Journal of Information Technology Management

Print ISSN: 2008-5893

Online ISSN: 2423-5059

A Novel Scheme for Improving Accuracy of KNN Classification Algorithm Based on the New Weighting Technique and Stepwise Feature Selection

Saeid Sheikhi

MSc, Department of Computer, Gorgan Branch, Islamic Azad University, Gorgan, Iran. E-mail: s.sheikhi@outlook.com

Mohammad Taghi Kheirabadi *

*Corresponding Author, Assistant Prof., Department of Computer, Gorgan Branch, Islamic Azad University, Gorgan, Iran. E-mail: mtkheirabadi383@gmail.com

Amin Bazzazi

Assistant Prof., Department of Computer, Gorgan Branch, Islamic Azad University, Gorgan, Iran. E-mail: bazzazi@gorganiau.ac.ir

Abstract

K nearest neighbor algorithm is one of the most frequently used techniques in data mining for its integrity and performance. Though the KNN algorithm is highly effective in many cases, it has some essential deficiencies, which affects the classification accuracy of the algorithm. First, the effectiveness of the algorithm is affected by redundant and irrelevant features. Furthermore, this algorithm does not consider the differences between samples, which led the algorithm to have inaccurate predictions. In this paper, we proposed a novel scheme for improving the accuracy of the KNN classification algorithm based on the new weighting technique and stepwise feature selection. First, we used a stepwise feature selection method to eliminate irrelevant features and select highly correlated features with the class category. Then a new weighting method was proposed to give authority value to each sample in train dataset based on neighbor categories and Euclidean distances. This weighting approach gives a higher preference to samples that have neighbors with close Euclidean distance while they are in the same category, which can effectively increase the classification accuracy of the algorithm. We evaluated the accuracy rate of the proposed method and analyzed it with the traditional KNN algorithm and some similar works with the use of five real-world UCI datasets. The experiment results determined that the proposed scheme (denoted by WAD-KNN) performed better than the traditional KNN algorithm and considered approaches with the improvement of approximately 10% accuracy.

Keywords: Data mining, KNN algorithm, Classification algorithm, Weighted KNN.

DOI: 10.22059/jitm.2020.296305.2455

Introduction

In recent years, the rapid development of the internet and information technology (IT) has produced a large amount of data in many areas. The complexity of storing and manipulating this amount of data has received significant attention from the information industries, companies, and many individuals to use data mining to process and extract valuable information. The information obtained in this manner can have various applications in market research, production control, and scientific research. In general, data mining is the method of analyzing large datasets to discover patterns and relationships between a variety of data to make decisions. Data mining includes several techniques used in mining operations and is one of the essential techniques for analyzing large datasets in data classification (Cheng et al, 2018).

Data classification, as one of the widely used data mining techniques, assigns categories to different objects identified by a set of features. Several researchers have developed new theories and solutions to solve classification problems in recent years (Abbasi et al, 2019). Among various algorithms for the classification of different varieties of data, and the k-nearest neighbors (KNN) algorithm is one of the essential nonparametric algorithms (Biswas et al, 2017; Serpen et al, 2018).

The KNN algorithm is a supervised classification algorithm that produces classification rules using training data. This algorithm predicts the class of an object according to the distance of the object to other records. Then, it forecasts the class of the test object based on the category of the majority of its neighbors (Wu et al, 2018). Despite several advantages of KNN algorithm, the traditional KNN algorithm still has some deficiencies, which are explained below:

A. The high complexity of computation: When the traditional KNN algorithm is applied to find the closest samples to a test one by calculating all similarities between training data. This is a long process that increases computations and process time. Also, this leads to lower efficiency in predicting a category (Kotenko et al, 2018).

B. Dependence on the training set: The KNN algorithm does not use additional information to define classification rules. Based on this, instances are classified according to the produced training data. In this way, the algorithm relies strongly on training data. For example, the algorithm must recalculate all data when small changes occur in a training set (Kotenko et al, 2018).

C. No difference between the data samples: All training samples are equal in the traditional KNN algorithm. Moreover, it does not distinguish between different training samples that do not correspond with the real world. Typically, samples are not similar and equal in the real world (Pan et al, 2017).

