three well-known Escherichia coli classification schemes suggested by biologists. Results indicated that the KNN outperformed naive KNN methods and showed competitiveness with support vector machine (SVM) algorithms. In a review paper on KNN algorithms, Dhanabal and Chandramathi (2011) explained the key ideas, merits, and demerits of 14 different types of KNN algorithms. Based on the literature study, a suggestion was made to use structure-based KNN techniques for small volumes of data and non-structure KNN techniques for large volumes of data.

Imandoust and Bolandraftar (2013) addressed the sensitivity of KNN to irrelevant or redundant features due to the contribution of all features to the similarity. This issue can be resolved by proper feature selection. The proposed model overcomes this problem by applying KNN to each group after selecting significant variables from the decision tree. Prasatha et al. (2019) reviewed the performance of KNN classifiers using numerous distance measures on clean and noisy datasets. They proposed an appropriate distance measure that can be used with KNN in general and on noisy data. Furthermore, it was observed that non-convex distance measures perform better when applied to most datasets compared to other test distances. Bhatia and Vandana (2010) observed that among structure-less and structure-based KNN algorithms, the structure-less method overcomes memory limitations, while structure-based techniques reduce computational complexity.

KNN secures a position in the top 10 methods in machine learning algorithms. As a non-parametric algorithm, KNN stands out as the best choice in prediction studies with little or no prior information about the underlying distribution of the data. Despite longer computation times, KNN remains extensively used in many fields due to its simplicity and reasonable accuracy.

The paper is organized as follows: Methodology, notations used in the article, and the flow chart of the proposed model are given in Section 2. The simulation setup to check the performance of the proposed model and its results are presented in Section 3. In Section 4, numerical analysis with a real-life application is considered. Section 5 concludes the paper.

2 Methodology

The primary drawback of Regression Tree regression is its tendency to assign the same predicted value, an average value, to all tuples satisfying the corresponding splitting criterion. Even though RMSE is minimized, the predicted response value holds significant importance for the decision-making process. Additionally, KNN is sensitive to irrelevant or redundant features since all features contribute to the similarities. In this paper, we propose an efficient hybrid regression model based on Regression Tree (RT) and KNN to address these issues. The proposed model tackles the problem associated with KNN by applying KNN to each group after selecting significant variables from the decision tree. The suggested Gini-based KNN regression model, instead of directly searching for K-nearest neighbors in the entire training data, first groups the elements based on the Gini Index criteria, and then KNN is applied to each group.

The proposed algorithm depends on two input parameters: 'dataset with variable list' and 'variable selection method.' Here, a heuristic procedure is used for selecting the best variables, employing the Gini index as the variable selection measure. The tree's construction follows the same principles as CART. The novelty of this work lies in the post-construction phase of CART. Instead of predicting the average of all tuples in a branch as the response variable's predicted value, for each branch, the data will be rearranged first, and then separate KNN regression is applied to obtain the predicted values.

2.1 Algorithm to generate hybrid RT-KNN tree

Input:

- *Data set* , Which consist of training tuples.
- *Variable list* , the set of variables related to study variables.

- *Variable Selection Method*, A method to determine the splitting variable and splitting point that best partition the dataset.

Output: RT-KNN tree which holds predicted values of response variable

Method

- Create node N
- If tuples in dataset T, are all same class C then return N as leaf node and apply KNN regression obtain the predicted values of the corresponding response variable.
- If the list is empty then apply *"Variable Selection Method"* to determine the best splitting variable and splitting point.
- Lable N with splitting criterion.
- For each outcome z of splitting criterion // partition the dataset T and grow subtree for each partition.
- Let T_z, be the set of datasets tuples in T satisfying outcome of z // a partition .
- If T_z is empty then attach a leaf labled with the majority class in T to node N and Re-arrange the tuples in T_z and Apply KNN regression to T_z and obtained the predicted values of the corresponding response variable.
- Else attach the node returned by 'Generate RT-KNN tree' to node N
  end for
- Return N

2.2 Procedure

1. Built a decision tree using CART algorithm
2. For first leaf node note down the position of the tuples satisfying the corresponding splitting criterion, re-arrange the both response variable and explanatory variables according to position noted.
3. Apply the KNN regression to the arranged dataset to get the predicted values of the response variable based on appropriate value of K.
4. Repeat step 2 and 3 for all the leaf node
5. Print the predicted values along with its original observation of the response variable.

