A Review of various KNN Techniques

Pooja Rani

Student, Department of Computer Science and Engineering, GJU S&T, Hisar, India

Abstract: K-Nearest Neighbor is highly efficient classification algorithm due to its key features like: very easy to use, requires low training time, robust to noisy training data, easy to implement, but alike other algorithms it also has some shortcomings as computation complexity, large memory requirement for large training datasets, curse of dimensionality which can't be ignored and it also gives equal weights to all attributes. Many researchers have developed various techniques to overcome these shortcomings. In this paper such techniques are discussed. Some of these techniques are structure based, like R-Tree, R*-Tree, Voronoi cells, Branch & Bound algorithm and these algorithms overcome computation complexity as well as also help to reduce the searching time for neighbors in multimedia training datasets. Some techniques are non-structure based like Weighted KNN, Model based KNN, distance based KNN, Class confidence weighted KNN, Dynamic weighted KNN, Clustering based KNN, and Pre-classification based KNN, and these algorithms reduce memory limitation, curse of dimensionality and time complexity.
Key words: K-Nearest Neighbor, Structure based KNN, Non-Structure based KNN, Weighted KNN.

I. INTRODUCTION

A. Data Mining

Each and every day the human beings are using a vast amount of data in different fields which may be generated during many operations in different place of countries by different small as well as big organizations. This data can be present in several formats like document, graphic, text, number, figures, may be audio or video and hypertext. This data is of no use if it doesn't provide useful information. To make this large amount of data useful and interesting we are required to process and scan useful information so that we can make desired decisions from this data. To analyze, manage and make a decision from this huge amount of data we need a technique called data mining. Data mining can be defined as a process of dig out useful information from huge volume of data and it is a powerful technology with great potential to help organizations emphasis on the most important information in their data depositories. Data mining tools predict future trends and behaviors which helps organizations to make proactive knowledge-driven decisions [1]. It is a multidisciplinary field, sketch work from areas including database technology, machine learning, statistics, pattern recognition, information retrieval, neural nets, knowledge-based systems, artificial intelligence, high-performance computing, and data conception[2]. Data mining has applications in multiple spheres like in retail industry, telecommunication sector, criminal investigation, finance, sales, medicine, marketing, banking, healthcare and insurance, customer segmentation and research analysis [3] [1]. Throughout the years several techniques have been developed to extract interesting and hidden knowledge for various required purposes from the provided dataset. Some famous methods of data mining include On-Line Analytical Processing (OLAP), Classification, Regression, Clustering, Time Series Analysis, Prediction, Summarization, Sequence Discovery, Spatial Mining, Web mining, Association Rule Mining etc.[1] [4]. In data mining, one of the most important tasks is classification which is a supervised learning because target class is predefined. Classification is the task of assigning each record to one of the several predefined categories or classes [5]. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks. Data classification is a two-step process, consisting of a learning step (where a classification model is constructed) and a classification step (where the already constructed model is used to predict class labels for the given data) [6]. Training set is the one whose class labels are known and along with the learning algorithm they are used to construct a classification model. The classification model also called a classifier is then applied to the Test set whose class labels are unknown [7]. Each training instance consists of a number of rows which are called records, tuples or instances with corresponding class labels. A class label must be discrete. An instance is characterized by a set of attribute vector [8]. There are different classification methods like decision tree induction, Bayesian classification, rule-based classification, classification by back-propagation, support vector machines, and classification based on association rule mining are all the examples of eager learners [2] [9]. Eager learners, when given a set of training tuples, will construct a generalization (i.e., classification) model before receiving new (e.g., test) tuples to classify. However, there are some classification algorithms which do not make a model, but make the classification decision by comparing the test set with the training set each time they perform classification. These algorithms are known as instance-based learning algorithms. These algorithms are also known as lazy learners because they simply store the instances and they do not do any work on instances until it is given a test tuple. Lazy learners do less work when a training tuple is presented and more work

while making a classification or prediction. K-nearest-neighbor classifiers and case-based reasoning classifiers are two such examples of lazy learners.

