2020 IEEE 9th Data Driven Control and Learning Systems Conference
November 20-22, 2020, Liuzhou, China

Granger Causality Detection Based on Neural Network

Jing-Ru Su[1], Jian-Guo Wang[1], Long-Fei Deng[1], Yuan Yao[2], Jian-Long Liu[3]

1. School of Mechatronical Engineering and Automation, Shanghai University, Shanghai Key Lab of Power Station Automation Technology, Shanghai, 200072, China
E-mail: jgwang@shu.edu.cn

2. Department of Chemical Engineering, National Tsing-Hua University, Hsin-Chu, 30013, Taiwan
E-mail: yyao@mx.nthu.edu.tw

3. Shanghai Minghua Electric Power Science &Technology Co., Ltd.Shanghai, 200092, China
E-mail: 15000386718@163.com

Abstract: Plant-wide oscillations are very common in industrial processes. When a control unit oscillates during the process, the oscillations will propagate through the connectivity between the units, which will cause poor product quality and higher energy consumption. It is important to diagnose the root cause of plant-wide oscillations. Generally, methods for estimating Granger causality use linear models such as autoregressive models. This paper proposes using Granger causality analysis based on the neural network for root cause diagnosis, which effectively solves the problem that Granger causality analysis based on linear models cannot handle non-linear data. The Granger causality detection model based on neural network is successfully applied to the plant-wide oscillation root location of industrial process, and the correct root cause is detected, which proves the feasibility and effectiveness of the method.

Key Words: Granger causality, root cause diagnosis, multilayer perceptron, long short-term memory network

1 Introduction

Granger causality quantifies how well past activity of one time series predicts another time series. We can study the entire time series system with a network that explains the interactions [1]. Generally, methods for estimating Granger causality use linear models such as vector autoregressive model (VAR) [2,3]. However, many real systems have non-linear dependencies between time series, so the use of linear models may lead to inconsistent estimates of Granger causality [4,5,6]. A common non-linear method for detecting interactions in time series is the use of additive models. Additive model is a non-parametric model, which is very flexible, because it does not need to assume some form of function like the parametric model, as long as the impact of the predictor on the response variable is independent, it is also called additive hypothesis [4,7,8]. However, additive model may also miss important non-linear interactions with variables, so it may not be able to detect some important non-linear Granger causality.

In order to solve this problem, a Granger causality research framework based on the neural network is proposed. This framework introduces the penalty of weights to increase the sparseness of the neural network to obtain the causality between variables. Because the nodes of the hidden layer in the neural network are connected to each other, the influence of the input is difficult to accurately quantify. Therefore, we usually use the neural network model as a predictive model rather than an explanatory model. One neural network model in this paper is an explanatory model. First, we consider a component architecture. The neural network is a multilayer perceptron (MLP), where each time series i is modeled separately using a separate MLP; the other neural network is a long short-term memory network (LSTM) [9], where a separate LSTM is also used to model each time series i separately. The purpose of this is to better represent the impact of the input's past sequence on a single output sequence. We refer to these sparse component models as componentwise MLP (cMLP) and componentwise LSTM (cLSTM). Second, we perform group Lasso penalties on specific weight groups [10,11]. These weights associate the past activity of each time series with an output sequence, allowing us to accurately select time series without a non-linear Granger causality. Finally, these two neural network models are applied to an industrial example to prove the feasibility and effectiveness of the proposed method.

2 Group Lasso Algorithm

Least absolute shrinkage and selection operator (Lasso) was first proposed by Robert Tibshirani in 1996. This method is a kind of compression estimation. It obtains a more refined model by constructing a penalty function, which makes it compress some coefficients and set some coefficients to 0. It can reduce bias and improve the accuracy of linear regression models.

The general linear model is:

$$Y = X\beta + \varepsilon \tag{2.1}$$

Where, the output variable $Y = (y_1, y_2, \cdots, y_n)^T$, the input variable $X = (X^1,\ X^2, \cdots, X^d)$. For every X^j there is $X^j = (X_1^j, X_2^j, \cdots, X_n^j)^T$ and every X_i^j is centralized and normalized, random error $\varepsilon_i \sim N(0,\ \sigma^2),\ i = 1,2,\cdots,n$,

[*] Corresponding authors.
E-mail addresses: yyao@mx.nthu.edu.tw(Y.Yao).