D. The accuracy of the KNN algorithm depends on many irrelevant and redundant features: When the number of unrelated features grows, the rate of prediction accuracy of classification algorithms decreases (Kumar et al, 2016). Removing these features can play an essential role in increasing the accuracy of the KNN algorithm and enhance its speed and accuracy. It can be concluded that the number of features and weighting techniques in the KNN algorithm is one of the most critical factors that affect the efficiency of the algorithm and eventually improve its accuracy (Zhao et al, 2016).

In the present study, we introduce a novel scheme to improve the classification accuracy of the KNN method by creating a new weight calculation technique based on stepwise feature selection. Moreover, the proposed scheme had investigated five datasets with different dimensions to ascertain the introduced scheme could be used in various applications. In our proposed method, the class label of the test sample is assigned based on distances and the category of N neighbors of each sample. However, the main contributions of WAD-KNN are given below:

- Since the performance of the KNN algorithm in various cases is affected by irrelevant features, the proposed scheme selects features with high discriminative power through a stepwise feature selection method.
- An efficient and practical classification method usually considers the area around the test sample and not only the distance proximity of K nearest neighbors.
- The proposed method gives authority value to each sample based on distances and categories of their N neighbors.
- Finally, using authority values, the algorithm inserts a stronger effect on the nearest neighbors that have neighbors with common categories and are closer to the test sample. Using such values helps the algorithm to produce more reliable and accurate predictions and have less sensitivity to the K parameter.
- Extended experiments on real-world datasets demonstrate that our proposed scheme has better classification accuracy in comparison to traditional KNN and other considered researches.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of the recent history of improvement in the KNN algorithm. Section 3 describes the methodology used for this study and presents a stepwise feature selection and a new weighting method. Section 4 describes the proposed KNN method. Section 5 describes the datasets and conditions used in the experiment and report achieved results by the proposed method. Finally, Section 6 discusses and analyzes the achieved results of the proposed approach. Finally, Section 5 sums up the paper with concluding remarks.

Related Works

K-Nearest Neighbor (KNN) algorithm is an instance-based method that computes the similarity between instances on the training data and considering the k top-ranking nearest samples to predict the category of the target sample. The implementation of this method is simple; however, the K value affects the result in some cases. In this section, we will briefly overview some recently improved KNN methods and discuss their advantages and deficiencies.

Bailey et al (1978) used the weighting method with the basic KNN algorithm and developed a new algorithm named the weighted KNN. This algorithm evaluates the distance between samples using the K value and calculates the weight of instances. The convolutional neural network (CNN) algorithm (Gowda et al, 1979; Alpaydinet al, 1997; Angiulliet al, 2005) employs patterns one by one and removes the duplicate patterns. It removes data points that do not add any information and are similar to other data samples. Gates (1972) presented the recurrent neural network (RNN) algorithm, which is an improved version of the CNN algorithm. RNN includes an additional step of eliminating patterns that do not affect the result of a training data set.

Another technique referred to as model-based KNN was presented by (Goa et al, 2003). This method selects the same criteria and then creates a similarity matrix according to given training data. In a different approach, (Bagui et al, 2003) improved the KNN algorithm using the concept of ranking. In this method, all observations belonging to different categories are collected in one location and each category is ranked in ascending order. Then, after counting and ranking all observations, a label is set for unknown instances. Parvin et al (2010) presented the modified KNN algorithm. In this method, the algorithm checks the neighbors of each sample such that if the neighbors of a sample are in the same class, then that sample receives more validity. Next, the validity value is multiplied by the value of calculated distances between samples and a new weighting technique is created.

Zeng et al (2009) introduced a new concept for classifying instances, referred to as pseudo-KNN. In this method, not the closest neighbor but a close neighbor is selected based on the total weight of unclassified patterns in each class. Then, distances between samples are calculated using the Euclidean method. Next, a pseudo neighbor with a higher weight is found and the unknown sample is classified based on that. The pseudo-KNN method attempts to find accurate answers by increasing calculations and ignoring the effects of outlier data.

Lin et al (2015) introduced a new method for improving the KNN algorithm with a combination of k-mean clustering and KNN algorithm to use it in intrusion detection. In training part of their proposed method, they used k-mean clustering for finding the centroid of each class and calculating the distances of each instance and its neighbors and the class centroid. They also employed a similar preprocessing method for the testing set.