B. K-Nearest Neighbor Classifier

The K-Nearest-Neighbor (KNN) classifier was first described in the early 1950s. KNN is applicable in many fields such as pattern recognition, mining of text, finance, agriculture, medicine etc [10]. KNN is a non-parametric algorithm. It does not require any prior knowledge about dataset and it assumes that instances in the datasets are independently and identically distributed, therefore, the instances which are close to each other have the same classification [5]. KNN is also known as lazy learner because in its learning phase it simply stores all the training tuples given to it as input without performing any calculations or does only a little processing and waits until a test tuple is given to it to classify. All the computations or processing apply at the time of classification of a test tuple. It classifies the unknown tuple by comparing it with training tuples that are similar to it. When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the 'k' nearest-neighbors or closest tuples to the unknown tuple. "Closeness" is defined in terms of a distance metric [2]. To find the 'k' closest training tuples to the unknown tuple various distance metrics are used like Euclidean distance, Minkowski distance, Manhattan distance. The standard Euclidean distance d(x, y) is often used as the distance function [11].

Euclidean distance function is given below which calculate the distance between two tuples x and y.

$$d(x, y) = \sqrt{\sum_{i=1}^{n}\left(a_i(x) - a_i(y)\right)^2}$$

Where 'm' is the number of total attributes and 'a' is the value of attribute in instances x (test tuple) and y.

When given an unknown tuple, a KNN classifier searches the pattern space for the 'k' training tuples that are closest to the unknown tuple and assigns the most common class of x's 'k' nearest neighbors as shown in equation given below.

$$c(x) = \arg\max_{c \varepsilon C} \sum_{i=1}^{k} \delta\big(c, c(y_i)\big)$$

Where $y_i = y_1, y_2 \ldots y_k$ are the 'k' nearest neighbors of x, k is the number of neighbors and $\delta\big(c, c(y_i)\big)$=1 if c= $c(y_i)$ and $\delta\big(c, c(y_i)\big)$=0 otherwise.

The choice of 'k' is very important in building the KNN model. It is one of the most important factors of this model that can strongly influence the quality of predictions. For any given problem, a small value of 'k' will lead to a large variance in predictions. Alternatively, setting 'k' to a large value may lead to a large model bias. Therefore, 'k' should be set to a value large enough to minimize the probability of misclassification and small enough (with respect to the number of tuples in the dataset) so that the 'k' nearest point is close enough to the query point. When 'k'=1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space. In binary (two class) classification problems, it is helpful to choose 'k' to be an odd number as this avoids ties between votes [21].The value of 'k' should not be chosen as a multiplier of the number of classes. It avoids ties when number of classes is greater than two.

C. K-Nearest Neighbor Tecniques

K-Nearest neighbor techniques can be classified into two categories (1) Structure based techniques and (2) Non-structure based techniques.

1) Structure based KNN Techniques: Traditional KNN consumes too much time in searching 'k' nearest neighbors when high dimensional or multimedia datasets are given to it. Another problem is memory limitation in case of large datasets. To overcome this type of limitations structure based techniques are developed. A Voronoi cells based method which gives high efficiency for uniformly distributed as well as for real datasets was developed by Berchtold *et al* [12]. Searching of 'k' nearest neighbors for Geographic information systems requires different search algorithm as compared to those which are used in search location or range queries. To solve these queries Roussopoulos *et al* proposed an efficient Branch and Bound R-Tree traversal algorithm [13]. Computer Aided Design and Geo-data applications use spatial datasets and to handle spatial data efficiently an index mechanism is required so that data items can be retrieved quickly according to their spatial locations. Antonn Guttmann proposed a dynamic index structure called R-Tree. It indexes the data points and by using depth-first traversal of the tree entries in the nodes are orders and pruned with some heuristics approaches. [14]. King Lum Cheung and

Ada Wai-chee Fu enhanced this R-Tree algorithm to R*-Tree which preserve the pruning power and reducing the computational cost [15].

