applied sciences

MDPI

Article

Interval Feature Transformation for Time Series Classification Using Perceptually Important Points

Lijuan Yan [1,2,*], Yanshen Liu [2] and Yi Liu [2]

[1] National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China
[2] Hubei Research Center for Educational Informationization, Central China Normal University, Wuhan 430079, China; yanshenliu@126.com (Y.L.); liuyi@mail.ccnu.edu.cn (Y.L.)
* Correspondence: yanlijuan@mails.ccnu.edu.cn

Received: 21 July 2020; Accepted: 3 August 2020; Published: 6 August 2020

Abstract: A novel feature reconstruction method, referred to as interval feature transformation (IFT), is proposed for time series classification. The IFT uses perceptually important points to segment the series dynamically into subsequences of unequal length, and then extract interval features from each time series subsequence as a feature vector. The IFT distinguishes the best top-k discriminative feature vectors from a data set by information gain. Utilizing these discriminative feature vectors, transformation is applied to generate new k-dimensional data which are lower-dimensional representations of the original data. In order to verify the effectiveness of this method, we use the transformed data in conjunction with some traditional classifiers to solve time series classification problems and make comparative experiments to several state-of-the-art algorithms. Experiment results verify the effectiveness, noise robustness and interpretability of the IFT.

Keywords: interval feature; transformation; time series classification; perceptually important points

1. Introduction

A time series is a sequence of time-indexed measurements and each measurement in practice can have its own statistical properties [1]. Time series exist in various real-life domains, such as finance, multimedia [2,3], medicine [4] and so on. Time series classification (TSC) has attracted significant interests in the data mining community. The TSC is different from the traditional classification problem [5], which is mainly due to the attributes in a time series are ordered. The TSC involves the learning of a function that maps a series into a class from a set of predefined classes [6]. To handle this problem, many methods have been proposed, which can be aggregated into two categories: instance-based method and feature-based method.

Many TSC studies have focused on the similarity measures between the two time series vectors for nearest neighbor (NN) classifiers, such as one-nearest-neighbor classifier with Euclidean distance (1NN-ED) [7,8] and one-nearest-neighbor classifier with dynamic time warping (1NN-DTW) [9–11]. Euclidean distance (ED) deals with time series of equal length, and it calculates time series point-to-point in the time axis but can't match similar shapes if they are out of phase in the time axis. Dynamic time warping (DTW) achieves "one-to-many" matching of time series data points through stretching or compressing the series. So, the 1NN-DTW is more robust to the distortion of the time axis. The experimental evidence has suggested that the 1NN-DTW can yield high classification performance for many data sets. However, these methods are computationally expensive.

Despite the evidence in favor of these instance-based methods, they are hard to provide understanding of why performance is good. In the field of applied scientific research, the most important work is to deeply understand patterns of time series in different classes which prompt

classification successfully. A spate of feature-based methods has been proposed in recent years, which reframe the problem in terms of interpretable features. In general, interpretable features can be derived from either simple statistic (e.g., mean and variance) of time intervals or just a subsequence of the time series (i.e., shapelets). Deng et al. [12] built a decision tree on three simple features: mean, standard deviation and slope. L. Ye and E. Keogh [13] constructed a decision tree classifier by recursively searching for discriminatory shapelets based on data split via an information gain measure, and also calculated the distance to the branching shapelets.

In recent years, some transformation methods have been proposed to solve the TSC problem. Time series transformation algorithms process series to create alternative data representations. Hills et al. [14] proposed shapelet transformation which is used to convert the raw time series data based on the shapelets to a different data representation. The representation contains features corresponding to a specific shapelet and its value is the minimal distance to the time series. The distances between the shapelets and the time series are computed, and then these distances, as features, are used to fit a logistic regression. Learning time-series shapelets (LS) learn the shapelets as well as the coefficients of the logistic regression [15]. Bag-of-words approaches are frequently used in time series classification. The Bag-of-SFA Symbols (BOSS) algorithm extracts words from a time series and builds features representing frequencies of each word for each time series [16]. The BOSS model transforms a time series into an unordered set of SFA words, which is fast and very robust to noise. Symbolic aggregate approximation (SAX) is a symbolic representation that allows for dimensionality reduction, lower bounding, and transformation of streamed data [17]. SAX represents time series as a string of characters. The SAX transformation is often combined with other methods to solve the problem of time series classification. Senin et al. [18] proposed an algorithm (SAXVSM) for time series classification based on SAX approximation of time series and vector space model. Studies have shown that the classifier can provide a greater level of performance improvement via creating a new classification data set independently. Such approaches can speed up run-time for the very long series and are not sensitive to noise data.

In this paper, we present a new transformation method based on the interval feature. Interval features are the temporal features calculated over time series intervals [19], which can capture the temporal characteristics. We propose the interval feature vector and evaluate their benefits to classification in order to get the discriminative feature vectors. The discriminative feature vectors are used to generate transformed data. The new transformation data set can be joined with any other classifier to solve the time series classification problem. Since interval features carry information of the time series, which is not based on individual time points, they are less sensitive to noise.

The remainder of this paper is organized as follows: Section 2 presents the related work. Section 3 describes the IFT method, including perceptually important points (PIP), interval feature vector and transform scheme. Section 4 demonstrates the effectiveness, noise robustness and interpretability of IFT by experimental studies. Discussion is given in Section 5.

2. Related Work

A feature-based method relies on extracting meaningful features from a time series. In time series classification tasks, some researchers choose to extract features over the global time domain while others prefer to extract features from data sets in which the raw time series is already segmented. Some time series classification problems involve class differences in time series properties that are restricted to specific discriminative time intervals. Interval classifiers seek to learn the location of discriminative subsequences and the features of separate different classes, which can be learned by computing simple features across many subsequences, and then building classifiers by searching over both features and time intervals [6].

Lu Wei et al. [20] fully exploited the amplitude and trends information of time series. In their method, the universe of the discourse of time series was firstly pre-divided into several intervals according to the predefined number of intervals to be partitioned. Then, information granules were

constructed in the amplitude-change space based on the data of the time series belonging to each of the intervals and their corresponding change (trends). Deng et al. [12] used statistics, such as the mean, slope, and variance to build a tree-based ensemble classifier that significantly outperformed one-nearest-neighbor classifiers with dynamic time warping. Ben D. Fulcher et al. [21] proposed a feature extraction approach that was beneficial for very large-size data, extracting approximately 9000 distinct features from data, including correlation structure, distribution, entropy, stationarity, and scaling properties, and fitted to a range of time-series models. Nanopoulos et al. [22] used the mean, standard deviation, skewness, and kurtosis of the time series and its successive increments to represent and classify control chart patterns. Jessica Lin et al. [23] introduced a histogram-based similarity measure, and this method counted the frequency of occurrences of each pattern in the time series, then compared the frequencies (or the histograms) of patters in the time series to obtain a (dis)similarity measure.

Since many time series are non-stationary with the variability of amplitude and frequency over time, new frequency domain features extracted from original time series may help to improve the performance of classification models. Techniques for frequency domain feature extraction include the Fourier transform (FT) [24], continuous wavelet transform (CWT) [25], Hilbert–Huang transform [26], ensemble empirical mode decomposition (EEMD) [27], the least-squares wavelet analysis (LSWA) [1] and so on.

3. Interval Feature Transformation

3.1. Perceptually Important Points (PIPs)

For long time sequences, it is more appropriate to measure the similarity by looking for features in specific discriminative time intervals, rather than point-to-point, local comparisons. In the time series mining system, the interesting and frequently appearing patterns usually can be abstractly represented by a few critical points. These points have perceptually important influences in the human vision [28]. By identifying the perceptually important points (PIPs) from the time domain, the time series can be segmented flexibly and effectively according to the needs of the users and the applications.

Tsinaslanidis et al. [29] used PIPs to dynamically segment price series into subsequences and used dynamic time warping (DTW) to find similar historical subsequences. Jimenez et al. [30] used the perceptually important points (PIP) process to represent and index each rime series of each station and used the rule set to classify the stations. Yu et al. [31] proposed a framework for mining emerging patterns from time series data, which transformed the time series data into a symbolic representation based on the SAX and PIPs algorithms. The previous studies have shown that the intuitive pattern matching can be carried out effectively by comparing time series segments of different lengths.

In this study, we use perceptually important points to segment the series dynamically into subsequences of unequal length. Now, we present the algorithm of constructing PIPs for time series segmentation.

A univariate time series is a sequence of data that is typically recorded in temporal order at fixed intervals. The number of real values is the length of the time series. A data set $T = \{T_1, T_2, T_3, \cdots T_n\}$ has n time series. Each time series T_i has m real-valued ordered data $\langle t_1, t_2, t_3, \cdots t_m \rangle$ and a class label l_i , then $T_i = \{t_1, t_2, t_3, \cdots t_m, l_i\}$.

For a given time series T_i, all the data points, $t_1, t_2, t_3, \cdots t_m$ in T_i will go through the PIPs identification algorithm. The algorithm starts by characterizing the first value t_1 and the last value t_m as the first two PIPs. Supposing we want to divide the time series into ω segments, we need to obtain $\omega - 1$ PIPs. Firstly, the algorithm calculates the distance between all remaining data points and the two initial PIPs, and also signifies as the third PIP the one with the maximum distance. Subsequently, the fourth PIP will be the point in T_i with the greatest distance to its two adjacent PIPs, either between the first and second PIPs or between the second and the last PIPs. The process of locating the PIPs continues until $\omega - 1$ PIPs are identified, as shown in Figure 1.

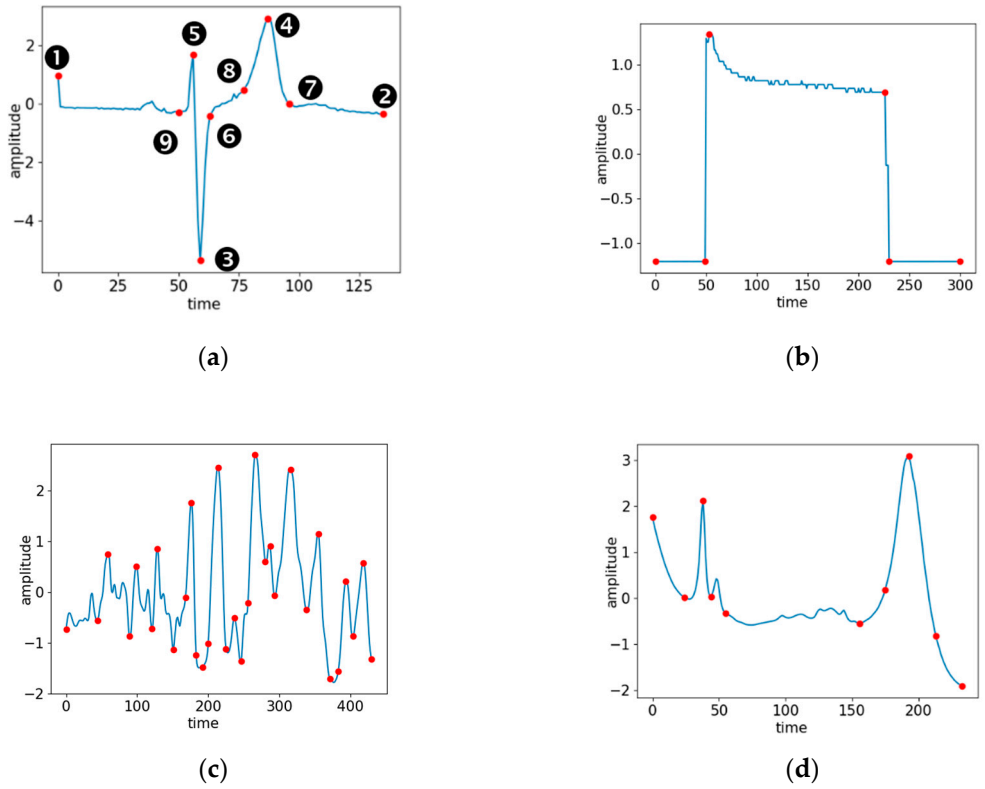


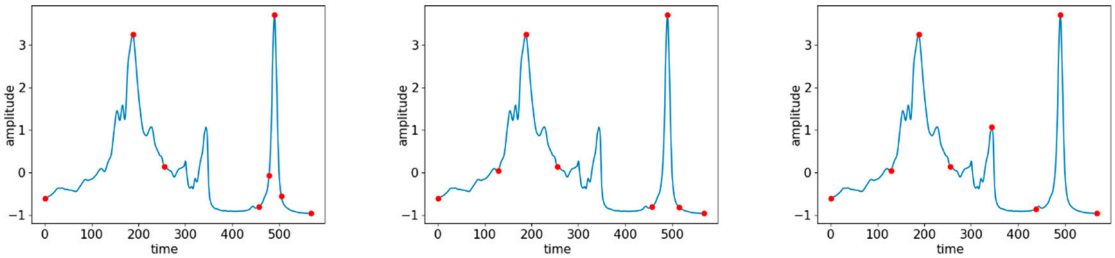
Figure 1. Examples of time series segments for four data sets, which are ECGFiveDays (ECG = electrocardiogram) (**a**), FreezerSmallTrain (**b**), Ham (**c**) and Wine (**d**), respectively.