After selecting the value of K, according to proposed mode prediction in each leaf node is the average of the k nearest neighbours outcome variable values:

$$\hat{m}_{L_pk}(x) = \frac{1}{n_p}\sum_{i=1}^{n_p} W_{L_pki}(x)Y_{L_pi} \tag{2.1}$$

L_p - p^{th} leaf node, $p = 1, 2, \dots, m$
M - number of leaf node
n_p - number observations in p^{th} leaf node
Y_{L_pi}- value of i^{th} observation of outcome variable in the p^{th} leaf node
$\{W_{k1}(x), W_{k2}(x), \dots, W_{kn}(x)\}$ is a sequence of weights defined through a set of indexes from p^{th} leaf node. Finally, the predicted values from each leaf node are combined and compared with original observations of the response variable.

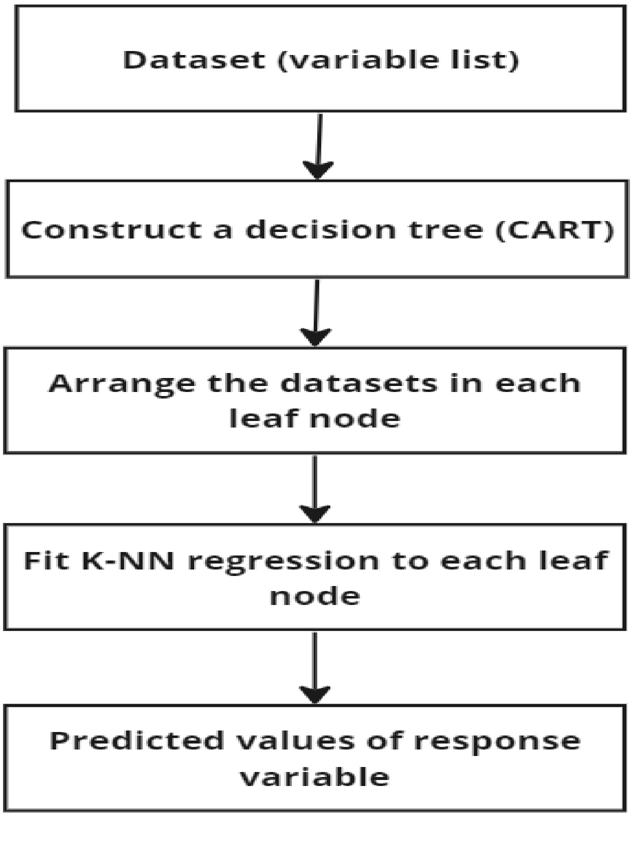
Dataset (variable list)
Construct a decision tree (CART)
Arrange the datasets in each leaf node
Fit K-NN regression to each leaf node
Predicted values of response variable

Figure 1: Diagrammatic representation of the proposed method

The RT-KNN hybridization approach effectively combines the strengths of regression trees (RT) and k-nearest neighbors (KNN). Initially, a regression tree is carefully constructed to identify the most relevant features for the task, effectively reducing dimensionality and improving model performance. Subsequently, a modified KNN algorithm is trained using the carefully selected features, considering the class distribution of nearest neighbors to ensure that predictions align seamlessly with the majority class in the neighborhood. Finally, the RT-KNN approach exhibits inherent resilience to noise and outliers owing to the combined ability of RT to filter out irrelevant features and KNN's ability to handle noisy or extreme data points.

RT-KNN hybridization offers several benefits over traditional RT and KNN models:

1. **Improved Stability:** The hybrid approach reduces the sensitivity of the KNN algorithm to changes in data distribution or outliers.
2. **Enhanced Generalization:** The incorporation of a regression tree helps the KNN algorithm generalize better to unseen data.
3. **Increased Robustness:** The combination of RT and KNN provides resilience to noise and outliers, improving the overall performance of the model.

Chomboon et al. (2015) utilized 11 distance measures, Todeschini et al. (2015),Todeschini et al. (2016) employed 18 distance measures, Lopes and Ribeiro (2015) used five distance measures, and Punam and Nitin (2015) employed three distance measures to investigate the effect of distance measures on the performance of KNN. They reported their best distance measures. Euclidean distance is the most widely used distance in the KNN algorithm. The KNN method can handle noisy data, and it relies mainly on measuring the distance or similarity between the test and training datasets. Given the availability of numerous distance measures, determining which one to use for KNN and the proposed model poses a significant question that researchers need to address. This paper attempts to answer this question by evaluating the performance of these two algorithms for regression using different distance measures.

The focus of this paper is to identify the top five best distance measures (Prasatha et al. (2019)) to be used for KNN and the proposed hybrid model, ensuring the highest possible prediction accuracy. This study also aims to address whether the proposed method is dependent on distance/similarity measures.

Distance measures

The distance function between any two vectors X and y is defined as a distance between both vectors as a non-negative real number which satisfies the properties (Elena Deza (2009)) of Identity of indiscernible and Symmetry and triangle inequality.

