Granger Causality for Multivariate Time Series Classification

Dandan Yang, $*$ Huanhuan Chen , Yinlong Song, Zhichen Gong

Abstract—**Multivariate time series, which is a set of ordered observations for multiple variables, is pervasively generated in air condition, traffic, entertainment, etc. Echo State Network has shown promising performance for processing multivariate time series due to its ability to approximate sequential dynamics. However, the intrinsic relationships among time series have not been generally analyzed in the previous Echo State Network based methods. These relationships may help reveal the intrinsic characteristics of multivariate time series and benefit the classification performance. In this paper, we propose a novel method for approximating the sequential dynamics and learning the relationship among multiple variables explicitly in a unified framework. We learn a model for each multivariate time series and evaluate the distance of the original multivariate time series by the distance of their models. The relationship among variables in a multivariate time series is learnt according to Granger causality. We further constrain the sparsity of the learnt time series models to find the Focal series which help explain all the series. Experiments on benchmark datasets demonstrate superior classification performance of the proposed method.**

Index Terms—**Granger causality, Echo state network, multivariate time series, focal series**

I. Introduction

A multivariate time series (MTS) contains a set of ordered observations at discrete time for multiple variables. MTS can be viewed as a collection of multiple univariate time series. MTS is ubiquitously generated in traffic prediction [1], air condition [2] and economy [3], etc. Although a lot of efforts have been made for processing MTS data, the implicit relationships among sub-series[1] in a MTS and possibly varying length still raise significant challenges to learn from MTS.

Echo State Network (ESN) is a variant of Recurrent Neural Network [4]. Due to ESN's design nature, the process of training ESN is more simple. With its ability to capture the order information of time series, ESN is widely employed [5]–[7]. Chen et al. [8] proposed model-based kernels for time series classification. In their approach, each time series is represented by the readout mapping of an ESN [8]. Then the learnt time series models are employed for classification. Wang et al. [9] employed ESN with adaptive differential evolution algorithm. The key idea is to transform MTS samples into different state clouds and use the state clouds as features. However, the problem for providing explainable insights for modeling MTS has been generally ignored. Basic ESN can not explicitly reveal the intrinsic relationships among sub-series in a MTS. Moreover, the fully connected readout layer may lead to overfitting.

Granger Causality (G-causality) refers to a predictive relationship among time series. Generally speaking, given two time series $\mathbf{X}$ and $\mathbf{Y}$, if it would be more favorable in predicting $\mathbf{X}$ with the incorporation of $\mathbf{Y}$'s historical information than using $\mathbf{X}$'s own historical information, that is, $\mathbf{Y}$ G-causes $\mathbf{X}$ [10]. G-causality is useful in finding the relationships for multiple variables, such as human action classification [11], anomaly detection [12] and stock analysis [13]. Previously Vector Autoregression (VAR) model is employed to reveal the G-causality between different time series [14]. The goal is to minimize the prediction error. However, in a VAR model, the historical information is limited to the size of a sliding window. In this case, the nonlinear information and dynamics of time series are largely ignored.

In this paper, we intend to take advantage of nonlinear approximating ability of ESN and G-causality in learning models for MTS. The ESN is employed as a general-purpose nonlinear temporal filter. For each MTS, we input each sub-series to an ESN iteratively. Then we learn the readout mapping of ESN by G-causality to reveal the nonlinear relationship among different sub-series. We further designed a sparsity-enhanced method for finding the focal series (FS) which are able to G-cause all the other sub-series. We employ ESN as the model for the input MTS. The distance of original MTS is evaluated by the distance of their models. The general architecture of our method is illustrated in Fig. 1.

There are three advantages of our approach:

1) The proposed method is able to classify multivariate time series of different length;
2) A principled distance between the time series models can be formulated analytically under a certain assumption;
3) The sparsity of the proposed method helps improve the generalization ability for classification. In addition, the focal series can be found, which endows more explanation ability for our method.

The rest of the paper is organized as follows. Section II introduces related works and background on time series classification. Section III details the proposed methods. In Section IV, we present our experiments and demonstrates the effectiveness of our proposed method. Finally, our work is concluded in Section V.

II. Background and Related Work

In this section, we first review some related works on MTS classification. Then we introduce background of the ESN and G-causality.

[1] The time series for a specific variable in a MTS.

978-1-5386-3120-1/17 $31.00 © 2017 IEEE
DOI 10.1109/ICBK.2017.36

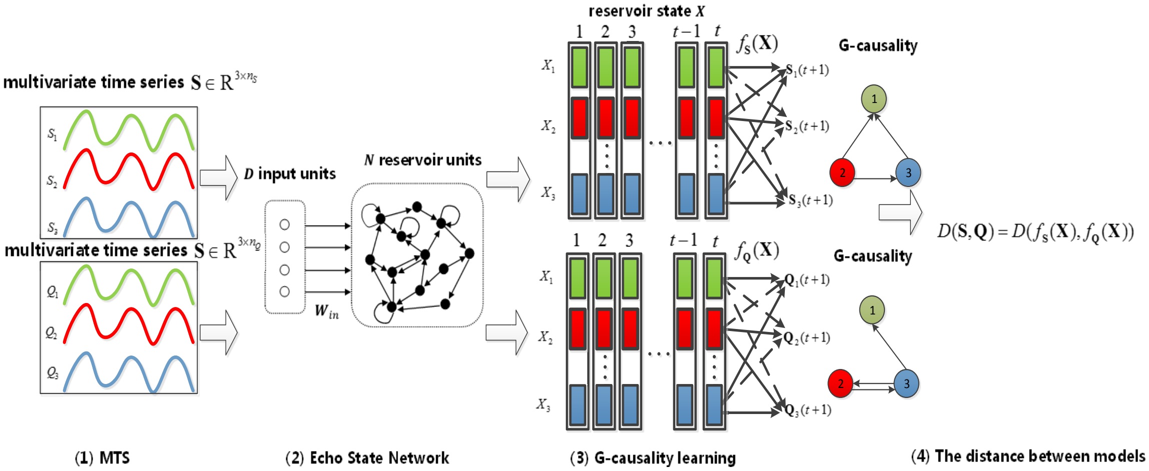


Fig. 1. Architecture of our proposed method. The whole process of our method contains four steps. (1) Each sub-series of a MTS is fed to ESN respectively. (2) Approximate the sequential dynamic of each sub-series by ESN and collect the corresponding reservoir state. (3) G-causality is applied to the readout mapping of ESN. We learn the readout mapping coefficient for focal series and we show the G-causality by G-causality graph. The dashed arrow from X_i to s_j represents that the learnt corresponding coefficient is zero, that is, there is no G-causality between two series. The solid arrow from X_i to s_j indicates that the learnt corresponding coefficient is nonzero, that is, there is G-causality between two series. (4) The distance between two models is calculated under a certain assumption.