Sun et al (2016) presented another algorithm that fills null features as a way of increasing the accuracy of the KNN algorithm. In general, the null attributes can harm the accuracy of the KNN algorithm. When an attribute does not have a value on a record, the value will be replaced based on the highest value of that attribute in other samples of the datasets. For example, if the age value in a record is null and the highest age value is 45, then the algorithm sets the null records age feature to 45 or 46. This process affects weight calculation. Xueli et al (2015) used the kernel method and the reduction in features to increase the accuracy of the KNN algorithm. This method has improved the accuracy of the KNN algorithm in some cases, but it is not accurate enough yet. Kafaf et al (2017) presented a new method called B-KNN to improve the efficiency of the KNN algorithm. In this method, they used a two-fold preprocessing with the notion of minimum and maximum points (MMP) and boundary subsets (BS). The results show suitable improvement in the efficiency of the algorithm on the entire training set while their method achieved a lower accuracy rate in comparison with the traditional KNN algorithm.

Overall, after analyzing the advantages and weaknesses of the other algorithms and studying the history and evaluation of their results, we conclude that the KNN algorithm is not yet sufficiently accurate and it can be improved farther. Thus, in this study, we proposed a novel scheme to improve the classification accuracy of the KNN algorithm, which can efficiently classify unknown instances in different datasets.

Methodology

The main aim of this study is to propose a novel KNN algorithm that can classify instances effectively with a high detection rate. This section describes the procedures and methods used in this investigation and explains the advantages of the proposed algorithm.

Normalization

The datasets used in this study are large and contain many features with different ranges of values. Preparing and processing of the original data is time-consuming and involves computational complexity. Moreover, their classification accuracy may not be accurate enough. Therefore, there are several techniques for normalizing numeric values such as decimal scaling, z-score, and min-max normalization. In the proposed scheme, we applied the min-max normalization method to improve the performance of the algorithm and reduce its computational complexity. The equation of min-max normalization method is expressed as follows:

$$X_s = \frac{X_i - Min}{Max - Min} \tag{1}$$

Where X_i is *i-th* data point, and min and max represent the minimum and maximum ranges of values in X_i this method converts the entire range of values of X_i to the range of 0 to 1.

Round Off Decimal Values

Rounding is usually done to achieve a value that is easier to analyze and communicate than the original. In the presented method, the data of datasets are rounded off so that the decimal values are changed to two decimal places.

Stepwise Feature selection

The main objective of feature selection is to select the subset of most correlated features from large datasets that can describe and represent the main characteristic of all the original features on the dataset (Reshi et al, 2019). The process of feature selection starts using an original dataset and selecting essential features stepwise. In general, the specific cases of stepwise feature selection are forward and backward selection algorithms, which are one of the standard and simplest feature selection methods. The main advantages of stepwise methods are their simplicity and applicability in various types of data.

However, in our proposed scheme, the stepwise forward selection was employed to improve the accuracy of the traditional KNN algorithm and reduce the complexity cost of the algorithm by eliminating irrelevant features. The stepwise forward method could improve the performance of our scheme since we tested the method using several datasets with different data and dimensions. However, we are not sure which feature can lead the algorithm to a higher accuracy rate.

Therefore, we are not able to eliminate many features to get the most optimal set of features and improve the performance of the KNN algorithm. Hence, the employed stepwise method starts with one set of features and adds features to it until the defined criterion is met. This method uses the traditional KNN algorithm to compute the accuracy rate of every set of features. The process of eliminating features in the proposed scheme described in Figure 1.

As shown in Figure 1, in the stepwise approach, we try to extract a subset of features with the highest accuracy.