978-1-7281-5922-5/20/$31.00 ©2020 IEEE

$\varepsilon = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_n)^T$, regression coefficient $\beta = (\beta_1, \beta_2, \cdots, \beta_d)^T$.

When X is a full rank matrix, the regression coefficient β can be obtained by ordinary least squares estimation method:

$$\hat{\beta}_{OLS} = \arg\min_{\beta \in \Re^d} \|Y - X\beta\|^2 = (X^T X)^{-1} X^T Y \quad (2.2)$$

When X is not a full rank matrix, the ordinary least squares method will no longer be suitable for solving regression coefficient β. At this time, we can apply the Lasso regression algorithm to solve the regression coefficient β. Lasso's basic idea is to minimize the sum of squared residuals under the constraint that the sum of the absolute values of the regression coefficients is less than a constant, so that some regression coefficients strictly equal to 0 can be generated, and an explanatory model can be obtained. Lasso regression algorithm can realize variable selection and parameter estimation at the same time. In the parameter estimation, the effect of variable selection is achieved by compressing some parameters to 0. For ordinary linear models, Lasso estimates as:

$$\hat{\beta}_{LASSO} = \arg\min_{\beta \in \Re^d} \left(\|Y - X\beta\|^2 + \lambda \|\beta\|_1 \right) \quad (2.3)$$

Where, λ is the penalty coefficient.

In many cases, there are natural groupings between variables, and variables that belong to the same group should be selected or eliminated as a whole. To solve this problem, Yuan proposed the group Lasso algorithm in 2006. By grouping all variables and then penalizing the L2 norm of each group in the objective function, the effect achieved is that a whole set of coefficients can be punished to 0 at the same time, that is, the entire set of variables is erased, and the objective function is :

$$\hat{\beta}_{group_2} = \arg\min_{\beta \in \Re^d} \|Y - X\beta\|^2 + \lambda \sum_{j=1}^{J} \|\beta_j\|_2 \quad (2.4)$$

3 Group Lasso-granger Algorithm

3.1 Granger Causality Based on Linear Autoregressive Model

Define $x_t \in \Re^p$ as a p-dimensional stationary time series, and its sequence length is T .Linear Granger causality is usually studied using autoregressive (AR) models [2]. In this model, the variable value x_t at time t can be linearly combined by the past k values of the time series.

$$x_t = \sum_{k=1}^{K} A^{(k)} x_{t-k} + e_t \quad (3.1)$$

Where, $A^{(k)}$ ($p \times p$ -dimensional matrix) is the coefficient matrix, which represents the degree of influence of the k-th order past value of the variable on the current value. K is the lag order. e_t is the error matrix. In this model, when $\forall k$, $A_{ij}^{(k)} = 0$, it can be derived that the time series j is not the Granger cause of time series i. Therefore, the Granger causality analysis in the autoregressive model can be converted to determine which values in $A^{(k)}$ are 0 on all k-order lags. Also, the problem can be solved by the group Lasso algorithm.

$$\min_{A^{(1)},\cdots,A^{(K)}} \sum_{t=K}^{T} \left(x_t - \sum_{k=1}^{K} A^{(k)} x_{t-k} \right)^2 + \lambda \sum_{ij} \left\| \left(A_{ij}^{(1)}, \cdots, A_{ij}^{(K)} \right) \right\|_2 \quad (3.2)$$

Where, $\|\bullet\|_2$ represents the L2 norm, which is used to punish all values of $\left(A_{ij}^{(1)}, \cdots, A_{ij}^{(K)} \right)$ to 0. λ is the penalty coefficient used to control the sparsity of the group Lasso.

3.2 Granger Causality Based on Nonlinear Autoregressive Model

The linear autoregressive model can be replaced by a non-linear model. At this time, the Granger causality analysis model can be expressed as:

$$x_t = g(x_{<t1,\cdots,}x_{<tp}) + e_t \quad (3.3)$$

Where, $x_{<ti} = (\cdots, x_{<(t-2)i}, x_{<(t-1)i})$ represents the past value of time series i. In addition, the nonlinear autoregressive function g can be expanded in a single time series:

$$x_{ti} = g_i(x_{<t1,\cdots,}x_{<tp}) + e_{ti} \quad (3.4)$$

For all $(x_{<t1,\cdots,}x_{<tp})$ and $x'_{<tj} \neq x_{<tj}$, equation (3.5) holds, that is, the value of $x_{<tj}$ has no effect on the function g_i. Then time series j is not the Granger cause of time series i.

$$g_i(x_{<t1}, \cdots, x_{<tj}, \cdots, x_{<tp}) = g_i(x_{<t1}, \cdots, x'_{<tj}, \cdots, x_{<tp}) \quad (3.5)$$

Our goal is to estimate the nonlinear Granger causality between variables using the group Lasso penalty optimization method.

