ISSN: 2350-0328

International Journal of AdvancedResearch in Science, Engineering and Technology

Vol. 6, Issue 4, April 2019

Development of New Method of Time Series Classification Based on k-Nearest Neighbors for Analysis of Large Data Flow by Artificial Intelligence

Muminov Bakhodir, Mukhamadiyeva Kibriyo

Professor, doctor of technical Sciences, Tashkent University of information technologies, Uzbekistan. Applicant, Lecturer of department "Information communication technology" Bukhara engineering-technological institute, Uzbekistan.

ABSTRACT: Time series classification is one of the most important problems when analyzing time series data. This problem has been attracting increasing attention of researchers in recent years. In this article we propose a new nonparametric method for time series classification. The proposed method, i.e. weighted local dynamic averaging of the baricenter by deformation *k* time.

KEYWORDS: time series, classification, *k* nearest neighbors, nonparametric method

I. INTRODUCTION

The time series is a sequence of observations of well-defined data elements derived from repeated measurements over time. Time series data exist in many areas, including economics, finance, production systems or medicine. Measurement of heartbeat per minute of a person in the form of an electrocardiogram can be considered as an example of time series. In the context of **Industry 4.0**, sensors and automation systems are the heart, and they transmit a huge amount of time series data. Among the problems of time series analysis, time series classification is one of the most popular and very useful for production technologies in the **Industry 4.0**.

Distance measurement for time series plays an important role and contributes to various aspects of time series classification. Several distance measurement functions have been proposed for time series in which the Euclidean Distance (ED) and Dynamic Time Deformation Distance (DTW) are most widely used. Time series classification has been deployed as a key component in many artificial intelligence applications, such as the classification of ozone concentration in the atmosphere, the classification of galaxies and quasars, or many applications in these energy systems.

So far, many distance-based methods have been proposed for time series classification. However, non-parametric algorithms have advantages over them in that they can be used with invisible and variable datasets because the trained models are not based on a training kit. K-Nearest Neighbors (*k-NN*), especially 1-Nearest Neighbor (1-NN), is nonparametric and is widely considered difficult to overcome when classifying time series.

However, since 1-NN is based on the distance from the request instance to all the tutorial instances to find the single closest instance, this can lead to greater classification risk.

In recent years, several algorithms have been proposed to improve the 1-NN algorithm. Instead of selecting a single closest point, *k-NN* selects the nearest k point from the request point. The improvement of *k-NN* is a local average value of k -Nearest Neighbor (LM *k-NN*). It tries to overcome the negative effect of emissions in the small sample preparation established by calculating the local average vector k Nearest specimens in each class. LM *k-NN* has been successfully applied in the classification of galaxies and quasars. Another improvement to *k-NN* is k Closest Neighborhood Centroids (*k-NCN*) based on a method called Nearest Centroid Neighbor (NCN). k- NCN was evaluated as an effective method for data sets with different sample sizes B **[2]** it was proposed to use a combination of LM *k-NN* and *k-NCN*, namely, the local average neighbors of k-neighbors (LM *k-NCN*), in order to outrun the previous two methods. The LM *k-NCN* is emission-resistant and effective for small sample sizes with the NCN concept. It should be noted that although *k-NN* is proposed for time series, LM *k-NN*, *k-NCN* and LM *k-NCN* have not yet been tested on this kind of data. **[4]**. A weighted local average of k- nearest neighbors (WLM *k-NN*) has been proposed, which improves LM *k-NN* by defining local average vectors more efficiently than LM *k-NN* by weighing their elements on the basis of

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

the distances from query instances to k nearest instances in each class. Our work in this article is an improvement of this method [4].

II. RELATED WORKS

Dynamic baricular time series averaging (DBA) was first presented [3] for clustering problems and was expanded to reduce time series data sets to provide faster and more accurate classification [3]. This algorithm is the best algorithm to date for averaging time series due to the use of the DTW matrix to make it more efficient and reliable than other averaging methods. It is effectively used in many applications, such as hyperspectral classification of images [1] and alignment of speech to the translation of languages with low resources [1].In this article, we first extend the DBA algorithms called Weighted DBA (WDBA) to better calculate local average vectors. Then we propose a new nonparametric method of time series classification, namely: weighted local dynamic deformation in baricenter time with averaging of k- nearest neighbors (WLDBA *k-NN*). This improvement of LM *k-NN* algorithm differs from LM *k-NN* in that it differs from LM *k-NN* by replacement of local average vectors with local DBA vectors calculated using WDBA. Experimentally, we show that the WLDBA *k-NN* is ahead of the WLM *k-NN*, and the WLM *k-NN* and WLDBA *k-NN* are ahead of 1-NN, *k-NCN*, LM *k-NN*, LM *k-NCN* in 85 time series data sets in the UCR time series classification archive [1]. The experimental results also show that the new local vectors used in the WLM *k-NN* and WLDBA *k-NN* make a significant contribution to improving the time series of performance classification.

A. TIMESERIES

Time series X is a sequence of actual numbers collected at regular intervals over a period:$X = x_1, x_2, \dots, x_n$, Collection periods are equal, so they are not important. Therefore, we can consider the time series as an n-dimensional instance in the metric space. Fig. 1 shows some time series data in the ECGFiveDays data sets of the UCR time series classification archive.

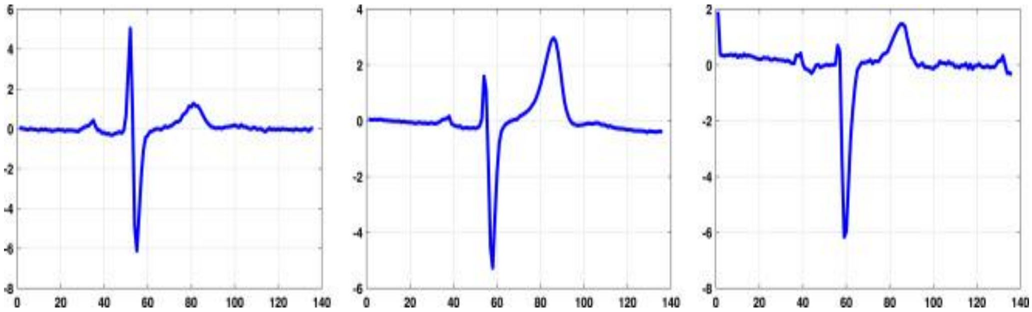

Fig. 1. Some time series in the ECGFiveDay dataset in the UCR time series classification archive.

B. DISTANCE MEASURES

When extracting data from time series, distance measurement is a crucial factor that has a huge impact on a number of problems, such as similarity search, classification, clustering, motive detection, anomaly detection, prediction. These questions require distance to assess the similarity between time series. Several distance functions have been proposed for time series, among which two popular distance measures are Euclidean distance and dynamic time deformation.

B.1 Euclidian distance

Euclidean distance between two time series $Q = q_1, q_2, \dots, q_n$and $C = c_1, c_2, \dots, c_n$is determined by formula (1)

$$.ED(Q, C) = \sqrt{\sum_{i=0}^{n}(q_i - c_i)^2} \quad (1)$$

Note that the two time series Q and C must have the same length n.

B.2 Dynamic time distortion

Dynamic Time Distortion (DTW) is a popular distance in time series. It calculates the optimal coincidence between two time series and fixes flexible similarities by aligning the coordinates within both time series. Note that the DTW can calculate the distance of two time series of different lengths. Taking into account two time series $Q = q_1, q_2, \dots, q_m$as well as $C = c_1, c_2, \dots, c_n$ to calculate the distance between these time series, matrix D of size m X n where each element $D(i, j)$is calculated by formula (2) .

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

$$D(i,j) = d(q_i, c_i) + min\begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}$$

In which the distance$d(q_i, c_j) = (qi - c_i)^2$, the dynamic deformation time of the two time series Q and C is the square root of the cumulative distance in cell D (m,n) , Table 1 shows the DTW matrix when using the DTW to calculate the distance between two time series Q=[4,6,7,7,10,9,6,6,4,4,5,6,6] and C=[5,6,6,9,10,8,5,5,4,4,5] , the distance between Q and C is $D\sqrt{(10,10)} = \sqrt{6}$ and curved path =[1,1,2,3,3,3,4,5,5,5,5,6].

	4	6	7	10	9	6	4	4	5	6
5	1	2	6	31	47	48	49	50	50	51
6	5	1	2	18	27	27	31	35	36	36
6	9	1	2	18	27	27	31	35	36	36
9	34	10	5	3	3	12	37	56	51	45
10	70	26	14	3	4	19	48	73	76	61
8	86	30	15	7	4	8	24	40	49	53
5	87	31	19	32	20	5	6	7	7	8
4	87	35	28	55	45	9	5	5	6	10
4	87	39	37	64	70	13	5	5	6	10
5	88	40	41	62	78	14	6	6	5	6

Table 1.DTW matrix of two time series Q and C.

