A Feature Extraction Method for Multivariate Time Series Classification Using Temporal Patterns

Pei-Yuan Zhou(✉) and Keith C.C. Chan

Department of Computing, The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
choupeiyuan@gmail.com, cskcchan@comp.polyu.edu.hk

Abstract. Multiple variables and high dimensions are two main challenges for classification of Multivariate Time Series (MTS) data. In order to overcome these challenges, feature extraction should be performed before performing classification. However, the existing feature extraction methods lose the important correlations among the variables while reducing high dimensions of MTS. Hence, in this paper, we propose a new feature extraction method combined with different classifiers to provide a general classification strategy for MTS data which can be applied for different area problems of MTS data. The proposed algorithm can handle data of high feature dimensions efficiently with unequal length and discover the relationship within the same and between different component univariate time series for MTS data. Hence, the proposed feature extraction method is application-independent and therefore does not depend on domain knowledge of relevant features or assumption about underling data models. We evaluate the algorithm on one synthetic dataset and two real-world datasets. The comparison experimental result shows that the proposed algorithm can achieve higher classification accuracy and F-measure value.

Keywords: Multivariate time series · Time series classification · Intra-temporal patterns · Inter-temporal pattern

1 Introduction

A multivariate time series (MTS) can be considered as made up of a collection of data values taken by a set of temporally interrelated variables monitored over a period of time at successive time instants spaced at uniform time intervals [4]. Effective classifying of such data can be applied into various problem domains. For example, in medicine and healthcare, the values taken by many variables representing different signs and symptoms may be temporally related or interrelated and they have to be monitored for a patient over a period of time for such relationship or interrelationship to be discovered. In financial analysis, as another example, the performance of a stock in terms of such variables as highs and lows, opening and closing prices, trading volumes, may also be temporally related or interrelated and for such relationship or interrelationship to be understood.

T. Cao et al. (Eds.): PAKDD 2015, Part II, LNAI 9078, pp. 409–421, 2015.
DOI: 10.1007/978-3-319-18032-8_32

Time series classification has received a great deal of attention in the past, and it also brings some new challenges to the data mining and machine learning community [1]. A number of different approaches have been proposed for univariate time series classification [9-11], however, few papers are found about multivariate time series classification in the literature [6].

Generally, to classify MTS data, feature extraction need to be performed for the original data, and then the classifiers, such as Support Vector Machine (SVM) or Artificial Neural Network (ANN), is used to classify the feature vectors [14-16]. The challenges of the classification process can be summarized as follows [2,7]: i) The most of the classifiers (e.g. decision trees, neural networks) can only take input data as a vector of features, but there are no explicit features in sequence data. ii) The dimensionality of feature space should be very high and the computation is costly. iii) The important correlations information among variables may be lost if the value of one variable is broken into MTS or each processed separately. Finally, the MTS can be of different lengths that cannot be extracted features by traditional method easily.

Hence, for satisfying the above requirement, we propose a new feature extraction algorithm combined with traditional classifier that provides a general classification strategy, called as Multivariate Time Series Classifier (MTSC). The algorithmic contributions of the proposed algorithm are: i) it can handle MTS data of high feature dimensions efficiently with unequal length; ii) it focuses on discovering the relationship in the same or among different variables; iii) it is a general method that can be applied to different problems on MTS data.

The structure of this paper is arranged as follows. In Section 2, we present a summary of existing work on feature extraction and classification for MTS data. In Section 3, we describe the details of the proposed algorithm. In Section 4, the results of the algorithm for both simulated and real world data sets are performed and presented. In the same section, we discuss results of the various tests carried out for effectiveness of its tasks. In the last section, we present a summary of the paper and the possible directions for future work.

2 Literature Survey

A multivariate time series (MTS) can be defined as a sequence of vectors, which may carry a class label [2]. For example, Electrocardiography (ECG) is a kind of MTS data, which is recorded from several sensors to describe the electrical activity of the heart, and it may come from either a healthy or ill person, labeled as "healthy" and "ill". The classification of MTS is the problem of classifying a set of MTS samples into a pre-defined set of classes [8]. To summary the existing algorithms of classification for MTS data, two main steps are considered: extract features using classical feature extraction method as pre-processing for MTS data, and classify feature vectors using classifier.