4 Granger Causality Detection Based on MLP

4.1 Introduction to MLP Principle

Multilayer Perceptron (MLP) is also called Artificial Neural Network (ANN). In addition to the output layer and input layer, there can be many hidden layers in it. The simplest MLP contains only one hidden layer. The three-layer structure (input layer, hidden layer, output layer) is shown in Fig. 1 below:

As can be seen from the above figure, the layers of the multilayer perceptron are fully connected, that is, any neuron in the upper layer is connected to all neurons in the next layer. The bottom layer of a multilayer perceptron is the input layer, the middle is the hidden layer, and the last is the output layer.

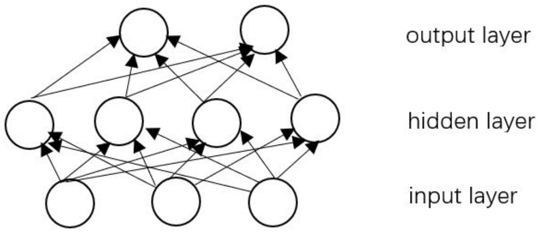

Fig. 1: Three-layer Perceptron

Assuming that the input variable of the input layer is x, the output of the hidden layer is:

$$h_{w,b}(x)=f(W_1^T x)=f(\sum_{t=1}^{p} w_t x_t + b_1) \qquad (4.1)$$

Where, w_t is the weight (also called the connection coefficient), b is the coefficient of the bias, and the activation function $f(\bullet)$ is usually a sigmoid function:

$$sigmoid(z)=1/(1+e^{-z}) \qquad (4.2)$$

This function is a strictly incremental function, which can better balance the relationship between linear and non-linear. It is a working model that is closer to the function of biological neurons. Its graph is shown in Fig. 2 below. In a single neuron, the activation function $f(\bullet)$ is used as a mapping between input and output.

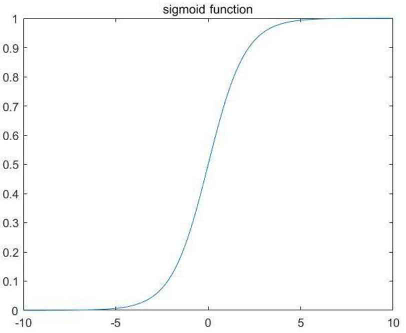

Fig. 2:sigmoid function

The hidden layer to the output layer can be regarded as a multi-class logistic regression, that is, softmax regression, so the output of the output layer is:

$$f(x)=soft\max(W_2^T h + b_2) \qquad (4.3)$$

4.2 Training MLP

MLP is used as a non-linear model of Granger causality analysis. In this model, the entire output set x_t is modeled using MLP, where the input is $x_{<t}=x_{(t-1):(t-K)}$ and K corresponds to the model lag order. Granger causality detection based on MLP has the following two problems: First, it is difficult to select sufficient conditions on the weights to allow time series j to affect time series i but not to sequence $i^{'}$ ($i \neq i^{'}$) because of the shared hidden layer; Second, the MLP model requires that all $g_i(\bullet)$ functions depend on the same lag order, but in practice each $g_i(\bullet)$ may have a different lag order dependency.

In order to better solve the above two problems, we use separate MLP to model each time series i separately, so that we can easily distinguish between input and output. We call this method componentwise MLP (cMLP). Fig. 3 is a schematic diagram of Granger causality modeling using cMLP.