Time Distortion (DTW) is a popular distance in time series. It calculates the optimal match between the two given time series and captures the flexible similarities by aligning the coordinates within the two time series. Note that DTW can calculate the distance of two time series of different lengths. Taking into account the two time series $Q = q_1, q_2, \dots, q_m$ and $C = c_1, c_2, \dots, c_n$ to calculate the distance between these time series, the matrix d of size m X n where each element D (i, j) is calculated by the formula (2).

C. DYNAMIC TIME AVERAGING OF THE BARICENTER

In this subsection we mention the best algorithm for averaging time series. The dynamic baricentral averaging (DBA) algorithm was presented **[3]**. This algorithm is based on a DTW matrix to improve the efficiency of time series averaging. As shown in **[3]**, the DBA algorithm surpasses all existing algorithms of averaging in all data sets in the UCR time series classification archive **[1]**. This algorithm has been successfully used to reduce the time series data to provide faster and more accurate classification **[3]** or clustering for time series **[3]**.

D. LOCAL MIDDLE K-NEIGHBOURHOOD

The local average value of *k -NearestNeighbor* (LM *k-NN*) is a simple and reliable classifier in cases of small sample size **[2]**. The goal of LM *k-NN* is to overcome the negative impact of emissions in the training kit. The justification for this method is the local average vectors, which are obtained from k nearest neighbors of the query instance in each class. Then the request is assigned to the class that has the local average vector closest to it. Each local average vector is calculated as follows:

Given the X set of M time series: $\{Xi\}_i = 1..M$where each $X_1 = \{x_{11}, x_{12}, \dots, x_{1n}\}$is a time series of length n, the average vectorX , is designated $\overline{X} = \{\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n\}$ where each $\overline{x}_j$, is defined as in the formula (3).

$$\overline{x}_j = (\sum_{i=1}^{M} x_{ij})/M \qquad (3)$$

The LM *k-NN* scheme looks like this:

Given the TL training kit and the subset $T_L^i(i = \overline{1,L})$of TL, which consists of all instances in class i , the q request instance label is determined by the following steps:

The LM *k-NN* scheme is as follows:

Given a TL training set and a subset ,$T_L$which consists of all the instances in class i. The label of the query instance q is defined by the following steps:

Step 1:for eachT_L^ifinds k nearest neighbors q from T_L^i, is called $\{uij\}(j = 1)k$k, and then puts them in the KNNi set$(i = \overline{1,L})$.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

Step 2: For each$KNN_i = \{u_{ij}\}\,(j = 1)\,k$, obtained in Step 1, calculates the local average vector ūi for class i using formula (3). Thus, each class i has one local average vector $\bar{u}_i$,
Step 3: Assigns q to the i class if the distance between q and the local average vector $\bar{u}_i$ is minimal in the Euclidean space.

Note that LM *k -NNN* is equivalent to 1-NN when K = 1. LM *k-NN* has been successful in classifying conventional data, but this has not yet been tested on time series.

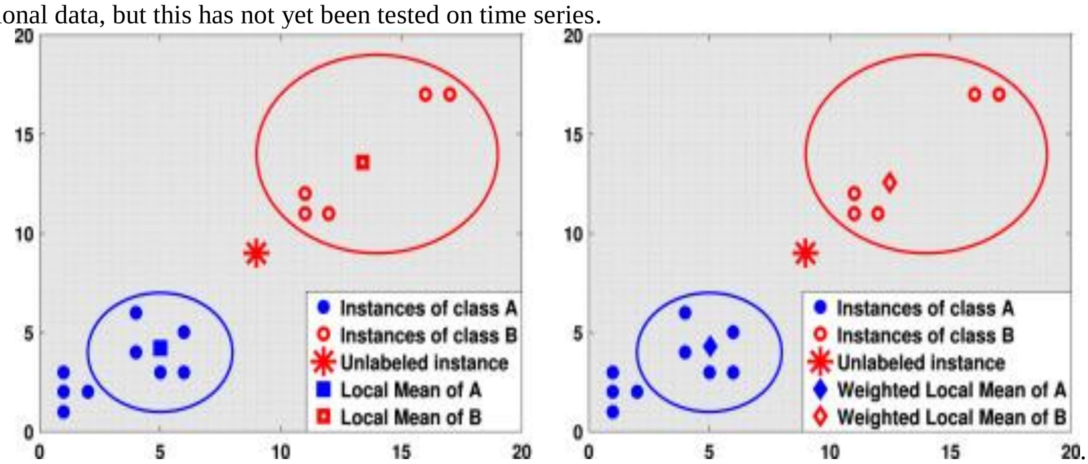

Fig. 2.The difference between the local mean vector LM *k-NN* (left) and the weighted local mean vector WLM *k-NN* (right).

The left image shows that Q is closer to the local average of class A than to class B. A correct image shows that Q is closer to the local average of class B than to class A.

E. WEIGHTED LOCAL MIDDLE K-NEIGHBORHOODS

LM *k -NNN* finds the average values by calculating the average K vectors of the nearest instances in each class. This can overcome the negative effects of emissions. **[4]**extended LM *k -NNN* by WLM *k* -NNN method. WLM *k -NNN* finds the average value using weighted local average vectors, where weights are calculated based on the distances from the query instance to its nearest *k* instances in each class. Therefore, this method can better cope with emissions.

Note that the WLM *k -NNN*, LM*k-NN* and 1-NN are the same when *k* is equal to 1 and you can use a different measure instead of aneuclidean distance.

The difference between WLM *k -NNN* and LM *k -NNN* is at step 2 of both algorithms when we calculate the local average vectors. LM*k-NN* simply finds the local average vector, calculating the usual arithmetic mean, while WLM*k-NN* uses the weighted arithmetic mean.

When using LM*k-NN* with *k=5* to D, classify q , k closest neighbors q in class A is D1, D2, D3, D4, D5 and k closest neighbors q in class B is D10, D11, D12, D13, D14. Thus, D13 and D14 influence the local middle vector of class B, and as a result it will assign the label A to q because the local middle vector of class A is closer to q than that of class B.

Using the proposed WLM*k-NN* method from k=5 to D classify q D13 and D14 are also selected as k nearest neighbors of q in class B , however, when we calculate the weighted local average vector of class B weights given for D13, D14, now much less than the weights given for other instances, because their distances up to q are much further away. The weighted local average vector of class B is much closer to q than that of class A , so it will assign the B label to q as we expected.

III. PROPOSED METHOD

In this section we present dynamic barycenter averaging (DBA) with dynamic time distortion for time series averaging. We then proposed a weighted DTW baricentric averaging (WDBA) method, which improves the DBA. Finally, we present our new proposed method called the Weighted local dynamic deformation of time *k* the nearest neighbor (WLDBA*k-NN*) for time series classification.
There are two expansion points

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

First, we do not use the ***medoid*** method to select the initial time series. Instead, we use an instance of the request (without a label) because the initial time series cause the local DBA vector to be more like an instance of the request.

Second, we use weights determined by the distance between the instances in the training set and the query instance to calculate a more efficient local DBA vector. The reason is that each copy in the training set has a different effect on the classification of the unmarked copy.

We call our modified version of DBA "Weighted Baricenter Averaging" (WDBA).

A nonparametric classifier based on *k-NN* calculates the distances between the query template and all training samples. According to Fukunaga (1990), such a nonparametric classifier usually suffers from existing emissions. To overcome the disadvantage of *k-NN*, Mitani and Hamamoto have proposed a nonparametric classifier (LM*k-NN*) that improves *k-KNNN* by classifying a query instance based on local medium vectors. Nearest training samples of the sample request. Thus, LM*k-NN* is resistant to emissions and belongs to the type of averaging methods. **[3]**proposed a method of averaging called DTW Barycenter Averaging (DBA), which shows that it achieves better results than the averaging methods in all experiments in proven data sets. Since the DBA processes all time series in the same way, we offer an averaging method called DTW Barycenter Averaging (WDBA), which extends the DBA with two extension points. First we use an instance of the request as the initial input. Secondly, we draw every time. We are experimentally proving that WDBA is a method of averaging as well as DBA. Therefore, it also calculates the local average vectors, which increases the efficiency of classification. This paper proposes a new nonparametric classifier based on WDBA, called the weighted dynamic time deformation of the *nearest* neighbors (WLDBA*k-NN*). This improves LM*k-NN* with WDBA to calculate local average vectors.

IV. SIMULATION&RESULTS

First we compare WLM*k-NN* to 1-NN, LM*k-NN*, *K-NCN*, LM*k-NCN*. After that we compare the proposed WLDBA*k-NN* with LM*k-NN* and LM*k-NCN* when using DBA to find local middle vectors. In addition, the WLDBA*k-NN* is also compared to WLM*k-NN* and 1-NN. All comparisons in terms of accuracy. Accuracy is the ratio of the total number of verified copies. In general, the higher the accuracy, the better the method. Note that in our experiments all methods use the Euclidean distance as a measure of distance.