A MTS dataset contains a set of sample-label pairs $\{\mathbf{S}^i, C^i\}_{i=1}^{N_{tr}}$. $\mathbf{S}^i = [\mathbf{s}_1^i; \mathbf{s}_2^i; \cdots; \mathbf{s}_m^i] \in R^{m*n}$ contains n sequential observations of m variables. Hence, each row in $\mathbf{S}_i$ is a sub-series. $\mathbf{s}^i \in R^n$ is the time series for the i_{th} variable, which has n samples. $C \in \{1, 2, \cdots, C_{label}\}$ is the label for the MTS. Roughly speaking, MTS classification methods can be categorized into three branches.

The first one transforms the original series to obtain the feature vectors. K-gram employs a short segment of k consecutive symbols as a feature [15]. In this way, traditional classifiers can utilize the pre-treated data for convenience. In [16], the authors used Gamma-test to search for homologies in nucleotide sequences by selecting a small sized informative features from the k-grams. In [17], the authors proposed time series shapelets that are able to represent a class maximally. However, they can only represent local properties of a sequence. These methods mainly require a complex preprocess to select features according to certain criteria [18].

The second category is based on the sequence distance. In this case, the definition of distance function is the critical for MTS classification. Dynamic time warping (DTW), as illustrated in Fig. 2, is a method that aligns two time series by warping along time axis such that their accumulative distance is minimized [19]. With nearest neighbor (1-nn) classifier, DTW has proved to be strong solution. DTW is more precise and elastic than Euclidean distance [20]. However, The time complexity of DTW is quadratic, which limits the application to long time series. Kernel methods also has widely applied to time series classification, such as Fisher kernel [21], Autoregressive kernel [22] and Model-based kernel [8]. However, more computation cost is required for building a desired kernel.

The third category of methods approximate time series with a generative model. Jebara et al. [23] demonstrated probability product kernels on hidden Markov model (HMM). It maps each time series to a HMM and makes the definition for the inner product between the corresponding HMM distributions. Srivastava et al. [24] used HMM to classify biological sequences. The key idea of that approach is to empoy HMM by optimising the discrimination threshold and modifying emission probabilities to represent the training data. HMM based methods are not sensitive to noise and the length of time series, but the Markov constraint limits the approximation ability to nonlinear time series.

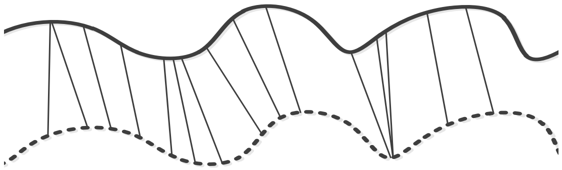


Fig. 2. Architecture of Dynamic time warping. The dashed line and the solid line represent two different time series. The thin line is the ”warping path” between two instances.

Granger et al. [10] proposed *Copular-Granger* to read the non-linearity of time series in high dimensions. Arnold et.al [25] applied G-causality to lasso regression and other linear regression models. The similarity measurement of the G-causality graph is provided. However they can not guarantee the sparsity in the G-causality graphs, which unable to insure that the regressions will not be overfitted .

III. Granger Causality Based Multivariate Time Series Model Learning

A. G-causality graph

Time series prediction is to employ the historical information to predict the future behavior of the time series.

Formally speaking, for two sub-series time series $\mathbf{X}$ and $\mathbf{Y}$, if the combination of the historical information of both $\mathbf{X}$

and $\mathbf{Y}$ leads to better prediction performance than that solely based on $\mathbf{X}$'s own historical information , it is defined that time series $\mathbf{Y}$ Granger causes (G-causes) time series $\mathbf{X}$. And the causal relationship between $\mathbf{X}$ and $\mathbf{Y}$ is called Granger causality (G-causality). A directed graph $G = \{\boldsymbol{V}, \boldsymbol{E}\}$, called G-causality graph is usually employed to visualize the G-causality between pairwise time series. In the G-causality graph, vertices represent the sub-series and the directed edge, directed from v_a to v_b, namely e_{ab}, represents that sub-series a G-causes sub-series b. Fig. 3 illustrates the G-causality graph.

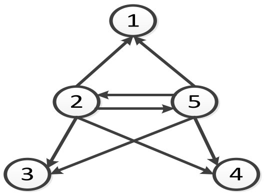


Fig. 3. An example of G-causality graph. The G-causality graph illustrates that time series 2 and time series 5 G-cause all the remaining time series. Thus, the focal series are time series 2 and time series 5 in this situation.

B. Echo State Network

Echo State Network (ESN) [8] is employed as the core model for approximating the sequential dynamics in our method. A typical ESN consists of three components: input layer, reservoir layer and output layer, which contains D, N and L neurons respectively. The connections between the input neurons and the reservoir neurons and the connections among the reservoir neurons are prefixed randomly. The reservoir provides versatile dynamical features for the input time series. The readout mapping maps the reservoir states to the desired outputs and is the only part which needs to be trained. In contrast with classical Recurrent Neural Network, the training process of ESN is more simple. The architecture of ESN is depicted in Fig. 4.

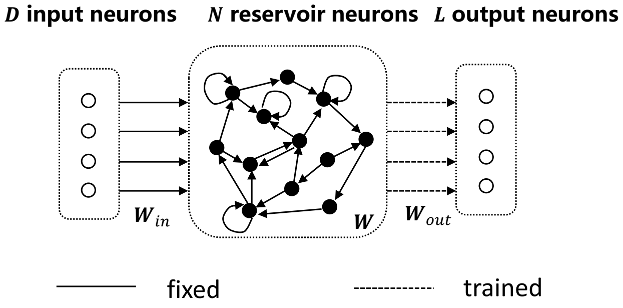


Fig. 4. Illustration of the architecture of basic Echo State Network assumed in this article.

The mechanism of Echo State Networks can be defined by the following:

$$\mathbf{x}(t) = h(\mathbf{W}\mathbf{x}(t-1) + \mathbf{W}_{in}\mathbf{s}(t)) \tag{1}$$
$$\mathbf{y}(t) = \mathbf{W}_{out}\mathbf{x}(t) = f(\mathbf{x}(t)) \tag{2}$$

where $\mathbf{s}(t) \in \mathbb{R}^D$ is the element of the input time series at time t, $\mathbf{x}(t) \in \mathbb{R}^N$ is the reservoir state, and $\mathbf{y}(t) \in \mathbb{R}^D$ is the target output. $\mathbf{W} \in \mathbb{R}^{N\times N}$ is the reservoir weight matrix, $\mathbf{W}_{in} \in \mathbb{R}^{N\times D}$ is the input coefficient, $\mathbf{W}_{out} \in \mathbb{R}^{D\times N}$ is the output coefficient. h is the nonlinear activation function and is set as tanh in the paper. Thus $\mathbf{x} \in [-1,1]^N$. The readout mapping is the only trainable component which can be trained by linear regression efficiently. In this paper, we train the readout mapping to predict the next input, i.e., $\mathbf{y}(t) = \mathbf{s}(t{+}1)$. In principle, the output weights can be estimated with squared error loss function, namely:

$$L(\mathbf{W}_{out}) = \sum_{t=1}^{T-1}(\mathbf{y}(t) - \mathbf{s}(t+1))^2 \tag{3}$$

The goal is to faithfully predict the next input by taking advantages of the approximation ability of ESN.

C. Sparse Focalised-Readout Mapping

In a MTS, a sub-series can be predicted by the its own input history, ignoring the information from other sub-series. In this case, each series has an arrow pointing to itself in the G-causality graph. In a MTS, if there exist several sub-series that are able to G-cause not only themselves but also the remaining series, we denote this subset of sub-series as Focal series (FS). FS provide information about the generation mechanism of the MTS. It has been shown that capturing the intrinsic information of data may help enhance the classification performance [26]. This motivates us to uncover the FS in our MTS classification approach.

In our method, we process each sub-series in a MTS separately with the same prefixed ESN. We train the readout mapping for each sub-series by using the states of all sub-series, as demonstrated in Fig. 5. Thus, each sub-series has its own coefficients for readout. We collect these coefficients for analyzing the G-causality. Following the typical setting of ESN, we denote $\mathbf{W}_{out}$ as the collection of all the readout parameters.

Given a MTS containing D sub-series, for each time point $t \in \{1, 2, \cdots, T{-}1\}$, the i_{th} sub-series $s_i(t) \in R$ is fed to the ESN to get the corresponding reservoir state matrix $\mathbf{x}_i(t) \in R^{N\times 1}$, where N is the reservoir size and $i \in \{1, \cdots, D\}$. All the state matrices $\mathbf{x}_i(t)$ are combined to predict the j-th sub-series with weight matrix $\mathbf{W}_{out_{ij}} \in R^{N\times 1}$:

$$\mathbf{y}_j(t) = \sum_{i=1}^{D} \mathbf{x}_i^T(t)\mathbf{W}_{out_{ij}} \tag{4}$$

where $\mathbf{y}_j(t) = \mathbf{s}_j(t+1) \in R$.

Similarly, $\mathbf{W}_{out_{ij}}, i, j \in \{1, \cdots, D\}$ is arranged into matrix $\mathbf{W}_{out}$ as follows

$$\mathbf{W_{out}} = \begin{pmatrix} \mathbf{W}_{out_{11}} & \mathbf{W}_{out_{12}} & \cdots & \mathbf{W}_{out_{1D}} \\ \mathbf{W}_{out_{21}} & \mathbf{W}_{out_{22}} & \cdots & \mathbf{W}_{out_{2D}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{W}_{out_{D1}} & \mathbf{W}_{out_{D2}} & \cdots & \mathbf{W}_{out_{DD}} \end{pmatrix} \in R^{ND\times D} \tag{5}$$

where $\mathbf{W}_{out_{ij}} \in R^{N\times 1}$.

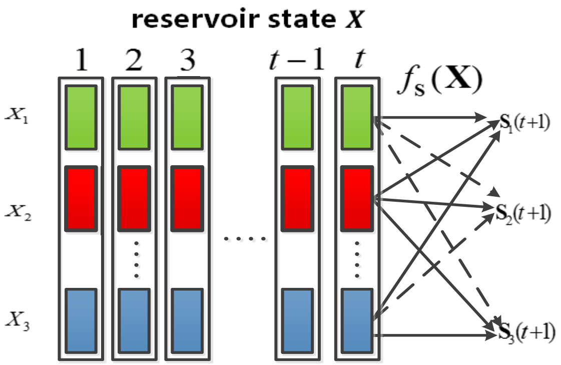


Fig. 5. Illustration of sparsity constrained G-causality in learning MTS models. The colored column represents the element of reservoir state at a moment. $\mathbf{X}_i$ represents the i_{th} collected reservoir state, $\mathbf{s}_j$ represents the desired output(the j_{th} time series). We set i_{th} reservoir state as the input and each $\mathbf{s}_j$ as the target respectively, and get corresponding $\mathbf{W}_{out_{i,j}}$.The dashed arrow from X_i to s_j represents that the learnt corresponding coefficient is zero, that is, there is no G-causality between two series. The solid arrow from X_i to s_j indicates that the learnt corresponding coefficient is nonzero, that is, there is G-causality between two series.

We decompose $\mathbf{W}_{out_{i,j}} = \Psi_{i,j}\mathbf{r}_{i,j}$ into the form of a scalar $\Psi_{i,j}$ and a same dimension vector $\mathbf{r}_{i,j} \in R^{N\times 1}$, $i,j \in \{1,\cdots,D\}$. Then the loss function is generalized as

$$L(\Psi,\mathbf{r}) = \sum_{j=1}^{D}\sum_{t=1}^{(T-1)}(\mathbf{y}_j(t) - \sum_{i=1}^{D}\mathbf{x}_i^T(t)\Psi_{i,j}\mathbf{r}_{i,j})^2 \quad (6)$$

If the learnt $\Psi_{i,j}$ is non-zero, the i_{th} time series G-causes the j_{th} time series. In this way, we are able to reveal the G-causality and learn the G-causality graph from the coefficient of $\mathbf{W}_{out}$. The process of learning the G-causality readout mapping is illustrated in Fig. 5. The algorithm is described in Algorithm 1 in detail.

Algorithm 1 G-causality-based ESN Algorithm

```
Input: a dataset of MTS $\{\mathbf{S}^k, C^k\}_{k=1}^{N_{tr}}$ $\mathbf{S}^k = \mathbf{s}_1^k,\cdots,\mathbf{s}_D^k$, $k = 1,\cdots,N_{tr}$; the number of reservoir units $N$;
Output: Distance matrix among MTS.
for each MTS $\mathbf{S} \in \{\mathbf{S}^k, C^k\}_{k=1}^{N_{tr}}$ do
    for each sub-series $\mathbf{s}_j$, $j = 1,\cdots,D$ do
        Update the reservoir states with the input univariate time series $\mathbf{s}_i$ and obtain the reservoir state $\mathbf{x}_i$ (Eq. 1).
        for each $\mathbf{x}_i$, $i = 1,\cdots,D$ do
            Learn the $W_{out_{ij}}$ by using the linear simplex constrained regression on $\mathbf{s}_j$ and $\mathbf{x}_i$ (Algorithm 2).
        end for
    end for
end for
Calculate the distance $L_2(f_i, f_j)$, $\forall i,j = 1,\cdots,N_{tr}$, via Eq. 9 (Sections $D$).
```

In this way, we provide a method to control the sparsity of the readout mapping by the $D \times D$ matrix Ψ instead of the $ND \times D$ matrix $\mathbf{W}_{out}$ itself. In our method, we transform the matrix Ψ into two $D \times D$ matrices: a sparse matrix $\mathbf{M}$ capturing the common part i.e. the focal series; and the second one is a diagonal matrix $\mathbf{Q} = \tau\mathbf{I}$. $\mathbf{M}$ represents the common part and $\mathbf{Q}$ represents the specific part. We tie the matrices together by $\Psi = \mathbf{M} - diag(\mathbf{M}) + \mathrm{Q}$ where we set $\tau = 1$ for convenience. Thus, the experimental results of our method are not meant to be optimal.

To this end, our objective is to minimize the constrained loss function Eq. 6.

$$\underset{\Psi,\mathbf{r}}{argmin} L(\Psi,\mathbf{r}) \quad s.t.\ \mathbf{1}'\overline{\mathbf{m}} = 1; \overline{\mathbf{m}} \geq \mathbf{0}; \|\ \mathbf{r}\ \|_F^2 \leq \epsilon \quad (7)$$

where $\|\cdot\|_F$ is the Forbenius norm, $\mathbf{1}$ is the a vector of D ones, $\overline{\mathbf{m}}$ is the a column vector of $\mathbf{M}$ and all elements in $\overline{\mathbf{m}}$ are initialized as $\frac{1}{D}$, where D is the number of sub-series in a MTS.

The simplex constraint on $\overline{\mathbf{m}}$ has an important impact on the sparsity for $\mathbf{W}_{out}$. We optimize the objective function by alternating descent for $\mathbf{r}$ and $\overline{\mathbf{m}}$ to find a local minimum. More details can be acquired in Algorithm 2.

Algorithm 2 Sparsity Controlling Algorithm

```
Input: $\mathbf{y}$,$\mathbf{x}$,$\lambda$;
Output: $\mathbf{r}$, $\overline{\mathbf{m}}$
repeat
    Steps for $\mathbf{r}$
        Initiate all columns of $\mathbf{M}$ as $\overline{\mathbf{m}} = \frac{1}{D}$;
        Get $\Psi = \mathbf{M} - diag(\mathbf{M}) + \mathbf{Q}$ where $\mathbf{Q} = \mathbf{I}$;
        Update $\mathbf{p}_{t,i,j} = \mathbf{x}_{t,i}\Psi_{i,j}$;
        Solve for each $j$ $\|\mathbf{y}_{.,j} - \mathbf{P}_j\mathbf{r}_{.,j}\|_2^2 + \lambda\|\mathbf{r}_{.,j}\|_2^2$;
    Steps for $\overline{\mathbf{m}}$
        Get $\mathbf{a}_{t,i,j} = \mathbf{x}_{t,i}^T \cdot \mathbf{r}_{i,j}$
        get residual by using the own history $\mathbf{h}_{t,j} = \mathbf{y}_{t,j} - \mathbf{a}_{t,j,j}$ ;
        concatenate $\mathbf{a}_{t,i,j}$ into $(T-1)\times D$ matrix $\mathbf{A}_i$ and replace the $i_{th}$ column in $\mathbf{A}_i$ by zeros.
        concatenate $\mathbf{A}_i$ into $D(T-1)\times D$ matrix $\mathbf{A}$.
        Get $\overline{\mathbf{m}}$ by $\mathbf{h} = \overline{\mathbf{m}}\mathbf{A}$ with simplex constraint.
until convergence
```

D. Distance between two models

Since the models is considered to capture the important information of the local data collections [27]. It contributes to more robust and more targeted learning on diverse data collections [28]. One may be interested in the distance in the function space of the readout model instead of the distance between the model parameterizations [8]. Therefore, we treat the readout mapping as representation of time series and use the distance of the readout mappings of ESN to measure the similarity between two MTS. The L_2 distance between the readout mapping $f_1(x)$ and $f_2(x)$ are defined as:

$$L_2(f_1, f_2) = \left(\int_{\delta} \|f_1(\boldsymbol{x}) - f_2(\boldsymbol{x})\|^2 \, d\mu(\boldsymbol{x})\right)^{1/2}, \quad (8)$$

where $\mu(x)$ is the probability density function on the input (reservoir) domain δ. Since the activation function is tanh, thus $\delta = [-1, +1]^N$.

We assume that $\boldsymbol{x}$ is uniformly distributed. The readout mapping recalls Eq. 2. Consider two readout mapping generated from the first sub-sequence and the second sub-sequence.

$$\begin{aligned} f_1(\mathbf{x}) &= \mathbf{W}_{out1}\mathbf{x} + \mathbf{b}_1, \\ f_2(\mathbf{x}) &= \mathbf{W}_{out2}\mathbf{x} + \mathbf{b}_2. \end{aligned}$$

Next,

$$L_2(f_1, f_2) = \left(\int_\delta \left(\|\mathbf{W}_{out}\mathbf{x}\|^2 + 2\mathbf{b}^T \boldsymbol{W}_{out}\mathbf{x} + \|\mathbf{b}\|^2 \right) d\mathbf{x} \right)^{1/2}$$

where $\mathbf{W}_{out} = \mathbf{W}_{out1} - \mathbf{W}_{out2}$ and $\mathbf{b} = \mathbf{b}_1 - \mathbf{b}_2$. Because $\delta = [-1, +1]^N$ and $\boldsymbol{b}^T \mathbf{W}_{out}\boldsymbol{x}$ is odd function, we have

$$\int_\delta \mathbf{b}^T \mathbf{W}_{out}\mathbf{x} \; d\mathbf{x} = 0.$$

Then the L_2 distance can be rewritten as

$$L_2(f_1, f_2) = \left(\int_\delta \left(\|\mathbf{W}_{out}\boldsymbol{x}\|^2 + \|\mathbf{b}\|^2 \right) d\mathbf{x} \right)^{1/2}$$

Finally, the L_2 distance is shown below

$$L_2(f_1, f_2) = \left(\frac{2^N}{3} \sum_{i=1}^{D} \sum_{j=1}^{N} w_{i,j}^2 + 2^N \|\mathbf{b}\|^2 \right)^{1/2} \quad (9)$$

where $w_{i,j}$ is the (i, j)-th element of $\mathbf{W}_{out}$.

After we obtain the distance matrix of MTS, we employ SVM and knn as the classifier for classification. The whole process of our method is shown below in Fig. 1.

IV. Experimental Studies

In this section, we perform experiments on benchmark datasets to evaluate the performance of our proposed methods.

We first introduce the employed datasets. We then conduct experiments to demonstrate the effectiveness of finding focal series in a MTS. After that, we compare our method with other state-of-the-art time series classification methods in terms of classification accuracy. Then we demonstrate the robustness of our method. Finally, we evaluate the influence of the size of the reservoir for classification performance and the necessity for preprocessing datasets.

A. Experimental Setup

Two distance-based classifiers, K Nearest Neighbor (*knn*) [29] and SVM [30] are applied to the G-causality classification methods. SVM and knn classifiers for our method are distinguished as G-SVM and G-knn respectively. We compare G-knn and G-SVM with DTW, Reservoir kernel (RV) [8], the G-causality-based VAR (G-linear) and Fisher kernel (FK) [31]. In the G-causality-based classification methods (including G-linear), all the hyperparameters are set as following: The number of nodes in the reservoir state is set to a fixed number as $N = 100$. SVM classifier is implemented by LIBSVM [32]. The hyperparameters of SVM, such as the slack-weight penalty $C \in 10^{-3}, 10^{-2}, \cdots, 10^{3}$, scaling parameter $\gamma \in 10^{-6}, 10^{-5}, \cdots, 10$ are selected by 5-fold cross validation (cv). The searching range of kernel width α is set the same as γ's. After determining these parameters on the training set, we use the same parameters for test set.

In the knn classifier, the k is set from 1 to 10 with the step length of 1, and k is selected according to the highest 5-fold cv accuracy on the training data. We use the selected k to evaluate the classification performance on the test set.

B. Multivariate Time Series Datasets

The proposed approach is empirically studied for the MTS classification. To demonstrate the feasibility and effectiveness of our proposed method, the experiments are performed on seven datasets, i.e., Brazilian sign language (*Libras*), handwrittern characters *handwrittern*, robot failure LP1 (LP1), robot failure LP2 (LP2), Japanese Vowels (JV), ECG and Wafer.

Data sets, JV, LP1 and LP3, are downloaded from the University of California at Irvine (UCI) [33] website[2]. Data sets, ECG and Wafer, are acquired from OLszewski's website[3]. Except *Libras*, LP1 and LP3, the rest of the data sets are MTS of various length. *Libras* and *handwrittern* are processed as the way in [22]. As for JV, LP1, LP3, ECG and Wafer, we randomly set a quarter of each raw dataset as test set and the rest three quarters of dataset as training set. The statistics of these MTS datasets are presented in Table I.

TABLE I
Statistics of the MTS datasets

Dataset	Dim	Length	Classes	Train	Test
Libras	2	45	15	360	585
handwrittern	3	60-182	20	600	2258
LP1	6	15	4	66	11
LP3	6	15	4	36	11
ECG	2	39-152	2	150	50
Wafer	6	104-198	2	896	298
JV	12	7-29	9	270	370

The ECG dataset collects medical time sequences that are recorded by an electrode. Each instance of ECG is a record of electrical potentials sequence of heartbeat. The heartbeat's type can be categorized to normal class and abnormal class. Supraventricular premature beats are selected as abnormal heartbeats. The normal heartbeats are randomly selected from these records.

The wafer dataset collects time sequences that are recorded by a vacuum-chamber sensor. The silicon wafer's information is collected during the manufacture of semiconductor microelectronics. Wafer's type can be categorized to normal class or abnormal class. The abnormal wafers are produced during semiconductor manufacturing. And the normal wafers are randomly selected from the process.

[2] https://archive.ics.uci.edu/ml/datasets.html

[3] http://www.cs.cmu.edu/~bobski/data/data.html

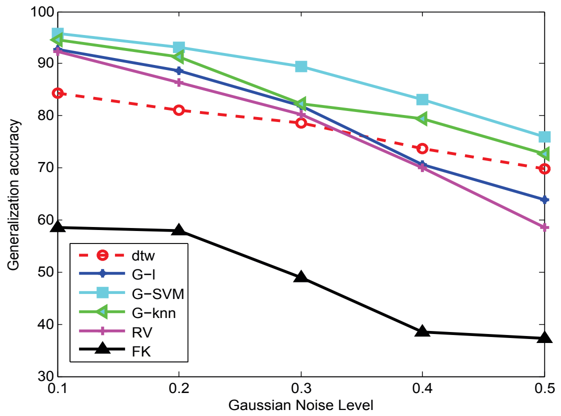


Fig. 6. The proposed method is more robust than others. The noisy is additive Gaussian noise whose mean is zero and standard derivation varies in [0.1:0.1:0.5]

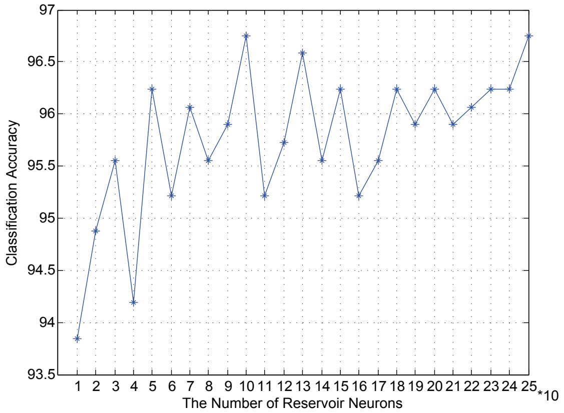


Fig. 8. Classification performances of our methods for different number of reservoir neurons. The experiments are conducted on the *libras* data set. The reservoir size ranges from 10 to 250 with the step size of 10.

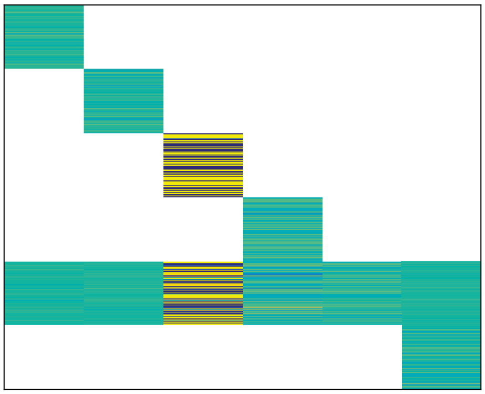


Fig. 7. Example of the learnt parameter matrix $\mathbf{W}_{out}$ on the data set of LP1. The value of the elements of $\mathbf{W}_{out}$ are indices of the default colormap that determine the color of each patch in Matlab. Non-white cells are the non-zero elements. The full-filled row represents the focal series.

TABLE II
CLASSIFICATION ACCURACY FOR DIFFERENT DATASETS OF DTW, RV, G-LINEAR,G-KNN, G-SVM AND FK (%). THE OPTIMAL PERFORMANCE HAS BOLDFACED FOR EACH DATA SET.

Dataset	DTW	RV	G-linear	FK	G-knn	G-SVM
Libras	94.19	93.25	92.82	94.93	95.21	**96.75**
handwrittern	88.67	89.41	78.61	88.44	83.70	**90.48**
LP1	61.90	38.10	80.95	53.36	**85.71**	80.95
LP3	72.73	63.64	81.82	**90.91**	**90.91**	**90.91**
ECG	82.00	62.00	76.00	78.00	82.00	**88.00**
Wafer	85.91	89.93	90.60	58.72	90.60	**91.28**
JV	64.00	89.93	85.68	7.84	87.03	**96.49**

The LP1 and LP2 datasets collect the information of force and torque measurements for a robot after failure detection. The instances are collected in regular time interval no sooner than the failures occur.

The *Libras* dataset collects the sequential information of hand movements in a period of time. The hand movements are represented by a bidimensional curve which were studied from videos of hand movements.

Handwrittern character dataset collects samples of pen tip trajectories of individual characters. All samples are collected from a same writer. Therefore, characters with a single pen-down segment are the only part need to be changed.

Japanese Vowels dataset collects audio streams of two Japanese vowels by Nine male speakers. It applies linear prediction analysis to the audio data to obtain a discrete-time series.

