International Journal of Applied Science and Engineering

Special issue: The 10th International Conference on Awareness Science and Technology (iCAST 2019)

Fast and interpretable transformation for time series classification: A comparative study

Hideotoshi Ito[1], Basabi Chakraborty[2*]

[1] *Graduate School of Software and Information Science, Iwate Prefectural University, Iwate, Japan*
[2] *Faculty of Software and Information Science, Iwate Prefectural University, Takizawa, Iwate, Japan*

ABSTRACT

This work is an extended version of the paper published by Ito and Chakraborty (2019). Time Series Classification (TSC) is gaining importance in the area of pattern recognition, as the availability of time series data has been increased recently. TSC is a complicated problem because of needs to consider the characteristics of temporal data; periodicity, time correlation, elasticity and unequal lengths of the time series. As all of those characteristics are usually not expressed simultaneously in raw data, design of a unified similarity metric for time series classification or clustering is difficult to achieve. In addition to traditional feature-based, model-based or distance-based algorithms for TSC, ensemble and deep neural network have been proposed recently, and deep neural network model like ResNet is known to be quite effective. However, deep neural network model requires enormous computing resources and computing time as well as large number of training samples. Feature based and distance based approaches till have potential to outperform them in computational time with reasonable classification accuracy. In this work, new temporal data transformation algorithms have been proposed and their combination with nearest neighbor classifier have been compared to existing time series classification methods. From the experimental results, the proposed algorithms with nearest neighbor classifier are found to be inferior to ResNet regarding classification accuracy though comparable to Dynamic Time Warping (DTW) but the computation is much faster than ResNet and DTW, and also the classification accuracy is better in case of small datasets which seems to be important for many real life applications with limited resources.

Keywords: Time series classification; feature extraction; deep neural network.

OPEN ACCESS

Received: May 19, 2020

Accepted: July 28, 2020

Corresponding Author:
Basabi Chakraborty
basabi@iwate-pu.ac.jp

Publisher:
Chaoyang University of Technology
ISSN: 1727-2394 (Print)
ISSN: 1727-7841 (Online)

1. INTRODUCTION

Time series is a sequence of data that describes the change of the observed phenomenon over time. Due to increased use of sensors, the improvement of computation power and decreased cost of storage, enormous temporal data are collected and stored from various application areas ranging from financial prediction to health care. Because of this high volume of data, the demand of analysis of big time series data is increasing. Among them, time series classification is an important task because many applications rely on it, for example, online signature verification (Tamilarasi and Nithya Kalyani, 2017), human gait recognition (Ebenezer et al., 2019) authentication problem, electroencephalogram (EEG) and electrocardiogram (ECG) analysis in medical field (Wang et al., 2012), stock price and exchange rate prediction in financial field (Fisher and Krauss, 2018) or human activity recognition (Lara and Labrador, 2013) in the area of healthcare. Time series classification is a challenging problem as traditional machine learning algorithms for static data are difficult to use directly for temporal data, because of elasticity, periodicity and unequal length of time series. Traditional time series classification algorithms can be roughly grouped into three types — model-based feature-based, and distance-based.

The model-based approaches make model of each class from raw time series data by fitting appropriate parameters and classify the given data according to the best fit of the model. The examples are Autoregressive (Kini and Sekhar, 2013), Markov and Hidden Markov Model (HMM) (Antonucci et al., 2015), Naive Bayes, or Neural Network models. Most of the models are based on statistical probability distribution. Autoregressive model is based on stochastic process in which the value at some point of time series depend on all previous values. On the other hand, Markov process is another stochastic process where the value at some point of time series depends on the previous one. HMM is an automaton in which the state transition occurs probabilistically. Naive Bayes is the simplest probability distribution based model, commonly used for text classification. Among neural networks, Recurrent Neural Network (RNN) (Smirnov and Nguifo, 2018) is suitable for time series classification because it considers variable length input and is dependent on previous values. Long Short-Term Memory (LSTM) (Karim et al., 2018), extended form of RNN, tends to have better performance due to consideration of long term dependency. Currently, it is found that convolutional neural network based models are also effective for time series classification.

The feature-based approaches extract features and transform the raw time series into a feature vector before classifying with traditional classification methods. The examples of feature extraction methods are Fourier Transform, Wavelet Transform, Shapelet (Ye et al., 2010), Time Series Bag of Features (TSBF) (Baydogan et al., 2013) etc.

The distance-based approaches compare raw time series by a distance metric and assign it to the class of the nearest class sample. Euclidean distance, Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) and its derivations are the popular metrics used for this purpose. Euclidean distance, though the simplest one, is not suitable for unequal length time series, because it compares the values of time series at the same time, and elasticity and periodicity of temporal data are not considered. On the other hand, DTW with k-nearest neighbor classifier shows better performance as it is capable of taking care of time distortion though it is computationally costly.

Recently ensemble based approaches have been developed in which different classifiers are combined to achieve higher performance. Some of the examples are Elastic Ensembles (Lines and Bagnall., 2015), COTE (Bagnall et al., 2015) and HIVE-COTE (Lines et al., 2018), an extended version of COTE. Though ensemble of classifiers can produce good classification accuracy, the computation time is very high even using recent high performance machines. A comparative study of several approaches can be found in Bagnall et al. (2017). Another recent development of time series classification algorithms is based on deep neural networks (DNN). A review of the DNN based time series classification approaches can be found in Fawaz et al. (2019). Among several DNN models, Convolutional neural networks (CNN) and Residual Network (RESNET) (He et al., 2016) are known to be the most successful in time series classifications. As very recent state-of-the-art time series classification algorithms based on ensemble algorithms or DNN requires too much computational resources, their versatile use is difficult in real life resource-constrained applications. Also deep neural network seems to have poor interpretability due to the black box architecture having a large number of layers of computing elements with no definite method for setting parameters. Though researchers are trying to interpret DNN models, it is still in its early phase.

In many real world applications of time series classification, fast and interpretable algorithms capable of producing reasonable classification accuracy with limited resources are needed and research on developing these algorithms is important. As there are various types of time series, for example, time series having seasonal trend like weather parameters, almost changeless and spike, or random, dealing all types of time series uniformly with one approach is unfortunately not the solution, the shape seems to be the only common feature of the most of the time series. Model based and Feature based traditional TSC algorithms possess better interpretability than deep network based models. Feature based algorithms are also faster than raw time series based approaches. Elastic distance based measure, Dynamic Time Warping (DTW) is the most popular similarity measure for distance based algorithms. In fact, the combination of DTW and k-nearest neighbor classifier (Bishop, 1995) is known to be an effective approach and was considered as the best algorithm for time series classification problems until a few years ago before the development of ensemble and deep network based algorithms.

Though the performance of DTW- kNN is the best regarding classification accuracy among traditional algorithms for most of the applications, the computational complexity is high, of the order of $O(n^2)$ where n the length of the time series. There are many proposed improvements for the computational bottleneck, however in all cases, ultimately it is required to calculate precise similarity. Besides, DTW executes matching two sequences with shortest path, while finding corresponding segment of one series to another series as shown in Fig 1. At that time, pairs that could not be matched in the same segment are summed as dissimilarity. In other words, similar segments in two sequences are equated, and the differences are counted as the distance. It can be regarded as geometrical difference.