We use composite and scatter plots to compare performance. In particular, when comparing method A with method V, each column in the bar graph with the accumulation is divided into three parts: the bottom, middle and top. Values such as b, m, and T are shown at the bottom, middle, and top, respectively, showing that 85 datasets are used in the experiments, A wins V (accuracy A is better than accuracy V) on datasets b, A, and V draw, and A loses V (accuracy A is worse than V) on datasets T. Fig. 3 shows a bar graph with accumulation comparing WLM*k-NN* to 1-NN. In the column with k = 4, a value of 58 at the bottom indicates that the WLM*k-NN* wins 1-NN in 58 datasets; a value of 2 in the middle section indicates that the WLM*k-NN* equals 1-NN in 2 datasets; and a value of 25 at the top indicates that the WLM*k-NN* loses 1-NN in 25 datasets. When comparing the method, use a scatter plot, where each point on the plot is a data set, with its own auxiliary value, accuracy and value. The points in the upper triangle illustrating V give a higher accuracy than A; while the points in the lower triangle show that A gives a higher accuracy than V. The points on the diagonal line indicate A and V with the same accuracy. For example, the institution shows that the scattering diagrams in the figure 4 indicate most of the numbers in the upper triangle. This means that WLM*k-NN* is superior to 1-NN in all experiments.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

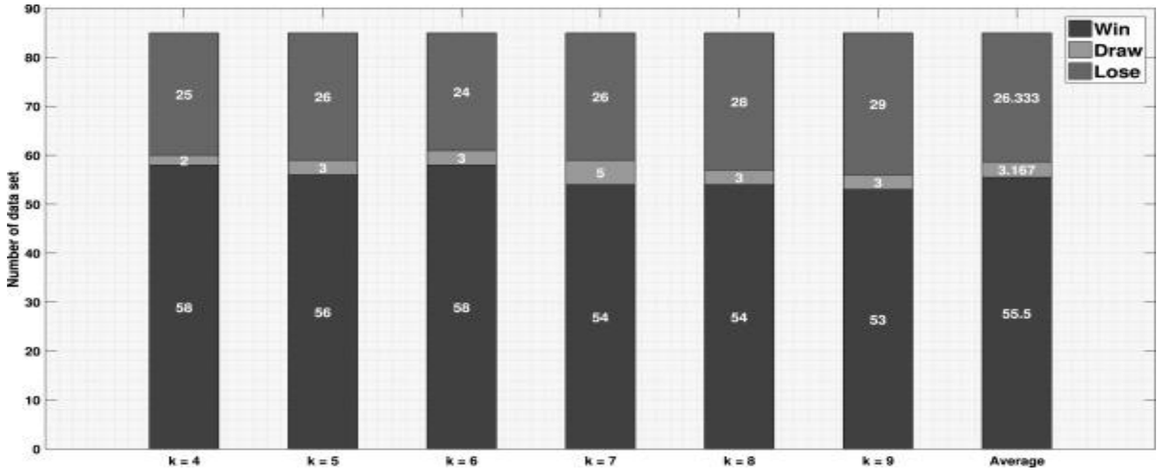

Fig. 3. A bar graph with accumulation compares WLM*k-NN* with 1-NN.

The numbers at the bottom, middle and top of each column represent the numbers of data sets in which WLM*k-NN*wins, win and loses 1-NN respectively. Victory: the number of datasets for which the accuracy of the WLM*k-NN* is better than that of 1-NN. Lose: the number of data sets for which the accuracy of WLM*k-NN* is worse than that of 1-NN. Draw: number of data sets for which the accuracy of WLM*k-NN* and 1-NN is the same.

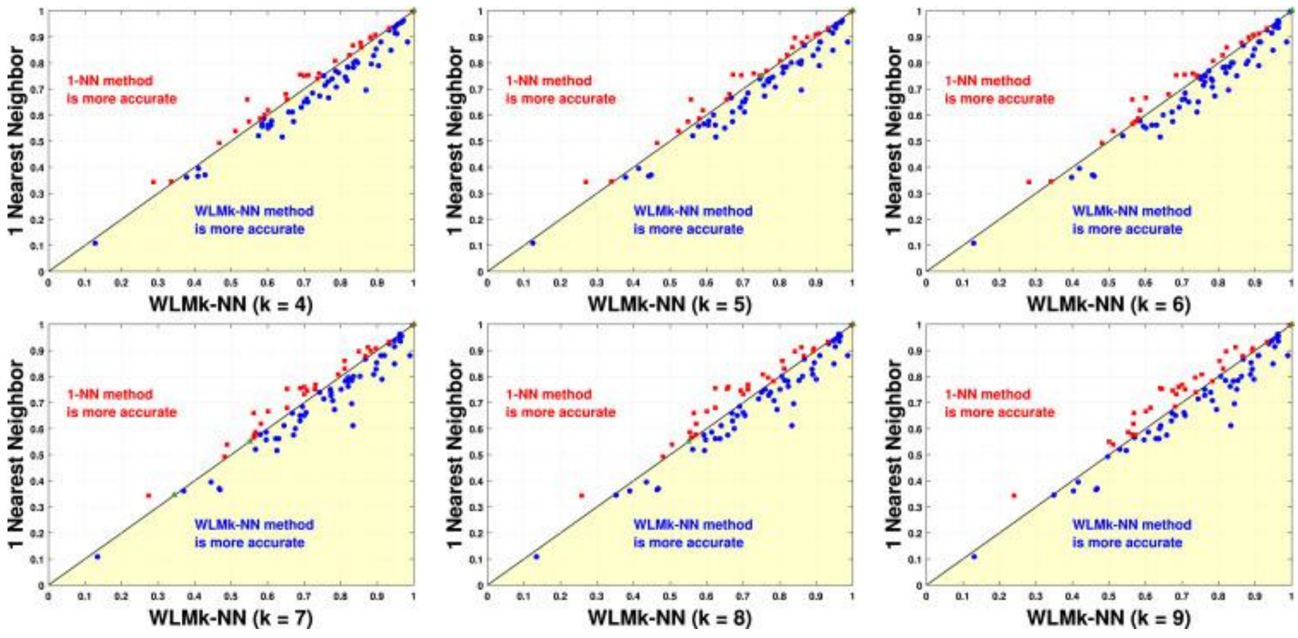

Fig. 4. Spot charts for comparing WLM *k -NNN* (*k* takes values from 4 to 9) with 1-NN.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

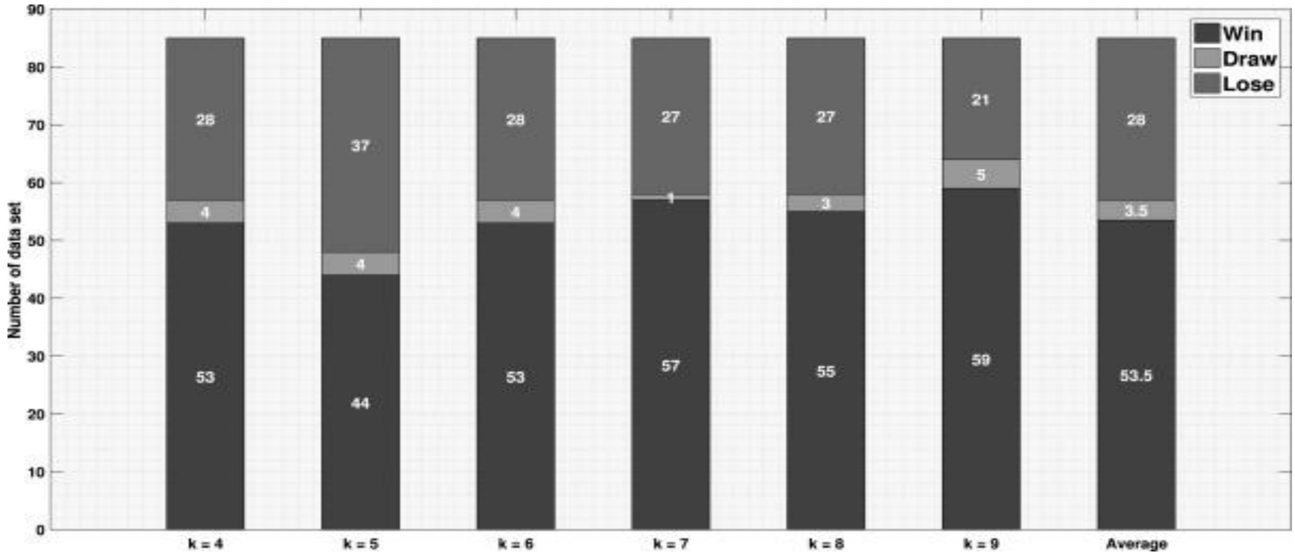

Fig. 5. A bar graph with accumulation compares WLM *k -NNN* with K -NCN.