1. **Euclidean (ED)**: This distance is the root of the sum of the square of differences between the two vectors x and y

$$ED(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2. **Manhattan (MD)** : The Manhattan distance is also known as city block distance which considers the distance is the sum of the absolute difference between the two vectors
$$MD(x, y) = \sum_{i=1}^{n} |x_i - y_i|$$

3. **Lorenztzian distance (LD)**: It is given by natural log of the absolute difference between two vectors.
$$LD(x, y) = \sum_{i=1}^{n} \ln(1 + |x_i - y_i|)$$
where ln is the natural logarithm and to avoid log of zero, one is added.

4. **Canberra distance (CanD)**: It is a weighted version of Manhattan distance. It is given computed as absolute difference between the values of two vector x and y is divided by the sum of absolute value of the attribute.
$$CanD(x, y) = \sum_{i=1}^{n} \frac{|x_i - y_i|}{|x_i| + |y_i|}$$

5. **Dice distance (DicD)**: This distance is a complementary to the dice similarity (Dice (1945)) and is obtained by subtracting the dice similarity from one.
$$DiceD(x, y) = 1 - \frac{2\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 + \sum_{i=1}^{n} y_i^2}$$

6. **Average distance (AD)**: It is a modified version of the Euclidean distance. This distance given by (Shirkhorshidi et al. (2015))
$$AD(x, y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2}$$

7. **Clark distance (ClaD)** : This distance was introduce by Clark (1952) and it is computed as square root of half of the divergence distance
$$ClaD(x, y) = \sqrt{\sum_{i=1}^{n}\left(\frac{x_i - y_i}{|x_i| + |y_i|}\right)^2}$$

8. **Divergence distance (DivD)** : It is computes as two times sum of square of difference divided by square of their sum.
$$DivD(x, y) = 2\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{(x_i + y_i)^2}$$

9. **Squared Chi- Squared distance (SCSD)**:

$$SCSD(x,y) = \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{|x_i| + |y_i|}$$

10. **Cosine distance (CosD)** : It is also known as angular distance. This distance is obtained by subtracting the cosine similarity from one

$$CosD(x,y) = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}$$

The model is built using training data, and its performance is subsequently assessed with a test dataset. Root Mean Square Error and Mean Absolute Error are computed to evaluate the predictive efficacy of the proposed method, and these metrics are compared with the results from KNN regression, SVR, and Regression Tree.

3 Simulation Study

Monte Carlo simulation serves as a robust tool for evaluating the RT-KNN hybridization approach, offering a comprehensive assessment of performance, uncertainty quantification, robustness, comparative analysis, and generalizability. The proposed hybrid model's performance was evaluated using 10 different types of distance measures and compared to that of several other popular machine learning models. Additionally, the model's robustness was assessed by evaluating its performance on data from the t-distribution. These thorough assessments ensure a comprehensive evaluation of the model's strengths, weaknesses, and applicability, establishing it as a valuable tool for validating the RT-KNN model in real-world applications.

In this section, a simulation study is conducted to highlight the distinctions between the proposed model (Hybrid RT-KNN), RT, Gaussian kernel-based SVR, and KNN models. The predictive performances of these models are compared in terms of Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) using R. The Regression tree was constructed using the rpart package. The covariates in the simulation included three continuous variables X1, X2 and X3 from a normal distribution with a mean vector (13, 14, 15) and a variance vector (4.5, 5.5, 6.5), respectively. The sample sizes used are 30, 40, 50, 80, 100, 200, 300, 400, 500, 1000, 2000, 3000. The tree was grown to consist of three leaf nodes, and the threshold value for the stopping parameter in RT is set to 0.01. For each case, 5,000 repetitions were performed, and in each simulation, the tree was constructed using training data, and its performance was evaluated using independently generated test data. A simulation study to check the robustness of the model is also conducted by generating observations from a multivariate t-distribution. Ten different distance measures were used in the study. The RMSE of each distance measure on each sample size is averaged over 5000 runs. The same technique is followed for both KNN and the proposed RT-KNN model. The results are summarized in Table 3.1, and the following are the observations. RStudio version 4.1.3 was used for the simulations and data analysis using the following packages: ggplot2, dplyr, tidyr, rpart, caTools, e1071, MASS, caret, tune, and kernlab.

Table 1: Prediction Performance of KNN & RT-KNN method for different distances.