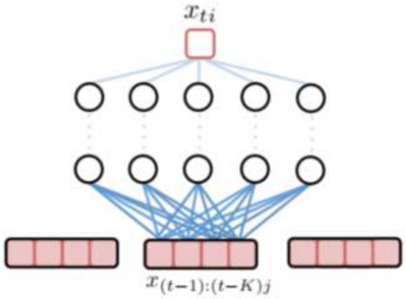

Fig. 3: Schematic diagram of Granger causality modeling using cMLP

Suppose that for each time series i, $g_i(\bullet)$ takes the form of MLP with L layers, and let the vector h_t^1 represents the value of the first hidden layer at time t, and define $W=\{W^1,\cdots,W^L\}$ as the weight matrix of each layer. The variable value of the first hidden layer at time t is:

$$h_t^1=\sigma\left(\sum_{k=1}^{K} W^{1k} x_{t-k} + b_1\right) \qquad (4.4)$$

Where $\sigma(\bullet)$ is the activation function and b_1 is the bias vector of the first hidden layer. As with the first hidden layer, subsequent hidden layers are given by a fully connected unit with an activation function of $\sigma(\bullet)$. At this time, the output x_{ti} at time t is:

$$x_{ti}=g_i(x_{<t})+e_{ti}=W_O^T h_t^L + e_{ti} \qquad (4.5)$$

In equation (4.4), if the j-th column of the first-layer weight matrix can be penalized to 0 for all lag orders k, then time series j is not the Granger cause of time series i. Therefore, similar to the AR model introduced above, the Granger causality can be selected by applying the group Lasso algorithm to the columns of the W^{1k} matrix of each $g_i(\bullet)$, similar to equation (3.5).

$$\min_{W}\sum_{\mathrm{t=k}}^{\mathrm{T}}\left(x_{it}-g_i\left(x_{(t-1):(t-K)}\right)\right)^2+\lambda\sum_{j=1}^{p}\left\|\left(W_{:j}^{11},\cdots,W_{:j}^{1K}\right)\right\|_F \qquad (4.6)$$

Where, λ is the penalty coefficient and $\|\bullet\|_F$ is the F norm. We use a near-end gradient descent method based on linear search to optimize equation (4.6).

5 Granger Causality Detection Based on LSTM

5.1 Introduction to LSTM Principle

Recurrent Neural Network（RNN） is a type of neural network used to process sequence data. The network structure diagram of RNN is shown in Fig. 4. It is a cyclic network that allows information to be persisted. RNN is particularly suitable for modeling time series. Compared with traditional time series models, it compresses the past information of the time series into a hidden state, enabling it to capture complex non-linearities in a longer time lag dependencies. One of the key points of RNN is that it can be used to connect previous information to the current task, such as using past video segments to infer the understanding of the current segment. But as the gap between the relevant information and the current predicted position keeps increasing, the RNN loses the ability to learn to connect such distant information.

Long Short-Term Memory (LSTM) is a special type of RNN that can learn long term dependent information. It is a time recurrent neural network suitable for processing and predicting important events with relatively long intervals and delays in time series. The repeating module in a standard RNN contains a single layer, as shown in Fig. 5. LSTM also has this structure, but the repeated modules have a different structure, as shown in Fig. 6. Unlike a single neural network layer, the LSTM has four repeating modules that interact in a very special way.

LSTM network uses a "gate" structure to remove or add information to neurons to protect and control the state of neurons. LSTM has three "gate" structures, namely input gate, forget gate and output gate.

The forget gate determines what information we will discard from the neuron state. The structure is shown in Fig. 7. The input of the forget gate is the previous hidden layer value h_{t-1} and the current sequence value x_t, and the output value f_t is between 0 and 1. 1 indicates that the current information is completely retained, and 0 indicates that the current information is completely discarded. The formula is as follows:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \tag{5.1}$$

The input gate is used to determine what new information should be stored in the cell state. The structure is shown in Fig. 8. i_t is the updated input value and $\overline{C_t}$ is a new candidate value vector for updating the neuron state. The formulas are as follows:

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \tag{5.2}$$

$$\overline{C_t} = \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \tag{5.3}$$

Once the discarded and updated information is determined, the state of the neuron can be updated. Multiply the old state C_{t-1} and f_t, discard the information we determined to be discarded, and add the updated information $i_t * \overline{C_t}$, this is the new neuron state value C_t. The structure is shown in Fig. 9. The formula is as follows:

$$C_t = f_t * C_{t-1} + i_t * \overline{C_t} \tag{5.4}$$

The structure of the output gate is shown in Fig. 10. We use a sigmoid layer to determine the output information o_t of the neuron state, while processing the neuron state C_t through the tanh layer and multiplying it with the output value o_t to obtain the hidden layer value h_t. The formulas are as follows:

$$o_t = \sigma\left(W_o[h_{t-1}, x_t] + b_o\right) \tag{5.5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{5.6}$$