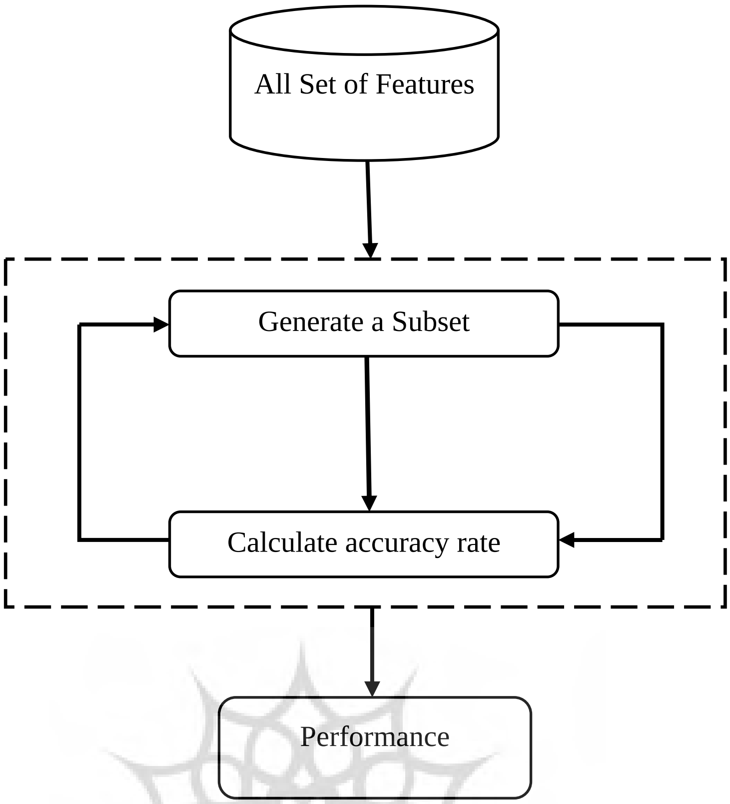

Figure 1. Process of selecting the best subset of features

In this way, the feature selection process starts from assigning one feature and then adding new features, in each iteration, until the addition of new features does not further improve the classification accuracy of the model. This approach helps select important features while eliminating irrelevant ones, and improves classification accuracy, and reduces the complexity of the model. We utilize the traditional KNN algorithm to compute the accuracy rate of each set of features to decide whether a subset of features improves the accuracy rate. Furthermore, to prevent the algorithm from eliminating the majority of features, we set limitations for the numbers of reduced features such that on every dataset, the algorithm can only eliminate 50% of the total features of the dataset.

Proposed Algorithm

In this study, we developed a novel, accurate, and efficient KNN algorithm classifier called WAD-KNN that can effectively classify a variety of data samples. The differences between WAD-KNN and all other improved KNN classifiers in the literature are rounding off decimal values and normalizing them, which has a good impact on the accuracy of the algorithm. Then, it uses a stepwise feature selection method to eliminate irrelevant features that help reduce redundancy and improve the efficiency and accuracy of the algorithm. Moreover, it provides a method for computing the authority of each instance in the dataset using their KNN and combining this technique with a new weight calculation. The calculation method combines the number of N nearest neighbors that have a similar category and sum of distances of N neighbors of each instance to improve the accuracy of the KNN algorithm. Besides, the algorithm utilizes a new weighting method to classify samples more accurately than other methods. The proposed algorithm pseudo code is presented by Algorithm 1 below.

Algorithm 1 WADKNN Algorithm

```
Procedure WAD-KNN ( train set, test set )
  Round_Values (train set);
  Normalization (train set);
  For i = 1 to all features do
  Features[i]= Calculate accuracy use first to i-th features;
  End For
  best set= Compare Features list and select optimal feature set with highest accuracy rate.
  For i = 1 to train size do
  Auth[i]= Calculate Authority of i-th sample;
  End For
  Result = Novel Weighted KNN(Auth, test set, best set);
  End Procedure
```

As explained in Algorithm 1, the proposed KNN contains a set of processes. In the training phase, first, the algorithm rounds off decimal values and normalizes them. Then, it measures the accuracy of each set of features using the stepwise feature selection technique to select the optimal set of features with the highest classification accuracy. Next; it calculates the authority of each instance based on its neighbors. In the testing phase, the algorithm identifies the class of test samples according to the new weighting calculation method with the help of authority values and selected subset of features. The processes of calculating the authority of each instance in the train set and finding the nearest neighbor in the testing phase is discussed in the following:

Authority calculation

In the proposed algorithm, the authority of each instance is calculated using the class of their neighbors. This value, which is calculated for all training samples, has a critical role in the testing phase.