2) *Non- Structure based KNN Techniques:* Traditional KNN is very simple, highly efficient, easy to implement, comprehensive, but it has some disadvantages like curse of dimensionality, distance metric, computational cost, deciding the value of 'k' parameter, biasing toward majority class, equal weight to all attributes etc. Non-structure based KNN techniques are used to meet these problems. To reduce the equal effect of those attributes which doesn't give any important information for data classification some weighting techniques are used where useful attributes give more weights than others irrelevant attributes. Two variations of weighting techniques devised by M. E. Syed, among them one was the normal weighting in which a single weight was calculated, while the other was class-wise weighting in which different weights were calculated for different classes for every attribute [5]. The class-wise weighting is very useful in recognizing the smaller classes and improving the rate of accuracy for smaller classes. Dynamic K-Nearest Neighbors Naïve Bayes with Attribute Weighted method to improve KNN's accuracy is suggested by L. Jiang [6]. Xiao and Ding suggested a weighting method based on weighted entropy of attribute value which enhance the accuracy of classification [16]. Li *et al* proposed an attribute weighting method where the irrelevant attributes are reduced and weight is assigned by the method of sensitivity which improves the efficiency of algorithm [17]. A Distance-weighted K-Nearest-Neighbor rule developed by Lan Du *et al* usin*g* the dual distance-weighted function [18]. It is robust to different choices of 'k' to some degree, and yields good performance with a larger optimal 'k' value, compared to the other state-of-art KNN-based methods. An instance weighting technique where the instances get more weight on the basis of their closeness to the new instance than such neighbors that are far away from the new instance had been presented by Schliep Hechenbichler [11]. It uses the kernel functions to give the weights to instances based on the distance from the unknown instance. By using this method the effect of 'k' parameter can be reduced up to some extent. K Maryam devised a new algorithm based on dynamic weighting to improve the classification accuracy of the KNN algorithm [19]. A weighting method based on information gain and extension relativity was proposed by Baobao *et al* to improve the conventional KNN algorithm [20]. This method improves the anti-jamming ability and accuracy of the KNN algorithm. The computing time is also reduced. In some techniques Eager learning and Lazy learning are combined together to improve the efficiency of the classifier. Liangxiao Jiang *et al* proposed a dynamic KNN Naïve Bayes with Attribute weighted algorithm. It learns the best value of 'k' eagerly at training time to fit the training data and at classification time for each given test instance, a local naïve Bayes within best 'k' nearest neighbor is lazily built. It significantly outperforms[21]. There are different ways to assign weights to features. A chi square statistical test based feature weighting method, used in conjunction with KNN classifier, was developed by D. P. Vivencio *et al* [22] .This weighting method has good performance in datasets with a large number of irrelevant features. A new method called Hybrid dynamic k-nearest-neighbor, distance and attribute weighted for classification was devised by Jia Wu *et al* [23]. KNN algorithm based on information entropy weighting of attribute was proposed by Shweta Taneja *et al* [24] . It improves the accuracy of classification but spent more time in classification when many categorical attributes are given. Presence of irrelevant information in the data set reduces the speed of classifier and quality of learning. The techniques of feature selection, irrelevant feature reduction reduces the amount of data needed and execution time. Rashmi Agrawal devised a method of relevant feature selection to select the relevant features and removes irrelevant features of the dataset automatically [25]. It gives better accuracy and also reduces the execution-time. By using robust neighbors in training data performance of KNN can be improved, was devised by Parvin *et al* [26]. This method employs a kind of preprocessing on training data. It adds a new value named "Validity" to train samples. A weighted KNN model-based data reduction and classification algorithm finds some more meaningful representatives to replace the original dataset for further classification proposed by Xuming Huang *et al* [27]. This method uses a similarity-based weighting technique to overcome the influence of different noisy data to representative tuple.