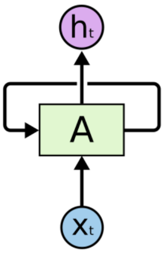

Fig. 4: Network structure of RNN

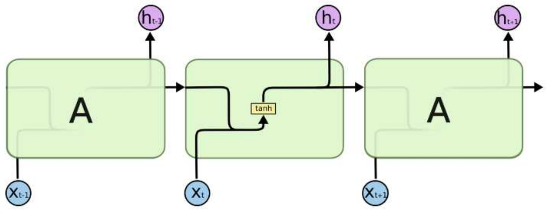

Fig. 5: Repeated module in a standard RNN contains a single layer

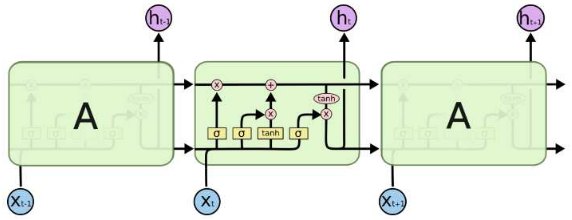

Fig. 6: The repeating module in LSTM contains four interactive layers

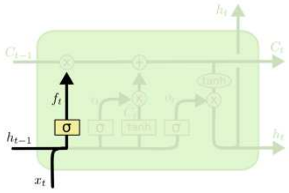

Fig. 7: Forget gate

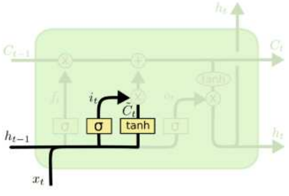

Fig. 8: Input gate

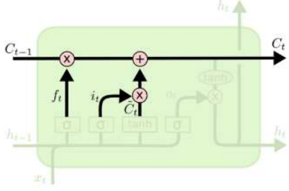

Fig. 9: Update neuron state

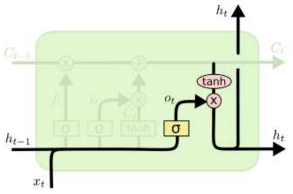

Fig. 10: Output gate

5.2 Training LSTM

LSTM is used as a non-linear model of Granger causality analysis. In this model, the entire output set x_t is modeled using LSTM, where the input is $x_{<t}$. Like MLP, it is difficult for the LSTM to select sufficient conditions on the weights to allow the time series j to affect the time series i but not the sequence $i^{'}$ ($i \neq i^{'}$) due to the shared hidden layer. Therefore, we follow the same strategy as MLP, using a separate LSTM to model each $g_i(\bullet)$ function. A schematic diagram of Granger causality modeling using componentwise LSTM (cLSTM) is shown in Fig. 11.

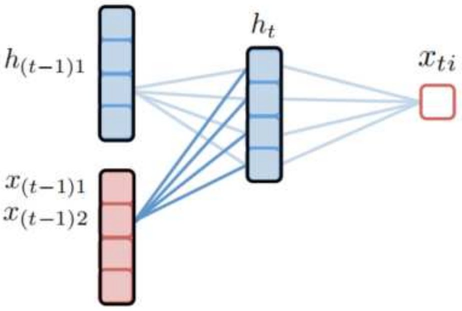

Fig. 11: Schematic diagram of Granger causality modeling using cLSTM

Let $h_{t-1} \in \Re^m$ be the state of the m-dimensional hidden layer at time $t-1$, which is used to represent the past information of the predictor x_{ti}. At this time, the state of the hidden layer at time t will be recursively updated as:

$$h_t = f(x_t, h_{t-1}) \qquad (5.7)$$

Where $f(\bullet)$ depends on the nonlinear function of the specific cyclic structure in the LSTM. The specific loop structure includes forget gate, input gate, unit update and output gate, the formulas are as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \qquad (5.8)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad (5.9)$$

$$\overline{C_t} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \qquad (5.10)$$

$$C_t = f_t * C_{t-1} + i_t * \overline{C_t} \qquad (5.11)$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o) \qquad (5.12)$$

$$h_t = o_t * \tanh(C_t) \qquad (5.13)$$

The dependence of $g_i(\bullet)$ on all past time series $x_{<t}$ is determined by the continuous iterative update of the hidden layer state h_t. The output of the model is:

