Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Polar Formed Histogram of Fast Fourier Transform Feature for Time Series Classification Problems

K. Kiaei, H. Omranpour*

Department of Electrical and Computer Engineering, Babol Noshirvani University of Technology, Babol, Iran.

Article Info

Article History:
Received 06 July 2021
Reviewed 15 August 2021
Revised 25 October 2021
Accepted 01 November 2021

Keywords:
Time series
Classification
Fast Fourier transform
Polar form
Random forest classifier

*Corresponding Author's Email Address:
h.omranpour@nit.ac.ir

Abstract

Background and Objectives: Time series classification (TSC) means classifying the data over time and based on their behavior. TSC is one of the main machine learning tasks related to time series. Because the classification accuracy is of particular importance, we have decided to increase it in this research.

Methods: In this paper, we proposed a simple method for TSC problems to achieve higher classification accuracy than other existing methods. Fast Fourier transform is a method that uses in raw time series data preprocess. In this study, we apply the fast Fourier transform (FFT) over the raw datasets. Then we use the polar form of a complex number to create a histogram. The proposed method consists of three steps: preprocessing using FFT, feature extraction by histogram computation, and decision making using a random forest classifier.

Results: The presented method was tested on 12 datasets of the UCR time series classification archive from different domains. Evaluation of our method was performed using k-fold cross-validation and classification accuracy. The experimental results state that our model has been achieved classification accuracy higher or comparable than related methods. Computational complexity has also been significantly reduced.

Conclusion: In the latest years, the TSC problems have been increased. In this work, we proposed a simple method with extracted features from fast Fourier transforms that is efficient to gain more high accuracy.

©2022 JECEI. All rights reserved.

Introduction

The amount of observations of a variable in many real-world applications depends on its value in earlier times. Such data are referred to as time series, that various researchers have used data mining techniques to predict and analyze such data.

The first step for time series identification and their nature determination is their past study. It can lead to an accurate and reliable prediction about its class [1].

Classification is a task that can be useful in many domains and applications. Any activity that requires thinking and judgment can be a matter of classification [2]. Medical imaging classification is an important classification problem because it plays a critical role in improving the diagnosis of many diseases [3], [4].

On the other hand, every problem using data that is registered taking into account some notion of order can be cast as a TSC problem [5].

In many real applications uses such as recognizing motion imagination by EEG signals, human activity classification based on sound recognition, not only sounds from human activities but also sounds from related objects [6], model prediction with electronic health record data is anticipated to drive personalized medicine and improve healthcare quality. If successful it could provide significant benefits not only for patient

safety and quality but also in reducing healthcare costs [7].

Different classification methods have been proposed so far, but what separates TSC problems from usual classification problems is that the order of features in time series is essential. For this reason, must use the methods that retain the best classification features to obtaining acceptable accuracy. The best classification features may be distorted by noise or preprocessing. As a result, classifying time series requires special techniques.

In general, solutions to TSC problems are separated into two separate groups: shape-based and structure-based. In solving shape-based problems, classifiers are classically used in combination with a similarity criterion. But the tendency in solving TSC problems is to discover small patterns to be representation time series classes. Between these patterns, shapelet-based techniques [8]-[10] have gained good classification accuracy. Structure-based methods extract feature vectors from time series, such as feature patterns, or convert a time series into various representations, which often have a high computational cost.

Preliminaries

In this section, we hold forth a few preliminaries and sum up the meaning of frequency notations in Table 1.

Table 1: Notations and their meaning

Notation	Meaning
T	Time series $T = (t_1, t_2, \dots, t_n)$ of the length n
D	Time series T_j without the class label
Z	FFT of $t_{ij}, 1 \le i, j \le n$
F	Fourier coefficient matrix of dataset D
C	the label set of dataset D
R	length of the Z
θ	The angle between Z and the real axis
H	Histogram matrix
n	Radius matrix is divided into n equal parts.
m	Angle matrix is divided into m equal parts.

A time series T is an ordered-value sequence $T = (t_1, t_2, \dots, t_i, \dots, t_n)$, where n is the length of T, t_i is the value observed at timestamp i.

A time series dataset D is a set of time series T_j with C_j = label(T_j); $j \in [1; m]$, where $C_j \in C$ is the class label of the dataset, $C = \{0,1,2, \dots, |C| - 1\}$, and $|C|$ denotes the number of classes.

The discrete Fourier transform is the primary transform used for numerical computation in digital signal processing. It a sequence of N complex numbers $\{x_n\} \coloneqq x_0, x_1, \dots, x_{n-1}$ into another sequence of complex numbers, $\{X_k\} \coloneqq X_0, X_1, \dots, X_{n-1}$ which is defined by:

$$\mathbf{X_k} = \sum_{j=0}^{n-1} x_j e^{\frac{-2\pi i}{n} kj} \quad (1)$$

These implementations usually employ efficient fast Fourier transform (FFT) algorithms. Applications of Fourier transform include data filtering [11], pattern extraction, and audio processing [12].

Related Work

The critical characteristic that separates TSC issues from the general classification task is that the ordering of the attributes is significant. The best discriminative features for classification may be concealed by the length of the series, embedded in the interaction of observations, or confounded by noise in the phase of the series. Hence, TSC, by and large, need strategies explicit to the nature of the issue.

There are many techniques for the TSC issue. [13] is a great review article that interested people can refer to it. In this section, we mainly survey two significant representative methods, called structure-based methods, and shapelet-based methods. This paper undertakes the structured-based approach with FFT.

A. Shapelet-Based Method

The shapelet-based approach utilizes some similarity criteria to gauge the distance between time series on raw numeric data.

A customary 1NN classifier based upon Euclidean distance [14] or DTW [15] distance is an example of this category. In [16] introduced time series shapelets and proposed the seminal work of it. Algorithms based on the time series shapelet can be interpretable but need high computation time. ST [17] suggests selecting shapelets by extracts the k best shapelets from a dataset in a single pass to represent the original time series. SVM and Neural Networks can be applied to determine shapelets. SD [18] proposed a quick shapelet discovery method that prunes the candidates based on a distance threshold to previously considered other similar candidates. FS [10] proposed an algorithm for shapelet discovery that is fast with transforming the raw time series data into SAX words. The run time is reduced when compared to [16]. However, accuracy is lower than other works [13]. SAX-VFSEQL [19] creates a pool of SAX words as the shapelets and iteratively optimizes the weights of the shapelet. However, this method still cannot produce shapelets, which there isn't in training data. Too, the discovered shapelets are too various and redundant, which makes interpreting the classification decision difficult.

Shape-based methods can achieve high accuracy, but

they have a lot of computational costs. So a few approaches are proposed to progress efficiency, such as early abandoning [20]. In addition, shape-based methods are sensitive to noisy data, particularly for long time series.

B. Structured-based method

There are a few Structure-based methods such as Piecewise Aggregate Approximation (PAA) [8], Symbolic Aggregate approximation (SAX) [19], [21], Symbolic Fourier Approximation (SFA) [22], Discrete Fourier Transform (DFT) [23] and other discretization techniques. The dictionary-based method is another significant example. These techniques extract higher-level feature vectors or build a model from the time series before the classification task using classification algorithms such as KNN, SVMs, or random forests. Bag-of-Patterns (BOP) [24] builds a classifier from SAX, calculating the Euclidean distance with the BOP representation of a new instance to specify its class. Another approach, SAXVSM [25], adopts SAX techniques, proposes a vector space model (VSM), and uses TF-IDF to rank time series patterns. In [19], combine various variable-length bag-of-symbolic-words representations and an efficient linear sequence learning approach (SAX-VSEQL [26]) for efficient TSC.

Structure-based methods have resistant to noisy data because of the smooth impact of time series representations. Because of the shortening of representations, these methods are more efficient than shape-based methods.

Some algorithms like BOSS VS[27], which transforms data from the time domain to the frequency domain, give high accuracy, but they can't interpret the classification decision.

Proposed Method

Details of the proposed method are provided in this section. Our method consists of 3 phases. In the first phase, we perform preprocessing on the raw dataset. In the second phase, we prepare a histogram of new data, and in the last phase, we obtain the accuracy of our classifications by using the appropriate classifier. The overview of the method showed in Fig. 1.

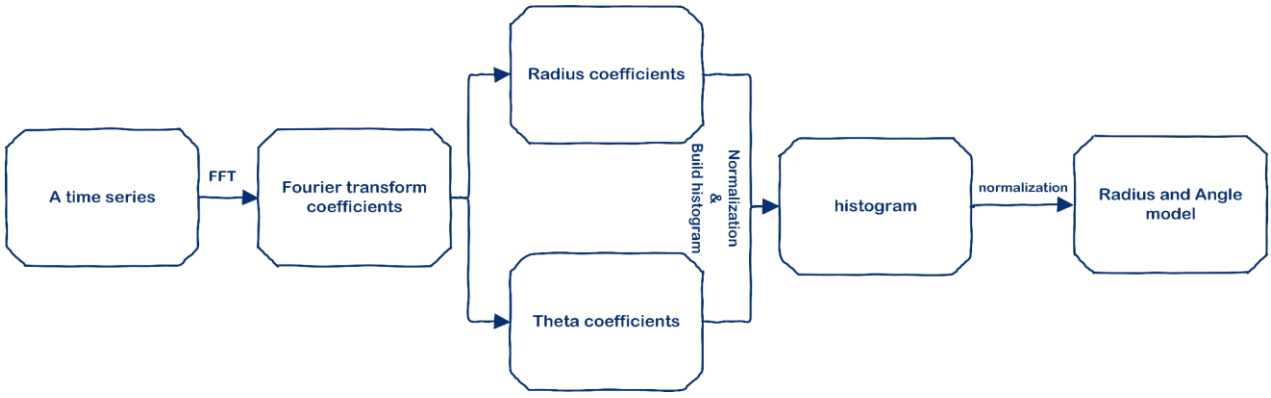

Fig. 1: The overview of our model.

A. Preprocessing

I) Fourier transform

A quick look at a time series shows us that it may be hard to interpret. It's because the underlying pattern behind the jagged peaks and troughs identify hardly in its raw form. Conversions on time series like scaling, shifting, time scaling, time-shifting, and dynamic time warping facilitate the discovery of flexible time series patterns. The high computational efficiency of Fourier transform has made Fourier analysis the preferred method for many applications. Fourier transform-based methods are very well used in all fields of science and engineering [28].

Therefore, in this paper, we use FFT on raw data in the preprocessing step.

In this case, we have a matrix of complex numbers (F) whose rows and columns are the original dataset size.

After applying FFT, we have a complex number $Z = a + bi$ in all elements of matrix F. Fig. 2 shows a complex number representation.

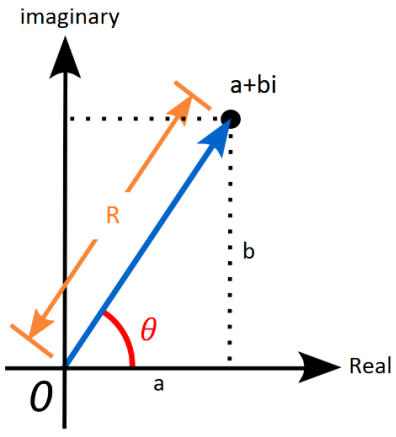

Fig. 2: Complex number representation.

The complex number can also be represented in polar form by (2).

$$\boldsymbol{Re^{i\theta}\ or\ R(cos\theta + isin\theta)} \tag{2}$$

As shown in Fig.2, R is the length of Z and θ is the angle between Z and the real axis. R and θ are obtained from formulas (3) and (4).

$$\boldsymbol{R = |Z| = \sqrt{(a^2 + b^2)}} \tag{3}$$

$$\theta = \begin{cases} \arctan\left(\frac{b}{a}\right) & if\ a > 0 \\ \arctan\left(\frac{b}{a}\right) + \pi & if\ a < 0\ and\ b \geq 0 \\ \arctan\left(\frac{b}{a}\right) - \pi & if\ a < 0\ and\ b < 0 \\ \frac{\pi}{2} & if\ a = 0\ and\ b > 0 \\ -\left(\frac{\pi}{2}\right) & if\ a = 0\ and\ b < 0 \\ indeterminate & if\ a = 0\ and\ b = 0 \end{cases} \tag{4}$$

In (3) and (4), a is the real part, and b is the imaginary part of a complex number. Now we derive the radius and angle from each element of F. In this case, we have two matrices, radius matrix and angle matrix, with the dimensions of the original matrix, one of which stores the R values and the other the θ values (Fig. 3).

II) Normalization

Normalization is a good technique used in situations where we do not know how to distribute the data. Normalize data is suitable when the data are at different scales. In addition, the method we use doesn't have any assumptions regarding our data. Change values without eliminating the difference between them are the goal of normalization. Normalization cause converting all the numeric data to the desired scale.

Equation (5) is the general formula for converting data to a range between 0 and 1:

$$\boldsymbol{x_{normalized} = \frac{x - min(x)}{max(x) - min(x)}} \tag{5}$$

Now all elements of radius and angle matrices are between 0 and 1.

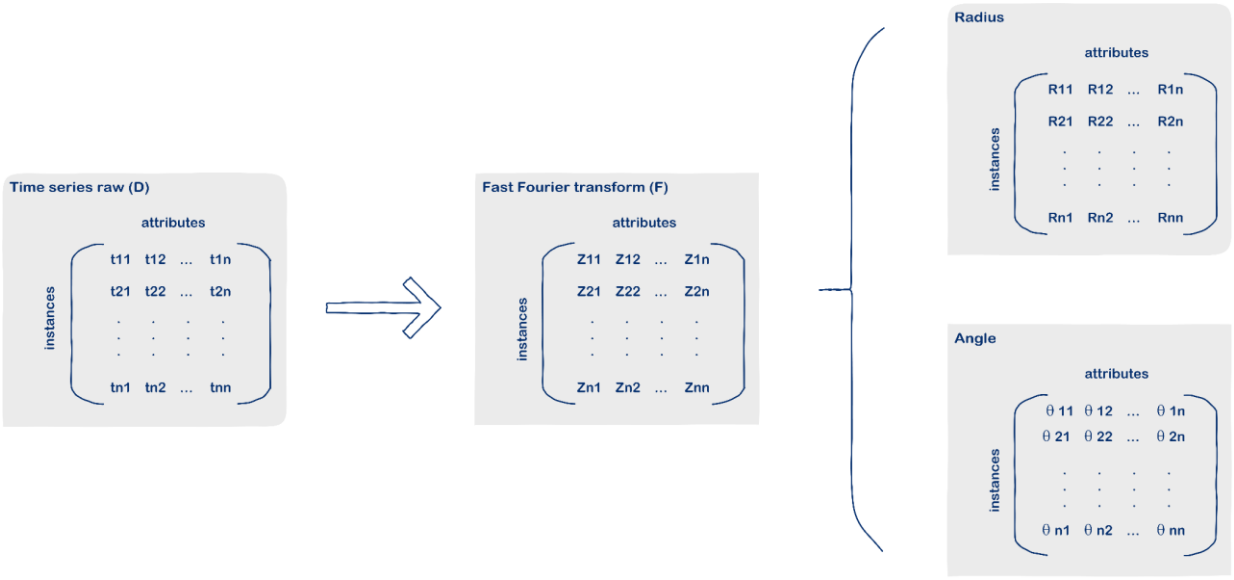

Fig. 3: Derive the radius and angle from F.

B. Histogram computation

A histogram gives us a visual interpretation of numerical data by appearing the count of data points inside a particular range called a bin. To histogram computation, we divided radius and angle matrices into n and m equal parts, respectively. Radius and angle matrices are normal, and the range is between 0 and 1, so the bin size for the radius matrix will be $\frac{1}{n}$, and the bin size for the angle matrix will be $\frac{1}{m}$. This work will usually reduce dimensions.

Equation (6) specifies each Z falls within which bin. We have to do this for all F elements.

$$\frac{\mathbf{j-1}}{\mathbf{n}} \leq \mathbf{R} \leq \frac{\mathbf{j}}{\mathbf{n}} \quad \& \quad \frac{\mathbf{k-1}}{\mathbf{m}} \leq \theta \leq \frac{\mathbf{k}}{\mathbf{m}} \tag{6}$$
$$\mathbf{for}\ \mathbf{1 \leq j \leq n} \quad \& \quad \mathbf{1 \leq k < m}$$

n and m are variables in (6). It is clear larger n and m, fewer data to be lost, but it also causes the uniformity of the histogram. That means the effect of the proposed feature will be reduced. The best n and m values that lead to high accuracy are listed in Table 3.

For implementation, we consider the matrix H, whose number of rows is the number of original dataset rows and the number of columns is $n * m$. Set the initial value of all H elements to 0. According to Algorithm 1, we have to place each element of row i of the original dataset in one of the rows i columns of the histogram. That means in row i and column j of the original dataset, we add a unit to the value of the element in row i and column $(j-1)*m+k$ from H. The result of this operation is the histogram calculation (Fig. 4).

Algorithm 1: Histogram computation

Input: $R_{1,:}, \theta_{1,:}, n, m$

Output: $H_{1,:}$

1. Initialize H = 0, c=size(R, 2)
2. For i=1 to c
3. For j=1 to n
4. For k=1 to m
5. If$\left(\frac{j-1}{n} <= R_{1,i} \,\&\, R_{1,i} < \frac{j}{n}\right) \& \left(\frac{k-1}{m} \leq \theta_{1,i} \,\&\, \theta_{1,i} < \frac{k}{m}\right)$
6. $H_{(j-1)*m+k}++$
7. End If
8. End For
9. End For
10. End For
11. Return $H_{1,i}$

C. Classifier

The classifiers used in this article are K-nearest neighbor for simple implementation, fast and efficient [29], support vector machine for the ability to deal with various classification problems like not linearly separable problems and high dimensional [30], and random forest for power, accuracy, can also be used to rate the importance of features in classification or regression problems [31]. Among these classifiers, the results of random forest were better than the others. Only the results of the random forest classification are given in the next section.

Algorithm 2: Feature-extraction

Input: time series dataset D

Output: H

1. Initialize $L = size(D, 2), r = size(normalR, 1)$
2. For $i = 1\ to\ L$
3. $A = FFT(D_{i,:})$
4. $R_{i,:} = |A|$ #Absolute value
5. $\theta_{i,:} = arg(A)$ # argument
6. End For
7. $normalR = (R - min(R))/(max(R) - min(R))$
8. $normal\theta = (\theta - min(\theta))/(max(\theta) - min(\theta))$
9. For $i = 1\ to\ r$
10. $H_{i,:} = Histogram - computation(normalR_{i,:}, normal\theta_{i,:}, n, m)$
11. End For
12. Return H

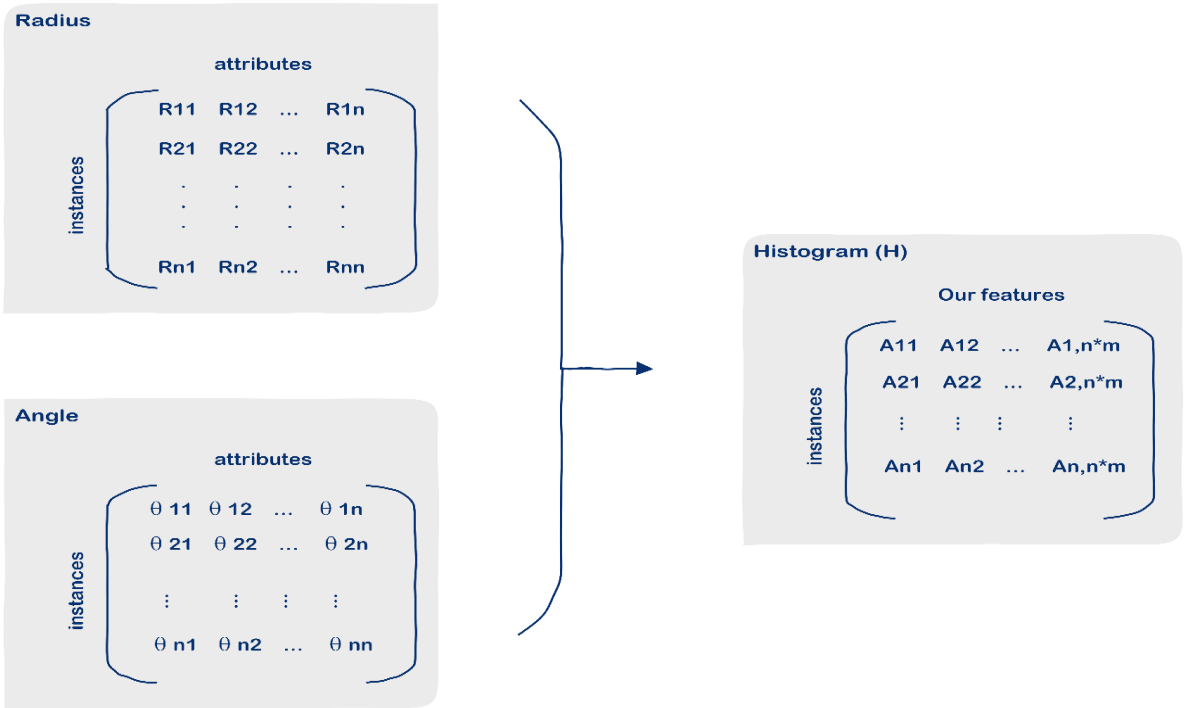

Fig. 4: The histogram matrix.

Results and Discussion

In this section, we show the results of the proposed method on the benchmark dataset used for TSC problems. The algorithm of this article is implemented with MATLAB. All the state-of-the-art methods were conducted on a machine with a Xeon E5-2630 v4 @ 2.2GHz (2S/10C) / 256GB RAM / 128GB SWAP, running on CentOS 7.6 (64-bit). Also, the proposed method was run on a computer with an Intel Core i7-4720HQ 4 x 2.6/ 16GB RAM DDR3-1600/ 256GB SSD, running on windows 8.1.