3) *Some other Variations in KNN:* While dealing with highly imbalanced data, the main drawback in standard KNN algorithms is that the class with more frequent samples tends to dominate the neighborhood of a test instance in spite of distance measurements, which leads to suboptimal classification performance on the minority class. To handle the imbalanced datasets Liu and Chawla defined a class confidence weighting strategy [28]. It uses the probability of attribute values of given class labels to weight prototypes in KNN. It reduces the biasing towards majority class prediction in traditional KNN. A single-class algorithm called Class Conditional Nearest Neighbor Distribution (CCNND), which mitigates the effects of class imbalances through local geometric structure in the data was addressed by Evan Kriminger *et al* [29]. CCNND maintains high sensitivity to the minority class. A Prototype Reduction approach for the K-Nearest-Neighbor algorithm learns the weighted similarity function for a KNN classifier by maximizing the leave-one-out cross-validation accuracy described by Tao Yang *et al* [30]. It

reduces the storage requirement and enhances the online speed while retaining the same level of accuracy for a K-Nearest Neighbor (KNN) classifier. One useful adaptive KNN classification algorithm which does not require a fixed value of nearest neighbors was proposed by Stefanos *et al* [31]. For this purpose it uses three early-break heuristics which use different conditions to enforce an early-break. It would save computational time of the classifier. Elena Marchiori investigated a decomposition of RELIEF (A popular method to generate a feature weight vector from a given training set, one weight for each feature) into class dependent feature weight vectors, where each vector describes the relevance of features conditioned to one class[32]. If we are using only one class to estimate the relevance of features is not beneficial for the classification performance. This work provides initial insights and results about the advantages of using feature relevance in a way that depends on the single classes. Correlation based K-Nearest-Neighbor algorithm which makes data classification based on the correlation calculation and uses a modified probability to improve the computational speed and prediction accuracy was described by Xinran Li [33]. It decreases the complexity of arithmetic and saves the accounting time. A cluster-based tree algorithm to accelerate k-nearest neighbor classification without any presumptions about the metric form and properties of dissimilarity measures was given by Bin Zhang et al [34].It improves the search efficiency with minimal accuracy loss. Zhao and Chen made a comparative study on weighted KNN classifiers for model selection. Attribute weighted KNN has better classification performance but it takes too long run time. When the hardware configuration is different, the algorithm runtime cost is different[35]. Jigang *et al* devised an adaptive K-NN rule, which is simply the K-NN rule with the conventional distance measure, be it the Euclidean or Manhattan metric, divided by the smallest distances from the corresponding training examples to training examples of different classes [36]. It significantly improves the performance of the KNN rule. Batista and Silva provides some useful information by their empirical study that Heterogeneous distance function is used to handle both qualitative and quantities attributes. The weighting functions cannot be applied when 'k'=1 and the performance of KNN will decreases for the higher values of 'k' [37]. The two aspects made by Bo *et al* (1) to improve the efficiency of classification by moving some computations from classification period to the training period, which leads to the great descent of computational cost (2) to improve the accuracy of classification by the contribution of different attributes and to obtain the optimal attribute weight sets using the quadratic programming method [38]. Huahua Xie suggested a novel improved KNN algorithm to reduce the time complexity in KNN with a better performance than traditional KNN [39]. A pre-classification is conducted and then the training set is divided into several parts, among which ambiguous instances are wiped out of the training set with a threshold. This method greatly reduces the time cost of the algorithm, but it also has better performance than traditional KNN. To overcome the sensitivity of the choice of the neighborhood size *k* and improving the classification performance Jianping Gou proposed a classifier which mainly employs the dual weighted voting function to reduce the effect of the outliers in the *k* nearest neighbors of each query object [40].This classifier always outperforms than other classifiers, especially in the case of a large value of neighborhood size 'k' .

4) *KNN with Evolutionary Computation:* James and Davis described a method for hybridizing a genetic algorithm and a K-Nearest-Neighbors classification algorithm. It uses the genetic algorithm and a training data set to learn real-valued weights associated with individual attributes in the data set. It requires computational capabilities above that of the KNN algorithm, but achieves improved classification performance in a reasonable time [41]. A fuzzy KNN decision rule and a fuzzy prototype decision rule with three methods for assigning membership values to the sample sets was devised by Keller *et al* [42].When an incorrectly classified sample will not have a membership in any class close to one while a correctly classified sample possesses a membership in the correct class close to one. Fuzzy nearest prototype classifier is computationally attractive and also produces membership assignments that are desirable. We also have an another technique which is an extension of Fuzzy based KNN where clusters are obtained at preprocessing step and the membership of the training data set is computed in reference with the centroid of the clusters addressed by Taneja *et al* [43]. Fuzzy-Rough NN classification approach presented by Haiyun Bian and Lawrence Mazlack performs better under partially exposed and unbalanced domain [44]. Selective Neighborhood based Naïve Bayes computes different distance neighborhoods of the input new object, lazily learns multiple Naïve Bayes classifiers and uses the classifier with the highest estimated accuracy to make decisions, was devised by Zhipeng XIE *et al* [45]. It improves the accuracy and computational efficiency.

II. CONCLUSION

In this paper we briefly reviewed the various KNN techniques. This review would be helpful to researchers to focus on the various issues of KNN like curse of dimensionality, memory limitations, computational cost, too much time consumption in searching 'k' nearest neighbors for a test tuple in large and multimedia training datasets, slow at classification process etc. Most of the previous

studies on KNN applications in various fields use the variety of data types like spatial, categorical, continuous, text, graphics, geo-data, moving objects etc. For these different data types different techniques are developed by many researchers. We conclude that the factors which don't let the work to be fully successful are curse of dimensionality, value of 'k' parameter and biasing towards majority class etc. The researchers were successful in making the outcomes of their research work fruitful but the problem of getting most accurate and general KNN classification model is still there. Attribute weighting, distance weighting and weighted class probability estimations methods are most commonly used to improve its performance.

REFERENCES

[1]   N. Padhy, "The Survey of Data Mining Applications and Feature Scope," *Int. J. Comput. Sci. Eng. Inf. Technol.*, vol. 2, no. 3, pp. 43–58, Jun. 2012.

[2]   J. Han and M. Kamber, *Data mining: concepts and techniques*, 2nd ed. Amsterdam ; Boston : San Francisco, CA: Elsevier, 2006

[3]   S. Bagga and G. N. Singh, "Applications of Data Mining," *Int. J. Sci. Emerg. Technol. Latest Trends*, vol. 1, no. 1, pp. 19–23, 2012.

[4]   S. Sethi, D. Malhotra, and N. Verma, "Data Mining: Current Applications & Trends," *Int. J. Innov. Eng. Technol. IJIET*, vol. 6, no. 14, pp. 667–673, Apr. 2006.

[5]   M. E. Syed, "Attribute weighting in k-nearest neighbor classification," University of Tampere, 2014.

[6]   L. Jiang, H. Zhang, and Z. Cai, "Dynamic k-nearest-neighbor naive bayes with attribute weighted," in *International Conference on Fuzzy Systems and Knowledge Discovery*, 2006, pp. 365–368.

[7]   I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques*, 2nd ed. Amsterdam ; Boston, MA: Elsevier, 2005.

[8]   L. Jiang, Z. Cai, D. Wang, and S. Jiang, "Survey of improving k-nearest-neighbor for classification," presented at the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007, vol. 1, pp. 679–683.

[9]   R. Kumar and R. Verma, "Classification algorithms for data mining: A survey," *Int. J. Innov. Eng. Technol. IJIET*, vol. 1, no. 2, pp. 7–14, 2012.

[10]   S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 605–610, 2013.

[11]   K. Hechenbichler and K. Schliep, "Weighted k-nearest-neighbor techniques and ordinal classification," Institue for statistik sonderforschungsbereich, 386, 2004.

[12]   S.Berchtold, Bernhard Ertl, Keim, and D.A., "A Fast nearest neighbor search in high-dimensional space," presented at the Proceedings of the International Conference on Data Engineering, 1998, vol. 31, pp. 209–218.

[13]   Roussopoulos, Kelley, and Vincent, "Nearest Neighbor Queries," in *Proceedings of the ACM SIGMOD International conference on Management of Data*, 1995, pp. 71–79.

[14]   Antonn Guttman, "R -Trees: A dynamic index structure for spatial searching," in *In proceedings of the ACM SIGMOD International Conference on Management of Data*, Berkeley, 1984, pp. 47–57.

[15]   King Lum Cheung and Ada Wai-chee Fu, "Enhanced Nearest Neighbour Search on R-Tree," in *ACM SIGMOD Record 27(3)*, 1998, pp. 16–21.

[16]   X. Xiao and H. Ding, "Enhancement of K-nearest neighbor algorithm based on weighted entropy of attribute value," presented at the Fourth International Conference on Advanced & Communication Technologies, 2012, pp. 1261–1264.

[17]   Z. Li, Z. Chengjin, X. Qingyang, and L. Chunfa, "Weigted-KNN and its application on UCI," presented at the International Conference on Information and Automation, 2015, pp. 1748–1750.

[18]   J. Gou, L. Du, Y. Zhang, T. Xiong, and others, "A new distance-weighted k-nearest neighbor classifier," *J Inf Comput Sci*, vol. 9, no. 6, pp. 1429–1436, 2012.

[19]   Maryam Kuhkan, "A Method to Improve the Accuracy of K-Nearest Neighbor Algorithm," *Int. J. Comput. Eng. Inf. Technol. IJCEIT*, vol. 8, no. 6, pp. 90–95, Jun. 2016.

[20]   W. Baobao, M. Jinsheng, and S. Minru, "An enhancement of K-Nearest Neighbor algorithm using information gain and extension relativity," presented at the International Conference on Condition Monitoring and Diagnosis, 2008, pp. 1314–1317.

[21]   Liangxiao Jiang, Harry Zhang, and Zhihua Cai, "Dynamic k-nearest-neighbor naive bayes with attribute weighted," presented at the International Conference on Fuzzy Systems and Knowledge Discovery, 2006, pp. 365–368.

[22]   D. P. Vivencio, E. R. Hruschka, M. do Carmo Nicoletti, E. B. dos Santos, and S. D. Galvao, "Feature-weighted k-nearest neighbor classifier," presented at the Foundations of Computational Intelligence,2007, 2007, pp. 481–486.

[23]   J. Wu, Z. hua Cai, and S. Ao, "Hybrid dynamic k-nearest-neighbour and distance and attribute weighted method for classification," *Int. J. Comput. Appl. Technol.*, vol. 43, no. 4, pp. 378–384, 2012.

[24]   S. Taneja, C. Gupta, K. Goyal, and D. Gureja, "An Enhanced K-Nearest Neighbor Algorithm Using Information Gain and Clustering," presented at the Fourth International Conference on Advanced Computing & Communication Technologies, 2014, pp. 325–329.

[25]   R. Agrawal, "A Modified K-Nearest Neighbor Algorithm Using Feature Optimization," *Int. J. Eng. Technol.*, vol. 8, no. 1, pp. 28–37, Mar. 2016.

[26]   Hamid Parvin, Hosein Alizadeh, and Behrouz Minaei-Bidgoli, "MKNN: Modified K-Nearest Neighbor," *OALib J.*, Oct. 2008.

[27]   X. Huang, G. Guo, D. Neagu, and T. Huang, "Weighted kNNModel-Nased Data Reduction and Classification," presented at the Fourth International Conference on Fuzzy Systems and Knowledge Discovery, 2007, 2007, vol. 1, pp. 689–695.

[28]   W. Liu and S. Chawla, "Class confidence weighted knn algorithms for imbalanced data sets," presented at the Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2011, pp. 345–356.

[29]   E. Kriminger, J. C. Príncipe, and C. Lakshminarayan, "Nearest neighbor distributions for imbalanced classification," presented at the International Joint Conference on Neural Networks, 2012, pp. 1–5.

[30]   T. Yang, L. Cao, and C. Zhang, "A novel prototype reduction method for the K-nearest neighbor algorithm with K≥ 1," presented at the Conference on Knowledge Discovery and Data Mining, 2010, pp. 89–100.

[31]   S. Ougiaroglou, A. Nanopoulos, A. N. Papadopoulos, Y. Manolopoulos, and T. Welzer-Druzovec, "Adaptive k-nearest-neighbor classification using a dynamic number of nearest neighbors," presented at the East European Conference on Advances in Databases and Information Systems, 2007, pp. 66–82.

[32]   E. Marchiori, "Class dependent feature weighting and k-nearest neighbor classification," presented at the International Conference on Pattern Recoginition in Bioinformatics, 2013, pp. 69–78.

[33]   X. Li and C. Xiang, "Correlation-based K-nearest neighbor algorithm," presented at the 3rd International Conference on Software Engineering and Service Science, 2012, pp. 185–187.

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

[35]   Ming Zhao and Jingchao Chen, "Improvement and Comparison of Weighted K-Nearest-Neighbors Classifiers for Model Selection," *J. Softw. Eng.*, pp. 1–10, 2016.

[36]   J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognit. Lett.*, vol. 28, no. 2, pp. 207–213, Jan. 2007.

[37]   G. Batista and D. F. Silva, "How k-nearest neighbor parameters affect its performance," presented at the Conference on Argentine Symposium on Artificial Intelligence, 2009, pp. 1–12.

[38]   B. Sun, J. Du, and T. Gao, "Study on the Improvement of K-Nearest-Neighbor Algorithm," presented at the International Conference on Artificial Intelligence and Computational Intelligence, 2009, pp. 390–393.

[39]   H. Xie, D. Liang, Z. Zhang, H. Jin, C. Lu, and Y. Lin, "A Novel Pre-Classification Based kNN Algorithm," presented at the 16th International Conference on Data Mining Workshops, 2016, pp. 1269–1275.

[40]   J. Gou, T. Xiong, and Y. Kuang, "A Novel Weighted Voting for K-Nearest Neighbor Rule," *J. Comput.*, vol. 6, no. 5, pp. 833–840, May 2011.

[41]   J. D. Kelly Jr and L. Davis, "A Hybrid Genetic Algorithm for Classification.," presented at the IJCAI, 1991, vol. 91, pp. 645–650.

[42]   James M. Keller, Michael R. Gray, James A. Givens, and JR, "A Fuzzy k-Nearest Neighbor Algorithm," *IEEE Trans. Syst. MAN Cybern.*, vol. 15, no. 4, pp. 558–585, Jul. 1985.

[43]   S. Taneja, C. Gupta, S. Aggarwal, and V. Jindal, "MFZ-KNN—A modified fuzzy based K nearest neighbor algorithm," presented at the International Conference on Cognitive Computing and Information Processing, 2015, pp. 1–5.

[44]   H. Bian and L. Mazlack, "Fuzzy-rough nearest-neighbor classification approach," presented at the 22nd International Conference of the North American on Fuzzy Information Processing Society, 2003, pp. 500–505.

[45]   Z. Xie, W. Hsu, Z. Liu, and M. L. Lee, "Snnb: A selective neighborhood based naive Bayes for lazy learning," presented at the Conference on Knowledge Discovery and Data Mining, 2002, pp. 104–114.

INTERNATIONAL JOURNAL
FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 ◎ (24*7 Support on Whatsapp)