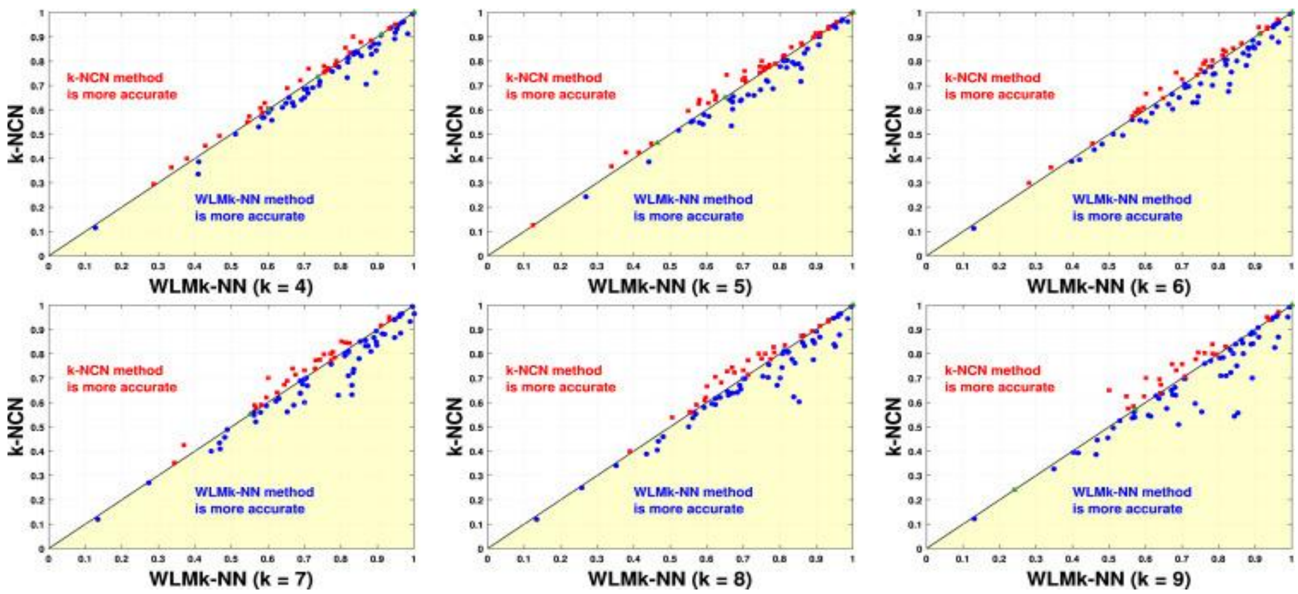

Fig. 6.Diagrams for comparing WLM*k-NN* (K 4 to 9) with *k-NCN*.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