Sample size	**Method**	**ED**	**MD**	**CanD**	**AD**	**LD**	**DicD**	**ClaD**	**DivD**	**SCSD**	**CosD**
30	KNN	13.94	14.44	16.74	14.00	18.23	14.68	17.98	17.98	14.22	43.76
	RT-KNN	9.79	10.10	11.15	9.86	11.39	10.42	11.54	11.54	9.85	17.23
50	KNN	12.24	12.61	15.14	12.26	16.25	12.82	16.68	16.68	12.53	47.38
	RT-KNN	9.26	9.51	10.64	9.30	10.68	9.84	10.78	10.78	9.16	15.65
80	KNN	11.01	11.33	13.71	11.04	14.73	11.43	14.97	14.97	11.36	50.98
	RT-KNN	8.80	9.05	10.24	8.86	10.05	9.31	10.35	10.35	8.64	14.44
100	KNN	10.39	10.69	12.98	10.43	13.81	10.78	13.92	13.92	10.83	52.36
	RT-KNN	8.48	8.73	9.86	8.53	9.66	8.96	9.91	9.91	8.47	13.95
200	KNN	9.08	9.31	11.21	9.10	11.85	9.34	11.58	11.58	10.19	58.92
	RT-KNN	7.88	8.13	9.25	7.93	9.01	8.33	9.24	9.24	8.13	13.39
300	KNN	8.51	8.68	10.34	8.52	10.80	8.69	10.43	10.43	10.12	61.82
	RT-KNN	7.58	7.79	8.95	7.62	8.64	7.97	8.88	8.88	8.08	13.42
400	KNN	8.18	8.31	9.70	8.20	10.12	8.33	9.87	9.87	10.92	64.84
	RT-KNN	7.46	7.66	8.69	7.46	8.49	7.88	8.73	8.73	8.74	14.14
500	KNN	7.94	8.09	9.42	7.95	9.78	8.07	9.42	9.42	11.01	67.78
	RT-KNN	6.85	7.05	8.13	6.90	7.87	7.19	8.17	8.17	8.33	14.73
800	KNN	7.49	7.60	8.76	7.50	8.89	7.59	8.66	8.66	12.22	71.36
	RT-KNN	6.23	6.43	7.48	6.29	7.39	6.52	7.44	7.44	10.06	16.97
1000	KNN	7.33	7.44	8.51	7.35	8.57	7.42	8.43	8.43	13.50	74.23
	RT-KNN	6.07	6.22	7.32	6.11	7.14	6.37	7.30	7.30	11.60	18.79
3000	KNN	8.24	8.25	8.55	8.24	8.38	8.23	8.49	8.49	13.78	15.66
	RT-KNN	7.57	7.62	8.53	7.58	8.04	7.55	8.86	8.86	12.13	10.89

1. The Euclidean distance measures outperformed all other distance measures irrespective of sample sizes, both for KNN and RT-KNN. For all types of distance measures, the proposed RT-KNN method performs better than the KNN method.

2. For a sample size of 200, the prediction performance of KNN and RT-KNN indicates that distance measures ED, MD, AD, and DicD form a group based on similar performance, while CanD, LD, ClaD, and DivD fall into another group.

3. Among the considered distance measures in the study, the performance based on CosD is very poor. The RMSE value of this distance measure is almost four times higher than all other methods up to n=100, and afterward, RMSE starts to increase. For a sample size of 1000, RMSE of CosD is ten times higher than all other methods in the case of KNN regression. For the proposed method, RMSE using CosD is higher than all other distances but better compared to the KNN method, irrespective of the sample size.

4. The RMSE of the KNN algorithm based on CosD increases with the sample size, whereas the RMSE of the proposed method based on CosD decreases up to a sample size of 200 and then increases. It is also observed that the RMSE of both KNN and the proposed method decreases with an increase in the sample size, indicating that prediction accuracy is high for large samples.

Top 5 distances in terms of RMSE

Euclidean distance outperforms all other distance measures across all sample sizes for both KNN and RT-KNN methods. This is followed by the distances measures AD, MD, DicD, and SCSD up to a sample size of 400. For sample sizes greater than or equal to 400, the top five distance measures are ED, AD, MD, DicD, and CanD.

Table 2: Prediction performance of RT- KNN (Euclidean distance), KNN, RT and SVR

Sample size	**Method**	**RT-KNN**	**KNN**	**RT**	**SVM**
30	RMSE	9.79	13.94	18.62	9.71
	MAE	7.74	10.75	14.81	7.65
50	RMSE	9.26	12.24	15.82	9.36
	MAE	7.31	9.38	12.40	7.45
80	RMSE	8.80	11.01	13.77	9.08
	MAE	6.97	8.44	10.72	7.26
100	RMSE	8.48	10.39	12.92	8.64
	MAE	6.42	7.98	10.05	6.90
200	RMSE	7.88	9.08	11.84	8.12
	MAE	6.25	7.02	9.27	6.47
300	RMSE	7.58	8.51	12.03	7.51
	MAE	5.97	6.59	9.47	6.02
400	RMSE	7.46	8.18	12.39	7.43
	MAE	5.87	6.36	9.75	5.92
500	RMSE	6.85	7.94	12.74	7.35
	MAE	5.29	6.18	10.04	5.62
800	RMSE	6.23	7.49	13.29	7.02
	MAE	4.84	5.86	10.47	5.45
1000	RMSE	6.07	7.33	13.59	6.78
	MAE	4.69	5.75	10.71	5.18
3000	RMSE	5.57	7.24	14.05	6.23
	MAE	4.57	6.51	11.12	4.84