$$x_{ti} = g_i(x_{<t}) + e_{ti} = W_O^T h_t + e_{ti} \qquad (5.14)$$

From the loop structure of LSTM, it can be concluded that the weight matrix of the neural network is $W = ((W_f)^T, (W_i)^T, (W_C)^T, (W_o)^T)^T$. The weight matrix determines how the past time series x_t affects the forget gate, input gate, unit update and output gate, and Granger causality between variables. For cLSTM, the sufficient condition that there is no Granger causality between the input variable j and the output variable i is that the element in the j-th column of the matrix W is 0, that is, $W_{:j} = 0$. Therefore, similar to the AR model introduced above, the Granger causality can be selected by applying the group Lasso algorithm to the columns of the W matrix of each $g_i(\bullet)$, similar to equation (3.5):

$$\min_W \sum_{t=2}^{T} (x_{ti} - g_i(x<t))^2 + \lambda \sum_{j=1}^{p} \|W_{:j}\|_2 \qquad (5.15)$$

Where, λ is the penalty coefficient and $\|\bullet\|_2$ is the L2 norm. We use a near-end gradient descent method based on linear search to optimize the equation. For proper λ, many columns in matrix W will be punished to 0, so that a sparse Granger causality connection set can be selected.

6 Analysis and Discussion of Experimental Results

Causality analysis is performed on these seven variables using the Granger causality detection method based on LSTM, and the result is shown in Fig. 12. The propagation paths between variables based on LSTM Granger causality detection can be obtained as shown in Fig. 13.

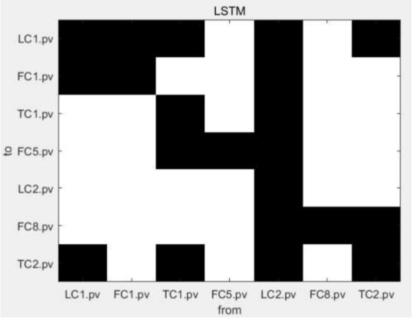

Fig. 12: Granger causality detection results based on LSTM

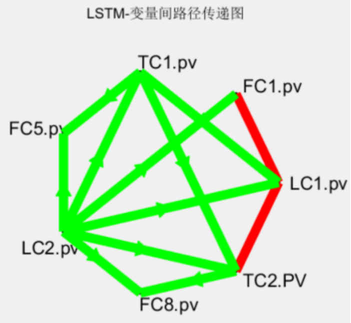

Fig. 13: Propagation paths between Granger causality detection variables based on LSTM

It can be obtained from Fig. 14 that the root cause detected by the LSTM-based Granger causality detection method is LC2, which is consistent with the correct root cause of industrial process analysis.

7 Conclusions

Most Granger causality tests are based on linear models, but there are non-linear causality relationships between variables in many applications such as neuroscience, economics, and industrial systems. In these cases, the use of linear models may lead to inconsistent estimates of Granger causality. In this paper, multilayer perceptron (MLP) and Long Short-Term Memory neural network (LSTM) are used as non-linear models to judge Granger causality. Based on this, the group Lasso algorithm is used to perform sparse induction punishment on the input weight groups to extract Granger causality between variables. And the two Granger causality detection models based on neural network are successfully applied to the plant-wide oscillation root location of industrial processes, and the correct root cause is detected, proving the feasibility and effectiveness of the method.

References

[1] Sumanta Basu, Ali Shojaie, and George Michailidis. Network Granger causality with inherent grouping structure. The Journal of Machine Learning Research, 2015.

[2] Helmut L¨utkepohl. New introduction to multiple time series analysis. Springer Science &Business Media, 2005.

[3] Aurelie C Lozano, Naoki Abe, Yan Liu, and Saharon Rosset. Grouped graphical Granger modeling methods for temporal causal modeling. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009.

[4] Timo Terasvirta, Dag Tjostheim, Clive WJ Granger, et al. Modelling nonlinear economic time series. OUP Catalogue, 2010.

[5] Howell Tong. Nonlinear time series analysis. In International Encyclopedia of Statistical Science. Springer, 2011.

[6] Bethany Lusch, Pedro D. Maia, and J. Nathan Kutz. Inferring connectivity in networked dynamical systems: Challenges using Granger causality. Phys. Rev. E, 2016.

[7] Trevor Hastie and Robert Tibshirani. Generalized additive models. Wiley Online Library, 1990.

[8] