IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Reduced Distance Computation *k* Nearest Neighbor Model

Preeti Nair[1*], Indu Kashyap[2]

[1, 2] Dept. of Computer Science and Engineering, Faculty of Engineering and Technology
Manav Rachna International Institute of Research and Studies

[*]*Corresponding Author: preeti.nair84@gmail.com*

DOI: https://doi.org/10.26438/ijcse/v7i5.658666 | Available online at: www.ijcseonline.org

Accepted: 22/May/2019, Published: 31/May/2019

***Abstract*---**In data mining k Nearest Neighbor (k NN) classification is one of a widely applied classification algorithm. The k NN is based on Nearest Neighbor (NN) search algorithm. One of the drawbacks in k (where k stands for the number of NN to be selected) NN method is that whenever a query point is given to be classified it has the propensity to search through each and every data point to get the minimum distance for finding the Nearest Neighbors. This increases the computational complexity when a large query set is given. So to reduce this complexity and improve the performance of k NN, a novel classification model called Reduced Distance Computation k Nearest Neighbor RDCkNN model is introduced in this paper. In RDCkNN two processes are combined, first the data is randomized and then an optimum percentage of subset is drawn from the randomized data hence reducing the overall quantum of distance finding tasks. This subset will act as the training point for the query set for performing k NN classification processes. The performance of RDCkNN is compared with standard k NN in terms of number of distance computed and accuracy. The experiments were employed on standard data sets, data sets with missing values and a very large dataset. It was also compared with a number of other well-known classification models in order to validate its efficacy. The results obtained during the experiments done here shows that the proposed model exponentially outperformed standard k NN as well as other classification models.

Keywords: *k NN, Complexity, Distance Computation, randomization, subset.*

I. INTRODUCTION

The *k* NN classification problem is based on distance similarity matrix. The task of *k* NN is to find closeness between a query instance and its neighbors. To explain further about *k* NN: Given a set T of training data points or instances T={(x_1,y_1),(x_2y_2).....(x_ny_n)} in a space M and an undefined query $q \in M$ then the task of the algorithm is to find the closest point from T to *q* using a distance formula such as Euclidian distance. Then also to select *k* closest points from q and hence majority class in *k* data points will be the category or label for *q* [1], [2]. The *k* NN classification is based on NN search and it has an extremely vital application in many areas of data analysis. It is also important in its ability to be feasible in a wide variety of extant data investigation scenarios. These include but are not limited to image recognition Li et al. [3], K. Q. Weinberger [4],data compression, pattern recognition and classification, machine learning, document retrieval systems, statistics and data analysis. Some other applications found to be feasible include finding the best match for local image features in large data sets Philbin et al. [5]; Zhou et al.[6]; secure transmission Fan et al. [7]; cloud computing Li et al. [8]; model analysis He et al. [9]; and water data analysis Wang et al. [10]. Peddinti et al. [11] An Optimal Route Search Using Spatial Keyword Query. Besides the fact that *k* NN is widely used and easy to implement, it is computationally expensive for working with large data set when it comes to computing distances from the test sample to the stored sample. Many methods have been introduced over the years to improve this issue of distance evaluation actually performed. The main contribution of this paper to propose a novel idea which can handle this issue more efficiently than the existing methods available so far.

The paper is organized as follows section II. Related works section III. Proposed model, section VI. Materials and Implementation, section V. Results and Discussions, section VI. Conclusion and Future work.

II. RELATED WORKS

Over the years a lot of research has been carried out to solve the problem of k NN complexity. Many methods have been introduced in the available research work some of them are discussed here.

O.F. Ertugrul et al. [12] have proposed a method for classifying data with distance similarity measure along with angular dependency between query and sample data in nearest neighbor method. A query is labelled by the data

sample present in a dependency region. The method when compared with other well-known methods showed good results in terms of accuracy and computational costs.

Many different types of approaches such as using different data structure have also been proposed to reduce the computational complexity and improve the Nearest Neighbor (NN) search algorithms some of them are partitioning tree methods, graph methods, hashing methods, probabilistic methods, some of them are discussed here. J.L. Bentley [13] has developed a concept of multidimensional search tree called k-d tree as a data structure for storage of information for high dimensional data. This kind of data structure is very efficient for information retrieval for associative searches, another advantage is that a single data can handle multiple queries. The average runtime for algorithm based on this structure is very less nO(logn). Haibo Hu et. al. [14] have proposed a method called range nearest-neighbor (RNN). It retrieves the nearest neighbor (NN) for every point in a range. The ranges is known as hyper-rectangles. They also introduced EXO-tree to speed up any type of NN query. EXO-tree is a state-of-the-art solution based auxiliary index which can be integrated to existing NN algorithm which gives competent in-memory processing and secondary memory pruning techniques for RNN queries in both 2D and high-dimensional spaces. Yewang Cheng et. al. [15] have proposed new RNN, NN and k NN algorithm based on k-d tree which reduces unnecessary visiting nodes and distance computation. They have used two techniques, first to filter all the query data inside the range for distance computation. Secondly they have used revised k-d tree to unnecessary visiting of the nodes for retrieving the descendants of any node. Fabian Gieseke et. al. [16] have proposed the concept of a buffer k-d tree for efficient k-d tree-based nearest neighbor search on GPUS. The method is designed for processing huge amounts of queries. Batch processing of query which belong to same leaf node gives much more efficient GPU implementation. In this method it also defines randomization which rotates the dataset randomly using tree for simultaneous searches. This speeds up the searches 3 times better with lower error rates. Shikha Bagui et. al. [17] have proposed a method which splits the training data into many buckets. A hashing function decides the number of buckets. Each bucket contains similar objects. In this approach of hashing, maximum probability of hash collision is obtained to preserve the locality sensitiveness. This method was implemented by the authors in hadoop map reduce environment. The use of the locality sensitive hashing approach took less time than conventional method. Kun Ding et. al. [18] proposes k NN based supervised hashing method which learns hash functions by maximizing the k NN accuracy of the Hamming embedded training data.

For larger data a factorized neighborhood representation is used to economically model the neighborhood relationships inherent in training data in order to make it scalable and kernelize the basic model for performance improvement with data which are inseparable. This makes for much less computation and storage cost. Adoni and Indyk [19] have surveyed many nearest neighbor algorithms based on locality sensitive hashing algorithm. In the second part of this paper they have discussed a hashing algorithm which performed notably near optimal among the locality sensitive algorithms.

III. PROPOSED MODEL

Algorithm of the proposed model is summarized in figure 1.

The proposed model is a combined process of two phases. The brief description of these two phases are Randomizing the dataset and Subset the randomized data. As *k* NN classification model is an instance based or lazy learning model in which given training instances are simply stored and model construction happens only if the queries are supplied, so that means no prior generalization of variables are made. It purely depends on similarity of the stored training instances and the test (query), for classifying the instances. Hence the query instances are totally unknown and independent points. Making the testing phase of *k* NN very costly [20]. So instead of taking distances between all of the training data point and q, a novel approach RDC*k*NN is introduced in this paper. In the first phase of RDC*k*NN the dataset is randomized. Randomizing or shuffling method makes a dataset more evenly distributed with respect to its classes. This will reduce the bias of class hence it greatly reduces over fitting of the model. A number of clinical studies done use randomizing techniques to reduce or completely remove bias in selection and protect against accidental bias. This also enhances the possibility of producing analogous groups and allows for the use of probability theory to plot the likelihood of chance as a factor for variance of the end result. [21], [22]. In the second phase an optimum size of subset is fetched from the randomized data. The similarity between *q* and the subset data are calculated using the Euclidean distance measure. An input parameter ζ (can be an integer or real number) decides the percentage of data to be fetched as subset. Depending upon the size of the dataset the ζ is defined. This subset acts as the training data for *q*. The distance computation task and *k* minimum points are selected by means of this subset. (*k* NN is performed on this subset). This reduced the amount of distance computation overheads. In this method missing values are also handled. The data sets with missing values are preprocessed before classification. We have eliminated the instances with missing values with a preprocessing function called clean data (Algorithm 1). The cleaned dataset is then classified using RDC*k*NN (Algorithm 2). The proposed model went well not only with small dataset having binary labels, but also with very large dataset like poker hand, multiple labels data sets like letter, and also data

sets with several features and it took much less distance computation calculations unlike standard *k* NN.

Notations

Let T $\subset \mathbb{R}^d$ where d is the dimension, *n* is the cardinality of T; S $\subseteq$ T;

1. Randomizing the dataset T to T' where T' is the shuffled dataset.
2. ζ is the percentage of the data to be fetched from the training dataset.
3. S is the subset of T and S $\subseteq T'$.
4. Set of Query is denoted by *q*. *Here for experiment we have considered each instance in T as undefined instance for classification. Then the predicted labels and actual labels are compared for determining the performance.*
5. The distances from the query is calculated using the Euclidean distance formula

$$\boldsymbol{d(q,x) = arg\min_{S} \sum_{i=0}^{n} \sum_{x \in S} \| q - x_i \|^2} \quad (1)$$

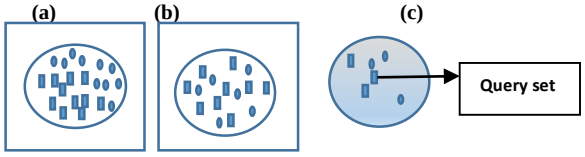


Figure 1. *(a) The original training dataset T (b) Shows the randomized training dataset T′ (c) Subset data selected from the training set which is used to find similarity distance measure from query set.*

When missing value data sets are to be classified **Algorithm1** is applied to the dataset for cleaning missing value instances from the data sets. A new file is created for classification. **(Table 1)**. **Algorithm 2.** Describes the RDC*k*NN process. **(Table 2).**

Table 1. The Algorithm for cleaning missing values in the dataset before classification process.

Algorithm 1:
Data: A training set T=$\{(x_1,y_1), (x_2y_2)\ldots\ldots\ldots(x_ny_n)\}$,$x_i \in \mathbb{R}^d$, n number of indices of instances (0….n-1) in T;
Result: clean dataset;
Repeat
check each column for missing values
if true then delete the instance
else write the instance to a new file;
until all instances are done

Table 2. The Algorithm of the proposed model RDCkNN.

Algorithm 2
Data: data ,k=3, ζ
Result: accuracy, no. of distances computed.
If (dataset T has missing values) then
call Algorithm 1
else
Input dataset and ζ percentage for *S*
The training data is shuffled-> T'
ζ percentage of subset *s* if fetched randomly from the dataset T'
q is the set of query
repeat for each *q*

Equation (1)// calculate the similarity measure from *q* to all instances in S;

Pick 3 nearest neighbor instances from q;
Classify *q* based on the majority number of class in *k*;
Until all queries are done
Calculate performance measure (accuracy)
Calculate the distances computed

IV. MATERIALS AND IMPLEMENTATION PROCEDURE

A. Data sets

In this study several data sets were chosen to validate the RDC*k*NN. The data sets were taken from UCI and KEEL dataset repository. The detailed description of the data sets are shown in the table. Each dataset employed here have different statistical characteristics and complexities. For validating the RDC*k*NN 12 benchmark standard dataset with different properties are taken such as dataset from two class problem to 26 class (letter dataset from KEEL), small to large number of attributes such as (2 to 61) attributes, 4 missing value dataset with missing value percentages 13.63%, 1.98%, 98.1%, 48.39% and a very large dataset-poker hand with more than 10 lakhs instances employed [23]. Table describes the properties of various dataset employed in this study.

Table 3. Dataset Descriptions.

Dataset	No. of Features	No. of classes	No. of Instances	Missing values %
Ionosphere	33	2	351	NMV
Mutagenesis-Atom	12	2	1618	NMV
New Thyroid	5	3	215	NMV
Page-blocks	10	5	5412	NMV
Phoneme	5	2	5404	NMV
Sonar	61	2	208	NMV
Vowels	13	11	990	NMV
Ring	20	2	7400	NMV
Satimage	36	7	6435	NMV
Banana	2	2	5300	NMV
Pima Diabetes	8	2	768	NMV
Letter	16	26	20000	NMV
Mammographic	5	2	961	13.63%
Cleveland	13	5	303	1.98%
Horse-colic	23	2	368	98.1%
Hepatitis	19	2	155	48.39%
Poker Hand	10	10	1025010	NMV

***NMV for no missing values.**

B. Validation procedure

The RDC*k*NN was validated by achieved accuracy against different existing well known classification models such as Standard *k* Nearest Neighbor (*k* NN), Bayesian network

(BN), naïve Bayes (NB), Logistic Regression (LR), Artificial Neural Network (ANN), Simple Logistic Regression (SL), Support Vector Machine (SVM), Decision Tree (DT). The standard *k* NN and RDC*k*NN are also compared with respect to number of distances computed for each dataset.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{n_{correct}}}{\boldsymbol{n}} * \mathbf{100}\ (\%)$$

Where *n* is the total number of samples and $n_{correct}$ is the number of samples which are correctly classified. The higher the accuracy denotes better performance.

Distance Computed = Total no. of distances taken between query and training samples

In classifying data sets a lower number of distance computation needed denotes better performance, less computational complexity, less time consumed and less storage. The computational work has been carried out using Intel core i5-430M processor, 2.26 GHz, 4GB RAM, PC. To determine the performance the process was run for 10-cross validation times. The proposed model was developed in Python 2.7.10. The experiment was also carried out using WEKA 3.8.3 data mining tool [24].

V. RESULTS AND DISCUSSIONS

The main contribution of this study is to achieve a solution for *k* NN complex distance computation as well as higher accuracy than other well-known models. So a combined process of randomizing the dataset and applying an optimum percentage of subset of data fetched from the randomized data. This subset acts as a training set in the proposed model. This subset is thus used for *k* NN classifier for classification. The predicted values are compared with the actual values of each dataset. The results obtained are shown below for standard dataset, missing value dataset and a very large dataset (poker hand). The bracket values present in the table denotes the percentage of data taken as a subset for RDCkNN. As mentioned above ζ is the input parameter for drawing out the percentage of subset that means only ζ% of randomized data are used for the whole process of proposed classification model therefore, reducing the computational complexities.

A. Results of Standard data sets and Missing Value Data sets

Table 4 shows performance of proposed classification model RDC*k*NN and other models in terms of accuracy for various standard dataset. **Figure 5.** Shows the graphical representation of some of the results of standard dataset. Table 5 shows the performance of various missing values data with various percentage of missing values present in it are shown in brackets. **Figure 2.** shows the graphical representation of classification performance with missing values data. Table 6 shows the number of distance computations that took place in each standard dataset with respect to standard *k* NN and RDC*k*NN. The percentage of difference between two models' number of distance computations is also shown in **Table 6. Figure 3** shows the graphical representation of distance computed by *k* NN and RDC*k*NN for standard dataset. The proposed model outperformed other classification models for both standard and missing value dataset.

Table 4. The Accuracy achieved by the proposed model RDC*k*NN and other classification models for standard data sets.

Dataset	***k* NN**	**BN**	**NB**	**LR**	**ANN**	**SL**	**SVM**	**DT**	**RDC*k*NN**
Ionosphere	87.00	90.00	83.00	90.00	91.40	89.00	90.7	86.42	91.00(65% data)
Mutagenesis.	62.75	69.39	64.60	66.76	68.00	67.23	67.7	67.38	81.21(45% data)
Thyroid	91.86	91.86	96.51	91.86	95.34	94.18	83.72	77.90	97.67(60% data)
Page	95.88	92.69	91.54	97.07	96.02	96.53	92.00	93.56	97.70(20% data)
Phoneme	88.62	77.71	76.09	74.23	80.34	74.28	75.99	74.79	91.00 (55% data)
Sonar	87.95	75.90	74.7	72.29	84.34	80.72	78.31	73.50	88.00 (65% data)
Vowels	92.42	52.27	62.88	68.69	75.76	65.90	61.11	15.40	98.08(75% data)
Ring	73.41	97.00	97.8	75.44	70.63	75.20	76.66	59.18	78.84(65% data)
Satimage	90.52	82.13	79.92	86.72	87.61	86.25	87.00	45.00	91.53(30% data)
Banana	86.60	69.10	61.5	56.84	71.99	56.84	55.76	58.35	90.43(50% data)
Pima	73.61	74.92	76.22	79.15	76.55	77.85	78.18	70.03	80.85(70% data)
letter	94.7	74.00	63.20	76.96	81.00	77.00	80.00	45.00	95.80(25% data)

Table 5. The accuracy achieved by the proposed model RDC*k*NN and other existing well-known models for classifying missing value data.

Dataset	***k* NN**	**BN**	**NB**	**LR**	**ANN**	**SL**	**SVM**	**DT**	**RDC*k*NN**
Mammography	74.74	82.81	83.00	81.00	82.00	83.00	81.50	82.80	85.00(57% data)
Cleveland	50.42	57.14	55.46	60.50	57.98	57.98	57.00	52.10	67.46(50% data)
Horse-colic	76.19	77.55	74.15	76.87	77.55	76.87	78.23	76.87	80.12(35% data)
Hepatitis	79.00	91.90	83.87	83.87	83.87	87.00	87.00	77.40	90.12(55% data)

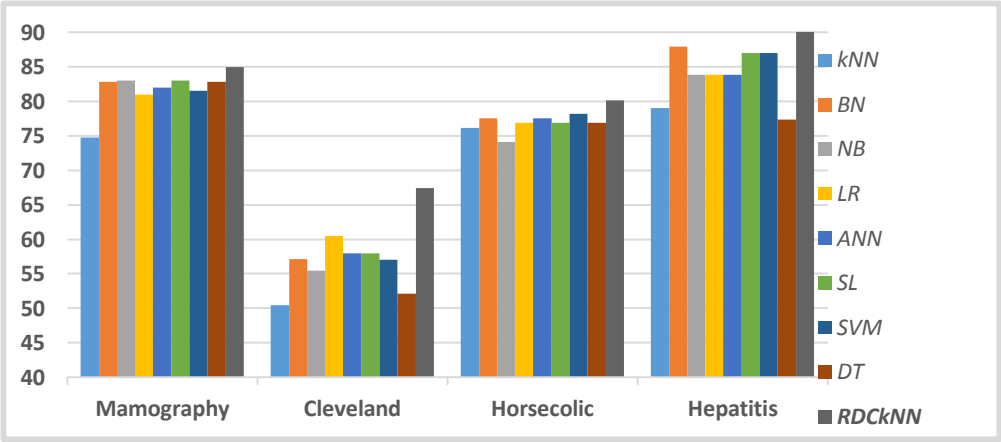


Figure 2. *Shows the accuracy values obtained by proposed model and other models for classifying missing value data.*

Table 6. Shows comparison of number of distance computed during standard *k* NN classification and RDC*k*NN classification for standard datasets.

Dataset	***k* NN**	**RDC*k*NN**	**% age DCD**
Ionosphere	123201	80379	65.24%
Mutagenesis .	2617924	1179822	45.07%
Thyroid	46225	27735	60.00%
Page	29942784	5991840	20.01%
Phoneme	29203216	16066092	55.01%
Sonar	43264	28288	65.38%
Vowel	980100	735570	75.05%
Ring	28090000	14045000	50.00%
Satimage	41409225	12425985	30.01%
Banana	28090000	14045000	50.00%
Pima	589824	413184	70.05%
letter	400000000	100000000	25.00%

As the values for distance computation for each dataset varies some selected data is taken for graphical visualization to explain this. In **figure 3** it is clearly seen that the difference between number of distance computed for standard *k* NN and RDC*k*NN is far larger. On an average there is almost 50% reduction of number of distance computation.

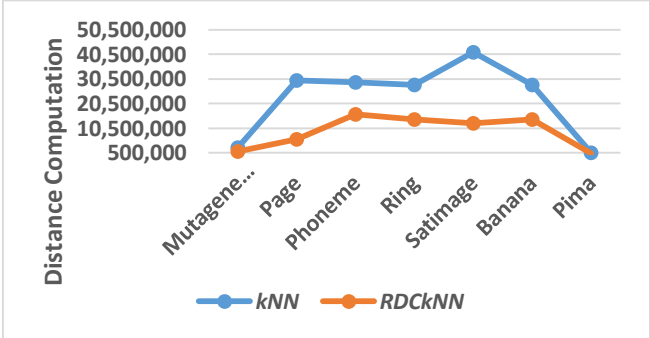


Figure 3. *Number of Distances Computed by RDCkNN and standard k NN for various standard datasets*.

B. Results of Poker Hand Dataset.

Table 7 shows the accuracy obtained by a large dataset- Poker Hand. **Table 8** shows the number of distances computed by standard *k* NN and RDC*k*NN for by a large dataset- Poker Hand and for the clear understanding of differences between the results obtained by standard *k* NN and RDC*k*NN the graphical view of it is shown in **figure 4.**

This dataset was executed on other classification models. The proposed model has outperformed the other classification models just by using ζ=7 where 7% of data out of whole data. In models such as ANN and LR, the results TLE indicates that time limit exceed (TLE) means that it took an extensive time to process the data yet couldn't cover the results.

Table 7. Poker dataset accuracy comparison between other models and RDC*k*NN.

Models	***k* NN**	**BN**	**NB**	**LR**	**ANN**	**SL**	**SVM**	**DT**	**RDC*k*NN**
Accuracy	50	47	50	TLE	TLE	45	40	48	55

Table 8. The Number of distance computed during standard *k* NN and RDC*k*NN Poker hand dataset.