Table 2 summarizes the performance of all three regression models, namely the proposed hybrid model, KNN, SVR, and RT, when the observations are from a normal distribution. As expected, the hybrid RT-KNN model has lower RMSE and MAE than all other methods. Thus, the proposed model demonstrates better predictive accuracy than all other methods, irrespective of the sample size. It is also observed that both RMSE and MAE for the proposed method decrease as the sample size increases, leading to higher prediction accuracy. However, for RT, the RMSE and MAE decrease up to a sample size of 200 and then start to increase with the sample size, which is a noticeable disadvantage. The proposed model overcomes this disadvantage of RT and performs better than all other methods under the RMSE and MAE criteria.

Robustness

To check the property of robustness of the proposed model, observations are generated from multivariate t –distribution, a fat-tailed distribution (Islam (2017)) and the results are summarised in table 3

Table 3: Performances of KNN and RT- KNN for different distance measure

Sample size	**Method**	**ED**	**MD**	**CanD**	**AD**	**LD**	**DicD**	**ClaD**	**DivD**	**SCSD**	**CosD**
30	KNN	9.20	9.29	9.90	9.22	9.59	9.19	9.98	9.98	13.09	14.57
	RT-KNN	6.00	6.06	6.32	6.03	6.14	6.04	6.41	6.41	6.83	6.97
50	KNN	8.83	8.91	9.51	8.84	9.21	8.80	9.50	9.50	13.41	14.91
	RT-KNN	5.96	6.00	6.26	5.98	6.12	5.98	6.35	6.35	6.68	6.66
80	KNN	8.74	8.81	9.37	8.76	9.12	8.70	9.29	9.29	13.63	15.24
	RT-KNN	6.06	6.12	6.32	6.08	6.21	6.07	6.45	6.45	6.86	6.76
100	KNN	8.65	8.70	9.25	8.65	8.99	8.62	9.13	9.13	13.61	15.25
	RT-KNN	6.14	6.20	6.45	6.16	6.30	6.13	6.54	6.54	6.97	6.78
200	KNN	8.51	8.54	9.03	8.52	8.81	8.47	8.94	8.94	13.76	15.53
	RT-KNN	6.32	6.41	6.70	6.35	6.54	6.29	6.86	6.86	7.30	7.21
300	KNN	8.53	8.57	9.02	8.55	8.85	8.52	8.96	8.96	13.84	15.72
	RT-KNN	6.54	6.60	6.89	6.56	6.72	6.46	7.05	7.05	7.74	7.54
400	KNN	8.41	8.42	8.87	8.41	8.64	8.39	8.79	8.79	13.78	15.65
	RT-KNN	6.61	6.66	6.87	6.64	6.74	6.62	7.07	7.07	7.94	7.62
500	KNN	8.39	8.41	8.85	8.40	8.64	8.37	8.76	8.76	13.79	15.59
	RT-KNN	6.87	6.96	7.26	6.86	7.15	6.74	7.50	7.50	8.50	8.02
800	KNN	8.31	8.32	8.75	8.31	8.54	8.30	8.67	8.67	13.78	15.70
	RT-KNN	7.02	7.04	7.62	7.03	7.26	7.00	7.80	7.80	9.17	8.69
1000	KNN	8.32	8.35	8.73	8.33	8.54	8.32	8.66	8.66	13.89	15.70
	RT-KNN	7.65	7.71	8.17	7.70	8.09	7.59	8.49	8.49	10.25	9.49
3000	KNN	8.24	8.25	8.55	8.24	8.38	8.23	8.49	8.49	13.78	15.66
	RT-KNN	7.57	7.62	8.53	7.58	8.04	7.55	8.86	8.86	12.13	10.89

Based on the results in table 3, the following observations are obtained .

1. For all types of distance measures, the proposed RT-KNN method performs better than the KNN method.

2. The Euclidean distance measure performs better than all other distance measures up to a sample size of 200 for both KNN and RT-KNN. For a sample size of 200, DicD outperformed all other distance measures.
3. The prediction performance of KNN and RT-KNN shows that distance measures ED, MD, AD, and DicD form a group based on similar performance, while CanD, LD, ClaD, and DivD fall into another group.
4. Among the considered distance measures in the study, the performance based on SCSD and CosD is very poor due to high RMSE values.
5. For the proposed method, RMSE using CosD is higher than all others but smaller compared to the KNN method, irrespective of the sample size.

Table 4: Prediction performance of RT- KNN(Euclidean distance), KNN, SVR and RT

Sample size	**Method**	**RT-KNN**	**KNN**	**RT**	**SVR**
30	RMSE	6.00	9.20	7.86	8.35
	MAE	4.85	7.24	6.13	5.86
50	RMSE	5.96	8.83	7.26	8.74
	MAE	4.81	6.96	5.64	6.27
80	RMSE	6.06	8.74	7.02	9.12
	MAE	4.89	6.90	5.43	6.74
100	RMSE	6.14	8.65	6.88	9.26
	MAE	4.95	6.83	5.34	6.82
200	RMSE	6.32	8.51	6.85	9.56
	MAE	5.02	6.72	5.34	7.12
300	RMSE	6.54	8.53	7.11	9.77
	MAE	5.09	6.67	5.49	7.39
400	RMSE	6.61	8.41	7.11	9.82
	MAE	5.14	6.64	5.57	7.45
500	RMSE	6.87	8.39	7.27	9.85
	MAE	5.27	6.62	5.69	7.50
800	RMSE	7.02	8.31	7.51	9.89
	MAE	5.38	6.57	5.89	7.62
1000	RMSE	7.65	8.32	7.62	9.94
	MAE	5.67	6.56	5.97	7.69
3000	RMSE	7.57	8.24	8.00	10.10
	MAE	5.57	6.51	6.26	7.84

Table 4 summarizes the performance of all four regression models when the observations are from a multivariate t-distribution. The proposed method demonstrates better predictive accuracy than all other methods, irrespective of the sample size. Only for the KNN method does RMSE and MAE decrease as the sample size increases. In the case of the Regression tree, RMSE and MAE decrease up to a sample size of 300, then start to increase with an increase in the sample size. RMSE and MAE of the proposed method decrease up to a sample size of 80, then start to increase with the sample size. Despite this nature, the proposed method exhibits minimum error compared to all other methods. This shows that the hybrid method is robust to distribution assumptions.

4 Real life application

Delhi, known for its pervasive pollution, is surrounded by a pollution layer stemming from various sources within the city, nearby regions, and even distant sources. The dataset utilized in this study pertains to air pollutant levels and temperature in Delhi, sourced from 'Giovanni.gsfc.nasa.gov' spanning from July 2017 to December 2020. The variables included in the study are Temperature (°C), Carbon monoxide (CO in μg/m^3), Benzene (Benzene in μg/m^3), Nitrogen dioxide (NO_2 in μg/m^3), Ozone (O_3 in μg/m^3), Ammonia (NH_3 in μg/m^3), Sulfur dioxide (SO_2 in μg/m^3), Particulate matter (PM_{10} and $PM_{2.5}$ in μg/m^3). The dataset is partitioned into a training set and a test set in an 80:20 ratio. Each experiment undergoes repetition five times with randomly assigned test and train sets, and the average performance over 5-fold validation is reported. The e1071 package is employed to fit a Gaussian kernel-based Support Vector Regression (SVR) model, utilizing most default arguments present in the packages.

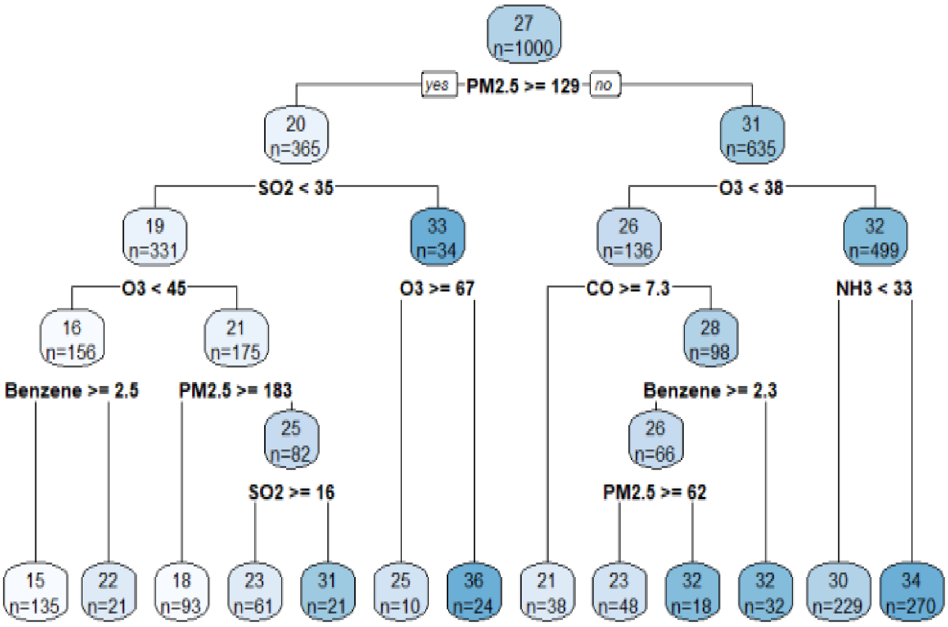

Figure 2: RT-KNN hybrid tree for Temperature

The above RT–KNN tree shows that PM_{10}, SO_2, CO, $PM_{2.5}$, Benzene, O_3 and NH_3 are the significant variables for Temperature. Also observe that hybrid model successfully capture the interaction effect among various air pollutants. The interpretation of these effects is straightforward.