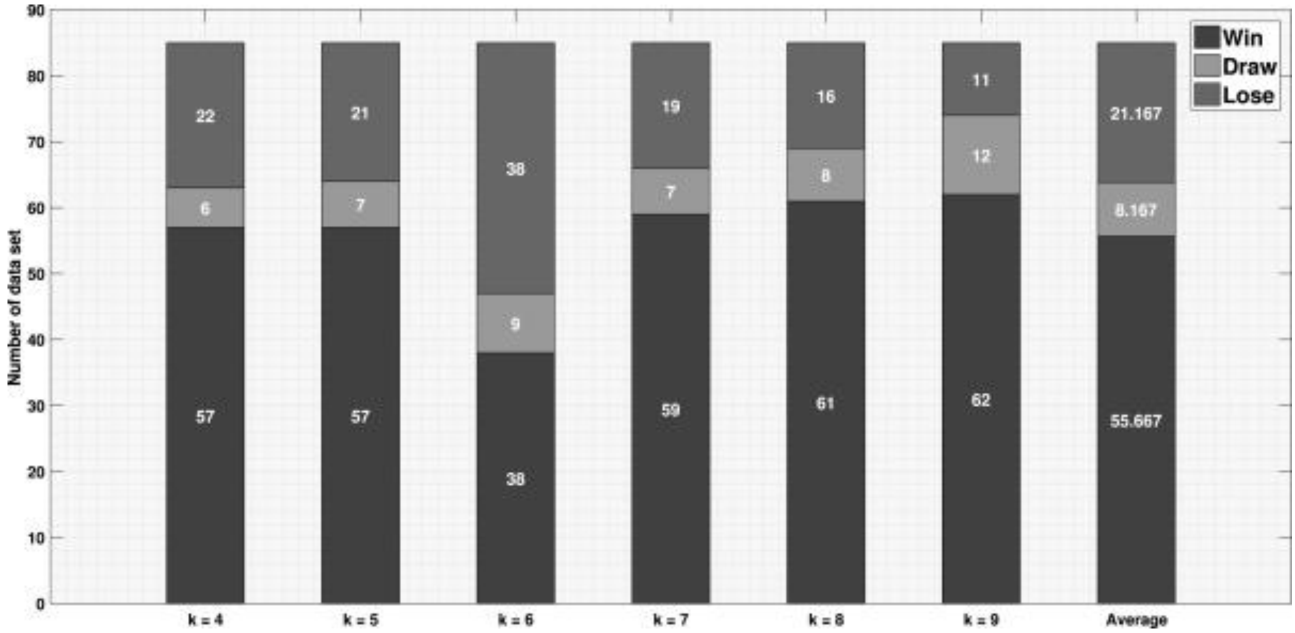

Fig. 7. A bar graph with accumulation compares WLM*k-NN* with LM*k-NN*

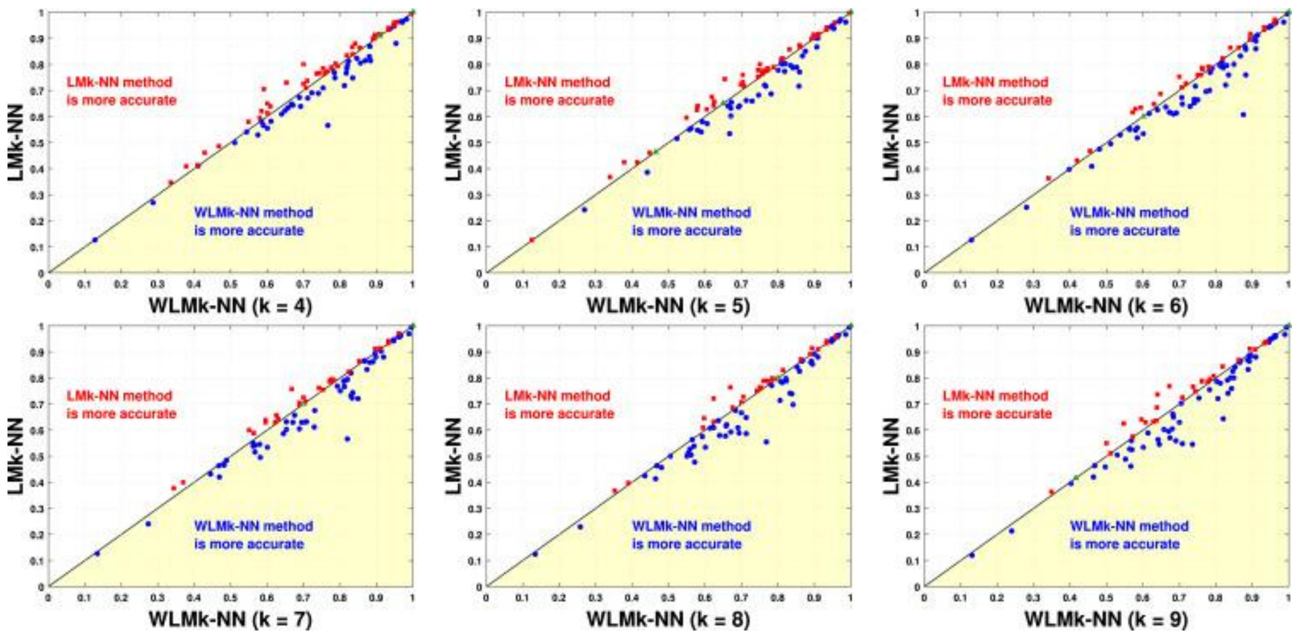

Fig. 8. Scatter plots for comparing WLM *k -NN* (K 4 to 9) with LM *k -NN*

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

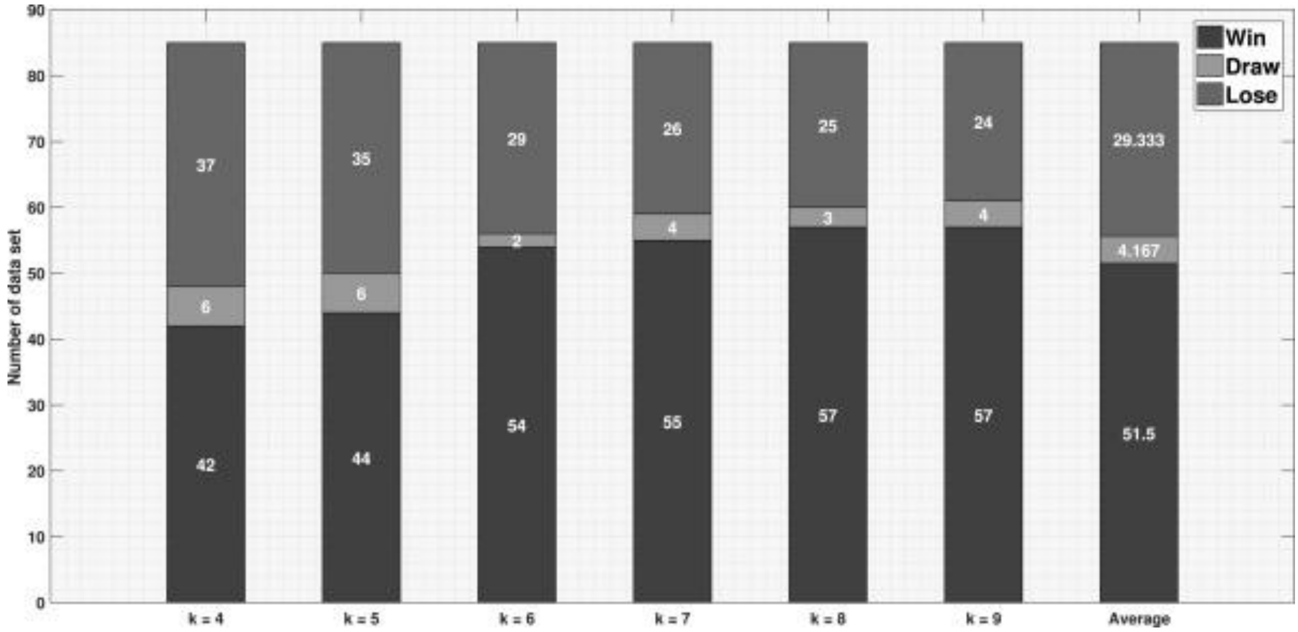

Fig. 9. A bar graph with accumulation compares WLM *k -NNN* with LM *k -NCN*.

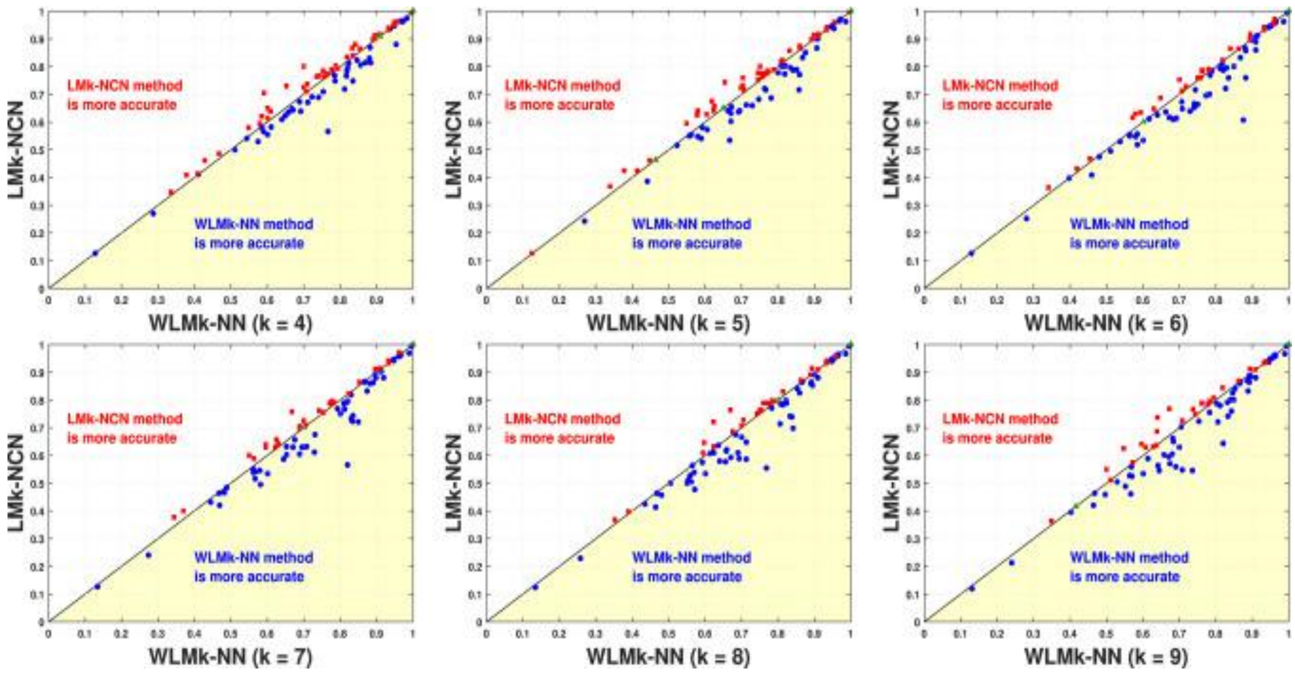

Fig. 10.Diagrams for comparing WLM *k -NNN* (K 4 to 9) with LM *k -NCN*.

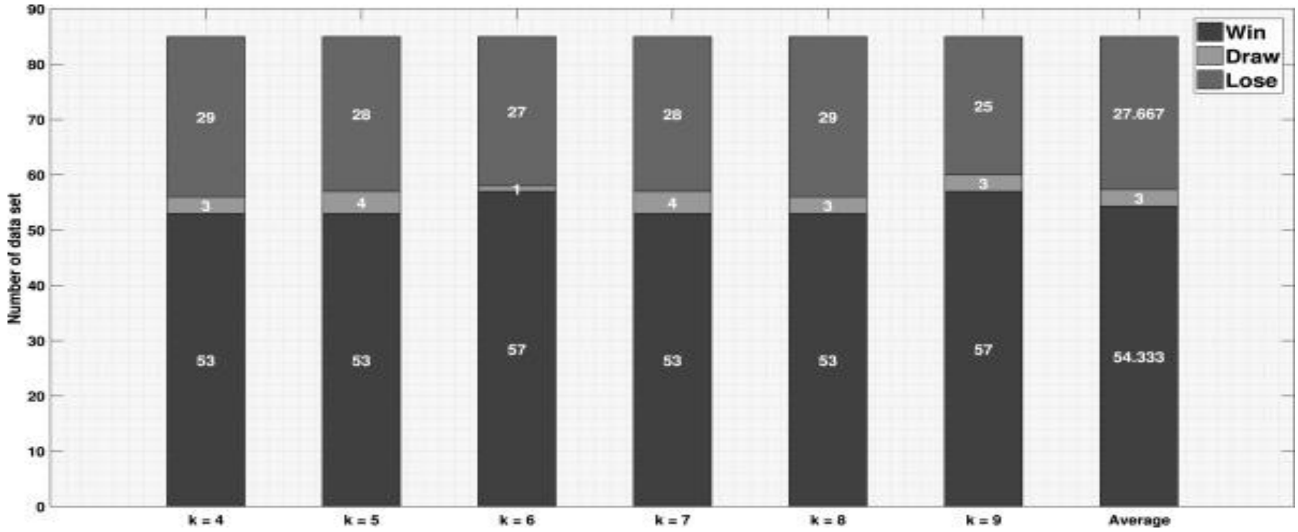

Fig. 11. A bar graph with accumulation compares the WLDBA*k-NN* with 1-NN.