In this work, an approach to reduce the computational cost of similarity computation of two unequal time series keeping the classification accuracy as high as possible, three new shape based characteristics transformation measures for similarity computation are proposed and their performance in TSC compared to state-of-the art popular algorithms is analyzed. In the next section a brief description of the popular similarity measures, feature extraction techniques and classification algorithms related to this work are presented followed by the section containing our proposed approach. Section 4 contains the simulation experiments and results while section 5, the final section, presents discussion and conclusion.

2. RELATED BACKGROUND ON TIME SERIES CLASSIFICATION

Among several traditional feature based, model based and similarity based time series classification algorithms, DTW-kNN, the combination of dynamic time warping (DTW) as a similarity measure and k nearest neighbor classification algorithm, is considered to be the best time series classification algorithm at least until very recently. As mentioned before in introduction, DTW incurs high computational cost and many proposals are already evolved to reduce the cost. Fast DTW (Salvador and Chan, 2007), Multiscale DTW (Keogh and Pazzani, 2000) or Sparse DTW (Al-Naymat et al., 2009) are some of the popular approaches already developed. The authors also previously proposed a few algorithms, DTW-GA (Chakraborty and Yoshida, 2017), DTE (dynamic translational error) (Chakraborty and Yoshida, 2016) and Edge-detectional DTW (Ito and Chakraborty, 2018) for improvement of computational cost of DTW without much sacrificing classification accuracy.

In this section a brief description of a few representative popular similarity metrics, feature based representation of time series and classification techniques used in this work are presented.

2.1 Similarity Metrics

In similarity based time series classification algorithms, the simplest measure is Euclidean distance, but it cannot be used for unequal time series. The most popular elastic similarity measure is Dynamic Time Warping.

2.1.1 Euclidean Distance

Euclidean distance is the simplest similarity measure that defined as below for two same length time series $X = \{x_i|0 \le i \le n\}$ and $Y = \{x_i|0 \le i \le n\}$. $O(n)$ is the order of computational complexity for the time series length n.

$$distance(X,Y) = \sqrt{\sum_{i}^{n}(x_i - y_i)^2} \tag{1}$$

2.1.2 Dynamic Time Warping

Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) is the most popular time series similarity measure. It computes the shortest matching path of two sequences, as shown in Fig. 1, while finding corresponding segment of one side to another side with a $(m+1) \times (n+1)$ cost matrix for $X = \{x_i|0 \le i \le m\}$ and $Y = \{y_i|0 \le i \le n\}$. The cost matrix D is computed with following procedure:

$$\begin{aligned} initially,\ D_{0\ldots i,0\ldots j} &= \infty \\ D_{0,0} &= 0 \\ then,\ D_{i,j} &= |x_i - y_j| + min\begin{cases} D_{i-1,j}, \\ D_{i-1,j-1}, \\ D_{i,j-1} \end{cases} \end{aligned}$$

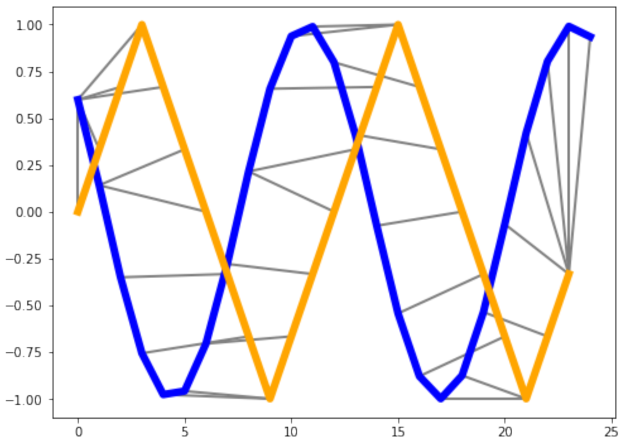

Fig. 1. Illustration of matching two time series with Dynamic Time Warping

2.2 Time Series Feature Extraction

In this section, some feature extraction methods for time series are introduced.

2.2.1 Time Series Bag of Features

Time Series Bag of Features (TSBF) (Baydogan et al., 2013) is a feature extraction method for time series like bag-of-words in case of natural language processing. It considers a local segment of time series as a codeword, and counts that in a time series, then represents the time series as a histogram of codewords.

2.2.2 Shapelet

Shapelet (Ye et al., 2010) is a feature extraction method which finds the subsequence within a time series that takes maximum information gain to represent the class, usually used with the decision tree whether the time series includes the shapelet or not. Let $T = t_1, \ldots, t_m$ be a time series, $S_T^l = \{\{t_p, \ldots, t_{p+l-1}\}|1 \le p \le m-l+1\}$ be a set of all subsequences of length l extracted from $T, d^*(T,S) = \min(d(S, S^* \in S_T^{[S]}))$ be a distance from the time series T

$$Gain(S, d_{OSP(D,S)}) \geq Gain(S, d^*_{th}), \quad (2)$$

$$Gain(\lambda(D), d_{OSP(D,\lambda(D))}) \geq Gain(S, d_{OSP(D,S)}), \quad (3)$$

to the subsequence S. For a shapelet candidate S of a dataset D which consists of two classes, a distance threshold, Optimal Split Point (OSP), is found to split a dataset D into D_1 and D_2, such that; Equation (2) for any other threshold d^*_{th}, a shapelet $\lambda(D)$ is defined with its corresponding OSP, Equation (3) for any other subsequence S, where $Gain(sp)$ is an information gain. Searching a shapelet is performed by the brute-force algorithm, with the theoretical worst-case complexity is $O(m^4)$ on the length of time series.

2.3 Time Series Classification

In this section, some time series classification methods used in this work have been introduced.

2.3.1 Residual Network

Residual Network (ResNet) (He et al., 2016) is a kind of deep convolutional neural network that enabled to learn many layers (more than 10 or 100 layers). Layers of plain networks approximate the function $H(x)$, but that of ResNet approximate $F = H(x) - x$ where x denotes the inputs to the first of these layers. In order to identify residual function F as original function $F + x$, connecting the inputs to the outputs by shortcuts. In the simulation experiments, the ResNet architecture of Hassan, et al (Fawaz et al., 2019), shown in Fig. 2 is used. The paper "strongly suggests to use ResNet instead of any other deep learning algorithm - it is the most accurate one with similar runtime to FCN (the second most accurate DNN)." (Fawaz et al., 2019) FCN is Fully Convolutional Neural Networks.