If $PM_{2.5} < 129$, $O_3 > 38$, $NH_3 > 33 \Longrightarrow$ 34°C (high temperature).

If $PM_{2.5} > 129$, $SO_2 > 35$, $O_3 > 67 \Longrightarrow$ 36°C(high temperature).

The primary goal of this study is not only to enhance the predictive model for forecasting global temperatures but also to identify significant causal variables and relationships. In the proposed model, the KNN regression model is applied to each leaf node to derive the fitted values of the response variable. Using the fitted model, predictions are made for the response variable in the test set, and both RMSE and R^2 are computed.

Table 5: Comparison of Prediction Performance of RT, SVR, KNN and proposed method (Average values of 5 -fold cross validations)

Method	**RMSE**	R^2
RT-KNN Hybrid	**14.01**	**76.88**
KNN	14.88	57.81
RT	15.28	64.56
SVR	14.26	70.61

From the table 5, t is evident that the proposed RT-KNN Hybrid model exhibits the minimum RMSE and maximum R2 value. Therefore, the proposed method outperforms KNN regression, Regression Tree, and SVR models. The novelty of the proposed work lies in predicting different values of the response variable instead of the same average value for all observations in a branch. This approach overcomes the problem associated with decision tree-based regression. Additionally, the proposed method addresses the sensitivity issue of KNN regression by applying KNN to each group after selecting significant variables from the Regression Tree. In addition to improved accuracy, the hybrid model assists administrators in taking necessary actions to mitigate air pollutant levels.

5 Conclusion

Regression Tree excels in interpretability and capturing non-linear relationships, providing transparency and adaptability. Meanwhile, K-Nearest Neighbors offers localized predictions and robustness to outliers, adding crucial flexibility for handling intricate dataset variations. The amalgamation of these strengths in the hybrid RT-KNN model yields a versatile tool that excels in handling diverse and complex data scenarios, enhancing overall prediction accuracy in regression tasks. KNN and Regression Tree can be powerful tools to analyze large volumes of data to obtain useful information for decision-making. These two techniques are simple, efficient toolboxes for researchers to deal with challenges processed by the rapid improvement in technologies in terms of time complexity and computational complexity. These two techniques have a wide variety of applications in many fields because of their additional flexibility.

In this paper, a hybrid RT-KNN model is proposed to improve prediction accuracy in regression. The proposed method overcomes the disadvantages of both Regression Tree and KNN methods. Even though RMSE is minimal, the predicted value of the response variable is of great interest for future decision-making processes. KNN is sensitive to irrelevant or redundant features because all features contribute to the similarities. The proposed model overcomes this problem by applying KNN to each group after selecting significant variables from the Regression tree. The focus of the study was on the performance of the proposed method to describe the relationship between a small number of covariates with a continuous outcome variable in the case of small sample sizes. The performance of the proposed model is also compared with KNN for different types of distance measures.

The results show that the Euclidean distance measure outperformed all other distance measures, irrespective of sample sizes for both KNN and proposed RT-KNN. For all types of distance measures, the proposed RT-KNN method performs better than the KNN method. The performance of the proposed method using the Euclidean distance is also compared with the Regression tree, SVR model, and KNN regression model in terms of the RMSE and MAE through a simulation study. The simulation results show the proposed model performs better than KNN, SVR model, and Regression Tree irrespective of sample size when the observations are from a normal distribution. A simulation study also shows that the proposed hybrid model is fairly robust. The working of the proposed method is illustrated for a real-life application to predict the global temperature of Delhi, and the result shows that the proposed model, along with overcoming the disadvantages, performs better than KNN and Regression tree. The hybrid model, along with improved accuracy, also helps administrators take necessary action to curb air pollutant levels. In the proposed hybrid model, overfitting is a potential limitation, especially when dealing with a small dataset. To address this, rigorous hyperparameter tuning, strategic tree pruning, and robust cross-validation techniques can be employed to effectively counter overfitting and enhance model performance.

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Adhatrao, K., Gaykar, A., Dhawan, A., Jha, R., and Honrao, V. (2013). Predicting students' performance using id3 and c4.5 classification algorithms. *International Journal of Data Mining & Knowledge Management Process*, 3.