Several importance evaluation methods have been proposed in this work. We chose vertical distance (VD) to evaluate the importance of the PIPs in a time series.

In this paper, we introduce a new parameter τ to make the PIPs well-distributed. The parament τ is a threshold for selecting important points that represents the minimum interval between adjacent PIPs. Once the interval between the new picked point with the biggest distance and the adjacent PIPs is less than this threshold, the point will be discarded and replaced with the point with the next biggest distance.

These two parameters, ω and τ, need be specified by the users manually. There are two ways to select the parameter ω. If we observe the sequence diagram, the waveform has obvious segmentation, we can directly specify the ω value. If not, we take 5 as the initial value and 5 as the step size. Through the test, the best parameter ω can be found. Figure 2 shows the influence of parameter τ selection for data set OliveOil. When the parameter τ is increased to 50, the extremum point which is circled in (c) can be captured very well. In Section 4.1, we give the combination of parameters which obtain the best results in our experiments.

After PIP, each time series $\langle t_1, t_2, t_3, \cdots t_m \rangle$ can be segmented into ω intervals $\langle s_1, s_2, s_3, \cdots s_\omega \rangle$.



(**a**) (**b**) (**c**)

Figure 2. Influence of parameter τ selection for data set OliveOil, which is $\tau = 1$ (**a**), $\tau = 25$ (**b**) and $\tau = 50$ (**c**), respectively.

3.2. Discriminative Feature Vector Selection

3.2.1. Internal Feature Vector

Interval features are calculated from a time series interval, e.g., "the interval between time 10 and time 30". Many types of features over a time interval can be considered, but one may prefer to define simple and interpretable features such as the mean and standard deviation [12]. Previous studies have shown that interval features are effective for the TSC problem [12,20–23].

In this paper, we adopt six statistical merits, the mean value μ, standard deviation σ,the sum of first difference absolute value (d), interquartile range (IQR), skewness ($SKEW$) and kurtosis ($KURT$). The first difference absolute value can reflect sensitively the contiguous data of quality indexes in the time sequence; however, the directional fluctuation is not considered. Skewness and kurtosis contain information on the shape of the distribution of the time series data. More precisely, $SKEW$ characterizes the degree of asymmetry of values around the mean value. $KURT$ measures the relative peakedness or flatness of the value distribution relative to a normal distribution. Suppose we have an interval between time p and time q ($p < q$), the interval features can be calculated by Equations (1)–(6) as shown below:

$$\mu = \frac{\sum_{i=p}^{q} t_i}{q-p} \tag{1}$$

$$\sigma = \sqrt{\frac{\sum_{i=p}^{q} (t_i - u)^2}{q-p}} \tag{2}$$

$$d = \sum_{i=1}^{n-1} \left| t_{i+1} - t_i \right| \tag{3}$$

$$IQR = Q_3 - Q_1 \tag{4}$$

$$SKEW = \frac{\sum_{i=p}^{q} (t_i - u)^3}{(q-p)\sigma^3} \tag{5}$$

$$KURT = \frac{\sum_{i=p}^{q} (t_i - u)^4}{(q-p)\sigma^4} - 3 \tag{6}$$

In this paper, we define a group of interval features with the above six statistical merits, $f_j(j = 1, 2, 3, \cdots \omega) = \left\langle \mu_j,\ \sigma_j,\ d_j, IQR_j, SKEW_j, KURT_j \right\rangle$ as an interval feature vector. In this way, every segment is replaced by the feature vector f. The original time series $\langle t_1, t_2, t_3, \cdots t_m \rangle$ can be represented by a set of feature vectors $\langle f_1,\ f_2,\ f_3, \cdots f_\omega \rangle$.

3.2.2. Selection Process

Interval feature vector selections are algorithms that attempt to identify and remove as much irrelevant and redundant information as possible prior to learning. In this paper, we try to extract k discriminative feature vectors. The corresponding algorithm contains three major steps:

Firstly, the algorithm performs a single scan of all candidate feature vectors and removes the duplicate and similar vectors, which are vectors with the same mean and variance values, or vectors with the same skewness and kurtosis values. Then it can obtain a final candidate list $\langle c_1,\ c_2,\ c_3, \cdots c_l \rangle$.

Secondly, the algorithm calculates the distance between each candidate c and each time series. In machine learning, cosine similarity is often adopted to express the similarity between two vectors. The algorithm measures the cosine of the angle between two vectors projected in a multi-dimensional space. Vectors that are very similar to each other have a cosine similarity that is close to 1. Vectors that are nearly orthogonal have a cosine similarity near 0. Vectors that point in opposite directions have a cosine similarity of −1. Let x and y be two feature vectors for comparison, the cosine similarity measure between x and y is given by Equation (7):

$$sim(x, y) = \frac{xy}{\|x\| \, \|y\|} \tag{7}$$

The cosine distance can be calculated by Equation (8):

$$cdis(x, y) = 1 - sim(x, y) \tag{8}$$

In this paper, the distance between candidate c and each time series T_i is defined as the minimum value of the cosine distance between c and each f in the T_i. If $c = \langle \mu,\ \sigma,\ d, IQR, \mathrm{SKEW}, KURT \rangle$ and $T_i = \langle f_{i1},\ f_{i2},\ f_{i3}, \cdots f_{i\omega} \rangle$, then the cosine distance from c to T_i is defined as Equation (9).

$$\mathrm{dist}(\mathrm{c}, T_i) = \min_{\mathrm{j} \in (0, \omega)} \left(\mathrm{cdist}\left(c, f_{ij} \right) \right) \tag{9}$$

After comparing the similarity of each candidate c and all the time series in T, the calculation result of each candidate will be ordered to search the split point and get the information gain (IG) as described in Section 3.2.3.

Finally, all the candidates are accessed, and they are sorted according to the IG_c. Then, we select the k feature vectors with the biggest IG_c as the discriminative features.

In general, discriminative features can be extracted by comparing the IG_c of every candidate.

3.2.3. Selection Measure

In probability theory and information theory, IG is asymmetric to measure the difference between the two probability distributions. IG has been approved to be effective as a similarity metric [32,33]. After calculating all the distances between a candidate and all the time series in T, it will get a list D_c with n distance values. D_c is sorted, and the IG at each possible split point sp is then assessed; here, a valid split point is defined as the average between any two consecutive distances in D_c. For each possible split point sp, as shown in Figure 3, IG is calculated by partitioning all elements in D_c into two sets, SP_h and SP_r. The IG at sp is calculated as Equation (10):

$$IG(D_c, sp) = H(D_c) - \left(\frac{|SP_h|}{|D_c|} H(SP_h) + \frac{|SP_r|}{|D_c|} H(SP_r) \right) \tag{10}$$

where $|D_c|$ is the length of the set D_c and $H(D_c)$ is the entropy of D_c. The $H(D_c)$ is defined as Equation (11).

$$H(D_c) = -\sum_{v \in V} p_v \log p_v \tag{11}$$

where v is the class label and p_v is the probability of each label.

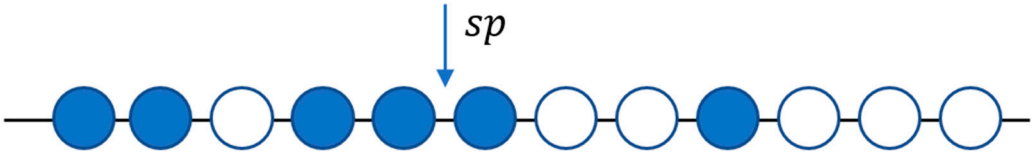


Figure 3. Split point.

The IG of candidate c is expressed as IG_c, which is calculated as Equation (12).

$$IG_c = \max_{sp \in D_c} IG(D_c, sp) \tag{12}$$

According to the order of IG_c value, the k feature vectors with the largest IG_c value are retained as the discriminative feature vectors to prepare for the next data transformation. The above procedure can be used to calculate the IG_c of each candidate in the binary classification problem. For the multiclass classification problem, we use a "one vs. all" strategy to obtain the IG_c of each candidate. Suppose we have a three-class classification problem, the class labels are 1, 2, 3, respectively. When we calculate the IG_c of the candidate in class 1, we set class 1 as the positive class and the other two classes as the negative class. So, we will calculate three times to get the IG_c of candidate c in each class label. In the above results, the k discriminative feature vectors can be retained in the multiclass classification problem.

3.3. Data Transform

The data transformation process is carried out using the cosine distance calculation described in Section 3.2.2. Feature set K, composed of the k discriminative feature vectors, is used to generate a new data set. For each instance of data T_i, the distance is computed between T_i and K_j, where $j = 1, 2 \ldots k$. The calculated k distances are used to form a new instance of transformed data, where each attribute corresponds to the distance between a discriminative feature vector and the original time series.

While the data is split into the training data set and test data set, interval feature vector selection (as described in Section 3.2) is carried out on the training data only to avoid bias; the selected discriminative features are then used to transform each instance of the training and test data to create transformed data sets, which can then be used with any traditional classifiers.

4. Experimental Studies

4.1. Experimental Design

To evaluate the IFT, we have selected 22 classification data sets in the University of California, Riverside time series classification archive (UCR) [34] (as listed in Table 1). All data sets are of labeled, univariate time series, without any preprocessing. We used the default train and test data splits. In our experiments, the parameter k is usually set as half of the time series length. When the length of some of the data sets is too large, in order to reduce the computing burden, the parameter k is selected according to the maximum candidate length. The parameter combinations used in the experiment are listed in Table 1.

Table 1. Summary of data sets and the combination of parameters in the experiment.

#	Dataset	Type	Train	Test	Length	Class	ω	τ	k
1	BirdChicken	Image	20	20	512	2	25	5	256
2	FreezerRegularTrain	Sensor	150	2850	301	2	10	2	150
3	ShapeletSim	Simulated	20	180	500	2	25	5	250
4	ToeSegmentation1	Motion	40	228	277	2	16	5	138
5	Worms	Motion	181	77	900	5	10	1	450
6	Rock	Spectrum	20	50	2844	4	5	1	100
7	Meat	Spectro	60	60	448	3	20	2	224
8	Beef	Spectro	30	30	470	5	40	5	235
9	InlineSkate	Motion	100	550	1882	7	8	1	400
10	Coffee	Spectro	28	28	286	2	15	5	143
11	DodgerLoopGame	Sensor	20	138	288	2	20	2	144
12	DodgerLoopWeekend	Sensor	20	138	288	2	20	2	144
13	ECGFiveDays	ECG	23	861	136	2	14	5	68
14	Ham	Spectro	109	105	431	2	20	10	215
15	Herring	Image	64	64	512	2	20	10	256
16	PowerCons	Power	180	180	144	2	24	2	72
17	Wine	Spectro	57	54	234	2	12	5	117
18	Yoga	Image	300	3000	426	2	5	2	213
19	FaceFour	Image	24	88	350	4	20	10	175
20	OliveOil	Spectro	30	30	570	4	7	50	200
21	Fish	Image	175	175	463	7	6	1	231
22	Plane	Sensor	105	105	144	7	10	1	72

The purpose of the experiments is to try to answer the following questions: (1) Does the IFT improve or reduce the classification result? Is the result sensitive to the noise? (2) How is the performance of the IFT classifier compared to the other classifiers? (3) Are the extracted features interpretable?

In the experiments, we use the IFT to transform the data set, and then utilize the transformed data in conjunction with common classifiers to solve the TSC problems. At first, we compare the performances of four classifiers on the interval feature transformed data to the performance of 1NN-DTW on the raw data. These classifiers include random forest (RF), support vector machine (SVM), eXtreme Gradient Boosting(XGBOOST), and gradient tree boosting (GBDT). Additionally, we complete a noise sensitivity analysis. Secondly, in order to test the performance of the IFT classifier, we compare the IFT classifier with the following baselines: 1NN-DTW, LS and BOSS, Time Series Forest(TSF) [12] and SAXVSM as mentioned in Section 1.

The comparisons to the above state-of-the-art methods are used to testify the effectiveness of the proposed method in this paper. We used the Python programming language with sklearn [35] and pyts [15] to implement all the algorithms in experiments.

4.2. Evaluating Indicator

To the classification problem, classification accuracy is the most important criterion to evaluate algorithm performance. In addition to accuracy, the Friedman test and Nemenyi test are widely used in machine learning to evaluate the performance of algorithms over multiple data sets. In order to consider the efficiency of the IFT method, we establish the multi-classifier comparison procedure over multiple data sets suggested by Demšar [36]. The Friedman test [37] is followed by the Nemenyi test [38] if the Friedman test shows a significant difference between the classifiers.

After getting the performance of the K algorithms on the N data set, the Friedman test ranks algorithms for each data set separately. When multiple algorithms have the same result for a data set,

the average rank is used. In this way, we can get a rank matrix of $N \times K$. r_{ij} is the rank mark of the ith data set on the jth algorithm, and the average ranges R_j are calculated as shown in Equation (13).

$$R_j = \frac{1}{N} \sum_{j=1}^{N} r_{ij} \tag{13}$$

Under the null hypothesis, all algorithms are equivalent, so their R_j should be equal. The Friedman statistics are defined as Equation (14),

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^{K} R_j^2 - \frac{K(K+1)^2}{4} \right] \tag{14}$$

which is according to χ_F^2 with $K-1$ degrees of freedom.

The research of Demiša et al. [36] shows that Friedman's statistics are too conservative, and proposed a better statistical formula such as Equation (15),

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2} \tag{15}$$

which is according to the F-distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom.

If the null hypothesis is rejected, indicating significant differences between these algorithms, the difference between the algorithms can be tested by the Nemenyi test to compare all the algorithms to each other. At a significance level of α, the critical difference (CD) value is defined as Equation (16).

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}} \tag{16}$$

All algorithms were divided into different groups by CD value, so that there was no significant difference in the performance of the algorithms in the group. In this way, performance differences between different algorithms can be represented by the critical difference diagram.

4.3. Results

Our first objective is to establish that IFT does not reduce classification performance. For comparison purposes, we test the performance of four traditional classifiers on the transformed data constructed by the IFT method and test the performance of 1NN-DTW on the raw data. Table 2 lists the accuracy results from the four classifiers on the interval feature transformed data. The win–lose–tie results of each IFT classifier compared to the 1NN-DTW on raw data, and the average rank of each classifier is also calculated.

As shown in Table 2, at least one IFT classifier out of four is better than 1NN-DTW on the raw data in 11/22 data sets. For the transformed data, the random forest classifier is the most sensitive classifiers on average.

There are several data sets where the interval feature transformation was detrimental to the accuracy of the classifier. For example, on the data set FaceFour, all the classifiers lose about 25% on accuracy. However, some motion data sets, image data sets and simulated data sets (Worms, ToeSegmentation1, BirdChicken, ShapeletSim) show good improvement with the IFT method implemented.

Table 2. Testing accuracy of one-nearest-neighbor classifier with dynamic time warping (1NN-DTW) on raw data, and four interval feature transformation (IFT) classifiers. RF: Random Forest; SVM: Support Vector Machine; XGBOOST: eXtreme Gradient Boosting; GBDT: Gradient Tree Boosting.

#	1NN-DTW	IFT + RF	IFT + SVM	IFT + XGBOOST	IFT + GDBT
1	0.7500	**0.8000**	**0.9000**	**0.7500**	**0.7500**
2	0.9042	0.8968	0.9035	0.8940	0.8958
3	0.6944	**0.9833**	**0.9944**	**0.9944**	**0.9778**
4	0.7368	**0.8728**	**0.8816**	**0.8246**	**0.8114**
5	0.4935	**0.6623**	**0.5974**	**0.6104**	**0.6234**
6	0.6400	0.6000	0.6200	0.5600	0.6000
7	0.9333	**0.9500**	0.8500	**0.9500**	0.9167
8	0.6667	**0.6667**	**0.6667**	0.4667	0.4667
9	0.3836	0.3491	0.3400	0.3582	0.3109
10	1.0000	0.9643	0.9643	0.8929	0.7857
11	0.8768	0.6693	0.6693	0.5433	0.6142
12	0.9493	0.8016	0.7778	0.7460	0.7778
13	0.7677	0.7027	**0.8281**	0.7247	0.7607
14	0.5905	**0.6381**	0.5619	0.5741	0.5238
15	0.5312	**0.6719**	**0.5625**	**0.6250**	**0.5938**
16	0.9667	0.9056	0.9222	0.9167	0.9333
17	0.5741	**0.7222**	**0.7037**	**0.7407**	**0.5926**
18	0.8363	0.7767	0.6720	0.7693	0.7553
19	0.8977	0.6250	0.6477	0.5568	0.4659
20	0.8333	0.7333	0.7667	0.7667	0.7333
21	0.8229	0.7600	0.8114	0.7886	0.7086
22	1.0000	**1.0000**	0.9810	0.9905	**1.0000**
Average	0.7659	0.7614	0.7556	0.7293	0.7090
Win/tie/lose		9/1/12	7/1/14	6/1/15	5/2/15

The better result for each classifier compared with 1NN-DTW is shown in bold.

We perform a noise sensitivity analysis. Additive white Gaussian noise with specified signal noise ratio (SNR) is added to the original signal. According to the SNR, the noise level is divided into 11 levels: 50, 45, 40, 35, 30, 25, 20, 15, 10, 5 and 0 dB. The test results on OliveOil and ShapeletSim data sets are shown in Figure 4. The results in Figure 4 show that the introduction of noise affects the accuracy of classification in a way. For example, in the ShapeletSim data set, when the SNR value is 0, the original signal is almost submerged by noise, and the accuracy is reduced by nearly 50%. However, when the SNR value is in the range from 15 to 50, the classification accuracy of the two data sets is stable relatively. This shows that ITF has a good noise robustness.

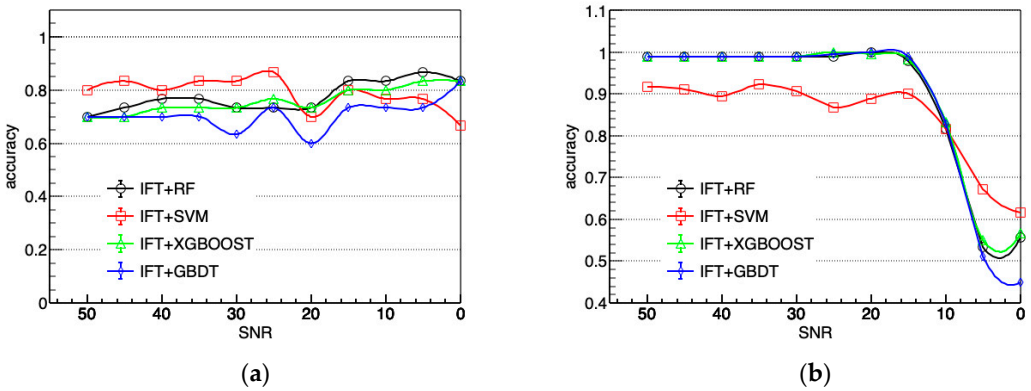


Figure 4. The results of a noise sensitivity analysis on two datasets, which are OliveOil dataset (**a**) and ShapeletSim dataset (**b**), respectively. IFT: Interval Feature Transformation; RF: Random Forest; SVM: Support Vector Machine; XGBOOST: eXtreme Gradient Boosting; GBDT: Gradient Tree Boosting; SNR: Signal Noise Ratio.

In this step, we confirm that the IFT method can improve the classification performance over several different data sets and is less sensitive to the additive noise.

The comparative experiment results are listed in Table 3. The average rank of each classifier is calculated. When the significance level is 0.05 and the degree of freedom is (5, 105), $F_F = 9.2269 > F_{0.05}(5105) = 2.301$. Therefore, at the significant level of 0.05, the six classifiers are significantly different. Then, according to Equation (14), $CD = 1.6076$ for $\alpha = 0.05$. As shown in Table 3, the IFT classifier achieves the best average rank, and best performance in six out of 22 problems. The average rank of IFT is very close to the TSF method, but the TSF has the best performance in more data sets.

Table 3. Classification performance as represented by accuracy values of the IFT classifier compared to 1NN-DTW and several feature-based methods over multiple time series data sets. LS: Learning Time-Series Shapelets; BOSS: Bag-of-SFA Symbols; TSF: Time Series Forest; SAXVSM: Symbolic Aggregate approXimation and Vector Space Model.

#	1NN-DTW	LS	BOSS	TSF	SAXVSM	IFT
1	0.7500	0.9000	0.9500	0.9000	0.5000	0.9000
2	0.9042	0.8079	0.8242	0.9958	0.6800	0.9035
3	0.6944	0.9889	0.6389	0.5000	0.6500	0.9944
4	0.7368	0.9167	0.7982	0.7632	0.8991	0.8816
5	0.4935	0.5325	0.6234	0.5714	0.2208	0.6623
6	0.6400	0.3200	0.4200	0.8800	0.1200	0.6200
7	0.9333	0.3333	0.8333	0.9333	0.7500	0.9500
8	0.6667	0.4000	0.5667	0.7667	0.5000	0.6667
9	0.3836	0.3000	0.2800	0.3673	0.1564	0.3582
10	1.0000	0.5000	0.8214	1.0000	0.9643	0.9643
11	0.8768	0.8583	0.5906	0.8031	0.6378	0.6693
12	0.9493	0.9762	0.7937	0.9841	0.7937	0.8016
13	0.7677	0.4971	0.6655	0.9872	0.7793	0.8281
14	0.5905	0.6190	0.5905	0.7524	0.5810	0.6381
15	0.5312	0.5938	0.6250	0.5625	0.4062	0.6719
16	0.9667	0.7833	0.9111	1.0000	0.7278	0.9333
17	0.5741	0.5000	0.4074	0.6667	0.5370	0.7407
18	0.8363	0.5473	0.6900	0.7463	0.6083	0.7767
19	0.8977	0.3977	0.5795	0.7614	0.9091	0.6477
20	0.8333	0.4000	0.8000	0.8667	0.5000	0.7667
21	0.8229	0.2229	0.5257	0.5429	0.3943	0.8114
22	1.0000	0.9905	0.9619	0.9429	0.9810	1.0000
Average rank	2.7045	4.3182	4.1818	2.5227	4.7727	2.5000
Win	6	1	1	9	1	6

The critical difference diagram is shown in Figure 5, and it shows that there is an obvious difference between the IFT method and Symbolic Aggregate approXimation and Vector Space Model (SAXVSM) and LS and BOSS, the difference between the IFT method, TSF and 1NN-DTW is found but not significant.

The proposed method is applied to several real-world applications. The ECGFiveDays data set from PhysioNet.Org [39] is a long ECG time series recorded on two different days with the same patient; a copy of this dataset can also be found at [34]. The dataset contains 890 objects, with 23 objects as training data and 861 as test data. As shown in Figure 6, the time series from two different classes are very similar, at least globally.

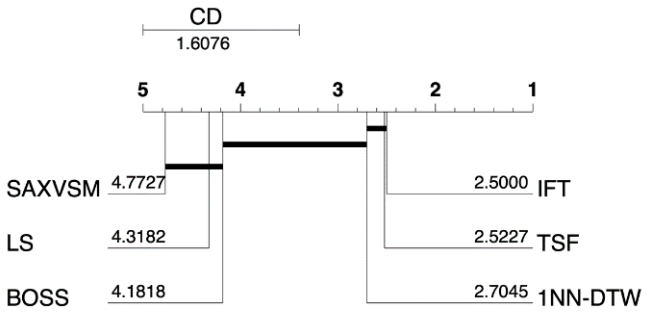


Figure 5. Critical difference diagram for six classifiers derived from results in Table 3. LS: Learning Time-Series Shapelets; BOSS: Bag-of-SFA Symbols; CD: Critical Difference; 1NN-DTW: One-Nearest-Neighbor Classifier with Dynamic Time Warping; TSF: Time Series Forest; SAXVSM: Symbolic Aggregate approXimation and Vector Space Model.

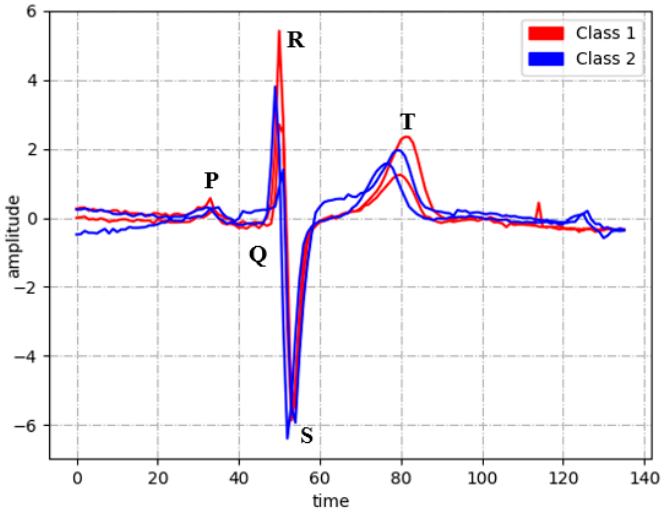
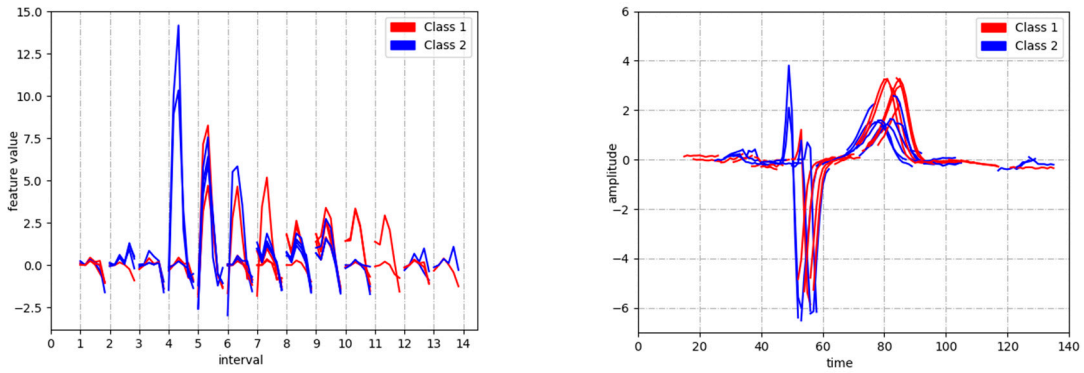


Figure 6. Two classes of time series from the ECGFiveDays.

An electrocardiogram (ECG) is a test that measures the electrical activity of the heartbeat. With each beat, an electrical impulse (or "wave") travels through the heart. This wave causes the muscle to squeeze and pump blood from the heart. A normal heartbeat on ECG will show the timing of the top and lower chambers. The right and left atria or upper chambers make the first wave called a "P wave", following a flat line when the electrical impulse goes to the bottom chambers. The right and left bottom chambers or ventricles make the next wave called a "QRS complex." The QRS complex duration is measured from the beginning of the Q wave to the end of the S wave. A complete QRS complex consists of a Q-, R- and S-wave. The final wave or "T wave" represents electrical recovery or return to a resting state for the ventricles. A healthy person's electrocardiogram has a certain pattern.

The experiment result of the feature extraction with our IFT method from the ECGFiveDays data set is shown in Figure 7. We extract 68 discriminative feature vectors from 14 intervals in each time series as the patterns of time series, as shown in Figure 7a. After converting these sets to the original subsequence, as shown in Figure 7b, we observe that our method can capture the feature pattern of the QRS complex and T wave in ECG very well. In addition to improvement of the classification performance, our proposed method can also provide interpretable features.



(**a**) (**b**)

Figure 7. (**a**) Sixty-eight feature vectors extracted by the proposed method. (**b**) Interval features found by the proposed method.

5. Discussion

Both the good performance and the interpretability are desirable for classifier in the TSC studies. Previous studies, for example the instance-based classifiers which can obtain an accurate performance but are limited to the interpretability. Interval features can be used to capture feature patterns for classification. The interval feature transformation (IFT) proposed in this paper provides a novel transformation method based on the interval feature. It transforms the original time series into a lower dimension representation, and any traditional classifier can be performed on the transformed data constructed by the IFT method to pursue the higher accuracy. In this study, we apply the IFT classifier to the 22 univariate data sets with current state-of-the art TSC methods, and the experimental results showing that the IFT classifiers have better performance and interpretable features are obtained at the same time.

Future work could address the improvement of the computation time for extracting the discriminative feature vector. The IFT uses six statistical features in this study, and we will reduce the number of statistical features and adopt a more powerful statistical merit in the future.

In addition, this study only involves the classification of univariate time series, and the proposed method can be further applied to the classification of multivariate time series. For example, a feature can be extracted from the interval of each dimension to form a feature matrix. By quantifying the classification ability of different dimensions, k dimensions with the strongest classification ability are finally selected, and the interval features of these k dimensions can be used to construct the new transformation data. The efficiency of the features and the quantitative standards need further experimental research support. As a flexible framework, IFT can be applied to different time series analysis scenarios.

Author Contributions: Software, L.Y.; writing—original draft preparation, L.Y.; supervision, Y.L. (Yanshen Liu); project administration, Y.L. (Yi Liu). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Hubei Research Center for Educational Informationization (Central China Normal University).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ghaderpour, E.; Pagiatakis, S.D. Least-Squares Wavelet Analysis of Unequally Spaced and Non-stationary Time Series and Its Applications. *Math. Geosci.* **2017**, *49*, 819–844. [CrossRef]
2. Abdel-Hamid, O.; Deng, L.; Yu, D. Exploring convolutional neural network structures and optimization techniques for speech recognition. *Interspeech* **2013**, *11*, 73–75.

3. Abdel-Hamid, O.; Mohamed, A.R.; Jiang, H.; Penn, G. Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing, Kyoto, Japan, 25–30 March 2012.
4. Wang, J.; Liu, P.; She, M.F.; Nahavandi, S.; Kouzani, A. Bag-of-words representation for biomedical time series classification. *Biomed. Signal Process. Control* **2013**, *8*, 634–644. [CrossRef]
5. Lines, J. Time Series Classification through Transformation and Ensembles. Ph.D. Thesis, University of East Anglia, Norwich, UK, 2015.
6. Fulcher, B.D. Feature-based time-series analysis. In *Feature Engineering for Machine Learning and Data*; CRC Press: Boca Raton, FL, USA, 2018.
7. Masip, D.; Vitrià, J. Boosted discriminant projections for nearest neighbor classification. *Pattern Recognit* **2006**, *39*, 164–170. [CrossRef]
8. Goldstein, M. k_n-nearest neighbor classification. *IEEE Trans. Inf. Theory* **2003**, *18*, 627–630. [CrossRef]
9. Ratanamahatana, C.A.; Keogh, E. Making time-series classification more accurate using learned constraints. In Proceedings of the Fourth SIAM International Conference on Data Mining, Lake Buena Vista, FA, USA, 22–24 April 2004. [CrossRef]
10. Jeong, Y.S.; Jeong, M.K.; Omitaomu, O.A. Weighted dynamic time warping for time series classification. *Pattern Recognit.* **2011**, *44*, 2231–2240. [CrossRef]
11. Yu, D.; Yu, X.; Hu, Q.; Liu, J.; Wu, A. Dynamic time warping constraint learning for large margin nearest neighbor classification. *Inf. Sci.* **2011**, *181*, 2787–2796. [CrossRef]
12. Deng, H.; Runger, G.; Tuv, E.; Vladimir, M. A Time Series Forest for Classification and Feature Extraction. *Inf. Sci.* **2013**, *239*, 142–153. [CrossRef]
13. Ye, L.; Keogh, E. Time series shapelets: A new primitive for data mining. *Knowl. Discov. Data Min.* **2009**, 947–956. [CrossRef]
14. Hills, J.; Lines, J.; Baranauskas, E.; Mapp, J.; Bagnall, A. Classification of time series by shapelet transformation. *Data Min. Knowl. Discov.* **2013**, *28*, 851–881. [CrossRef]
15. Faouzi, J.; Janati, H. pyts: A python package for time series classification. *J. Mach. Learn. Res.* **2020**, *21*, 1–6.
16. Schäfer, P. The BOSS is concerned with time series classification in the presence of noise. *Data Min. Knowl. Discov.* **2015**, *29*, 1505–1530. [CrossRef]
17. Patel, P.; Keogh, E.; Lin, J.; Lonardi, S. Mining Motifs in Massive Time Series Databases. In Proceedings of the 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, 9–12 December 2002.
18. Senin, P.; Malinchik, S. SAX-VSM: Interpretable Time Series Classification Using SAX and Vector. In Proceedings of the 2013 IEEE 13th International Conference on Data Mining, Dallas, TX, USA, 7–10 December 2013.
19. Rodríguez, J.J.; Alonso, C.J.; Boström, H. Boosting interval based literals. *Intell. Data Anal.* **2001**, *5*, 245–262. [CrossRef]
20. Lu, W.; Chen, X.; Pedrycz, W.; Liu, X.; Yang, J. Using interval information granules to improve forecasting in fuzzy time series. *Int. J. Approx. Reason.* **2015**, *57*, 1–18. [CrossRef]
21. Fulcher, B.D.; Jones, N.S. Highly Comparative Feature-Based Time-Series Classification. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 3026–3037. [CrossRef]
22. Nanopoulos, A.; Alcock, R.; Manolopoulos, Y. Feature-based Classification of Time-series Data. *Int. J. Comput. Res.* **2001**, *10*, 49–61.
23. Lin, J.; Li, Y. Finding Structural Similarity in Time Series Data Using Bag-of-Patterns Representation. In *International Conference on Scientific and Statistical Database Management*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 461–477. [CrossRef]
24. Sutcliffe, P.R. Fourier transformation as a method of reducing the sampling interval of a digital time series. *Comput. Geosci.* **1988**, *14*, 125–129. [CrossRef]
25. Hariharan, G. Wavelet Analysis—An Overview. In *Wavelet Solutions for Reaction–Diffusion Problems in Science and Engineering*; Forum for Interdisciplinary Mathematics; Springer: Singapore, 2019.
26. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [CrossRef]

27. Ren, H.; Wang, Y.L.; Huang, M.Y.; Chang, Y.L.; Kao, H.M. Ensemble empirical mode decomposition parameters optimization for spectral distance measurement in hyperspectral remote sensing data. *Remote Sens.* **2014**, *6*, 2069–2083. [CrossRef]
28. Yu, J.; Yin, J.; Zhou, D.; Zhang, J. A Pattern Distance-Based Evolutionary Approach to Time Series Segmentation. In *Intelligent Control and Automation*; Springer: Berlin/Heidelberg, Germany, 2006.
29. Tsinaslanidis, P.E.; Kugiumtzis, D. A prediction scheme using perceptually important points and dynamic time warping. *Expert Syst. Appl.* **2014**, *41*, 6848–6860. [CrossRef]
30. Jiménez, P.; Nogal, M.; Caulfield, B.; Pilla, F. Perceptually important points of mobility patterns to characterise bike sharing systems: The Dublin case. *J. Transp. Geogr.* **2016**, *54*, 228–239. [CrossRef]
31. Yu, H.H.; Chen, C.H.; Tseng, S. Mining Emerging Patterns from Time Series Data with Time Gap Constraint. *Int. J. Innov. Comput. Inf. Control* **2011**, *7*, 5515–5528.
32. Ye, L.; Keogh, E. Time series shapelets: A novel technique that allows accurate, interpretable and fast classification. *Data Min. Knowl. Discov.* **2011**, *22*, 149–182. [CrossRef]
33. Mueen, A.; Keogh, E.; Young, N.E. Logical-Shapelets: An Expressive Primitive for Time Series Classification. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21 August 2011; pp. 1154–1162.
34. Hoang, A.D.; Eamonn, K.; Kaveh, K.; Chin-Chia, M.Y.; Yan, Z.; Shaghayegh, G.; Chotirat, A.R.; Chen, Y.P.; Hu, B.; Nurjahan, B.; et al. The UCR Time Series Classification Archive. Available online: https://www.cs.ucr.edu/~{}eamonn/time_series_data_2018/ (accessed on 1 October 2018).
35. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2012**, *12*, 2825–2830.
36. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
37. Friedman, M. A comparison of alternative tests of significance for the problem of m rankings. *Ann. Math. Stat.* **1940**, *11*, 86–92. [CrossRef]
38. Nemenyi, P.B. Distribution-Free Multiple Comparisons. Ph.D. Thesis, Princeton University, Princeton, NJ, USA, 1963.
39. Physical Activity Monitoring for Aging People. Available online: http://www.pamap.org (accessed on 1 February 2011).