Feature extraction greatly affects the design and performance of the classifier and feature extraction is to use the existing feature parameters to comprise a lower-dimensional feature space, map useful information contained by original features to a small number of features, ignoring redundant and irrelevant information [15]. The classical feature extraction methods are based on statistical analysis. As the representative, there are some classical methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [16], Factor Analysis [16], and so on [15]. In addition, for improving the performance of classification MTS data, some new feature extraction methods are proposed. Li et al. [7] proposed a feature extraction method, Singular Value Decomposition (SVD), to reduce the different length of data to feature vectors, and then apply SVM on the feature vectors to classify MTS data. Weng et al. [6] project original MTS into PCA subspace by throwing away the smallest principal components firstly, and then MTS samples in the PCA subspace are projected into a lower-dimensional space by using supervised Locality Preserving Projection (LPP). However, the above existing extraction method may lose the dependency relationship information among different univariate time series.

Hence, to extract the features among the same and different variables can retain more significant information for further classification. For proving the efficiency of the proposed algorithm, we compare the proposed algorithm with classical feature extraction methods, PCA. We focus on discovering the relationship within the same variable (intra-temporal patterns) and between different variables (inter-temporal patterns) at different time points, and combine the degree value of all patterns as feature vector.

After extracting features from original MTS data, a classifier, such as Support Vector Machine (SVM) [18] and Artificial Neural Networks (ANN) [19] can be applied to classify output feature vectors. The SVM transforms original input data into a higher dimensional space using a nonlinear mapping and then searches for a linear separating hyper-plane [20]. Considering the input vector is a $m \times n$ matrix, m is the number of MTS, n is the number of features, in order to classify MTS, it applies a kernel function to the original input data. ANN is composed of interconnecting artificial neurons that can compute values from inputs. Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) are two of the most widely used neural network architecture in literature for classification or regression problems. RBF is a local type of learning which is responsive only to a limited section of input space, and MLP is a distributed approach [21].

3 Methodology

With the above requirements in mind, we developed a feature extraction method to catch the dependency relationship between different variables, and classify MTS data based on feature vectors. Combining the proposed feature extraction method and classical classifier, the proposed strategy is called as Multivariate Time Series Classifier (MTSC).

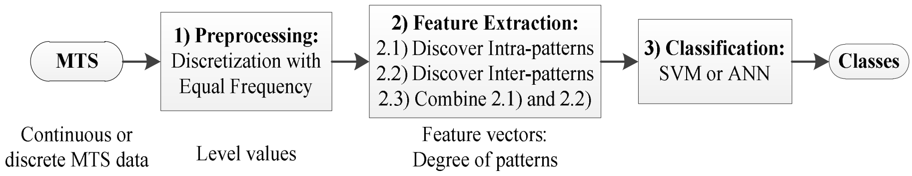

Fig. 1. The process of the proposed algorithm

The task of the proposed algorithm is to uncover the temporal relationship or interrelationships between different variables. These temporal relationship or interrelationships constituent what we call intra- and inter- temporal patters respectively. The proposed algorithm performs its tasks in several steps: 1) discretize MTS data into level value using Equal Frequency; 2) extract features from MTS data which contains three sub-steps: 2.1) discover intra-temporal patterns within each component time series in each MTS; 2.2) discover inter-temporal patterns between different component time series within each MTS; 2.3) combine the value of degree of patterns discovering in 2.1) and 2.2); 3) classify MTSs based on feature vectors outputting in 2.3) using SVM with RBF-kernel function or MLP ANN. The structure of our proposed algorithm is shown in Figure 1. The definitions and notations are given in the Section 3.1, and then Section 3.2 specifies the proposed algorithm in detail.

3.1 The Problem Definition and Notations

Let S represent a set of MTS with the following characteristics:

1. **S** consists of *m* MTS represented as **S**= $\{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^m\}$.
2. For *i*th MTS, $\mathbf{S}^i, i = \{1, \ldots, m\}$ consists of *n* components univariate time series (variables) that can be represented as S_j^i, *j*=*1*, ….,*n*, so that $\mathbf{S}^i = \{S_1^i, S_2^i, \ldots, S_n^i\}$, and S_j^i represent the *j*th univariate time series in *i*th MTS.
3. The values in the vector S_j^i, j =1, …., n, takes on the time instants of $1, \ldots, p$ can be represented as $S_j^i = (s_{j,1}^i,\ s_{j,2}^i, \ldots, s_{j,t-\tau}^i,\ \ldots, s_{j,t}^i,\ \ldots, s_{j,p}^i), 1 \leq \tau \leq p, \tau \in \mathbb{Z}^+$.
4. The domains of the value of variable, $V_j, j = 1, \ldots, n,$, are represented as $(V_j) = [L_{V_j}, U_{V_j}]$, $j = 1, \ldots, n,$ L_{V_j} represent the lower bound and U_{V_j}represent the upper bound of the values that V_j can take on.

Given a set of MTS, ***S***, with characteristics as described above, they are pre-classified into k classes, where, $C_1 = \{\mathbf{S}_{C_1}^{(1)}, \mathbf{S}_{C_1}^{(2)}, \ldots, \mathbf{S}_{C_1}^{(n_1)}\}$, $C_2 = \{\mathbf{S}_{C_2}^{(1)}, \mathbf{S}_{C_2}^{(2)}, \ldots, \mathbf{S}_{C_2}^{(n_2)}\}$, …, $C_k = \left\{\mathbf{S}_{C_k}^{(1)}, \mathbf{S}_{C_k}^{(2)}, \ldots, \mathbf{S}_{C_k}^{(n_k)}\right\}$. To classify these MTS, we need to find their class labels C_1 to C_k.

3.2 The Proposed Algorithm

3.2.1 Discretization

Before discovering patterns from MTS data, the preprocessing is needed. For reducing and simplifying the original data, numerous values of a continuous variable is always be replaced by a small number of interval labels, which leads to a concise, easy-to-use, knowledge-level representation of mining results [22]. Data discretization is a frequently used technique to partition the value space of a continuous variable into a finite number of intervals and assigning a nominal value of each of them [23, 24]. *Equal Width* and *Equal Frequency* are two simplest discretization methods. However, if uncharacteristic extreme values (outliers) exist in the data set, *Equal Width* can hardly handle this situation. Hence, in our case, we transform original numerical $s_{j,t}^{i}$ which represents the *i*th MTS of *j*th variable at *t* time point into $D_{j,t}^{i}$ using *Equal Frequency* [17] algorithm. And we set the number of bins is three, that is to say, the original numerical data is transformed into three levels (such as {high, medium, low}).

3.2.2 Discover Intra-/Inter-Temporal Patterns

After preprocess, discovering intra- and inter-temporal patterns among one MTS is applied. As one MTS contains several variables (univariate time series) which are temporally interrelated, the value that a particular variable take on at any time instant can be related to the variable's previous values or to the previous values of other variables. These interrelationships constitute, respectively, the intra- (within one univariate time series) and inter- (between different univariate time series) temporal patterns in one MTS. We use the proposed *significant discrepancy* measures to evaluate these patterns, and use a set of value of these measures to represent MTS. This process can be treated as feature extraction process from MTS, and the value of degree that describes dependency patterns can be treated as Feature Vectors. Two main sub-steps are specified as following.

Step One: Discovering Intra-Temporal Patterns.

Given a value, say, $S_{j,t}^{i}$, which represents the value on the time point *t* in *j*th time series in *i*th MTS, the magnitude of the difference between the conditional probability $\Pr(s_{j,t}^{i} | s_{j,t-\tau}^{i})$, $1 \le \tau \le \tau_{max} < t$, and the a priori probability $Pr(s_{j,t}^{i})$ can be estimated as follows:

$$Pr\left(S_{j,t}^{i} \middle| S_{j,t-\tau}^{i}\right) = \frac{freq\left(S_{j,t}^{i},\ S_{j,t-\tau}^{i}\right)}{freq\left(S_{j,t-\tau}^{i}\right)} \tag{1}$$

where $freq(S_{j,t}^{i},\ S_{j,t-\tau}^{i})$ is the number of instants of the value, $S_{j,t}^{i}$, being preceded at τ instants ahead by $S_{j,t-\tau}^{i}$ and $freq(S_{j,t-\tau}^{i})$ is the number of instants of the value $S_{j,t-\tau}^{i}$ that appear in S_{j}^{i} in $\boldsymbol{S^{i}}$. As for $Pr(S_{j,t}^{i})$, it can be estimated as follows:

$$P\,r\left(S_{j,t}^{i}\right) = \frac{freq\left(S_{j,t}^{i}\right)}{p} \ (p \text{ is the total number of time points}) \tag{2}$$

Given these probability estimations, the magnitude of the difference between the conditional probability $Pr\left(S_{j,t}^{i} \middle| S_{j,t-\tau}^{i}\right)$ and the apriori probability $Pr(S_{j,t}^{i})$ can be defined simply as: $Pr\left(S_{j,t}^{i} \middle| S_{j,t-\tau}^{i}\right) - Pr(S_{j,t}^{i})$. And the differences in the two probabilities are normalized using (3) [23].

$$d_{j,\tau}^{i} = \frac{p*\left(Pr\left(s_{j,t}^{i}\middle|s_{j,t-\tau}^{i}\right)-Pr\left(s_{j,t}^{i}\right)\right)}{\sqrt{p_{ij}\,Pr\left(s_{j,t}^{i}\right)Pr\left(s_{j,t-\tau}^{i}\right)\left(1-Pr\left(s_{j,t}^{i}\right)\right)\left(1-Pr\left(s_{j,t-\tau}^{i}\right)\right)}} \quad (3)$$

The significance of the temporal relationship depends on the magnitude of normalized difference, $d_{j,\tau}^{i}$ [23] which can be either ≥ 0 or ≤ 0. If $|d_{j,\tau}^{i}|$ is large, the presence or absence of $S_{j,t-\tau}^{i}$ would likely imply that at τ time instants later, the component univariate time series will or will not take on the value $S_{j,t}^{i}$, respectively. The magnitude of the normalized differences in conditional captures the strength of the temporal relationships [23] and they constitute the intra-temporal patterns for S_{j}^{i}. Hence, $d_{j,\tau}^{i}$ can be defined as the *significant discrepancy measure* to evaluate the relationship of two values within the same variable.

Step Two: Discovering Inter-Channels Temporal Patterns.
Similar to Step One, the inter-temporal patterns can be also discovered. The inter-temporal patterns are defined between two different variables say, S_{j}^{i}, and $S_{j'}^{i}$, both within $\mathbf{S}^{i}$, to consist of a set of temporal relationships or interrelationships detected between a value of S_{j}^{i} at a particular time instant, t, and those that it takes on at an earlier time instant, τ, 1≤$\tau \le \tau_{max} < t$. These temporal relationships or interrelationships can be determined as follows.

Algorithm 1. The Proposed Feature Extraction Method

```
Input: S = {S^1, S^2, ..., S^m}(m is the number of MTS)
       S^i = {S^i_1, S^i_2, ..., S^i_n} (n is the number of variable)
       τ(the time window 1•τ< t)
Output: finalResult(feature vector for one MTS)

for each MTS S^i in S
    Discretization using Equal Frequency for S^i
    for each univariate time series S^i_j in S^i
        Calculated^i_{j,τ}
        Result + = d^i_{j,τ} /*degree vector of intra-patterns*/
    end
    for each channels s^i_{j',t} ≠ s^i_{j,t}  in MTS
        Calculate d^i_{jj',τ}
        Result + = d^i_{jj',τ} /*degree vector of inter-patterns*/
    end
    finalResult += Result
end /*finalResult is a set of degrees of all patterns*/
```

Given a value, say, $S_{j,t}^{i}$, and another value, say, $S_{j',t-\tau}^{i}$, The magnitude of the differences can be determined by first estimating the two probabilities $Pr(S_{j,t}^{i}|S_{j',t-\tau}^{i})$ and

$Pr(S^i_{j,t})$, $1 \le \tau \le \tau_{max} < t$, based on the data and calculated using (1) and (2). And similarly, the differences in the two probabilities, $d^i_{jj',\tau}$, are normalized using same equation of (3). As the *significant discrepancy measure* can evaluate the relationship between two variables S^i_j and $S^i_{j'}$ in MTS, $d^i_{jj',\tau}$ is used to represent degree of relationship between S^i_j and $S^i_{j'}$. The pseudo code of the whole process of the feature extraction method is shown in Algorithm 1.

3.2.3 Classification Using SVM or ANN

Once all the temporal patterns are discovered, for MTS, we get a set of intra-temporal patterns measure $d^i_{j,\tau}$, (i = 1, 2, …, m, j = 1, 2, …, n, and τ = 1,2,… p), and a set of inter-temporal patterns measure $d^i_{jj',\tau}$ (i = 1, 2, …, m; jj' = 1, 2, …, n, j ≠j' and τ = 1,2,… p_{ij}), associated with it. These intra- and inter- temporal patterns forms feature vectors: $\{d^i_{1,\tau}, \dots d^i_{j,\tau}, d^i_{1\,2,\tau}, d^i_{1\,3,\tau}, \dots, d^i_{jj',\tau}, \dots d^i_{m-1\,m,\tau}\}$, where n is the total number of variables, $1 \le \tau \le \tau_{max}$ and i = 1, 2, …, m. In addition, each $d^i_{jj',\tau}$ may contain a set of value for when $t^i_{j,t}$ equal to different discrete value. And then the classifier, SVM with RBF or MPL ANN, can be used to classify these feature vectors. The detail of classification algorithm has been specified in Section 2.

4 Experimental Result

To evaluate the performance of MTSC, a number of different experiments were carried out using both synthetic and real world data. In order to prove the proposed algorithm can handle both categorical and numerical data, the synthetic data set is a discrete data set which was generated by embedding different temporal patterns in the data to see if MTSC can discover them for classification. The other two real world data sets included EMG: Physical Action data set and ECG data set. For the purpose of performance evaluation, the test samples are compared with the known class labels using two performance measures: F-measure and classification accuracy. In the following, we describe the data set in section 4.1; and then the experimental results using the proposed algorithm are provided in section 4.2; finally, a comparison result between the proposed algorithm and traditional methods are given in section 4.3.

4.1 Data Set Description

4.1.1 Discrete Data Set: Synthetic Data Set

The synthetic data set is a discrete dataset that consists of 45 MTS generated randomly. Each of these MTS consists, in turn, of 5 variables, $v_1, \dots v_5$, and $v_i = (A_i, B_i, C_i)$, i= 1,2,…5. There are total of 500 data values are generated for each variable to make all 5 univariate time series consists of 500 time points. Hence the synthetic data set is a 45 (MTS) × 5 (variables) × 500 (time points) dimensions data set.

Table 1. The Inserted Rules for Synthetic Dataset

Classes	Rules
Class 1	1. v_1 and v_5 are totally random. 2. v_4 takes on "A_4" at every interval of 2 time units and then $V_{2,}$ at next time point, is generated to be "B_2", others are "A_2" or "C_2" randomly. 3. If v_4 not to be "B_4", $v_3$takes on "A_3" at the next time instant 50% or "C_3" at the next time instant 50% of the time.
Class 2	1. v_2 and v_5 are totally random. 2. If $v_2$in A_2 then v_1 in B_1, others are random 3. If $v_2$in B_2,$v_3$takes values in C_3, others are random 4. If $v_2$in C_2, then $v_4$in A_4 others are random
Class 3	1. v_3 and v_5 are totally random. 2. If v_3 in A_3 then v_2 in C_2, others are random 3. v_1 takes on A_1at every interval of 3 time units and then at the next time points, v_4 is generated to be within A_4, others are random.

The different rules that belong to three classes are shown in Table 1. For example, in Class 1, we generate v_1 to v_5 randomly that takes on the value of $v_i = (A_i, B_i, C_i)$, $i = 1,2,\ldots 5$ firstly. The Rule 2 in Class 1 means, we insert patterns into v_4 to make it take on "A_4" at every interval of 2 time units, and 1hen Variable 4 takes on value of "A_4", v_2 is generated to be "B_2" at next time point. Similarly, Rule 3 means, if value of v_4 is not equal to "B_4", the value of v_3 is generated as "A_3" at the next time instant 50% or "C_3" at the next time instant 50% of the time. Hence, for Class 1 v_1 and v_5 are noise data since they are totally random.

4.1.2 Numerical Real-World Data Set 1: EMG Physical Action Dataset

Physical Action Dataset (EMG) which is a benchmark data set from UCI repository [12]. The subjects are three male and one female (age 25 to 30), who have experienced aggression in scenarios such as physical fighting, took part in the experiment. Throughout 20 individual experiments, each subject had to perform ten normal and ten aggressive activities. 8 skin-surface electrodes correspond 8 input time series, muscle channel 1 to muscle channel 8, placed on the upper arms and upper legs to detect the position of actions of muscle. Each time series contains ~10000 samples (time points) with sampling frequency of 200Hz. Hence the EMG data is a 80 (MTS) × 8 (channels) × ~10000 (time points) dimensions data set.

4.1.3 Numerical Real-World Data Set 2: ECG Dataset

The other real-world dataset is ECG data set [13]. This data set comprises a collection of time-series data sets where each file contains the sequence of measurements recorded with two electrodes by one electrode during one heartbeat. Each heartbeat has an assigned classification of normal or abnormal. It contains 200 data sets where 133 were identified as normal and 67 were identified as abnormal. Hence, in this dataset, there are 200 MTS in total, each MTS contains 2 univariate time series, and 39-152 records are collected for each univariate time series with sampling 200HZ. The dimension of this data set is 200 (MTS) × 2 (electrodes) × ~150 (time points).

4.2 Experiment Process and Evaluation

The dataset can be processed as described in the methodology section. For the purpose of performance evaluation, 80% data are selected randomly as training data and the rest 20% as testing data. Two evaluation measures are used F-measure [14] and Classification Accuracy (CA).

The traditional F-measure or balanced F-score (F1 score) is the harmonic mean of precision and recall [14] that can be defined as below. Let C_p, p $\in\{1, \ldots,k\}$, be a class after classification using MTSC and C_q, q $\in\{1, \ldots,k\}$, be the class that is previous known, (k is the number of classes), so the F-measure is defined for C_p and $C_q$as equal (4) shows [14].

$$F(C_p, C_q) = \frac{2\times Recall\ (Cp,Cq) \times Precision(Cp,Cq)}{Recall(Cp,Cq)+Precision(Cp,Cq)},$$

$$\text{Recall}(C_p, C_q) = \frac{freq(C_p,C_q)}{freq(C_p)}, \text{Precision}(C_p, C_q) = \frac{freq(C_p,C_q)}{freq(C_q)} \tag{4}$$

$freq(C_p, C_q)$ represents the number of MTS with the cluster label C_p in the discovered cluster, C_q, $freq(C_p)$ is the number of records with class label C_p and $freq(C_q)$, is the number of records in the predicted class label C_q. Given this definition, F-measure therefore takes on values in the interval [0, 1]. The large its value is, the better the classification quality it reflects. In addition to the F-measure, due to the classes are pre-known, classification accuracy (CA) can be used evaluate how accurate MTSC is and the definition of CA is shown in equation (5)

$$\text{CA} = \frac{Total\ Number\ of\ mvts\ in\ the\ correct\ classes}{Total\ number\ of\ MTS} \tag{5}$$

4.3 Experimental Result for the Proposed Algorithm

In synthetic data set, we only insert rules for one time intervals. So, we consider the intra-temporal patterns within one variable and inter-temporal patterns among different variables only between the previous time point and next one time point (τ=1). We use the proposed feature extraction to process the MTS data firstly and then use SVM or ANN to classify them. Table 2 summarizes the classification result for synthetic dataset using different classifier. The result table shows the value of Mean Acc. (the average of classification accuracy), Highest Acc. (the highest classification accuracy) and F-measure (the average value of F-measure). When MTSC is applied for ANN classifier, the highest accuracy is 100% and the average of classification accuracy is 98.6% with F-measure is 0.99 for synthetic data set and the average of accuracy is more than 75% with F-measure is 0.72 for SVM classifier. Hence, in this experiment, ANN can get the higher value for both classification accuracy and F-measure than SVM.

Table 2. The result of synthetic dataset using the proposed algorithm

Evaluation	Mean Acc. (Highest Acc.)	F-measure
SVM	76.67% (87.5%)	0.72
ANN	**98.6% (100%)**	**0.99**

In addition, for real-world data sets, considering the dependency or relationship within one variable or between different variables may not only in one time interval, we set τ=1 to 5 for intra-/inter-temporal patterns. Table 3 and Table 4 show the result for the two real data sets using MTSC with SVM or ANN classification algorithm for different time intervals.

In summary, ANN can get higher classification accuracy and F-measure than SVM for the most of classification result. In EMG dataset, when τ=5, the proposed algorithm can achieve highest average classification accuracy of 90.6% with the average F-measure being 0.89 for SVM classifier, and accuracy of 91.78% with the average F-measure being 0.92 for ANN classifier. Hence, the intra- and inter- relationship between variables are the most significant for classification in 5 time intervals. In ECG data set, when τ=4, the proposed algorithm can achieve a slightly higher average classification accuracy of 77.37% with the average F-measure being 0.72 for SVM classifier, and accuracy of 75.89% with the average F-measure being 0.7 for ANN classifier. Hence, when setting time interval equal to 4, the relationship within the same variable and between different variables can distinguish different samples best.

Table 3. The result of EMG dataset using the proposed algorithm with different time intervals

	Evaluation	τ=1	τ=2	τ=3	τ=4	**τ=5**
SVM	**Mean Acc.**	73.92%	74.94%	74.97%	81.94%	**90.60%**
	(Highest Acc.)	(92.31%)	(76.92%)	(84.62%)	(100%)	**(100%)**
	F-measure	0.79	0.74	0.69	0.89	**0.89**
ANN	**Mean Acc.**	89.80%	88.41%	85.01%	83.00%	**91.78%**
	(Highest Acc.)	(100%)	(100%)	(87.50%)	(92.31%)	(**93.75%**)
	F-measure	0.85	0.88	0.81	0.83	**0.92**

Table 4. Theresult of ECG dataset using the proposed algorithm with different time intervals

	Evaluation	τ=1	τ=2	τ=3	τ=4	**τ=5**
SVM	**Mean Acc.**	75.45%	71.32%	70.30%	**77.37%**	73.75%
	(Highest Acc.)	(80.56%)	(73.17%)	(73.17%)	(**81.40%**)	(80.65%)
	F-measure	0.73	0.71	0.66	**0.72**	0.72
ANN	**Mean Acc.**	78.21%	68.12%	69.08%	**75.89%**	70.09%
	(Highest Acc.)	(88.89%)	(70.30%)	(73.17%)	(**80.00%**)	(72.97%)
	F-measure	0.783	0.60	0.57	**0.7**	0.70

4.4 Comparison Experimental Result

For performance benchmarking, we compare the proposed MTSC algorithm with 1) classification using SVM or ANN classification without feature extraction method (No FE), 2) SVM or ANN classification with PCA for MTS data. The Principle Component Analysis (PCA) method can reduce high dimensions of MTS data and transform original data into a feature vector. However, PCA can only applied into MTS with equal length. So we cut unequal length time series into minimum length. And then we use the same classifiers, SVM with RBF kernel function and MPL ANN to classify feature vectors. Table 5 and Table 6 summarize the comparison result in classification accuracy and F-measure.

Table 5. Comparison of the average of classification accuracy between different algorithms

Data Sets	SVM			ANN		
	No FE	PCA	MTSC	No FE	PCA	MTSC
Synthetic	50.46%	67.2%	**76.67%**	68%	76.45%	**98.6%**
EMG	85.71%	86.73%	**90.6%**	74.73%	78.64%	**91.78%**
ECG	75%	75.05%	**76.52%**	75%	74.96%	**77.37%**

Table 6. Comparison of the average of F-measure between different algorithms

Data Sets	SVM			ANN		
	No FE	PCA	MTSC	No FE	PCA	MTSC
Synthetic	0.46	0.60	**0.72**	0.69	0.77	**0.99**
EMG	0.71	0.85	**0.89**	0.70	0.78	**0.92**
ECG	0.707	0.67	**0.75**	0.727	0.66	**0.7**

For synthetic dataset, when no feature extraction method is applied, the classification accuracy is 50.46% and 68% with F-measure of 0.46 and 0.69 for SVM classifier and ANN classifier respectively. PCA+SVM can only achieve an accuracy of 67.2% with average F-measure of 0.41, and PCA+ANN can achieve a higher accuracy of 76.45% with F-measure of 0.67. When comparing with the proposed algorithm, the performance of traditional algorithm is worse. Similarly, for two real-world datasets, MTSC can get higher both classification accuracy and F-measure than two other traditional methods. We can conclude from the result table, the value of the average of classification accuracy using MTSC is higher than the result using PCA with SVM or ANN for all data sets.

Besides the classification performance comparison, the complexity analysis is another significant target. Suppose that we have m MTS with n variables and t time points for each univariate time series (n<<t). If there is no feature extraction method is used, the classifier has to process for mnt dimensions data. When some feature extraction method, such as PCA, is used, the run-time complexity of the PCA is $mO(t^2)$, and the run-time of complexity of our proposed algorithm could be $mO(t * n^2)$. Generally, the MTS contains very high dimensions of time points with less variables, so in this case, the advantage of MTSC just need to count once for the value of each time points.

5 Conclusion

This paper has presented a classification strategy that combining the proposed feature extraction method and classifier for classifying MTS data. Unlike many existing methods, it is able to handle multivariate time series that may consist of either continuous or discrete data or both. As the proposed feature extraction method can perform its tasks without requiring any special assumption about data models, it is generic and application-independent. Given that MTSC performs classification for MTS data by discovering patterns within each time series independently of the others, it can also handle time series of different length. For performance evaluation, FEMTS was tested with both artificial and real data. The results show that it can be a promising algorithm for multivariate time series classification. The future work could be investigated into the possibility of improving the current work in three aspects: 1) after discovering intra- and inter- patterns in MTS, the dimensions of each MTS need to be reduced using attribute selection method to make algorithm speed up; 2) for improving the classification accuracy, fuzzy data/classes can be investigated so that MTS which belongs to overlapping classes can be discovered; 3) the more verification to verify how the algorithm can be more generally applicable.

References

1. Spiegel, S., Gaebler, J., Lommatzsch, A.: Pattern recognition and classification for multivariate time series. In: 5th International Workshop on Knowledge Discovery from Sensor Data, pp. 34–42. ACM (2010)
2. Xing, Z., Pei, J., Keogh, E.: A Brief Survey On Sequence Classification. ACM SIGKDD Explorations Newsletter, 40–48 (2010)
3. Ho, P.G.P.: Multivariate Time Series Support Vector Machine or Multispectral Remote Sensing Image Classification. In: Geoscience and Remote Sensing, ch.16 (2009)
4. Yang, K., Shahabi, C.: A PCA-Based similarity measure for multivariate time series. In: 2nd ACM International Workshop on Multimedia Databases, pp. 65–74. ACM Press (2004)
5. Weng, X.: Classification of multivariate time series using supervised neighborhood preserving embedding. In: 25th Control and Decision Conference (CCDC), Guiyang, pp. 957–961 (2013)
6. Weng, X., Shen, J.: Classification of Multivariate Time Series Using Locality preserving projection. Knowledge-based System. **21**(7), 581–587 (2008)
7. Li, C., Khan, L., Prabhakaran, B.: Feature selection for classification of variable length multi-attribute motions. In: Petrushin, V.A., Khan, L. (eds.) Multimedia Data Mining and Knowledge Discovery, ch. 7. Springer (2007)
8. Cai, D., He, X., Han, J.: Orthogonal Laplacianfaces for Face Recognition. IEEE Transactions on Image Processing. **15**(11), 3608–3614 (2006)
9. Zhang, H., Ho, T., Huang, W.: Blind feature extraction for time-series classification using haar wavelet Transform. In: Wang, J., Liao, X., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3497, pp. 605–610. Springer, Heidelberg (2005)
10. Rodrgueza, J.J., Alonsob, C.J., Maestro, J.A.: Support Vector Machines Of Interval-Based Features For Time Series Classification. Knowledge-Based Systems. **18**, 171–178 (2005)

11. Hayashi, A., Mizuhara, Y., Suematsu, N.: Embedding time series data for classification. In: Perner, P., Imiya, A. (eds.) MLDM 2005. LNCS (LNAI), vol. 3587, pp. 356–365. Springer, Heidelberg (2005)
12. Murph P.M., Aha, E.W.: UCI Repository of Machine Learning Databases.(1999). http://archive.ics.uci.edu/ml/datasets/EMG+Physical+Action+Data+Set
13. Olszewski, R.T.: ECG Database. http://www.cs.cmu.edu/~bobski/
14. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 16–22. ACM Press (1999)
15. Ding, S., Zhu, H., Jia, W., Su, C.: A Survey On Feature Extraction for Pattern Recognition. Artificial Intelligence Review **37**(3), 169–180 (2012)
16. Zhu, D., Wu, C., Qin, W.: Multivariate statistic analysis and software SAS. Southeast University Press, Nanjing (1999)
17. Wong, A.K.C., Wang, C.C.: DECA–A Discrete-Valued Ensemble Clustering Algorithm. IEEE Trans on Pattern Analysis and Machine Intelligence **PAMI-1**(4), 342–349 (1979)
18. Cortes, C.: VapnikV.: Support-Vector Networks. Machine Learning **20**(3), 273–297 (1995)
19. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice Hall, Upper Saddle River (1998)
20. Han, J., Kamber M.: Data Mining: Concepts and Techniques, 2nd edn. Elsevier Inc. (2007)
21. Cohen, S., Intrator, N.: A study of ensemble of hybrid networks with strong regularization. In: 4th International Conference on Multiple Classifier Systems, pp. 227–235 (2003)
22. Liao, T.W.: Clustering of Time Series Data - A Survey. Pattern Recognition **38**, 1857–1874 (2005)
23. Ma, P.C.H., Chan, K.C.C., Chiu, D.K.Y.: Clustering and re-clustering for pattern discovery in gene expression data. Journal of Bioinformatics and Computational Biology **3**(2), 281–201 (2005)
24. Dimitrova, E.S., McGee, J.J., Laubenbacher, R.C.: Discretization of Time Course Data. Journal of Computational Biology. **17**(6), 853–868 (2010)