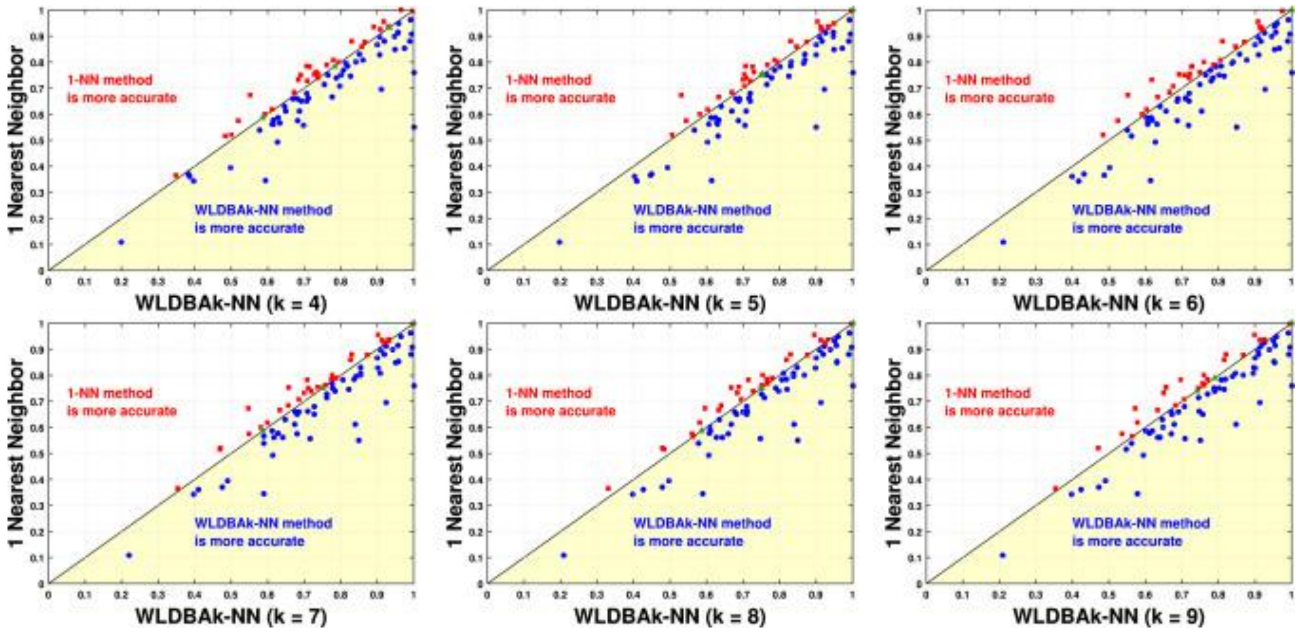

Fig. 12: Spot charts for comparing the WLDBA*k-NN* (K 4 to 9) with 1-NN.

V. DISCUSSION

A. DATA SET

The experiments were carried out on more than 85 data sets from the UCR time series classification archive **[2]**. Each data set consists of many fields such as medicine, engineering, astronomy or signal processing. We have several examples of different types in the UCR 4.2 time series classification archive. Evaluation of WLM*k-NN*

A.1 COMPARISON BETWEEN WLM*K-NN* AND 1-NN

We compare WLM*k-NN* with 1-NN, a widely used algorithm in the field of time series data. It's like rice. 3 WLM*k-NN* exceeds 1-NN in many data sets with different K values. In particular, with *K-sign* it is equal to 6 more than 1-NN on 58 data sets, simply loses on 24 data sets and equals to 1-NN on 3 data sets. Other K values also show that the WLM*k-NN* is better than 1-NN in many data sets. The details of the comparison are shown in Fig. 4, where we build six scatter plots to compare WLM*k-NN* with 1-NN at different K values: from 4 to 9. Each point of the circle on the

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

chart represents one set of data with its accuracy WLM*k-NN*, and its value is its accuracy of 1-NN. The dots in the lower triangle indicate data sets that the WLM*k-NN* exceeds 1-NN in accuracy. In the upper triangle, the points illustrate datasets that are 1-NN more accurate than WLM*k-NN*. The diagonal line of the point is indicated by datasets that WLM*k-NN* and 1-NN have the same accuracy. In general, the WLM*k-NN* method is better in the lower triangle. Note that from this moment on, the value of bar graphs and scattering diagrams in other experiments remains unchanged.

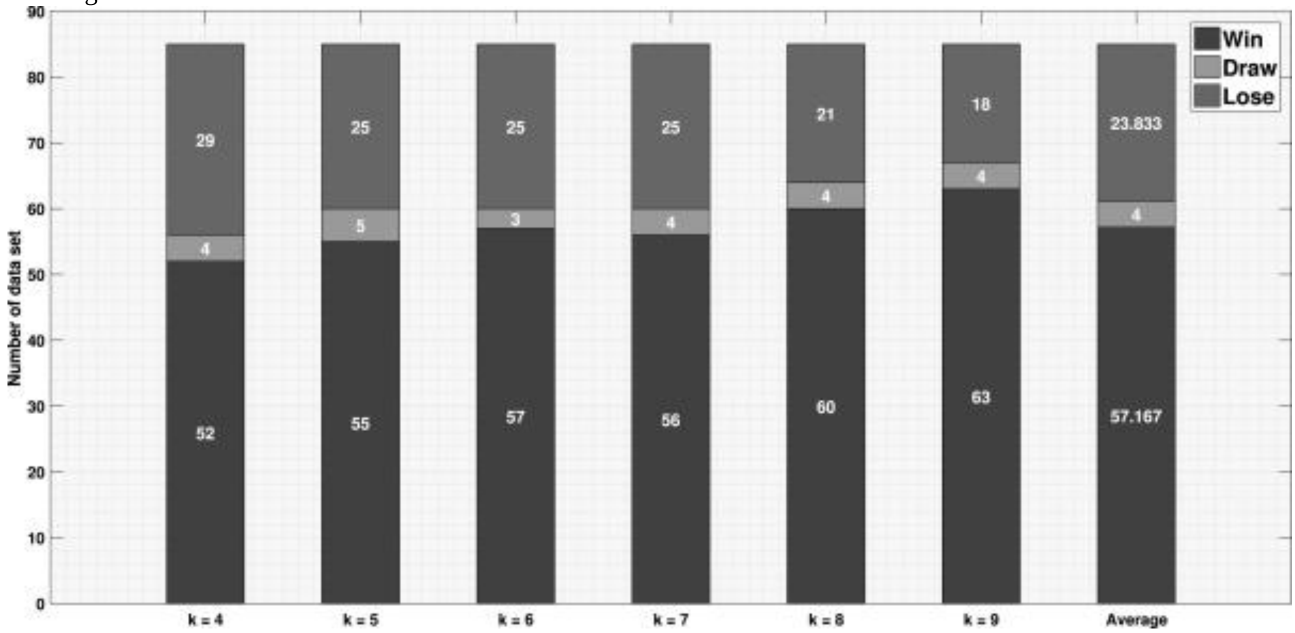

Fig. 13: A bar graph with accumulation compares WLDBAk -NNN with LDBAk -NNN.

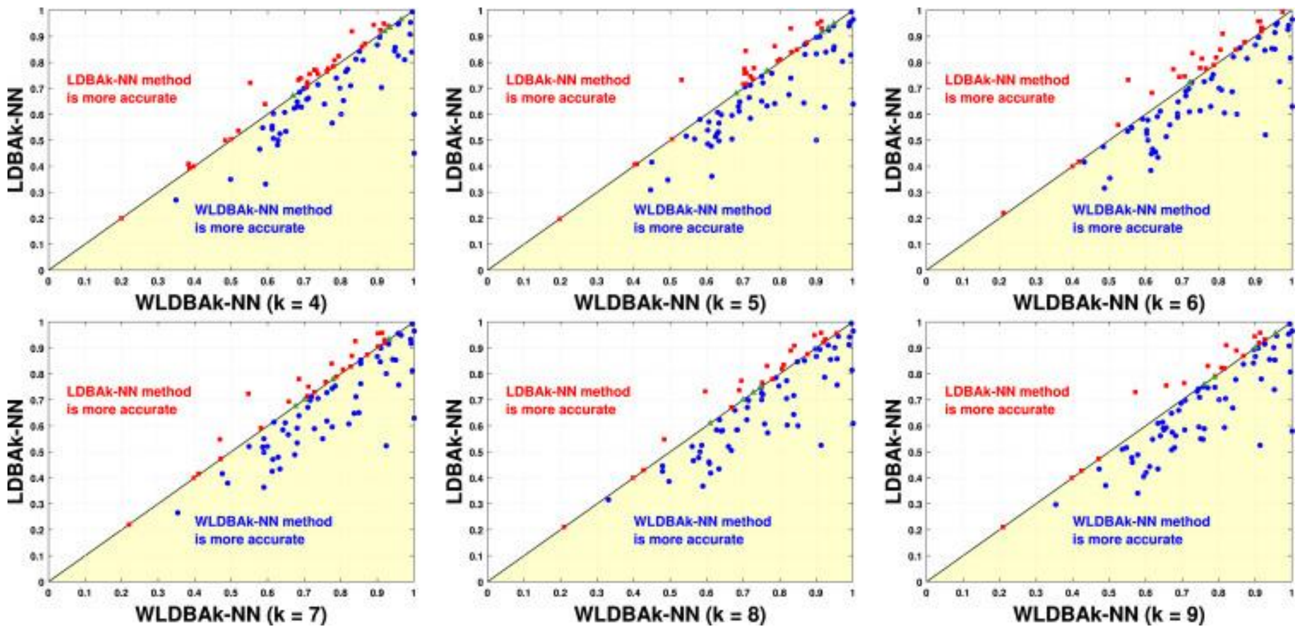

Fig. 14: Spot charts for comparing the WLDBA*k-NN* method with the LDBA*k-NN* method.

ISSN: 2350-0328
International Journal of AdvancedResearch in Science, Engineering and Technology
Vol. 6, Issue 4, April 2019

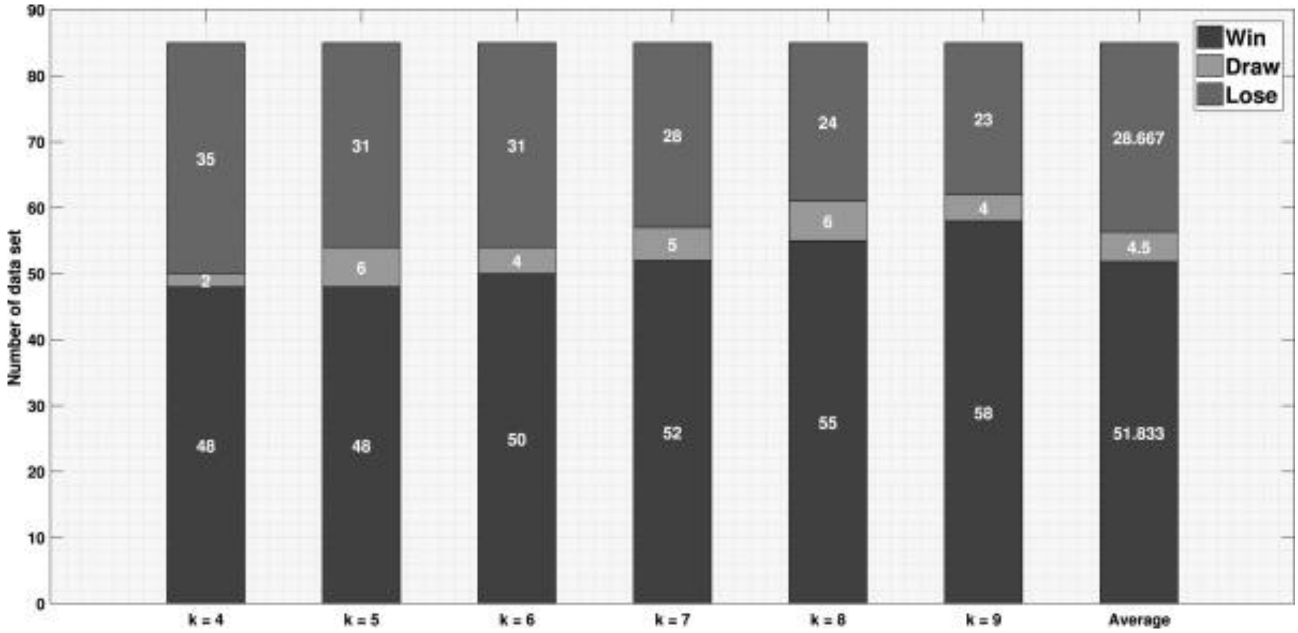

Fig. 15. A bar graph with accumulation compares the WLDBA*k-NN* with the *LDBAk-NCN*.

A.2 Comparison between WLM*k-NN* and *k-NCN*

In this experiment we compare WLM*k-NN* with *K-NCN*. The K-character values are equal to (4.9) . A brief comparison is shown in Fig. 5. As we can see, with *K-sign* equal to 9, WLM*k-NN* wins *k-NCN* on 59 data sets, loses in 21 data sets and pulls from *k-NCN* on 5 data sets. Other *K* values also show that the WLM*k-NN* exceeds *k-NCN* in many data sets. The details of the comparison are shown in Fig. 6.

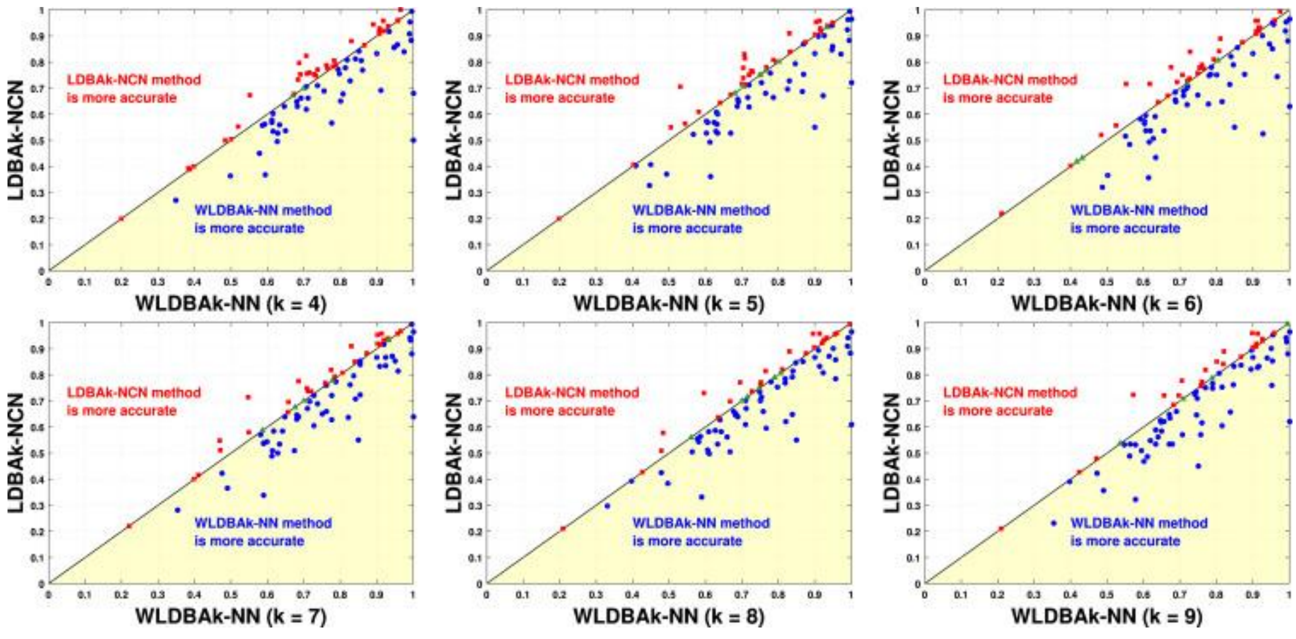

Fig. 16: Spot charts for comparing the WLDBA*k-NN* (K 4 to 9) with the LDBA*k-NCN*.

ISSN: 2350-0328

International Journal of AdvancedResearch in Science, Engineering and Technology

Vol. 6, Issue 4, April 2019

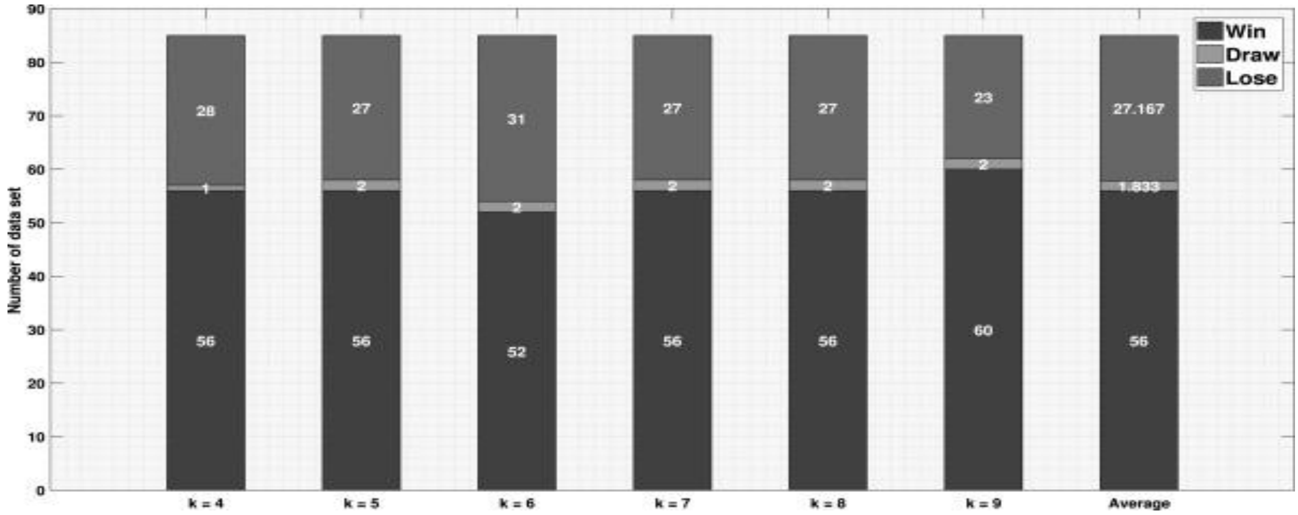

Fig. 17: A bar graph with accumulation compares the WLDBA*k-NN* with WLM*k-NN*.