C. Experimental Results

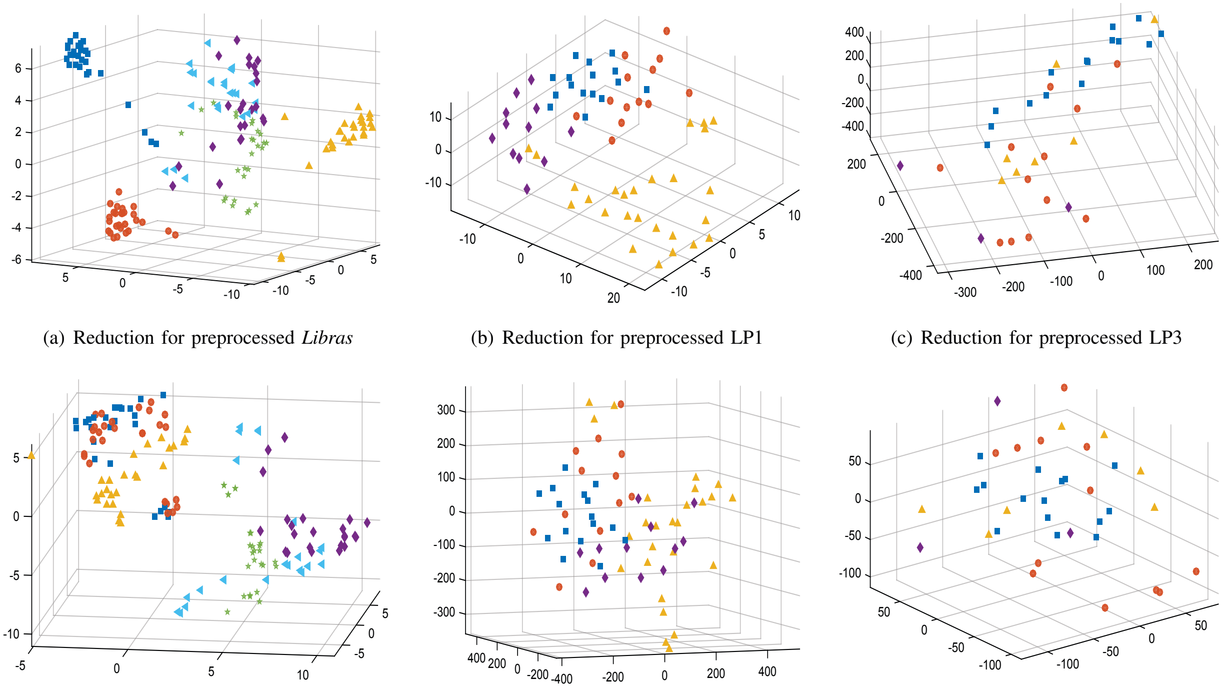
The the results of our experiments are illustrated in this subsection.

1) Focal Series and Sparsity: In order to demonstrate the sparsity of the learnt $\mathbf{W}_{out}$ and prove the exist of FS, we operate an experiment on LP1 dataset. The experiment is based on the G-SVM framework. We color the element values in the first sample of the learnt $\mathbf{W}_{out}$. Fig. 7 depicts that the learnt metric is sparse. And the value of diagonal elements is not zero, which indicates that this method exploit series' own historical information. Moreover, the full-filled row captures the common part shared by all the other series, which means that there are indeed focal series in this sample and our methods do capture the G-causality. Simplex constraint and the FS play an important role in sparsity. For reasons that the amount of the FS is in small amount, the learnt $\mathbf{W}_{out}$ is obviously sparse.

2) Multivariate Time Series Classification: In order to evaluate the classification performances of our method, we compare our methods with RV, DTW, G-linear and fisher kernel. Table II demonstrates the classification accuracies in test set. The performances of the G-causality-based methods are often superior to DTW, RV and FK. It indicates that G-causality indeed exists among different sub-series and G-causality is helpful in the process of predicting other time series.

Moreover, the non-linear G-causality-based methods outperforms G-linear, at least not worse. Since ESN can fit time series better than the linear model.

Compared with our methods, none of the baseline has all



(a) Reduction for preprocessed *Libras*

(b) Reduction for preprocessed LP1

(c) Reduction for preprocessed LP3

(d) Reduction for original *Libras*

(e) Reduction for original LP1

(f) Reduction for original LP3

Fig. 9. Map the preprocessed data (the reservoir state collected in our method) and the original data into the a three-dimensional space to compare the results of the classification performance on the datasets of *Libras*, LP1 and LP3.

the advantages that our methods have. DTW, RV and FK can not reveal the insightful relationships among sub-sequences. The G-linear can revel the relationships but lacks the satisfied merit of enhancing classification accuracy.

3) Robustness: We carry out further experiment to evaluate the robustness of the proposed method on *Libras* dataset. The *Libras* is distorted by zero-mean Gaussian noise whose standard derivation change from 0.1 to 0.5 with step length as 0.1. Fig. 6 shows that all the methods follow the descending tendency when the Gaussian noise becomes larger, but G-SVM and G-knn decline much slower. In other words, G-SVM and G-knn enjoy better robustness and are not so sensitive to noise.

4) Influence of the Reservoir Size: To demonstrate the influence of the number of reservoir neurons on the classification performance, we perform a group of experiments on Libras dataset by varying the number of the reservoir neurons from 10 to 250 with the step length of 10. For each reservoir size, we repeat the experiment 25 times and report the general results in Fig. 8. According to Fig. 8, we can make two main observations. First, the classification accuracy of our method follows approximately growing trend as the number of the reservoir neurons is increasing. We attribute this observation to the fact that larger reservoir usually maintains more approximation ability to input sequences. Second, the accuracy only varies within a small range from 93.5 to 97. Thus, our method seems to be robust to the reservoir size.

5) Visualization: In order to evaluate whether the preprocessed data (the reservoir state) collected in our method contributes to classification performance, the experiment is carried on the three datasets that occupy MTS of the same length, including *Libras*, LP1 and LP3. We map both the raw datasets and the managed datasets into the three-dimensional space. We utilize the t-distributed stochastic neighbor embedding (t-SNE) [34] for dimensionality reduction. T-SNE is a nonlinear dimensionality reduction technique that is particularly well-suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualized in a scattered plot. Alternatively, an initial solution obtained from another dimensionality reduction technique may be specified in initial solution. The labels of the data are used to color intermediate plots. Similar classes are close together and have the same color. The Matlab code of T-SNE comprises three important parameters. The first one is the ”initial_dims”, which is the reduced the dimensionality for preprocessing by PCA [35]. The second one is the ”no_dim” which is the target reduced dimension. The third one is the ”perplexity”, which is the perplexity of the Gaussian kernel. We adjust the angle of view of the mapping illustration to find the best view by hand.

In Fig. 9, in the reduced dimensionality space, figure visually shows that the plots of processed data with same label distribute closer than the plots of the original data, that is, when experiments are conducted by the same classifier, the result on dealt data outperforms that on unprocessed data.