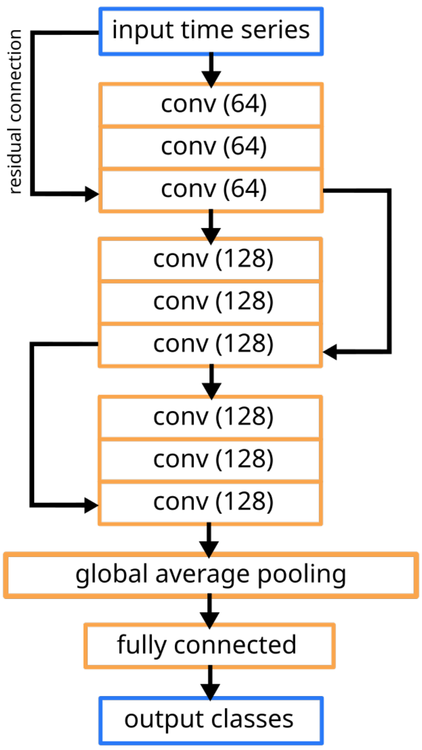

Fig. 2. Illustration of ResNet architecture

2.3.2 k-Nearest Neighbor

k-Nearest Neighbor (k-NN) (Bishop, 1995) is one of the most basic classification method that decides the class of the data by majority classes of k of the closest training samples. Distance metrics are used with this classifier to measure the distance between the given data and training samples. The number of k should be odd number of greater than or equal to one, specifically in case of $k = 1$, it is called Nearest Neighbor or 1-NN.

3. PROPOSED TRANSFORMATIONS FOR TSC

Time series has variety of characteristics such as elasticity and periodicity, and most of those are expressed as geometric features on the plot. In this work, two new shape based transformation technique of time series for similarity based classification have been proposed. These are named as Fold Count (FC) and Time Axis Area (TAA), basically an improvement of Fold Count that considers elasticity, periodicity an unequal length of time series for similarity calculation and also resembles shapelet, a feature extraction method. All of them take four parameters, original time series (T), its lower and upper limit of values ($\mathcal{L}$ and $\mathcal{U}$) and n, an user defined parameter that represents the number of divisions or folding between $\mathcal{L}$ and $\mathcal{U}$. The algorithms are described below successively.

3.1 Fold Count

Fold Count (Ito and Chakraborty, 2019) is a time series transformation method that counts the overlaps of time series values along the time, while the time series is approximated as a line plot. The concept is illustrated in Fig. 3. The folding values P are computed by user chosen parameter $n \in N$ according to the following equation.

$$P(n, \mathcal{L}, \mathcal{U}) = \bigcup_{0 \leq i < n} \left\{ \frac{i(\mathcal{U} - \mathcal{L})}{n} + \mathcal{L} \right\}. \quad (4)$$

For the time series $T = t_1, \ldots, t_m$, foldings between t_i and t_{i+1} are counted for all time intervals to the end of the time series; for all $p \in P(n, \mathcal{L}, \mathcal{U})$. Fold Count dissimilarity is defined as sum of differences of $FC(p)$ for all p for two time series. The pseudo code is shown in Algorithm 1 and Algorithm 2.

$$FC(p) = \sum_{i}^{m-1} \text{ if } (t_i \leq p \leq t_{i+1}) \text{ then } 1 \text{ else } 0, \quad (5)$$

Algorithm 1 Fold Count Transformation

Input: Time-series $\mathbf{T}$[0,...,m], the lower $\mathcal{L}$ and the upper $\mathcal{U}$ bounds, number of dividing folding positions $\mathbf{n}$.
Output: FoldCount(T, $\mathcal{L}$, $\mathcal{U}$, n)
Let a vector FC be the accumulated folding counts such that $FC[0,\ldots,n]$, P be the folding positions such that $P[0,\ldots,n]$.

for $i = 0$ to n **do**
  FC[i] := 0
end for
for $i = 0$ to $r - 1$ **do**
  P[i] := $i * (\mathcal{U} - \mathcal{L})/n + \mathcal{L}$
end for
for $i = 0$ to $m - 1$ **do**
  T_{lower}, T_{upper} = sorted(T[i], T[i+1])
  for $j = 0$ to n **do**
    if $T_{lower} \leq$ P[j] & P[j] $\leq T_{upper}$ **then**
      FC[j] += 1
    end if
  end for
end for
return FC

Algorithm 2 Fold Count Dissimilarity

Input: Time-series $\mathbf{X}$[0,...,p], $\mathbf{Y}$[0,...,q], the lower $\mathcal{L}$ and the upper $\mathcal{U}$ bounds, number of dividing folding positions $\mathbf{n}$.
Output: FoldCountDissimilarity(X, Y, $\mathcal{L}$, $\mathcal{U}$, n)

dissimilarity = 0
FC_X = FoldCount(X, $\mathcal{L}$, $\mathcal{U}$, n)
FC_Y = FoldCount(Y, $\mathcal{L}$, $\mathcal{U}$, n)
for $i = 0$ to n **do**
  dissimilarity += abs(FC_X[i]-FC_Y[i])
end for
return dissimilarity

Algorithm 3 Time Axis Area

Input: Time-series $\mathbf{T}$[0,...,m], the lower $\mathcal{L}$ and the upper $\mathcal{U}$ bounds, number of dividing folding positions $\mathbf{n}$.
Output: TimeAxisArea(T, $\mathcal{L}$, $\mathcal{U}$, n)
Let a vector TAL be the accumulated length such that $TAL[0,\ldots,n]$, a vector TS be the timestamps such that $TS[0, \ldots, n]$, P be the folding positions such that $P[0,\ldots,n]$.

for i = 0 to n **do**
  TAL[i] := 0
  TS[i] := -1
end for
for $i = 0$ to $r - 1$ **do**
  P[i] := $i * (\mathcal{U} - \mathcal{L})/n + \mathcal{L}$
end for
for $i = 0$ to $m - 1$ **do**
  T_{lower}, T_{upper} = sorted(T[i], T[i+1])
  for $j = 0$ to n **do**
    if $T_{lower} \leq$ P[j] & P[j] $\leq T_{upper}$ **then**
      if TS[j] ≥ 0 **then**
        TAL[j] += i - TS[j]
      end if
      TS[j] = i
    end if
  end for
end for
return TS

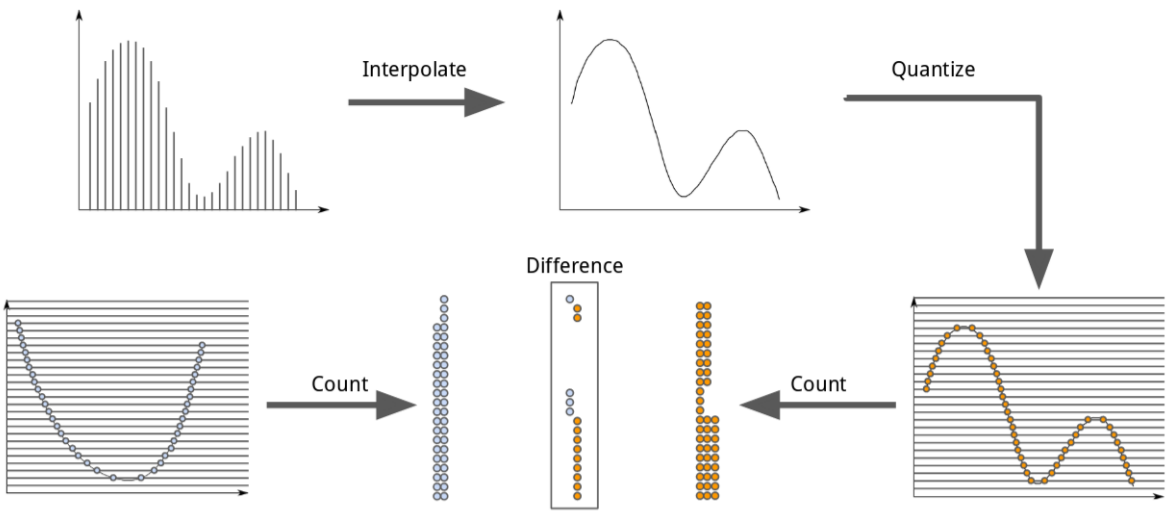

Fig. 3. Illustration of Fold Count Dissimilarity Algorithm (H. and B., 2019)