To calculate the authority of each instance in the training set, we considered *N* numbers of neighbors for each instance. The value of *Auth(x)* represents the authority value of sample *x*, which was computed based on the *N* numbers of sample *x* neighbors that are not in the same category as the *x*. The authority values were measured using Eq.2.

$$Auth(x) = Att(x) \times \sum_{i=1}^{N} E\left(Class(x).Class\big(N_i(x)\big)\right) \tag{2}$$

In Eq.2, the *Att(x)* represents the attraction value of instance *x*. In this function, we calculate the sum of Euclidean distance values of *N* neighbors of the *instance(x)*. Here, *N* denotes the number of considered neighbors of each instance and is a positive number that can change as similar as the value of K. *Distance* $(N_i\ (x))$ shows the distance of the *i-th* neighbor of sample *x*. Eq.3 expresses this function.

$$Att(x) = \sum_{i=1}^{N} Distance\big(N_i(x)\big) \tag{3}$$

In Eq.2, *Class (x)* gives the category of sample *x*, and *Class(* $N_i\,(x)$*)* returns the category of the *i-th* neighbor of sample *x*. The *E* function measures the similarity between the class of sample *x* and its *i-th* nearest neighbor. In this function, if sample x and its *i-th* nearest neighbor belong to a similar category, then the function returns 0, and if they have different categories, it returns 1. Eq.4 expresses this function.

$$E(x.y) = \begin{cases} 1 & x \neq y \\ 0 & x = y \end{cases} \tag{4}$$

As mentioned earlier, in the proposed weighting method, we first calculated the weight of *N* neighbors of the *instance(x)* on train set using the Euclidean distance method, by which we multiplied the value with the sum of *N* neighbors classes calculated by Eq.4. In this method, instances that have closer distance neighbors with the same category achieve a higher priority compared to instances that have closer neighbors with diverse categories. One of the significant achievements of this method is about combining the final authority value with the sum of neighbors' distance. Hence, when *instance(x)* has *N-1* neighbors with the same category, and *instance(y)* has the same number of neighbors with a similar category. Then, the instance that has neighbors with closer Euclidean distance values achieves more priority than another one.

Weight calculation

An easy and efficient method to solve class distributions in the KNN classifier is implementing a weighting method. The class of each K nearest neighbors is multiplied by an efficient weighting method that guarantees the nearer neighbors contribute more to the final weight than the distant ones. In the proposed method, the distance between points was calculated using Eq.5, which is based on the Euclidean distance.

$$D = \sqrt{\left(\sum_{i=1}^{n}(X_i - Y_i)^2\right)} \tag{5}$$

Where n denotes the number of total instances in the dataset, and D represents computed Euclidian distance between points X_i and$Y_{i.}$ In the WAD-KNN algorithm, first, the weight of every neighbor is calculated using Eq.5, which is based on Euclidian distance. Next, the authority values of each instance are added to the calculated Euclidian distance weight. The final weight of each neighbor is computed using the Eq.6.

$$W(x) = Auth(x) + \sqrt{\left(\sum_{i=1}^{n}(X_i - Y_i)^2\right)} \tag{6}$$

Where *Auth(x)* represents the authority value of the *i-th* nearest neighbor. We then added Euclidean distance value to the computed authority of sample *x* to calculate the final value. This method has more influence on the nearest neighbors that have neighbors with common categories and are closer to the test sample. With this weighting method, the algorithm can predict unknown instances with higher performance since unlike the traditional KNN algorithm, it gives priority to nearest neighbors of the sample instance base of the class of their neighbors. Hence, it decreases the distance of nearest neighbors which have neighbors that are in the same class from the unknown sample. This technique can overcome the weakness of regular weight calculation and can classify instances with higher reliability.

Experiments

This section includes experimental conditions and investigation of experiment outcomes over the different datasets. Subsection 5.1 describes the used datasets in the experiments. Subsection 5.2 reports the performance of the proposed method and compares achieved results by the WAD-KNN with the traditional KNN algorithm to determine the detection efficiency of different instances over datasets.

Datasets

We used the five UCI datasets for evaluating the performance of the proposed method. The selected datasets have different dimensions varying from small to large and contain just numerical attributes apart from the class attribute. A brief description of the five datasets used for the experiments is given in Table 1.

Table 1. Datasets description.

Dataset	Number of Instances	Number of Attributes
ILPD (Indian Liver Patient)	583	10
Wine	178	13
Diabetes	768	8
BTSCD (Blood Transfusion Service Center Dataset)	748	5
Yeast	1484	8

Experiment results

In our experiments, we examined the accuracy of the WAD-KNN compared with the basic KNN algorithm. We conducted experiments using 70% of each dataset records as the training set and the other 30% as the testing set. We selected three values for K (3, 5, and 7), which are standard numbers for the value of K in most of the recent studies. The proposed algorithm and basic KNN were evaluated 100 times for each dataset with a different value of K. The results of these examinations are reported in Table 2.

Table 2. Comparison of the classification accuracy between WAD-KNN and KNN

DataSet	Method	K=3	K=5	K=7
ILPD	KNN	64.16	62.24	61.84
	WAD-KNN	86.17	89.37	87.42
Wine	KNN	96.22	98.11	94.33
	WAD-KNN	100	100	100
Diabetes	KNN	73.47	74.78	73.04
	WAD-KNN	80.86	76.95	77.82
BTSCD	KNN	68.00	73.33	72.44
	WAD-KNN	75.11	76.33	75.89
Yeast	KNN	56.55	57.58	54.49
	WAD-KNN	65.80	64.01	63.49

The experiment results show that the proposed method is more accurate than the basic KNN using different values of k over different datasets. The proposed method showed that the average improvement achieved in the five datasets was approximately 10%. For the Indian Liver Patient (ILPD) dataset, the accuracy achieved by the proposed method was approximately 24% higher than the results achieved using the traditional KNN algorithm.

Discussion

In the experiments, we evaluated the accuracy of WAD-KNN compared with traditional KNN running on the five UCI datasets. Figure 2 illustrates the improvement in classification accuracy over the different datasets. From this graph, we can see that the lowest improvement value is achieved using the Wine dataset while the most considerable improvement was achieved based on the ILPD dataset. However, the average improvement of the proposed algorithm compared with basic KNN was approximately 10%.

The results (Table 2) indicate that the WAD-KNN algorithm outperformed the traditional KNN algorithm with the highest accuracy of 100% while running on the Wine dataset provided the lowest one for the Yeast dataset with accuracy levels of 65.80%, 64.01%, and 63.49%. The average achieved results of both algorithms over the different datasets are summarized in Figure 2.

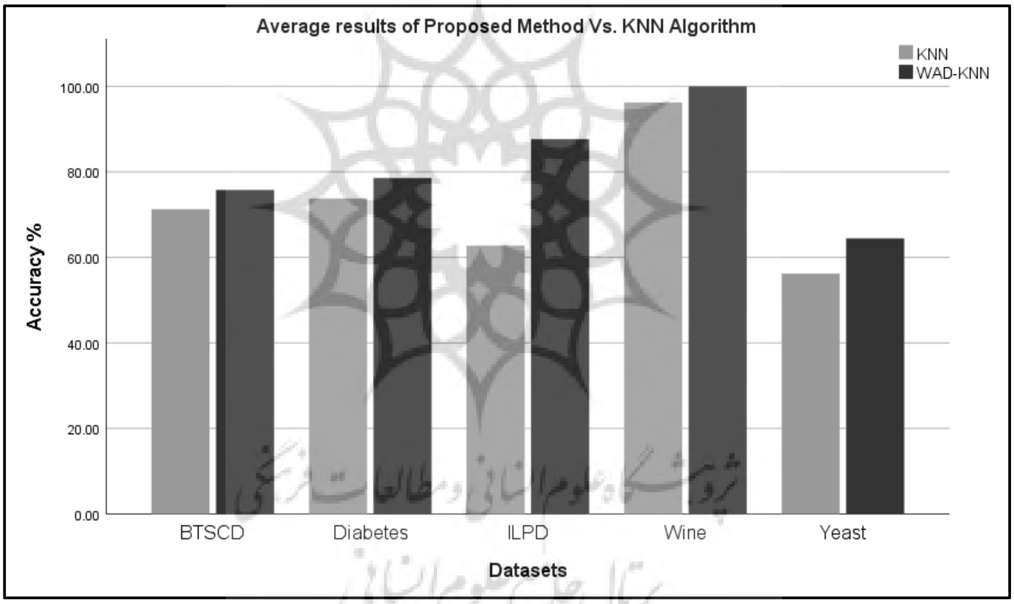

Figure 2. Comparison of accuracy values with KNN algorithm.