V. Conclusion

In this paper, we propose a novel approach for MTS classification by incorporating ESN with G-causality. Each sub-series is employed to the Echo state network to obtain dynamical information. Then the model of the MTS is trained according to G-causality to integrate the sequential dynamics and the G-causality between sub-series. The dissimilarity between MTS is evaluated by the distance between their models. Experimental results on benchmark datasets demonstrate that the proposed method achieves better classification performance than state-of-the-art MTS classification approaches. In addition, our method also shows more robustness. The training process based on G-causality facilitates the sparsity of the time series models and helps to find the focal series, which G-causes the other time series. In this way, our method is able to reveal the sub-series that are deemed to be the most important for explaining the behavior of MTS.

Our method has limitation in classifying MTS in high dimensionality. In the future, we will focus on reducing the time complexity and optimizing the method.

Acknowledgment

This work is supported by the National Key Research and Development Program of China with grant number 2016YFB1000905, the National Science Foundation of China with grant numbers 91546116, 61673363 and the Fundamental Research Funds for the Central Universities with grant number WK2150110001.

References

[1] S. Ma, Y. Zheng, and O. Wolfson, "Real-time city-scale taxi ridesharing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 7, pp. 1782–1795, 2015.

[2] S. De Vito, G. Fattoruso, M. Pardo, and F. Tortorella, "Semi-supervised learning techniques in artificial olfaction: A novel approach to classification problems and drift counteraction," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3215–3224, 2011.

[3] Y. Fu, H. Xiong, Y. Ge, Z. Yao, Y. Zheng, and Z. H. Zhou, "Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1047–1056, 2014.

[4] M. D. Skowronski and J. G. Harris, "Automatic speech recognition using a predictive echo state network classifier.," *Neural Networks*, vol. 20, no. 3, pp. 414–423, 2007.

[5] C. Sheng, J. Zhao, Y. Liu, and W. Wang, "Prediction for noisy nonlinear time series by echo state network based on dual estimation," *Neurocomputing*, vol. 82, no. 4, pp. 186–195, 2012.

[6] D. Li, M. Han, and J. Wang, "Chaotic time series prediction based on a novel robust echo state network.," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 787–799, 2012.

[7] M. D. Skowronski and J. G. Harris, "Minimum mean squared error time series classification using an echo state network prediction model," in *IEEE International Symposium on Circuits and Systems, 2006. ISCAS 2006. Proceedings*, pp. 4 pp.–3156, 2006.

[8] H. Chen, F. Tang, P. Tino, and X. Yao, "Model-based kernel for efficient time series analysis," pp. 392–400, 2013.

[9] L. Wang, Z. Wang, and S. Liu, "An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm," *Expert Systems with Applications An International Journal*, vol. 43, no. C, pp. 237–249, 2016.

[10] R. J. Granger, "An examination of the concept of potential evaporation," *Journal of Hydrology*, vol. 111, no. 1C4, pp. 9–19, 1989.

[11] S. Yi and V. Pavlovic, "Sparse granger causality graphs for human action classification," pp. 3374–3377, 2012.

[12] H. Qiu, Y. Liu, N. A. Subrahmanya, and W. Li, "Granger causality for time-series anomaly detection," pp. 1074–1079, 2012.

[13] C. Hiemstra and J. D. Jones, "Testing for linear and nonlinear granger causality in the stock price-volume relation," *The Journal of Finance*, vol. 49, no. 5, pp. 1639–1664, 1994.

[14] M. Gregorova, A. Kalousis, S. Marchandmaillet, and J. Wang, "Learning vector autoregressive models with focalised granger-causality graphs," *Computer Science*, 2015.

[15] P. F. Brown, P. V. Desouza, and R. Mercer, "Della pietra vj, lai jc. class-based n-gram models of natural language. computational linguistics," 1992.

[16] N. Chuzhanova, A. J. Jones, and S. Margetts, "Feature selection for genetic sequence classification.," *Bioinformatics*, vol. 14, no. 2, pp. 139–143, 1998.

[17] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," pp. 947–956, 2009.

[18] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," *SIGKDD Explorations*, vol. 12, no. 1, pp. 40–48, 2010.

[19] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," pp. 359–370, 1994.

[20] X. Xi, E. Keogh, C. Shelton, L. Wei, and C. A. Ratanamahatana, "Fast time series classification using numerosity reduction.," *Icml*, pp. 1033–1040, 2006.

[21] R. She, F. Chen, K. Wang, M. Ester, J. L. Gardy, and F. S. L. Brinkman, "Frequent-subsequence-based prediction of outer membrane proteins," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, Dc, Usa, August*, pp. 436–445, 2003.

[22] M. Cuturi and A. Doucet, "Autoregressive kernels for time series," *Statistics*, 2011.

[23] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels.," *Journal of Machine Learning Research*, vol. 5, no. 5, pp. 819–844, 2004.

[24] P. K. Srivastava, D. K. Desai, S. Nandi, and A. M. Lynn, "Hmm-mode c improved classification using profile hidden markov models by optimising the discrimination threshold and modifying emission probabilities with negative training sequences," *BMC Bioinformatics*, vol. 8, no. 1, p. 104, 2007.

[25] A. Arnold, Y. Liu, and N. Abe, "Temporal causal modeling with graphical granger methods," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 66–75, 2007.

[26] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[27] H. Chen, P. Tino, A. Rodan, and X. Yao, "Learning in the model space for cognitive fault diagnosis.," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, p. 124, 2014.

[28] K. H. Brodersen, T. M. Schofield, A. P. Leff, C. S. Ong, E. I. Lomakina, J. M. Buhmann, and K. E. Stephan, "Generative embedding for model-based classification of fmri data.," *PLOS Computational Biology*, vol. 7, no. 6, 2011.

[29] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[30] T. Joachims, "Making large-scale svm learning practical," tech. rep., Technical Report, SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen, Universität Dortmund, 1998.

[31] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *Advances in Neural Information Processing Systems*, vol. 11, no. 11, pp. 487–493, 1998.

[32] C. C. Chang and C. J. Lin, "Libsvm: A library for support vector machines," *Acm Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, article 27, p. 27, 2007.

[33] C. Blake, "Uci repository of machine learning databases," 1998.

[34] V. D. M. Laurens, "Accelerating t-sne using tree-based algorithms," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3221–3245, 2014.

[35] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2002.