Al-Thanoon, N. A., Qasim, O. S., and Algamal, Z. Y. (2019). A new hybrid firefly algorithm and particle swarm optimization for tuning parameter estimation in penalized support vector machine with application in chemometrics. *Chemometrics and Intelligent Laboratory Systems.*

Algamal, Z. Y., Qasim, M., H., M., and Mohammad Ali, H. T. (2021). Improving grasshopper optimization algorithm for hyperparameters estimation and feature selection in support vector regression. *Chemometrics and Intelligent Laboratory Systems*, 208(3):104196.

Anuradha, C. and Thambusamy, V. (2015). A comparative analysis on the evaluation of classification algorithms in the prediction of students performance. *Indian Journal of Science and technology*, 8(15):974–6846.

Bhatia, N. and Vandana (2010). Survey of nearest neighbor techniques. *International Journal of Computer Science and Information Security*, 8(2):302–305.

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and regression trees.* CRC Press, Boca Raton.

Chakraborty, T., Chakraborty, A., and Mansoor, Z. (2019). A hybrid regression model for water quality prediction. *OPSEARCH*, 56:1167–1178.

Chomboon, K., Pasapichi, C., Pongsakorn, T., Kerdprasop, K., and Kerdprasop, N. (2015). An empirical study of distance metrics for k-nearest neighbor algorithm. In *The 3rd International Conference on Industrial Application Engineering 2015*, pages 280–285.

Clark, P. J. (1952). An extension of the coefficient of divergence for use with multiple characters. *Copeia*, 1952(2):61–64.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27.

Dhanabal, S. and Chandramathi, S. (2011). A review of various k-nearest neighbor query processing techniques. *International journal of Computers and Applications*, 3.

Dice, L. R. (1945). Measures of the amount of ecologic association between species. *Ecology*, 26:297–302.

Elena Deza, M. M. D. (2009). *Encyclopedia of Distances.* Springer Berlin, Heidelberg.

Imandoust, S. B. and Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. *International Journal of Engineering Research and Applications*, 3(5):605–610.

Islam, M. Q. (2017). Estimation and hypothesis testing in multivariate linear regression models under non normality. *Communications in Statistics - Theory and Methods*, 46(17):8521–8543.

Ismael, A. and Benbouziane, M. (2020). Improving harris hawks optimization algorithm for hyperparameters estimation and feature selection in v-support vector regression based on opposition-based learning. *Journal of Chemometrics*, 34(4):3311–3324.

Liu, Y., Ning, Y., and Roozbeh, A. (2021). A gray wolf algorithm for feature and parameter selection of support vector classification. *International Journal of Computing Science and Mathematics*, 13(1):93–102.

Loh, W. and Shih, Y. (1997). Split selection methods for classification trees. *Stat Sin*, 7(4):815–40.

Lopes, N. and Ribeiro, B. (2015). On the impact of distance metrics in instance-based learning algorithms. volume 9117, pages 48–56.

Muhamad Safiih, L., Ramlee, M., Gunalan, S., Zainuddin, N., Zakariya, R., Idris, M., and Khalil, I. (2016). Improved the prediction of multiple linear regression model performance using the hybrid approach: A case study of chlorophyll-a at the offshore kuala terengganu, terengganu. *Open Journal of Statistics*, 06:789–804.

Prasatha, V. B. S., Alfeilat, H. A. A., Hassanat, A. B. A., Lasassme, O., Tarawneh, A. S., Alhasanat, M. B., Eyal, H. S., and Salman (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: A review. *Big Data*, 7.

Punam, M. and Nitin, T. (2015). Analysis of distance measures using k-nearest. *International Journal of Science and Research*, 7(4):2101–2104.

Shih, Y. (2004). A note on split selection bias in classification trees. *Comput Stat Data Anal*, 45(3):457–66.

Shirkhorshidi, A. S., Aghabozorgi, S., and Wah, T. (2015). A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, 10.

Todeschini, R., Ballabio, D., and Consonni, V. (2015). *Distances and other dissimilarity measures in chemometrics*. Encyclopedia of Analytical Chemistry.

Todeschini, R., Consonni, V., Grisoni, F. G., and Ballabio, D. (2016). A new concept of higher-order similarity and the role of distance/similarity measures in local classification methods. *Chemometrics and Intelligent Laboratory Systems*, 157:50–57.

Yao, Z. and Ruzzo, W. L. (2006). A regression-based k nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinformatics*, 7:S11 – S11.

Zhang, Q., Li, D., Ye, Y., and Wang, Q. (2021). A new adaptive algorithm for v-support vector regression with feature selection using harris hawks optimization algorithm. *Journal of Physics: Conference Series*, 1897(1):012057.