We can adjust the histogram computation parameters easily since they are efficient at using memory. Cross-validation has been used to evaluate performance.

We compared this method with a few baselines, including BSPCOVER [32] that is both efficient and accurate, RotForest [13] and 1NN-DTW [15] that used as benchmark classifier, and ST [17], ELIS [8], Fast shapelets [10], Scalable Shapelet Discovery [18] that are shapelet-based methods and COTE [33].

A. Datasets

This paper used the UCRARCHIVE [34], the important resource of TSC datasets, to evaluate the model.

We just chose 12 datasets from the various types like Image (4), Motion (2), Spectro (2), Stimulated (2), Sensor (1), ECG (1) because of space limitations. The number of each type is shown in parentheses.

Table 2 shows us information about each dataset, such as the number of instances in the train and test set, number of features, number of classes.

B. Experiments

The classification accuracy results of previous methods and proposed method on 12 datasets are given in Table 4.

I. Analysis of the proposed idea parameters

By repeating the experiments, we tuned the best value for the parameters of the proposed model. In each dataset, we tested numbers smaller than 50 with an interval of 5. Increasing one or two units has little effect on improving accuracy, and this distance allows us to find the best values in the fastest possible time. The values n and m are selected from (7).

$$\boldsymbol{n, m = 5i} \qquad \boldsymbol{i = 1, 2, \dots, 10} \tag{7}$$

Table 2: Datasets

Dataset	Train set	Test set	Length	Class	Type
BeetleFly	20	20	512	2	IMAGE
ChlorineConcentration	467	3840	166	3	SIMULATED
Coffee	28	28	286	2	SPECTRO
DiatomSizeReduction	16	306	345	4	IMAGE
DistalPhalanxOutlineCorrect	600	276	80	2	IMAGE
Earthquakes	322	139	512	2	SENSOR
ECG5000	500	4500	140	5	ECG
Haptics	155	308	1092	5	MOTION
InlineSkate	100	550	1882	7	MOTION
Mallat	55	2345	1024	8	SIMULATED
Meat	60	60	448	3	SPECTRO
Symbols	25	995	398	6	IMAGE

Table 3: Best values of m and n

dataset	BeetleFly	Chlorine Concentration	Coffee	DiatomSizeReduction	DistalPhalanxOutline Correct	Earthquakes	ECG5000	Haptics	InlineSkate	Mallat	Meat	Symbols
n	10	20	15	35	5	5	10	10	5	20	25	15
m	5	10	10	5	20	20	10	15	10	20	40	10

Table 4: Classification accuracy of the proposed method and state-of-the-art methods on UCRARCHIVE

Dataset	ST – 2012	FS – 2013	COTE – 2015	SD – 2016	DTW Rn 1NN – 2017	RotF – 2017	ELIS – 2018	ResNet – 2019	BSPCOVER – 2020	Proposed method
BeetleFly	90	70	80	75	65	90	85	85	90	90
ChlorineConcentration	69.97	54.64	72.71	55.3	65	84.74	27.39	84.4	61.22	90.08
Coffee	96.43	92.86	100	96.1	100	100	96.43	100	100	98.21
DiatomSizeReduction	92.48	86.6	92.81	89.6	93.46	87.25	89.86	30.1	87.25	96.89
DistalPhalanxOutlineCorrect	77.54	75	76.09	71.7	72.46	75.72	57.83	77.1	83.17	98.58
Earthquakes	74.1	70.5	74.82	63.6	72.66	74.82	77.64	71.2	81.68	80.69
ECG5000	94.38	92.27	94.6	92.4	92.51	94.58	72.69	93.4	94.44	93.6
Haptics	52.27	39.29	52.27	35.6	41.56	43.83	41.56	51.9	45.13	50.11
InlineSkate	37.27	18.91	49.45	38.5	38.73	37.09	35.46	37.3	38.73	51.23
Mallat	96.42	97.61	95.39	92.6	91.43	94.93	81.58	97.2	76.8	96.71
Meat	85	83.33	91.67	93.3	93.33	96.67	55	96.8	75	99.17
Symbols	88.24	93.37	96.38	90.1	93.77	79.3	78.29	90.6	93.37	93.14
Total best acc	2	1	4	0	1	2	0	1	3	6
Rank mean	4.16	6.58	3	6.5	4.91	4.33	7.08	4	4.25	2.08
final rank	4	9	2	8	7	6	10	3	5	1

In the training phase, the best values of n and m are chosen, given in Table 3.

Fig. 5 shows the classification accuracy for different values of n and m parameters in each dataset examined in this paper. According to the results, increasing n and m increases the accuracy, but this does not mean that this always happens. Not only do very high values of n and m not help improve accuracy, but they also reduce the effect of histograms on data classification, resulting in less accuracy.

The highest values are different for each dataset, but on average, $n, m = 5i \quad i = 1,2,3,4$ have the best results.

II. Analysis of the proposed features

As described in the previous section, our method is applied in 3 steps: preprocessing, feature extraction, and classifier. To ranking the importance of features in classification issues can use a random forest that rating features based on two criteria called Mean Decrease Impurity (MDI) [35] and Mean Decrease Accuracy (MDA) [35]. MDI is based on the total decrease in node impurity from splitting on the variable, averaged over all trees. Also, MDA is based on the concept that if a feature is not significant, so by changing its value, the accuracy of the prediction should not decrease [31]. To evaluate this effectiveness method, we measured the importance of our features after applying a random forest classifier to the training and test data set.

a) Beetle Fly

Fig. 6 shows how our prepared method discriminates the different classes of the BeetleFly dataset. Fig. 6 (a) a time series plot is presented to visualize data points at consecutive time intervals. To make it clear, we consider five random samples from each class. We extracted ten important features from our model [35] and showed them in Fig. 6 (b), respectively.

The three most important are features 8, 6, and 3, respectively. We show the scatter plot by combining these three properties in Fig. 7 (a)-(b)-(c) that tell us the proposed feature was able to separate the two classes to increase the accuracy of the classification.

b) Chlorine Concentration

Fig. 8 shows how our prepared method discriminates the different classes of the Chlorine Concentration dataset.

Fig. 8 (a) A time series plot is presented to visualize data points at consecutive time intervals. To be clear, we consider five random samples from each class. Fig. 8(b) shows ten features of the model that are more important in predicting data class [35]. The diagram of random forest classification on our model is shown in Fig. 9 that tells us the data of one class have different values than other classes, and they are similar to themselves in the proposed properties. Therefore, it can be concluded that the proposed feature can be a convenient feature for TSC problems. The accuracy we obtained for this dataset is 90.08, significantly higher than all the methods reviewed. In this dataset, after obtaining the importance of each feature [35], we added them to the model in the most important. After adding each feature, we measured the accuracy using the KNN and random forest classifiers. In Fig. 10, we showed the results of the first 100 features.

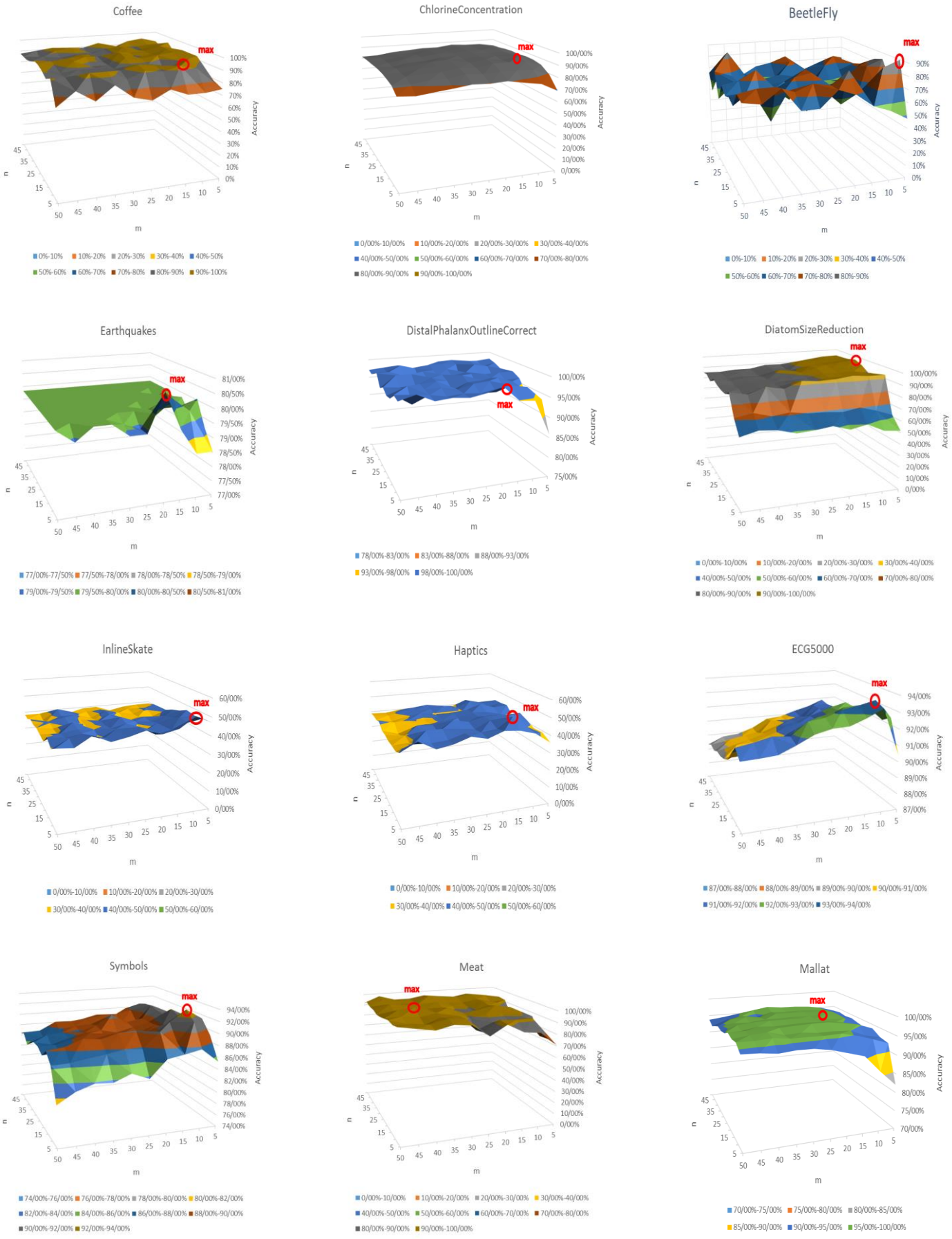

Fig. 5: Accuracy by varying n and m on different datasets.

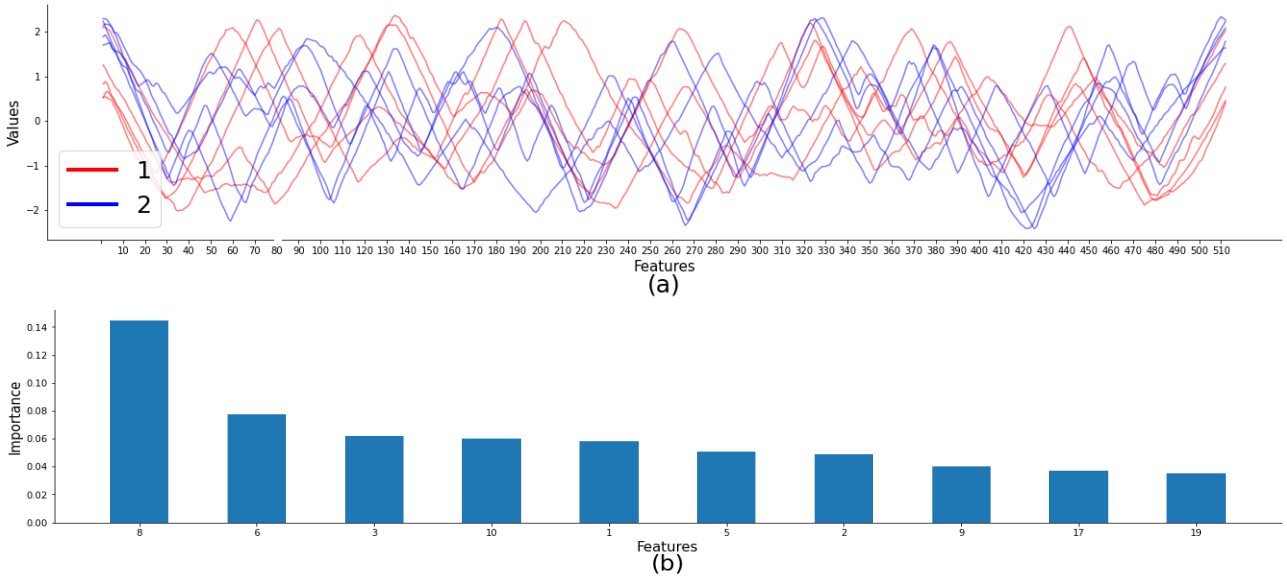

Fig. 6: BeetleFly dataset; (a) Train and test data. To make it clear, five random samples were taken from each class. (b) Feature importance obtained from random forest classifier.

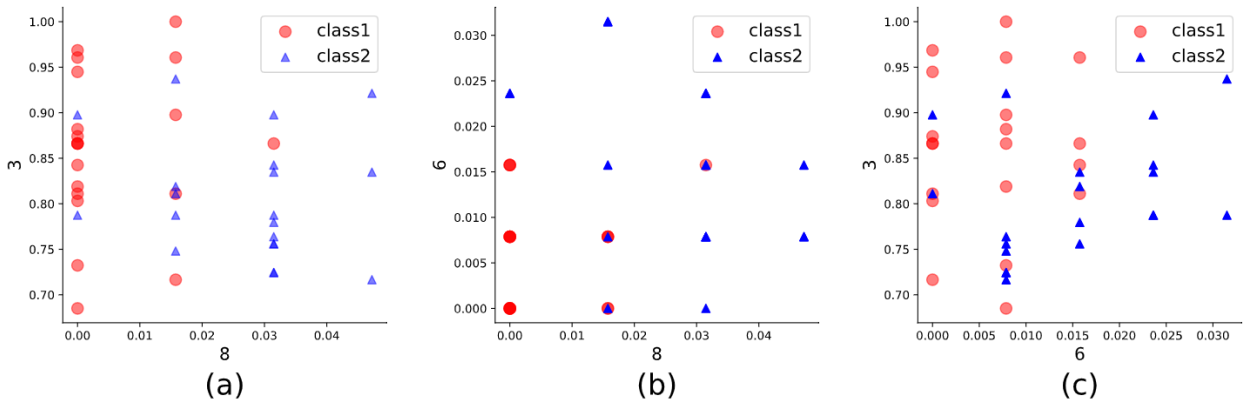

Fig. 7: Scatter plot of BeetleFly dataset; (a) Scatter plot by features 8 and 3. (b) Scatter plot by features 8 and 6. (c) Scatter plot by features 6 and 3.

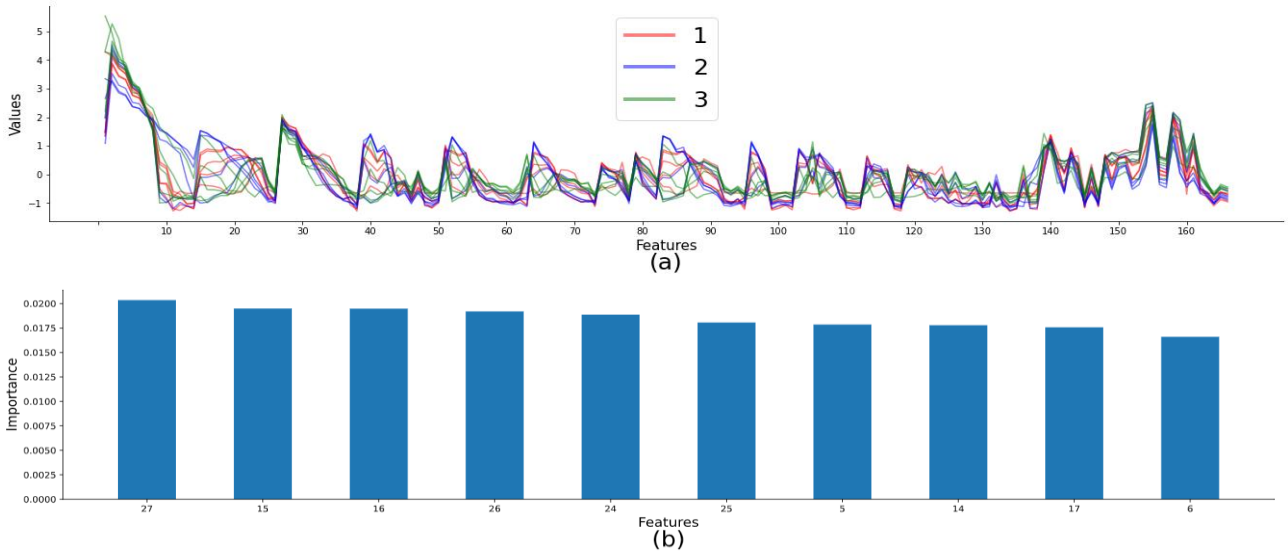

Fig. 8: ChlorineConcentration dataset; (a) Train and test data. To make it clear, five random samples were taken from each class. (b) Feature importance obtained from random forest classifier.

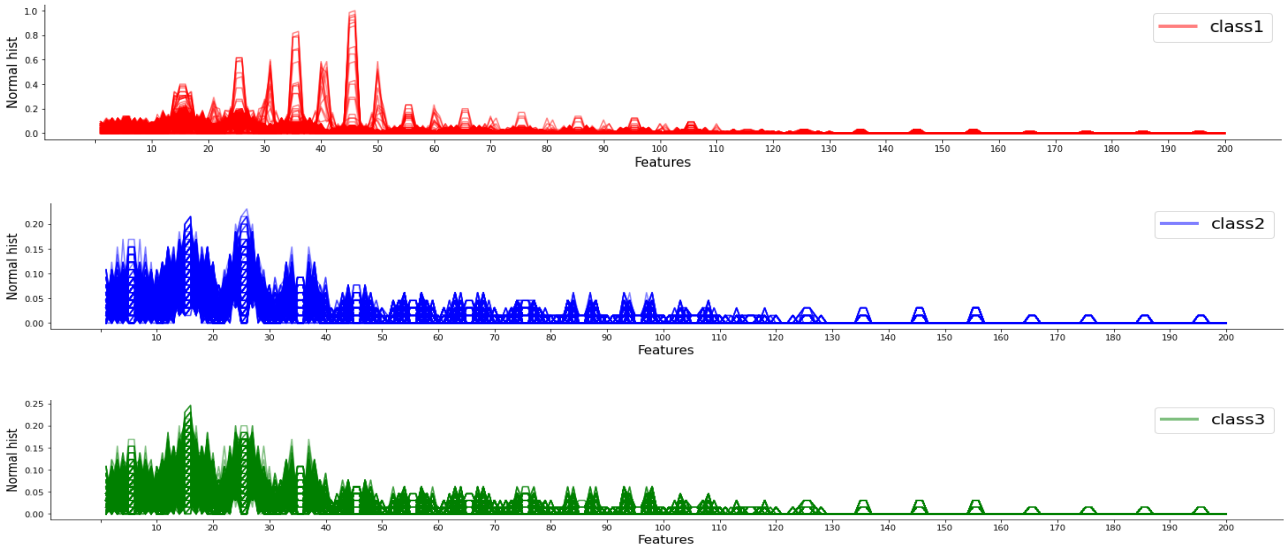

Fig. 9: The diagram of random forest classification in each class.

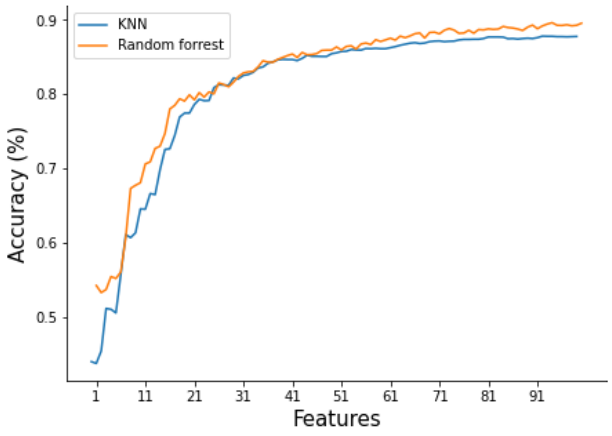

Fig. 10: Accuracy of the 100 most important features on Chlorine Concentration dataset.

As can be seen in Fig. 10, with the addition of features one by one, the accuracy is ascending, which is a good indication of the effect of the features of our proposed model on the final accuracy. With only 40 out of 200 features, we achieved higher accuracy than all the methods studied.

III. Comparison with state-of-the-art

The proposed method is compared with nine different methods that have the best performance. We will only briefly introduce each method here due to space constraints. Interested readers can refer to the main article for more details.

- DTW-Rn-1NN and RotF [13]. The two benchmark classifications are more competitive than many methods, so we added these two to our comparison.

Shapelets are a group of methods that focus on discovering short patterns that define a, and yet can appear anywhere in the series. These independent phase patterns are commonly identified as shapelets. A class is then distinguished by the presence or nonpresence of at least one shapelet someplace in the entire series. [17], [10], [18], and [8] are shapelet-based algorithms.

- Shapelet Transformation (ST) [17]. This algorithm combines a weighted ensemble of standard classifiers that new time series are classified with a weighted vote.
- Fast Shapelets (FS) [10]. FS combines SAX words and random masking techniques to raise efficiency.
- Scalable shapelet Discovery (SD) [18]. SD filters out candidates improving classification with an online clustering/ pruning technique. The dimensionality reduction ratio r is set to 1=2, and pruning distance percentile p is 25 for simplicity if there is no support from.
- ELIS [8]. ELIS is the present efficient shapelet-based algorithm. It utilizes PAA and TF-IDF to improve the efficiency of finding shapelet candidates. The logistic regression classifier is applied to adjust the shapelets. This paper follows ELIS [8] to adjust parameters. Otherwise, the parameters are set as follows: the iteration number is 1000, the regularization λ=0.01, and the learning rate η=0.1.
- COTE [33]. COTE is a combination of 35 TSC algorithms, including EE and ST and actually is the meta ensemble method. COTE works based on two principle ideas. First, the easiest approach to improvement for TSC issues is to convert into a new data space where discriminatory features identify simplify. Second, with a single data representation, improved accuracy could be achieved through simple ensemble schemes.
- ResNet [36]. The primary Property of ResNet is the residual shortcut connection between consecutive

convolutional layers. The network includes three residual blocks followed by a GAP layer and a final softmax classifier whose number of neurons is equal to the number of classes in a dataset. Each residual block comprises three convolutions whose output is added to the residual block's input and then fed to the next layer.

- BSPCOVER [32]. Creates many candidates by Symbolic Aggregate approximation with sliding window, then prunes identical and highly similar candidates by Bloom filters and similarity matching, respectively.

a) Accuracy

The experiment accuracy results for the method proposed in this paper and related methods are measure by cross-validation. The accuracy results for 12 datasets are presented in Table 4. On the dataset tested in this paper, our method is more accurate than other methods and even slightly ahead of COTE, which has been the most accurate classifier ever. Our proposed method has the highest classification accuracy among 12 datasets, so that with an average rank of 2.08, it is in the first final rank. The ranking of the state-of-the-art methods is shown in the last row of Table 4.

b) Efficiency

According to the results of [8], it can be seen that ELIS is about twice as fast as LTS. On the other hand, according to the results of [13], it can be understood that LTS is faster than COTE and ST. Therefore, we compare FS, SD, ELIS, and BSPCOVER with the proposed method for runtime evaluation.

We present the runtime of FS, SD, ELIS, BSPCOVER, and our method, in Table 5. (The unit of the numbers is seconds, and we use h and d to denote for hours and days, respectively.)

According to the results, both FS and SD methods have less execution time compared to the others, but on the other hand, a significant drop in accuracy than the others. The ELIS method has the longest runtime of all the methods and sometimes even takes a few days. The BSPCOVER method has both good runtime and high accuracy.

Table 5: Runtime of the proposed method and state-of-the-art methods on UCRARCHIVE

Dataset	FS	SD	ELIS	BSPCOVER	Proposed method
BeetleFly	13.96	1.62	122.87	42.92	578
ChlorineConcentration	128.56	1.82	31.8h	173.86	2.44h
Coffee	4.1	0.98	53.3	10.96	594
DiatomSizeReduction	4.33	1.02	749.03	30.04	815
DistalPhalanxOutlineCorrect	15.2	4.05	1.1h	52.39	1302
Earthquakes	1152.06	39.75	2.7h	2957.36	1244
ECG5000	36.89	6.95	7.5h	600.37	2.26h
Haptics	2086.17	6.45	3.6d	3.2h	1773
InlineSkate	2.1h	5.42	8.5d	4.2h	2627
Mallat	419.98	4.34	3.9h	2896.15	1.62h
Meat	34.62	1.94	2975.9	44.02	752
Symbols	16.17	1.04	2315.93	90.43	2387

Table 6: Wilcoxon signed rank test results on UCRARCHIVE

Proposed model VS	p-value	Test number	Superior	Inferior	Similar
ST	2.45E-02	12	9	2	1
FS	6.71E-03	12	10	2	0
COTE	8.05E-02	12	8	4	0
SD	5.37E-03	12	12	0	0
DTW Rn 1NN	1.24E-02	12	10	2	0
RotF	1.66E-02	12	9	2	1
ELIS	3.6E-03	12	12	0	0
ResNet	2.08E-02	12	9	3	0
BSPCOVER	5.79E-02	12	7	4	1

Although the method presented in this paper has a longer runtime compared to other related methods, but because the runtime is not very long in any of the datasets, the average runtime of this method is close to BSPCOVER. For example, the average runtime for our method is 2906 seconds and for BSPCOVER is 2794 seconds.

The complexity of this method is directly related to the values of n, m, the instances number and the features number of each dataset. If we call the FFT calculation time T_{FFT}, the instances number r and features number c, the order of complexity according to algorithm1-2 will be $O(T_{FFT} + rcnm)$.

We performed the Wilcoxon signed rank test between the nine state-of-the-art methods and our proposed method to find significant differences between each pair of algorithm behavior [37]. The p-value represents how significant differences the result is: If the value of p is small, the evidence is strong. As Table 6 states, the number of functions on which the proposed method performs better than ST, FS, COTE, SD, DTW Rn 1NN, RotF, ELIS, ResNet, and BSPCOVER are 9, 10, 8, 12, 10, 9, 12, 9 and 7, respectively.

We found out that p-value of all results are smaller than 0.05, except COTE, BSPCOVER (larger than 0.05). Therefore, the overall results show our method is better than related methods.

Conclusion

We present a simple method for time series classification in this paper. Our study of time series processing has led us to find a way to improve the accuracy of TSC problems. The proposed method has three steps. It applies FFT on raw data and then normalized them in preprocessing, histogram computation, extract features, the end using random forest classifier to predict the label of the test set and find accuracy. This method takes a minimum of time to build the model by reducing the dimensions it applies. Using this idea helped us to increase the accuracy of the classification to an acceptable level.

The implementation results of our proposed model show that it has achieved the highest accuracy in half of the datasets, and in total, it is in the first place among ten methods. These results were evaluated using the k-fold cross-validation on the UCR time series classification archive.

In future work, it is suggested that the method presented in this paper be applied to real data, which is usually time-consuming. Also, modifications should be made to the model to use in multivariate time series.

Author Contributions

Kiana Kiaei: Programmer, Software, Validation, Conceptualization, Visualization, Investigation, Writing - Reviewing and Editing, Writing - Original draft preparation. Hesam Omranpour: Supervision, Project administration, Conceptualization, Methodology, Visualization, Investigation, Writing - Reviewing and Editing, Funding acquisition, Programmer, Writing - Original draft preparation.

Acknowledgment

The authors gratefully thank the anonymous reviewers and the editor of JECEI for their useful comments and suggestions.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

TSC	Time Series Classification
FFT	Fast Fourier Transform
EEG	Electroencephalography
ED	Euclidean Distance
DTW	Dynamic Time Warping
SVM	Support Vector Machine
ST	Shapelet Transformation
SD	Scalable shapelet Discovery
FS	Fast Shapelets
SAX	Symbolic Aggregate approXimation
SFA	Symbolic Fourier Approximation
DFT	Discrete Fourier Transform
KNN	K Nearest Neighbor
BOP	Bag of Patterns
VSM	Vector Space Model
TF-IDF	Term Frequency — Inverse Document Frequency
BOSS	Bag of Symbolic Fourier Approximation Symbols
ELIS	Efficient Learning Interpretable Shapelets
COTE	Collective Of Transformation-based Ensembles
ECG	Electrocardiogram
MDI	Mean Decrease Impurity
MDA	Mean Decrease Accuracy
RotF	rotation forest

References

[1] H. Madsen, Time series analysis. CRC Press, 2007.

[2] M. Langkvist, L. Karlsson, A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," Pattern Recognit. Lett., 42: 11–24, 2014.

[3] Z. Turani et al., "Optical radiomic signatures derived from optical coherence tomography images improve identification of melanoma," Cancer Res., 79(8): 2021–2030, 2019.

[4] S. Adabi et al., "Universal in vivo textural model for human skin based on optical coherence tomograms," Sci. Rep., 7(1): 1–11, 2017.

[5] J.C.B. Gamboa, "Deep learning for time-series analysis," arXiv Prepr. arXiv1701.01887, 2017.

[6] M. Jung, S. Chi, "Human activity classification based on sound recognition and residual convolutional neural network," Autom. Constr., 114: 103177, 2020.

[7] A. Rajkomar et al., "Scalable and accurate deep learning with electronic health records," NPJ Digit. Med., 1(1): 1–10, 2018.

[8] Z. Fang, P. Wang, W. Wang, "Efficient learning interpretable shapelets for accurate time series classification," in Proc. 2018 IEEE 34th International Conference on Data Engineering (ICDE): 497–508, 2018.

[9] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, "Learning time-series shapelets," in Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 392–401, 2014.

[10] T. Rakthanmanon, E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in proc. 2013 SIAM International Conference on Data Mining: 668–676, 2013.

[11] B. Wang et al., "Removal of 'strip noise' in radio-echo sounding data using combined wavelet and 2-D DFT filtering," Ann. Glaciol., 61(81): 124-134, 2020.

[12] G.R. Schleder, A.C. M. Padilha, C.M. Acosta, M. Costa, A. Fazzio, "From DFT to machine learning: recent approaches to materials science--a review," J. Phys. Mater., 2(3): 32001, 2019.

[13] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," Data Min. Knowl. Discov., 31(3): 606–660, 2017.

[14] X. Xi, E. Keogh, C. Shelton, L. Wei, C.A. Ratanamahatana, "Fast time series classification using numerosity reduction," in Proc. 23rd international conference on Machine learning: 1033–1040, 2006.

[15] H. Sakoe, S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," IEEE Trans. Acoust., 26(1): 43–49, 1978.

[16] L. Ye, E. Keogh, "Time series shapelets: a new primitive for data mining," in Proc. 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 947–956, 2009.

[17] J. Lines, L.M. Davis, J. Hills, A. Bagnall, "A shapelet transform for time series classification," in Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 289–297, 2012.

[18] J. Grabocka, M. Wistuba, L. Schmidt-Thieme, "Fast classification of univariate and multivariate time series through shapelet discovery," Knowl. Inf. Syst., 49(2): 429–454, 2016.

[19] T. Le Nguyen, S. Gsponer, G. Ifrim, "Time series classification by sequence learning in all-subsequence space," in Proc. 2017 IEEE 33rd international Conference on Data Engineering (ICDE): 947–958, 2017.

[20] T. Rakthanmanon et al., "Searching and mining trillions of time series subsequences under dynamic time warping," in Proc. 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 262–270, 2012.

[21] J. Lin, E. Keogh, L. Wei, S. Lonardi, "Experiencing SAX: a novel symbolic representation of time series," Data Min. Knowl. Discov., 15(2): 107–144, 2007.

[22] P. Schäfer, U. Leser, "Fast and accurate time series classification with weasel," in Proc. 2017 ACM on Conference on Information and Knowledge Management: 637–646, 2017.

[23] R. Agrawal, C. Faloutsos, A. Swami, "Efficient similarity search in sequence databases," in Proc. International Conference on Foundations of Data Organization and Algorithms: 69–84, 1993.

[24] J. Lin, R. Khade, Y. Li, "Rotation-invariant similarity in time series using bag-of-patterns representation," J. Intell. Inf. Syst., 39(2): 287–315, 2012.

[25] P. Senin, S. Malinchik, "Sax-vsm: Interpretable time series classification using sax and vector space model," in Proc. 2013 IEEE 13th International Conference on Data Mining: 1175–1180, 2013.

[26] G. Ifrim, C. Wiuf, "Bounded coordinate-descent for biological sequence classification in high dimensional predictor space," in Proc. 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining: 708–716, 2011.

[27] P. Schäfer, "Scalable time series classification," Data Min. Knowl. Discov., 30(5): 1273–1298, 2016.

[28] "The Fast Fourier Transform (FFT) as a statistical tool for time-series analysis."

[29] S. Zhang, X. Li, M. Zong, X. Zhu, D. Cheng, "Learning k for knn classification," ACM Trans. Intell. Syst. Technol., 8(3): 1–19, 2017.

[30] A. A. Soofi, A. Awan, "Classification techniques in machine learning: applications and issues," J. Basic Appl. Sci., 13: 459–465, 2017.

[31] G. Biau, E. Scornet, "A random forest guided tour," Test, 25(2): 197–227, 2016.

[32] G. Li, B.K.K. Choi, J. Xu, S.S. Bhowmick, K.-P. Chun, G.L.H. Wong, "Efficient shapelet discovery for time series classification," IEEE Trans. Knowl. Data Eng., 2020.

[33] A. Bagnall, J. Lines, J. Hills, A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," IEEE Trans. Knowl. Data Eng., 27(9): 2522–2535, 2015.

[34] H. A. Dau et al., "The UCR time series archive," IEEE/CAA J. Autom. Sin., 6(6): 1293–1305, 2019.

[35] L. Breiman, A. Cutler, "Manual--setting up, using, and understanding random forests V4. 0. 2003,".

[36] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.A. Muller, "Deep learning for time series classification: a review," Data Min. Knowl. Discov., 33(4): 917–963, 2019.

[37] J. Derrac, S. García, D. Molina, F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," Swarm Evol. Comput., 1(1): 3–18, 2011.

Biographies



Kiana Kiaei received her Bachelor's degree in computer engineering from Babol Noshirvani of Technology (BNUT). She is currently a master's student in the field of computer engineering majoring in computer software at (BNUT).

- Email: k.kiaei@stu.nit.ac.ir
- ORCID: NA
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA



Hesam Omranpour received B.Sc. degree in Computer Engineering-Software Engineering from Iran, University of Science and Technology and M.S. degree and Ph.D. degree in Artificial Intelligence from Amirkabir University of Technology, Tehran, Iran, in 2006, 2009 and 2016 respectively. He is currently an assistant professor with the department of electrical and computer engineering, Babol Noshirvani University of Technology, Iran. His current research interests include optimization, machine learning, pattern recognition.

- Email: h.omranpour@nit.ac.ir
- ORCID: 0000-0003-4253-0811
- Web of Science Researcher ID: NA
- Scopus Author ID: 26423137100
- Homepage: https://ostad.nit.ac.ir/home.php?sp=391011

Copyrights
©2022 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.

How to cite this paper:
K. Kiaei, H. Omranpour, "Polar formed histogram of fast fourier transform feature for time series classification problems," J. Electr. Comput. Eng. Innovations, 10(2): 273-286, 2022.

DOI: 10.22061/JECEI.2021.8048.468

URL: https://jecei.sru.ac.ir/article_1628.html