From the above graph, we can see that the average accuracy results achieved by WAD-KNN are significantly higher than the KNN algorithm. This higher accuracy is because, unlike WAD-KNN, the traditional KNN algorithm classifies instances without the training section, which does not give weight or priority to instances. This is one of the reasons that reduce the classification accuracy of this algorithm. WAD-KNN outperformed traditional KNN in terms of accuracy because the new weighting technique gave the authority to each instance using the category of nearest neighbors and then calculated distances between neighbors and chose the nearest neighbors to the sample. Another salient point in favor of WAD-KNN performance is that it is affected by the normalization values of the dataset that helps make a better prediction with the highest efficiency and low complexity.

Next, to better analyze the effectiveness of the proposed method, we ran a comparative study between the proposed method and several similar approaches. The considered algorithms are MKNN and SWKNN. Figure 3 shows the comparative results for the concerned methods over five selected datasets. This comparison was made under similar conditions and the accuracy results of algorithms were computed used three different K numbers of 3, 5, and 7.

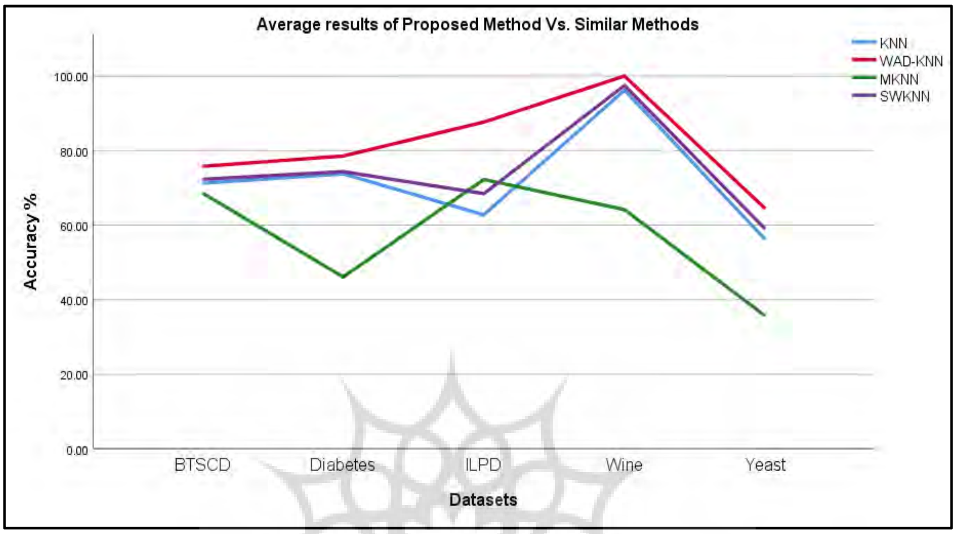

Figure 3. Comparison of accuracy values with similar algorithms

From Figure 3, it can be inferred that the overall accuracy of the proposed technique is better than other considered methods. So, it indicates that the proposed method was successful in distinguishing the various types of records in comparison with similar approaches. This improvement in classification accuracy was obtained using the presented weighting technique, which provides prioritizes samples that have neighbors with close Euclidean distances while they are in the same category. This idea helps the algorithm recognize the most important neighbors and make the decision using them.

Conclusion

The KNN algorithm is a branch of data mining classification algorithms that classify quickly based on the categories of their K nearest neighbors (KNNs). This algorithm has been studied in the past years by various researchers in real-world domains such as pattern recognition, medical diagnoses, text classification, and intrusion detection. In this paper, we developed a novel accurate KNN algorithm called WAD-KNN to improve the classification accuracy of the traditional KNN algorithm. First, we used a stepwise feature selection technique to choose features with high discriminative power; then, we applied a new weighting technique based on category and distances of N neighbors samples to produce more reliable results and overcome the weakness of regular weight calculation. We performed extensive experiments on five UCI real-world datasets in terms of accuracy to evaluate the performance of the proposed method. The accuracy of the introduced method was compared with those of KNN, MKNN, and SWKNN and found that it significantly improves the classification performance with high accuracy. The findings of this study also indicate that the proposed weighting method has a positive impact on the classification accuracy of the KNN algorithm. In the future, we will utilize the WAD-KNN scheme for various real-world applications.