3.2 Time Axis Area

Time Axis Area (TAA) (Ito and Chakraborty, 2019) is a modification of Fold Count algorithm which adds the concept of elasticity in FC. Time Axis Area is the area bounded by the time series if the number of folding positions n is taken as infinity. The pseudo code is shown in Algorithm 3.

4. SIMULATION EXPERIMENTS AND RESULTS

In this section simulation experiments and results with benchmark data sets for evaluation of the efficiency of the proposed technique have been presented.

4.1 Simulation Experiments and Data sets

Simulation experiments have been done to compare the performance of proposed transformations compared to raw time series for time series classification problems. The performance of nearest neighbor classifier (1NN) with proposed FC and TAA, 1NN-FC and 1NN-TAA, is compared with ResNet and 1NN-DTW by simulation. 88 benchmark datasets of UCR Time Series Classification Archive 2018 (Dau et al., 2019) shown in Table 4 are used for simulation experiments. Each dataset is already split into training samples and test samples, thus each classifier is trained with all training samples, then classified with all test samples.

The parameters of FC and TAA are as follows; the lower L and the upper U bounds are minimum value and maximum value of the training samples; the number of folding positions n is 128. For ResNet, the architecture of the model resembles the model as in He et al. (2016), the batch size is 64, 300 epochs learning have done, and the simulation experiment is repeated for ten times, the results in the tables are medians of all simulation results.

The simulation experiments have been performed on all thread of Intel Core i7-6700 CPU (3.40GHz) with Ubuntu 18.04, and NVIDA GeForce GTX970 have used for training ResNet. The experimental program has been implemented by Python 3 and Keras for ResNet, C ++ for 1NN-DTW, 1NN-FC and 1NN-TAA.

4.2 Simulation Results and Discussion

All simulation results are shown in Table 5, classification accuracies and computational times for classification of proposed methods as well as DTW-1NN and ResNet with raw time series for all the 88 datasets have been presented. The computational time shown in the table is in milliseconds. From the experimental results, it is seen that classification accuracies with ResNet are almost always better than other methods, but simpler algorithms like 1NN-based classifiers are better for some datasets that have few training samples, such as DiatomSizeReduction and Fungi. It is to be noted that data augmentation or fine tuning are required for deep neural model if there is only few training samples.

But the computational times are widely different between the classification results of ResNet and 1NN-based classifiers. Especially, FC and TAA based approaches are much faster to use in real-time situation compared to ResNet or DTW based classifier. Each method's results have also been compared using the Wilcoxon signed-rank two-sided test (Wilcoxon, 1945) in Table 1 and Table 2. Wilcoxon signed-rank test is a non-parametric statistical hypothesis test to compare two related samples for check whether there is any difference between the two samples. For instance, the null hypothesis "there is no difference between 1NN-FC and 1NN-TAA for classification accuracy" cannot be reject in significant level 1% < 17.8%. Other pairs are less than 1%, that is, there are significant differences for classification accuracy and predict time.

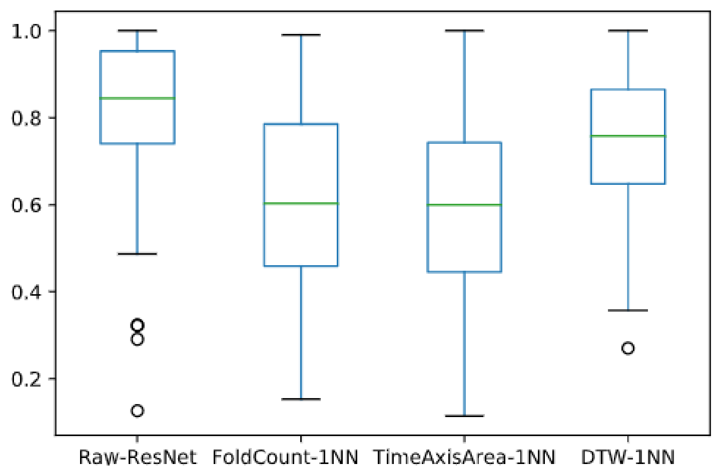

Fig. 4. Box-plot of classification accuracies

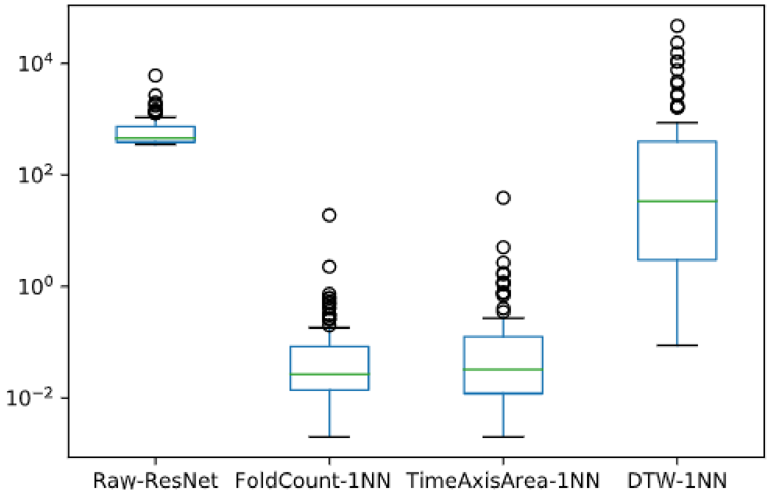

Fig. 5. Box-plot of prediction time

Table 1. Wilcoxon signed-rank test's p values of classification accuracies

	1NN-FC	1NN-TAA	1NN-DTW
ResNet	0.000	0.000	0.000
1NN-FC		0.178	0.000
1NN-TAA			0.000

Table 2. Wilcoxon signed-rank test's p values of computational times

	1NN-FC	1NN-TAA	1NN-DTW
ResNet	0.000	0.000	0.000
1NN-FC		0.000	0.000
1NN-TAA			0.000

From the comparative study it is found that on the average of 88 data sets, ResNet achieves quite higher classification accuracy compared to our proposed technique though it is seen to be comparable to 1NN-DTW. Table 3 represents 20 data sets for which the classification accuracy difference computed by ResNet and 1NN-FC (proposed technique) is not very large. The negative values indicate 1NN-FC is better. It is noted from the description of those data sets that in most of them, the shape of the time series plays critical role in classifying the time series. So for shorter time series, time series having characteristic shape and in case of smaller number of training samples, the proposed technique can have better effect in classification task.