	Distance computed
k NN	1050645500100
RDC*k*NN	63038730006

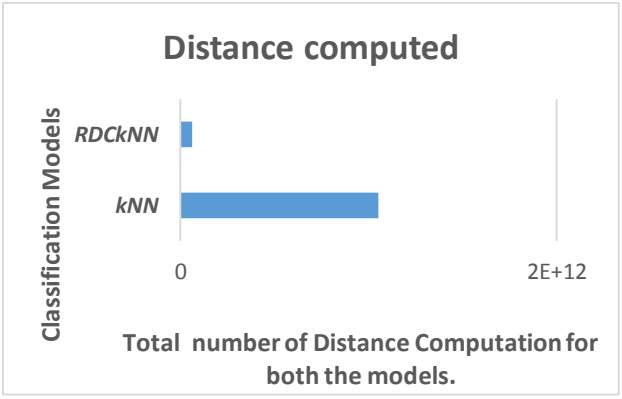
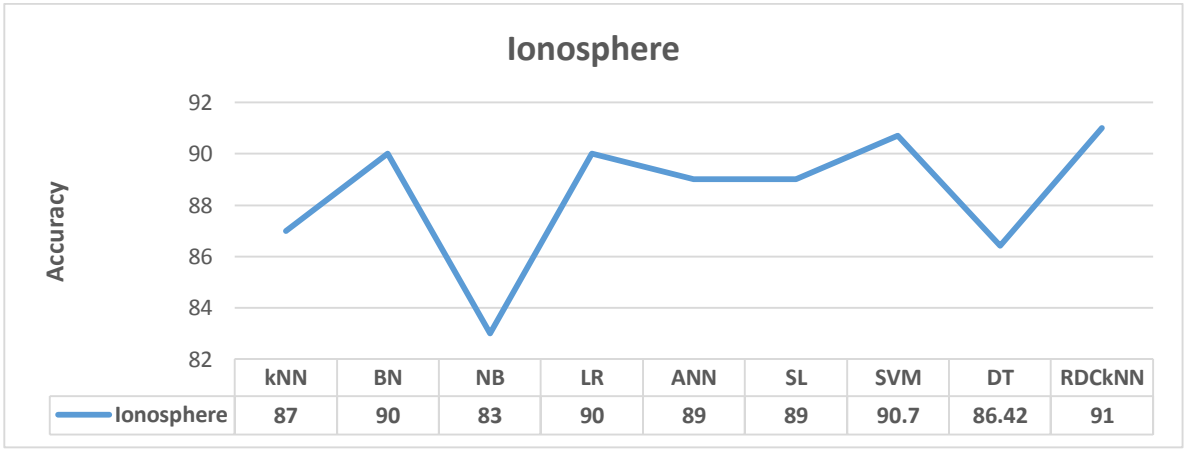
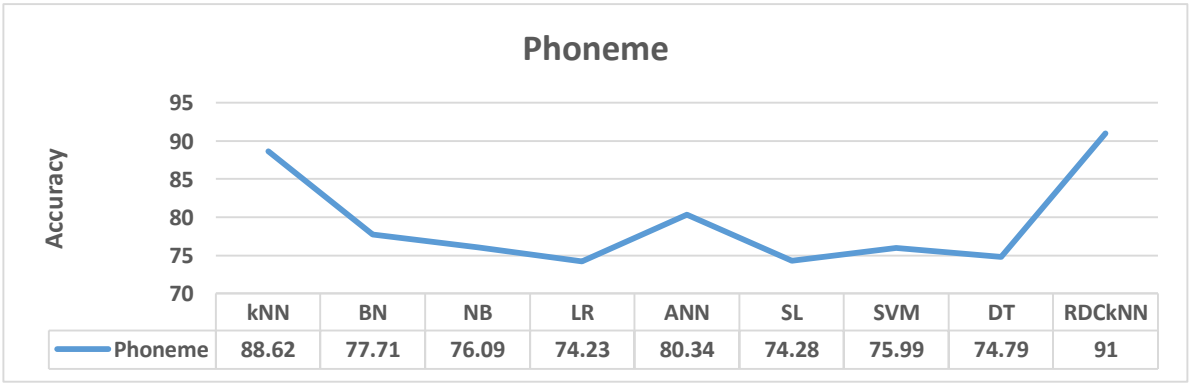
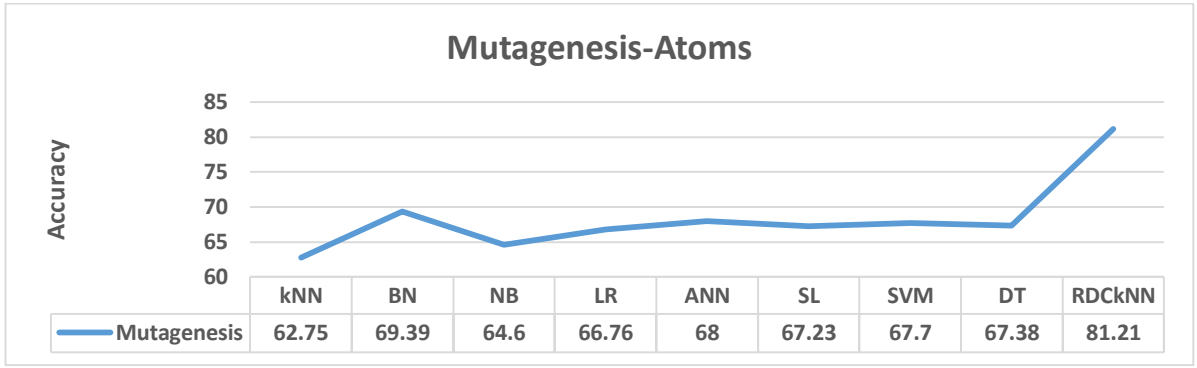
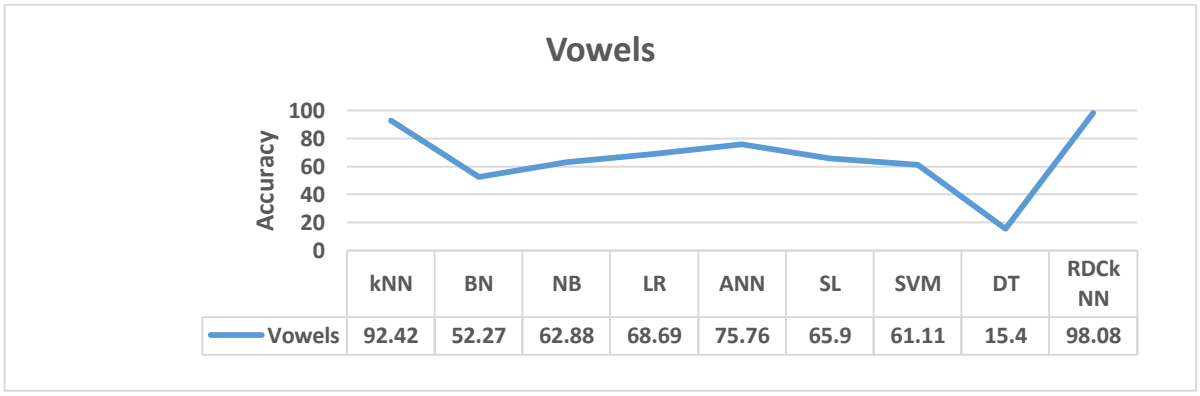