A.3 Comparison between WLM*k-NN* and LM*k-NN*

In this experiment we compare WLM*k-NN* with LM*k-NN* at different values. As you can see in the rice. 7, WLM*k-NN* exceeds LM*k-NN* in many data sets with different K values. Especially with *K-sign* it wins LM*k-NN* on 62 data sets, loses on 11 data sets and pulls with LM*k-NN* on 12 data sets. With other values K WLM*k-NN* also surpasses LM*k-NN* in the vast majority of data sets. The details of the comparison are shown in Fig. 8.

A.4 Comparison between WLM*k-NN* and LM*k-NCN*

Recent experiments should compare WLM*k-NN* with LM*k-NCN*. In Fig. 9, we show the final comparison that the WLM*k-NN* is ahead of the LM*k-NCN* with 51.5 data sets. In particular, at K = 9 WLM*k-NN* wins in 57 data sets, pulls with LM*k-NCN* in 4 data sets and loses in 24 data sets. And with other K values, WLM*k-NN* also gives better performance than LM*k-NCN*. We show the details of this comparison in Fig. 10, which also uses the scatter plot. In general, the experimental results are presented in sections 4.2.3 Comparison between WLM, 4.2.4 Comparison between WLM and WRC.

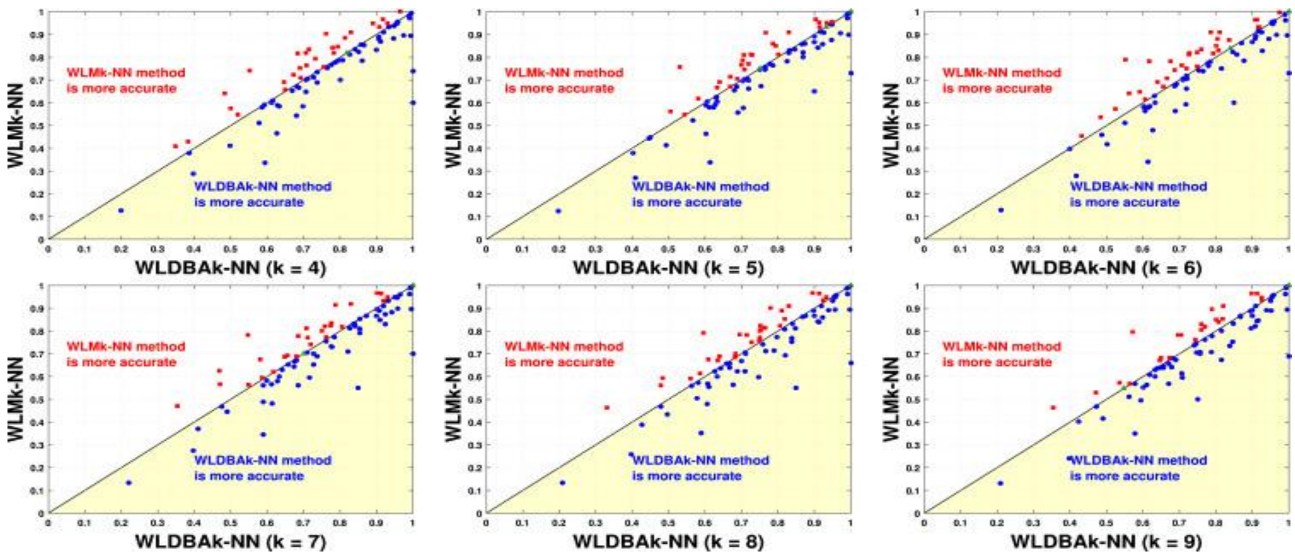

Fig. 18: Spot charts compare the WLDBA*k-NN* method (*K sign* equals 4.9) with the WLM*k-NN* method.

B. WLDBA*k-NN* assessment

B.1 Comparison between the WLDBA*k-NN* and 1-NN

In this subsection we compare WLDBA*k-NN* with 1-NN. The results of the experiment in Fig. 11 show that the number of winnings in the WLDBA*k-NN* is greater than in 1-NN. Especially, as shown in Fig. 11, the WLDBA*k-NN* wins 1-NN in 57 data sets, loses in 25 data sets and pulls from 1-NN in 3 data sets. The scattering graphs in Fig. 12 give a detailed view of the comparison.

B.2 Comparison between WLDBA*k-NN* and LDBA-NN

In this experiment we reintroduced LM*k-NN* using DBA to find local medium vectors and called it LDVA-NN. Experimental results show that the WLDBA*k-NN* exceeds the LDBA*k-NN* at different *K sign* values of 4.9‾. As shown in Fig. 13, the WLDBA*k-NN* exceeds LDBA*k-NN* in many data sets with different K values. In particular, with *K-sign* 9 WLDBA*k-NN* wins LDBA*k-NN* on 63 data sets, loses on 18 data sets and pulls with LDBA*k-NN* on 14 data sets. In other values of K, the proposed WLDBA*k-NN* also surpasses the LDBA*k-NN* in the vast majority of data sets. The details of the comparison are shown in the scattering charts in Fig. 14.

B.3 Comparison between the WLDBA*k-NN* and LDBA-NCN

In this experiment, we reintroduced the LM*k-NCN*, using the DBA to find the local medium vectors and called it the *k-NCN* LDBA. The diagram in Fig. 15 shows the number of winnings, losses and draws of the WLDBA*k-NN* *compared* to the LDBA*k-NCN*. As shown in Fig. 15, the average winning speed of the WLDBA*k-NN* is higher than that of the LDBA*k-NCN*, which is noticeable in almost two thirds of the data sets. We will show a detailed comparison in Fig. 16 on point charts.Based on the experimental results presented in sections 5.3.2 Comparison between WLDBAs, 5.3.3 Comparison between WLDBAs, we found that WDBAs outperform DBAs when used to classify time series.

B.4 Comparison between WLDBA*k-NN* and WLM*k-NN*.

The experimental results presented in section 4.2 showed that the WLM*k-NN* exceeds 1-NN, *k-CNNN*, LM*k-NN*, LM*k-NCN*. In this latest experiment, we estimate the WLDBA*k-NN* *compared to* the WLM*k-NN* as shown in Fig. 17. The experimental results show that the WLDBA*k-NN* exceeds the WLM*k-NN*. In particular, with *K-sign* equal to 9, WLDBA*k-NN* wins in 60 data sets, uses 2 data sets and loses in 23 data sets. With other KWLDBA*k-NN* values, *k-NN* also gives better performance than WLM*k-NN*. Details of comparison are shown in Fig. 18 on point diagrams.

VI. CONCLUSION

In this article we proposed a new nonparametric method of time series classification based on *k* - nearest neighbors and dynamic averaging of Baricenters with time deformation. The proposed method is called the weighted local DTW-baricenter of averaging *k* - nearest neighbors (WLDBA*k-NN*). We also proposed a new method of time series averaging by improving the method of averaging Baricenters of dynamic deformation by time and called a new method of weighted dynamic averaging of Baricenters by time (WDBA). WLDBA *k -NNN* calculates the local average vector value using WDBA. We evaluate the proposed method for 85 data sets in the big archive series classification compared to WLM *k-N*, 1-NN, LM*k-NN*, *k-NCN* and LM *k-NCN* on the same data sets. The results of the experiments show that the WLDBA*k-NN* gives the best results among the methods used in the experiments. When we look inside the experimental results, we see that the WLDBA*k-NN* and WLM*k-NN* surpasses 1-NN, LM*k-NN*, *k-NCN* and LM*k-NCN* on almost all experimental data except for the image of data sets. Note that, although the experiments in this article are conducted on data sets built for time series classification, the WLDBA*k-NN* can be applied to other tasks of time series. As far as future work is concerned, we plan to further study remote measures different from Euclidov or dynamic time distortion and evaluate the WLDBA*k-NN* by the diversity of time series problems.

REFERENCES

[1]Camps-Valls, G., Tuia, D., Bruzzone, L., Benediktsson, J.A., 2014. Advances in hyperspectral image classification: earth monitoring with statistical learning methods. IEEE Signal Process. Mag. 31 (1), 45–54.

[2]Gou, J., Yi, Z., Du, L., Xiong, T., 2012. A local mean-based k-nearest centroid neighbor classifier.Comput. J. 55, 1058–1071.

[3]Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y., Keogh, E., 2014. Dynamic time warping averaging of time series allows faster and more accurate classification. In: Proceedings of the 2014 IEEE International Conference on Data Mining. In: ICDM ’14, IEEE Computer Society, Washington, DC, USA, pp. 470–479.

[4]Tran, T.M., Le, X., Vinh, V., Nguyen, H., Nguyen, T., 2017. A weighted local mean-based k-nearest neighbors classifier for time series. In: Proceedings of The 9th International Conference on Machine Learning and Computing. In: ICMLC 2017, ACM, pp. 157–161.