In the proposed approach, raw time series is transformed into FC and TAA, but effects of preprocessing are not verified. Therefore, further modification or preprocessing of the algorithm can possibly improve performance. Regarding interpretability of an algorithm, the proposed technique is interpretable and can be applied efficiently to a particular class of time series data sets in contrast to ResNet or any deep network based algorithm in which the correlation between the algorithm and the data set cannot be assessed.

Table 3. Accuracy differences (ResNet -1NN-FC)

	Accuracy Difference
Fungi	-0.669
DiatomSizeReduction	-0.567
Wine	-0.241
BirdChicken	0.000
InlineSkate	0.009
Plane	0.010
Lightning2	0.016
Wafer	0.030
ECG5000	0.034
Mallat	0.037
HandOutlines	0.039
Strawberry	0.045
WormsTwoClass	0.058
Symbols	0.060
Trace	0.060
RefrigerationDevices	0.065
Earthquakes	0.072
Meat	0.083
StarLightCurves	0.088
BeetleFly	0.100

International Journal of Applied Science and Engineering

Ito et al., International Journal of Applied Science and Engineering, 17(3), 269–280

Table 4. Overview of datasets

Datasets	Train	Test	Class	Length	Train Range	Test Range
Adiac	390	391	37	177	-1.988 – 2.625	-2.053 – 2.464
ArrowHead	36	175	3	252	-2.257 – 2.554	-2.549 – 2.487
Beef	30	30	5	471	-3.290 – 3.721	-3.386 – 3.151
BeetleFly	20	20	2	513	-2.517 – 2.506	-2.515 – 2.408
BirdChicken	20	20	2	513	-2.825 – 2.124	-3.096 – 2.442
CBF	30	900	3	129	-2.317 – 3.245	-3.547 – 3.793
Car	60	60	4	578	-2.210 – 1.991	-2.246 – 2.142
ChlorineConcentration	467	3840	3	167	-11.839 – 7.442	-12.419 – 12.633
CinCECGTorso	40	1380	4	1640	-8.594 – 10.536	-11.213 – 11.733
Coffee	28	28	2	287	-2.064 – 2.177	-2.115 – 2.104
Computers	250	250	2	721	-3.748 – 21.596	-1.611 – 26.387
CricketX	390	390	12	301	-4.766 – 11.494	-5.373 – 12.653
CricketY	390	390	12	301	-9.775 – 6.839	-10.199 – 7.414
CricketZ	390	390	12	301	-4.758 – 11.924	-5.125 – 12.707
DiatomSizeReduction	16	306	4	346	-1.773 – 1.985	-1.979 – 2.447
DistalPhalanxOutlineAgeGroup	400	139	3	81	-1.994 – 2.058	-1.914 – 2.025
DistalPhalanxOutlineCorrect	600	276	2	81	-2.159 – 2.446	-2.180 – 2.460
DistalPhalanxTW	400	139	6	81	-1.994 – 2.058	-1.897 – 2.000
ECG200	100	100	2	97	-2.617 – 4.199	-3.014 – 4.148
ECG5000	500	4500	5	141	-5.798 – 4.058	-7.090 – 7.402
ECGFiveDays	23	861	2	137	-6.511 – 5.421	-7.108 – 6.033
Earthquakes	322	139	2	513	-0.886 – 7.863	-0.730 – 7.728
ElectricDevices	8926	7711	7	97	-9.696 – 9.696	-6.811 – 9.696
FaceAll	560	1690	14	132	-4.485 – 4.876	-4.823 – 9.189
FaceFour	24	88	4	351	-4.688 – 5.908	-4.252 – 5.345
FacesUCR	200	2050	14	132	-3.959 – 8.739	-4.823 – 9.189
FiftyWords	450	455	50	271	-2.354 – 5.018	-2.522 – 5.281
Fish	175	175	7	464	-1.951 – 2.126	-1.790 – 15.053
FordA	3601	1320	2	501	-4.618 – 5.059	-4.557 – 4.315
FordB	3636	810	2	501	-5.539 – 5.090	-4.088 – 4.930
FreezerRegularTrain	150	2850	2	302	-2.229 – 5.022	-2.230 – 17.148
FreezerSmallTrain	28	2850	2	302	-2.227 – 1.425	-2.230 – 17.148
Fungi	18	186	18	202	-1.494 – 80.786	-2.050 – 85.056
GunPoint	50	150	2	151	-2.369 – 2.053	-2.500 – 2.320
Ham	109	105	2	432	-2.054 – 8.033	-1.778 – 9.431
HandOutlines	1000	370	2	2710	-3.218 – 2.090	-2.891 – 1.778
Haptics	155	308	5	1093	-11.147 – 3.123	-14.860 – 3.903
Herring	64	64	2	513	-2.186 – 2.134	-2.209 – 2.074
InlineSkate	100	550	7	1883	-2.263 – 4.339	-2.519 – 3.827
InsectWingbeatSound	220	1980	11	257	-1.082 – 6.421	-1.305 – 6.590
ItalyPowerDemand	67	1029	2	25	-1.991 – 2.425	-2.393 – 3.294
LargeKitchenAppliances	375	375	3	721	-1.575 – 26.796	-1.107 – 25.703
Lightning2	60	61	2	638	-1.396 – 23.131	-1.447 – 22.683
Lightning7	70	73	7	320	-1.781 – 17.413	-1.728 – 16.641
Mallat	55	2345	8	1025	-1.607 – 2.762	-1.704 – 2.936
Meat	60	60	3	449	-1.542 – 3.390	-1.493 – 3.399

Ito et al., International Journal of Applied Science and Engineering, 17(3), 269–280

Table 5. Overview of datasets (Continued)

Datasets	Train	Test	Class	Length	Train Range	Test Range
MedicalImages	381	760	10	100	-2.392 – 7.222	-2.831 – 8.034
MiddlePhalanxOutlineAgeGroup	400	154	3	81	-1.641 – 1.924	-1.719 – 1.722
MiddlePhalanxOutlineCorrect	600	291	2	81	-1.660 – 2.067	-1.719 – 1.876
MiddlePhalanxTW	399	154	6	81	-1.719 – 1.924	-1.585 – 1.712
MoteStrain	20	1252	2	85	-8.409 – 2.468	-8.638 – 8.544
NonInvasiveFetalECGThorax1	1800	1965	42	751	-5.732 – 4.794	-5.750 – 5.195
NonInvasiveFetalECGThorax2	1800	1965	42	751	-5.416 – 4.677	-5.360 – 5.634
OSULeaf	200	242	6	428	-3.157 – 3.669	-3.427 – 3.400
OliveOil	30	30	4	571	-1.001 – 3.719	-1.000 – 3.732
PhalangesOutlinesCorrect	1800	858	2	81	-2.159 – 2.446	-2.180 – 2.460
Phoneme	214	1896	39	1025	-13.324 – 8.091	-11.190 – 12.524
Plane	105	105	7	145	-2.113 – 2.911	-2.115 – 2.924
ProximalPhalanxOutlineAgeGroup	400	205	3	81	-1.483 – 1.903	-1.442 – 1.824
ProximalPhalanxOutlineCorrect	600	291	2	81	-1.483 – 1.903	-1.442 – 1.824
ProximalPhalanxTW	400	205	6	81	-1.483 – 1.903	-1.469 – 1.850
RefrigerationDevices	375	375	3	721	-5.466 – 5.590	-4.952 – 7.041
ScreenType	375	375	3	721	-2.922 – 26.796	-7.834 – 26.796
ShapeletSim	20	180	2	501	-1.811 – 1.892	-1.868 – 1.874
ShapesAll	600	600	60	513	-3.220 – 2.819	-3.290 – 2.846
SmallKitchenAppliances	375	375	3	721	-5.067 – 26.795	-3.875 – 26.795
SonyAIBORobotSurface1	20	601	2	71	-2.727 – 3.626	-3.626 – 4.002
SonyAIBORobotSurface2	27	953	2	66	-4.138 – 4.231	-4.021 – 4.502
StarLightCurves	1000	8236	3	1025	-2.678 – 5.288	-2.625 – 5.459
Strawberry	613	370	2	236	-2.328 – 3.682	-2.128 – 3.723
SwedishLeaf	500	625	15	129	-3.412 – 3.222	-2.940 – 3.289
Symbols	25	995	6	399	-2.311 – 2.205	-2.595 – 2.869
SyntheticControl	300	300	6	61	-2.454 – 2.412	-2.619 – 2.605
ToeSegmentation1	40	228	2	278	-3.583 – 3.932	-6.682 – 4.822
ToeSegmentation2	36	130	2	344	-2.636 – 3.926	-3.679 – 5.550
Trace	100	100	4	276	-2.221 – 3.967	-2.392 – 3.937
TwoLeadECG	23	1139	2	83	-3.149 – 1.870	-3.797 – 1.929
TwoPatterns	1000	4000	4	129	-1.939 – 1.939	-1.933 – 1.918
UWaveGestureLibraryAll	896	3582	8	946	-4.434 – 7.628	-5.683 – 6.509
UWaveGestureLibraryX	896	3582	8	316	-4.438 – 4.434	-5.709 – 6.515
UWaveGestureLibraryY	896	3582	8	316	-4.103 – 7.654	-3.867 – 5.228
UWaveGestureLibraryZ	896	3582	8	316	-3.548 – 4.779	-4.296 – 4.864
Wafer	1000	6164	2	153	-3.054 – 11.787	-2.984 – 12.127
Wine	57	54	2	235	-1.943 – 3.201	-1.922 – 3.192
WordSynonyms	267	638	25	271	-2.261 – 5.003	-2.522 – 5.281
Worms	181	77	5	901	-4.311 – 4.859	-4.887 – 4.196
WormsTwoClass	181	77	2	901	-4.311 – 4.859	-4.887 – 4.196
Yoga	300	3000	2	427	-2.419 – 2.405	-2.854 – 2.438

Table 6. Experimental results

	Raw-ResNet			FoldCount-1NN		TimeAxisArea-1NN		DTW-1NN	
Dataset	Learning Time	Predict Time	Accuracy	Predict Time	Accuracy	Predict Time	Accuracy	Predict Time	Accuracy
Adiac	125029.849	444.423	**0.827**	**0.021**	0.251	0.032	0.501	25.044	0.532
ArrowHead	148301.445	433.662	**0.849**	0.007	0.514	**0.006**	0.611	1.269	0.697
Beef	74596.158	359.827	**0.633**	**0.003**	0.500	**0.003**	0.467	2.849	0.567
BeetleFly	78856.258	356.226	**0.850**	**0.002**	0.750	**0.002**	0.700	1.510	0.700
BirdChicken	79547.187	357.597	**0.850**	**0.002**	**0.850**	**0.002**	0.450	0.573	0.750
CBF	381734.281	488.821	0.994	**0.015**	0.661	0.016	0.672	2.559	**0.997**
Car	95595.689	373.218	**0.908**	**0.008**	0.433	0.011	0.417	15.390	0.767
ChlorineConcentration	417639.460	972.358	**0.843**	**0.201**	0.550	0.269	0.538	379.241	0.626
CinCECGTorso	1185173.577	1839.012	**0.791**	0.256	0.562	**0.250**	0.560	1746.230	0.692
Coffee	120962.963	355.021	**1.000**	0.014	0.786	**0.008**	0.964	0.210	**1.000**
Computers	220331.440	484.950	**0.816**	**0.043**	0.680	**0.043**	0.632	406.484	0.660
CricketX	175164.436	454.710	**0.785**	**0.031**	0.438	0.039	0.321	79.083	0.772
CricketY	178266.521	454.294	**0.796**	**0.032**	0.408	0.048	0.328	78.443	0.728
CricketZ	182047.549	455.306	**0.805**	**0.034**	0.390	0.044	0.331	78.305	0.795
DiatomSizeReduction	447592.985	441.892	0.322	0.021	0.889	**0.015**	0.879	2.685	**0.958**
DistalPhalanxOutlineAgeGroup	92462.856	366.252	0.723	0.013	0.590	**0.012**	0.640	2.050	**0.748**
DistalPhalanxOutlineCorrect	107191.593	393.706	**0.795**	**0.018**	0.630	0.021	0.678	7.493	0.699
DistalPhalanxTW	104673.435	376.804	**0.676**	**0.007**	0.475	0.021	0.597	2.734	0.633
ECG200	74915.238	358.606	**0.875**	0.010	0.770	**0.004**	0.740	0.095	0.810
ECG5000	395824.734	1015.393	**0.935**	**0.170**	0.902	0.350	0.893	279.297	0.929
ECGFiveDays	517680.967	486.477	**0.984**	**0.015**	0.822	0.026	0.720	1.659	0.758
Earthquakes	174916.121	400.325	**0.727**	**0.029**	0.655	0.033	0.669	141.670	0.662
ElectricDevices	1326089.383	1370.004	**0.729**	**18.884**	0.617	38.606	0.543	4318.372	0.650
FaceAll	217793.082	612.111	**0.836**	**0.075**	0.451	0.125	0.411	127.226	0.769
FaceFour	133267.802	373.250	**0.955**	0.009	0.534	**0.005**	0.341	2.687	0.841
FacesUCR	207754.868	672.602	**0.948**	**0.056**	0.566	0.062	0.512	54.994	0.934
FiftyWords	183946.519	470.401	0.701	**0.034**	0.343	0.043	0.215	76.551	**0.714**
Fish	145846.387	416.694	**0.971**	**0.022**	0.566	0.024	0.497	87.021	0.811
FordA	1471013.018	883.699	**0.936**	**0.543**	0.638	1.751	0.505	7591.564	0.567
FordB	1416649.492	670.404	**0.817**	**0.403**	0.596	0.678	0.531	4768.061	0.605
FreezerRegularTrain	501813.981	1078.156	**0.999**	**0.117**	0.896	0.125	0.948	191.802	0.920
FreezerSmallTrain	1625337.370	1075.469	**0.932**	**0.098**	0.786	0.110	0.756	35.802	0.719
Fungi	338452.501	388.140	0.126	**0.007**	0.796	**0.007**	0.565	0.129	**0.876**
GunPoint	115023.791	420.115	**0.990**	**0.004**	0.680	**0.004**	0.827	0.662	0.893
Ham	101389.183	383.766	**0.743**	**0.014**	0.543	**0.014**	0.533	29.831	0.552
HandOutlines	1807790.820	983.649	**0.874**	0.419	0.835	**0.405**	0.819	23443.569	0.865
Haptics	278112.929	584.279	**0.511**	0.059	0.266	**0.056**	0.325	765.729	0.357
Herring	98230.247	370.198	**0.656**	0.015	0.453	**0.010**	0.547	14.159	0.547
InlineSkate	534305.185	1023.797	0.291	0.138	0.282	**0.132**	0.367	2622.469	**0.376**
InsectWingbeatSound	326130.248	836.139	**0.487**	**0.075**	0.153	0.090	0.184	149.834	0.435
ItalyPowerDemand	265975.108	463.250	**0.961**	0.013	0.836	**0.011**	0.805	0.127	0.925
LargeKitchenAppliances	329012.960	571.575	**0.891**	**0.062**	0.752	0.066	0.757	387.708	0.835
Lightning2	91499.968	378.611	0.770	**0.010**	0.754	0.011	0.754	19.054	**0.803**
Lightning7	81425.826	367.812	**0.822**	0.025	0.562	**0.006**	0.425	6.335	0.767
Mallat	1194586.581	1982.350	**0.968**	**0.268**	0.931	0.269	0.728	734.422	0.915
Meat	84440.694	367.656	**0.883**	**0.006**	0.800	0.012	0.850	8.894	0.867

Table 7. Experimental results (Continued)

	Raw-ResNet			FoldCount-1NN		TimeAxisArea-1NN		DTW-1NN	
Dataset	Learning Time	Predict Time	Accuracy	Predict Time	Accuracy	Predict Time	Accuracy	Predict Time	Accuracy
MedicalImages	132081.156	458.402	0.751	**0.029**	0.616	0.034	0.604	18.041	**0.766**
MiddlePhalanxOutlineAgeGroup	92717.344	365.205	**0.578**	**0.008**	0.396	0.011	0.487	3.025	0.494
MiddlePhalanxOutlineCorrect	105576.528	386.936	**0.818**	**0.015**	0.529	0.023	0.636	7.836	0.636
MiddlePhalanxTW	104712.317	391.568	**0.510**	0.011	0.305	**0.010**	0.448	2.017	0.468
MoteStrain	696450.722	499.046	**0.922**	0.017	0.772	**0.014**	0.790	0.350	0.890
NonInvasiveFetalECGThorax1	1279491.548	1447.469	**0.953**	**0.491**	0.428	1.189	0.623	10819.214	0.758
NonInvasiveFetalECGThorax2	1278670.116	1431.964	**0.947**	**0.500**	0.516	1.644	0.721	10783.542	0.843
OSULeaf	162491.195	436.816	**0.977**	**0.023**	0.566	0.036	0.298	48.315	0.632
OliveOil	83127.123	363.010	0.567	**0.014**	0.367	0.019	**0.833**	3.313	0.633
PhalangesOutlinesCorrect	249551.291	449.700	**0.837**	**0.071**	0.565	0.234	0.649	72.732	0.679
Phoneme	839740.375	1731.913	**0.324**	**0.255**	0.177	0.270	0.114	2823.801	0.270
Plane	108051.158	395.829	**1.000**	0.012	0.990	**0.004**	0.848	0.457	**1.000**
ProximalPhalanxOutlineAgeGroup	82162.928	372.267	**0.839**	0.016	0.654	**0.013**	0.737	3.548	0.785
ProximalPhalanxOutlineCorrect	93458.148	385.890	**0.919**	**0.019**	0.667	0.027	0.698	8.306	0.729
ProximalPhalanxTW	88023.604	369.745	**0.776**	0.021	0.556	**0.012**	0.600	3.065	0.741
RefrigerationDevices	310522.109	548.520	**0.535**	**0.063**	0.469	0.070	0.435	435.121	0.477
ScreenType	312790.892	553.425	**0.628**	**0.064**	0.461	0.078	0.400	393.783	0.413
ShapeletSim	181992.606	422.037	**0.931**	0.022	0.800	**0.014**	0.478	13.939	0.778
ShapesAll	346662.006	599.892	**0.920**	**0.079**	0.522	0.090	0.407	529.949	0.773
SmallKitchenAppliances	330047.979	573.234	**0.752**	**0.063**	0.621	0.067	0.563	390.006	0.696
SonyAIBORobotSurface1	374001.286	423.091	**0.966**	0.010	0.805	**0.008**	0.787	0.088	0.732
SonyAIBORobotSurface2	578033.665	464.232	**0.978**	**0.013**	0.621	**0.013**	0.600	0.309	0.843
StarLightCurves	2588520.640	6062.343	**0.974**	**2.250**	0.886	5.023	0.861	46778.334	0.887
Strawberry	180661.494	436.475	**0.966**	**0.035**	0.922	0.043	0.935	64.682	0.938
SwedishLeaf	136632.578	448.516	**0.959**	**0.030**	0.528	0.038	0.610	31.475	0.762
Symbols	633664.205	695.532	0.926	**0.045**	0.866	0.046	0.855	20.277	**0.949**
SyntheticControl	90752.972	410.284	**0.997**	0.016	0.563	**0.013**	0.573	3.204	0.983
ToeSegmentation1	150702.977	408.871	**0.954**	**0.010**	0.842	0.015	0.737	3.494	0.798
ToeSegmentation2	120645.311	386.151	**0.908**	0.010	0.800	**0.007**	0.785	7.684	0.846
Trace	79660.783	376.322	**1.000**	0.010	0.940	**0.008**	**1.000**	8.487	0.990
TwoLeadECG	651991.678	489.810	**1.000**	0.018	0.861	**0.014**	0.763	0.279	0.922
TwoPatterns	367034.745	925.783	0.983	**0.623**	0.268	1.163	0.270	511.754	**1.000**
UWaveGestureLibraryAll	1320920.413	2673.218	0.846	**0.614**	0.342	1.087	0.384	15355.783	**0.918**
UWaveGestureLibraryX	610795.255	1293.986	**0.773**	**0.309**	0.391	0.754	0.405	1625.963	0.729
UWaveGestureLibraryY	614737.090	1284.347	**0.663**	**0.306**	0.317	0.769	0.326	1618.227	0.644
UWaveGestureLibraryZ	616038.114	1292.523	**0.749**	**0.310**	0.381	0.770	0.414	1620.951	0.658
Wafer	545608.004	1285.151	**0.997**	**0.731**	0.967	2.664	0.981	798.986	0.984
Wine	84334.573	362.018	0.500	**0.015**	**0.741**	0.017	0.611	1.323	0.685
WordSynonyms	187079.914	515.874	0.610	**0.033**	0.315	0.047	0.251	63.568	**0.674**
Worms	182298.571	398.691	**0.734**	**0.026**	0.610	0.028	0.481	147.322	0.519
WormsTwoClass	184298.922	396.439	**0.747**	**0.027**	0.688	**0.027**	0.584	147.939	0.636
Yoga	647321.432	1374.364	**0.873**	**0.181**	0.698	0.251	0.687	862.279	0.839
#MEAN	409982.369	715.908	**0.809**	**0.338**	0.608	0.688	0.595	1644.598	0.743
#MEDIAN	185689.418	454.502	**0.845**	**0.026**	0.603	0.033	0.600	33.639	0.758
#SD	472049.674	724.275	0.182	2.019	0.202	4.144	0.201	5967.368	0.162

5. CONCLUSION

In this work, a hybrid feature based and similarity based time series classification approach is proposed and its performance compared to the most popular (DTW-kNN) and recent deep network based (ResNet) algorithms have been investigated. The proposed approach is very fast with moderate classification accuracy and can be implemented for real life applications in resource constrained platforms. With the recent increase of sensor technologies and smartphone platforms, many smartphone based health care applications are developing. Those applications require low computational cost and resources for their implementation. The proposed approach of time series classification is low cost and suitable for such applications.

The comparative study with benchmark data sets also shows that for some of the data sets, the classification accuracy of the proposed approach is not statistically very different from DTW-kNN while computational cost is far less. Though the classification accuracy of the proposed approach on the average is quite poor compared to ResNet but implementation of ResNet is restricted to high computational resources and large number of training samples. Also the black box nature of ResNet hinders the interpretability of the classification process while it is easier to find out the characteristics of the applicable data sets with our proposed approach. We have not experimented with ensemble classifier due to their high computational burden. As our proposed approach seems to be quite fast and interpretable, it can be used for many practical applications where light computational burden is the primary criterion.

REFERENCES

Al-Naymat, G., Chawla, S. Taheri, J. 12/2009. Sparse DTW: A novel approach to speed up dynamic time warping, The 2009 Australasian Data Mining, vol. 101, Melbourne, Australia, ACM Digital Library, 117–127.

Antonucci, A., DeRosa, R., Giusti, A. et al., 2015. Robust classification of multivariate time series by imprecise hidden Markov models, International Journal of Approximate Reasoning, 56(B), 249–263.

Bagnall, A., Bostrom, A., Large, J, Lines, J. 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Mining and Knowledge Discovery, 31, 606–660.

Bagnall, A., Lines, J., Hills, J., Bostrom, A. 2015. Time series classification with COTE: The collective of transform-based ensembles, IEEE Trans. on Knowledge and Data Engineering, 27, 2522–2535.

Bagnall, A.J., Bostrom, A., Cawley, G.C., Flynn, M., Large, J., Lines, J. 2018. Is rotation forest the best classifier for problems with continuous features?, ArXiv, vol. abs/1809.06705.

Baydogan, M.G., Runger, G., Tuv, E. 2013. A bag-of-features framework to classify time series, IEEE Trans. on PAMI, 35, 2796–2802.

Bishop, Christopher, M. 1995. Neural networks for pattern recognition. Oxford university press.

Chakraborty B., Yoshida, S. 2016. Proposal of a new similarity measure for time series classification, Proc. ITISE 2016, Spain.

Chakraborty, B., Yoshida, S. 2017. A novel genetic algorithm based similarity measure for time series classification, Proceedings of ITISE 2017, 536–547.

Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.-C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Chen, Y., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G., Hexagon-ML, 2019. The UCR time series classification archive. URL https://www.cs.ucr.edu/~eamonn/time_

Deng, H., Runger, G.C., Tuv, E. Martyanov, V. 2013. A time series forest for classification and feature extraction, Inf. Sci., 239, 142–153.

Ebenezer, R.H.P., Isaaca, Susan Elias, Srinivasan Rajagopalan, Easwarakumar, K.S. 2019. Trait of Gait: A Survey on Gait Biometrics', https://arxiv.org/pdf/1903.10744.pdf

Fawaz, H.I., Forestier, G., Weber J., Idoumghar, L., Muller, P.A. 2019. Deep learning for time series classification :a review, Data Mining and Knowledge Discovery, 33, 917–963.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A. 2019. Deep learning for time series classification :a review, Data Mining and Knowledge Discovery, 1–47.

Fisher, T., Krauss, C. 2018. Deep learning with long short-term memory networks for financial market predictions. Eur. J. Oper. Res., 270, 654–669.

He, K., Zhang, X., Ren, S., Sun, J., NV, 2016. Deep residual learning for image recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 770–778.

Ito, H., Chakraborty, B. 2018. A proposal for cost aware edge-detectional dynamic time warping for time series classification, in Proceedings of iCAST 2018, 150–154.

Ito, H., Chakraborty, B. 2019. A proposal for shape aware feature extraction for time series classification, 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), Morioka, Japan, 1–6.

Karim, F., Majumdar, S., Darabi, H., Chen, S. 2018. LSTM fully convolutional networks for time series classification. In: IEEE Access 6, 1662–1669.

Keogh, E.J., Pazzani, M.J. 2000. Scaling up dynamic time warping for datamining applications, Proceedings of KDD, 285–289.

Kini, B.V., Sekhar, C.C. 2013. Large margin mixture of AR models for time series classification, Applied Soft Computing, 13, 361–371.

Lara, O.D., Labrador, M. 2013. A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. 15, 1192–1209.

Lines, J., Bagnall, A. 2015. Time series classification with ensembles of elastic distance measures, Data Mining and Knowledge Discovery, 29, 565–592.

Lines, J., Taylor, S., Bagnall, A. 2018. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation based ensembles, ACM Transactions on Knowledge Discovery from Data, 12, 1–52.35.

Sakoe, H., Chiba, S., Feb 1978. Dynamic programming algorithm optimization for spoken word recognition, in IEEE Transactions on Acoustics, Speech, and Signal Processing, 26, 43–49.

Salvador, S., Chan, P. 2007. FastDTW : Toward accurate dynamic time warping in linear time and space, Intell. Data Anal., 11, 561–580.

series_data_2018/

Smirnov, D., Nguifo, E.M. 2018. Time series classification with recurrent neural networks', https://project.inria.fr/aaldt18/files/2018/08/aaltd18rnn.pdf

Tamilarasi, K., Nithya Kalyani, S., April 2017. A survey on signature verification based algorithms. In Proceedings of the IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE), Karur, India, 27–28, 1–3.

Wang J., Liu P., She, M., Nahavandi S., Kouzani A. 2012. Bag-of-Words representation for biomedical time series classification. Biomedical Signal Processing and Control. 8. 10.1016/j.bspc.2013.06.004.

Wang, Z., Yan, W., Oates, T., May, 2017. Time series classification from scratch with deep neural networks: A strong base line. In Proceedings of IEEE IJCNN, Alaska, USA, 14–19, 1578–1585.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. Biometrics Bulletin, 1, 80–83.

Ye, L., Keogh, E.L., Ye, Keogh, E.J. 2010. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data Mining and Knowledge Discovery, 22, 149–182.