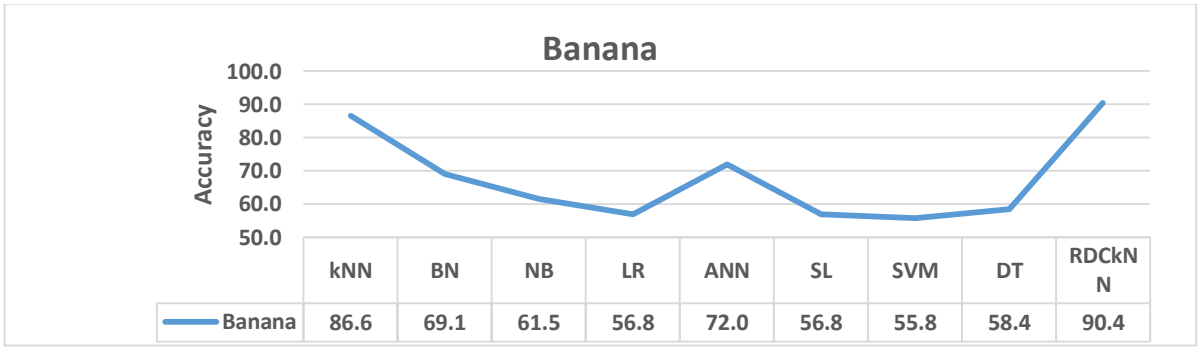
Figure 4. *Shows the number of distances computed by standard k NN and RDCkNN for Poker Hand Dataset.*











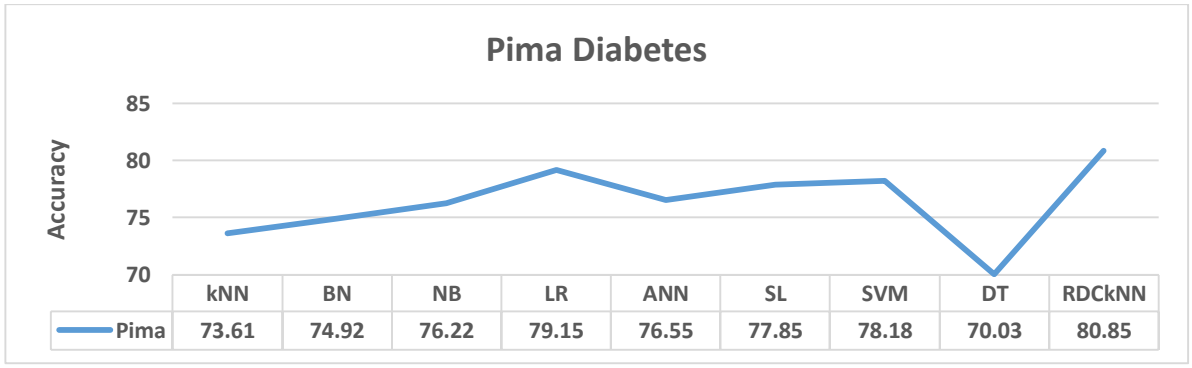


Figure 5. *Shows the graphical visualization of Accuracy obtained by the proposed model RDCkNN and other classification models for various standard datasets.*

C. For optimum RDCkNN parameter

Like other machine learning techniques the better accuracy is achieved by RDCkNN depends upon the input parameters

such as percentage of subset value and k value. Here in the entire experiments we have taken k=3. The parameter depend upon trails and error method or through prior experience. The datasets with large number labels may have to be given reasonably larger percentage of subset values. As larger values will give smoother decision boundaries. The dataset with huge number of instances has been given a smaller percentage of subset values and it should be gradually increased until a better accuracy is obtained, as it consumes a lot of time for computation.

VI. CONCLUSION AND FUTURE WORK

In this study a novel model has introduced called RDC*k*NN, the performance of RDC*k*NN is compared with standard *k* NN in terms of number of distance computed and accuracy. The experiments were employed on standard datasets, datasets with missing values and a very large dataset. It was also compared with number of other well-known classification models in order to validate its efficacy. The results obtained during experiments showed that the proposed model outperformed standard *k* NN as well as other classification models significantly. The proposed model is capable of handling only numeric data. So, it can be further enhanced to handle other kinds of data as well.

REFERENCES

[1]. P. Nair, N. Khatri, N and I. Kashyap, "*A novel technique: ensemble hybrid 1NN model using stacking approach*", International Journal of Information Technology, Springer **doi.org/10.1007/s41870-018-0109-0; 2018.**

[2]. S. Jabeen Begum, B. Swaathi,*"A Survey for identifying Parkinson's disease by Binary Bat Algorithm",* Computer Science and EngineeringVol.**7**, Issue.**2**, pp.**17-23, 2019.**

[3]. Y. Li, G. Wang, L. Nie, Q. Wang, W. Tan. "*Distance metric optimization driven convolutional neural network for age invariant face recognition*". Pattern Recognition; Vol. **75,** Issue **C**, pp **51-62, 2018.**

[4]. K. Q. Weinberger, L. K. Saul, "*Distance Metric Learning for Large Margin Nearest Neighbor Classification*", Journal of Machine Learning Research 10; **207-244; 2009**.

[5]. J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, "*Object retrieval with large vocabularies and fast spatial matching*", In: Computer Vision and Pattern Recognition, IEEE Conference on. IEEE; pp **1–8; 2007.**

[6]. Z. Zhou, M. Dong, K. Ota, G. Wang, L.T. Yang, "*Energy-efficient resource allocation for d2d communications undelaying cloud-RAN-based LTE-A networks*", IEEE Internet of Things Journal; Vol. **3**, Issue **3**, pp **428-38; 2016.**

[7]. L. Fan, X. Lei, N. Yang, T. Q. Duong, G.K. Karagiannidis, "*Secrecy cooperative networks with outdated relay selection over correlated fading channels*", IEEE Transactions on Vehicular Technology, Vol. **66,** Issue **8, pp 7599–603, 2017.**

[8]. J. Li, Z. Liu, X. Chen, F. Xhafa, X. Tan, D.S Wong, "*L-EncDB: A lightweight framework for privacy-preserving data queries in cloud computing*", Knowledge-Based Systems; Vol. **79:18–26; 2015.**

[9]. P. He, Z. Deng, C. Gao, X. Wang, J. Li, "*Model approach to grammatical evolution: deep-structured analyzing of model and representation*" Soft Computing Vo. **21**, Issue **18**, pp **5413–23**, **2017.**

[10]. H. Wang, W. Wang, Z. Cui, X. Zhou, J. Zhao, Y. Li, "*A new dynamic firefly algorithm for demand estimation of water resources*" , Information Sciences; Vol. **438,** pp **95–106, 2018.**

[11]. H.K.N. Peddinti, J.A.Lavanya, G.K. Chakravarthi, "*An Optimal Route Search Using Spatial Keyword Query using Keyword Nearest Neighbor Expansion*", Computer Science and EngineeringVol.**4**, Issue.**3**, pp.**30-33**, **2016**

[12]. O.F. Ertugrul, M.E. Tagluk, "*A novel version of k nearest neighbor: Dependent nearest neighbor*", Applied Soft Computing, Vol. **55, 480-490, 2017.**

[13]. J.L, Bentley, "Multidimensional binary search trees used for associative searching", Commun. ACM Vol. **18**, Issue **9**, pp **509-517, 1975.**

[14]. H Hu, D. L. Lee, "*Range nearest-neighbor query*". IEEE Transactions on Knowledge and Data Engineering; Vol. **18**, Issue **1**, pp **78–91, 2006.**

[15]. Y. Chen, L. Zhou, Y. Tang, J. P. Singh, N. Bouguila, C. Wang, H. Wang, J. Du, "*Fast Neighbor Search By Using Revised K-D Tree*", In-formation Sciences Volume **472**, pp **145-162, 2019.**

[16]. F. Gieseke, J. Heinermann, C.E. Oancea, C. Igel, "*Buffer kd trees: Processing massive nearest neighbor queries on GPUS*" in Proceedings of the 31st International Conference on Machine Learning, pp**. 172–80, 2014.**

[17]. S. Bagui, A. K. Mondal, S. Bagui, "*Using locality sensitive hashing to improve the KNN algorithm in the mapreduce framework*" , Proceedings of the ACMSE Conference Article **No. 32; 2018**

[18]. K. Ding, C. Huo, B. Fan, C. Pan, "*kNN Hashing With Factorized Neighborhood Representation*", IEEE International Conference on Computer Vision (ICCV), pp. **1098-1106; 2015.**

[19]. A. Andoni, P. Indyk, "*Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*", Communications of the ACM; Vol. **51, Issue 1, 117–22; 2008.**

[20]. J. Han, M. Kamber, J. Pei, "Data Mining Concepts and Techniques, Waltham, MA", USA: Third edition, Morgan Kaufmann, pp **423-425, 2012.**

[21]. C. Lim, J. In, "*Randomization in clinical studies*", Korean Journal of Anesthesiol, **DOI: https://doi.org/10.4097/kja.19049; 2019.**

[22]. K. P Suresh, "*An overview of randomization techniques: An unbiased assessment of outcome in clinical research*", Journal of Human Reproductive Sciences Vol. **4** Issue **1**, **2011.**

[23]. J. Alcalá-Fdez, A. Fernandez, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera. "*KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework*", Journal of Multiple-Valued Logic and Soft Computing Vol.**17**, **255-287, 2011.**

[24]. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "*Data Mining: Practical Machine Learning Tools and Techniques*", Morgan Kaufmann, Fourth Edition, pp **35-40, 2016.**

Author's Profile

Preeti Nair has pursed Master of computer applications from Bangalore University (2008). She is currently pursuing Ph.D. from Manav Rachna International Institute of Research and studies (MRIIRS), under the Faculty of Engineering and Technology. Her main research work focuses on Big Data Analytics, Data Mining.

Dr. Indu Kashyap has more than eleven years of experience in teaching. She has done M.Tech and Ph.D. in Computer Science and Engineering. She has several publications to her credit in various leading International and National Journals in the various areas like, Wireless Networking, Databases, and Cloud Computing etc. Currently, she is working as a Professor in the Faculty of Engineering and Technology (FET), MRIIRS and also acting as a Ph.D. coordinator for Engineering Programme. She is a member of many technical committees