References

Abbasi, F., Khadivar, A., &Yazdinejad, M. (2019). A Grouping Hotel Recommender System Based on Deep Learning and Sentiment Analysis. Journal of Information Technology Management, 11(2), 59-78.

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

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

Bagui, S. C., Bagui, S., Pal, K., & Pal, N. R. (2003). Breast cancer detection using rank nearest neighbor classification rules. Pattern recognition, 36(1), 25-34.

Bailey, T. (1978). A note on distance-weighted k-nearest neighbor rules. Trans. on Systems, Man, Cybernetics, 8, 311-313.

Biswas, N., Chakraborty, S., Mullick, S. S., & Das, S. (2018). A parameter independent fuzzy weighted k-nearest neighbor classifier. Pattern Recognition Letters, 101, 80-87.

Cheng, Y., Chen, K., Sun, H., Zhang, Y., & Tao, F. (2018). Data and knowledge mining with big data towards smart production. Journal of Industrial Information Integration, 9, 1-13.

Dua, D., & Graff, C. (2017). UCI machine learning repository (2017). URL http://archive. ics. uci. edu/ml.

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

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

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

Kafaf, D. A., Kim, D. K., & Lu, L. (2017). B-knn to improve the efficiency of kNN. In Proceedings of the 6th international conference on data science, technology and applications. Science and Technology Publications (pp. 126-132).

Kotenko, I., Saenko, I., &Branitskiy, A. (2018). Framework for Mobile Internet of Things Security Monitoring Based on Big Data Processing and Machine Learning. IEEE Access, 6, 72714-72723.

Kumar, M., Rath, N. K., &Rath, S. K. (2016). Analysis of microarray leukemia data using an efficient MapReduce-based K-nearest-neighbor classifier. Journal of biomedical informatics, 60, 395-409.

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. Knowledge-based systems, 78, 13-21.

Pan, Z., Wang, Y., & Ku, W. (2017). A new general nearest neighbor classification based on the mutual neighborhood information. *Knowledge-Based Systems*, 121, 142-152.

Parvin, H., Alizadeh, H., &Minati, B. (2010). A modification on k-nearest neighbor classifier. Global Journal of Computer Science and Technology.

Reshi, J. A., & Singh, S. (2019). Investigating the Role of Code Smells in Preventive Maintenance. *Journal of Information Technology Management*, 10(4), 41-63.

Serpen, G., &Aghaei, E. (2018). Host-based misuse intrusion detection using PCA feature extraction and kNN classification algorithms. *Intelligent Data Analysis*, 22(5), 1101-1114.

Sun, C., Yao, C., Shen, L., & Yu, X. (2016). Improving classification accuracy using missing data filling algorithms for the criminal dataset. *International Journal of Hybrid Information Technology*, 9(4), 367-374.

Wu, X., Yang, J., & Wang, S. (2018). Tea category identification based on optimal wavelet entropy and weighted k-Nearest Neighbors algorithm. *Multimedia Tools and Applications*, 77(3), 3745-3759

Xueli, W., Zhiyong, J., &Dahai, Y. (2015, September). An Improved KNN Algorithm Based on Kernel Methods and Attribute Reduction. In 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC) (pp. 567-570). IEEE.

Zeng, Y., Yang, Y., & Zhao, L. (2009). Pseudo nearest neighbor rule for pattern classification. *Expert Systems with Applications*, 36(2), 3587-3595.

Zhao, M., & Chen, J. (2016). Improvement and comparison of weighted k nearest neighbors classifiers for model selection. *Journal of Software Engineering*, 10(1), 109-118.

Bibliographic information of this paper for citing:

Sheikhi, Saeid, & Taghi Kheirabadi, Mohammad, & Bazzazi, Amin (2020). A Novel Scheme for Improving Accuracy of KNN Classification Algorithm Based on the New Weighting Technique and Stepwise Feature Selection. *Journal of Information Technology Management*, 12(4), 90-104.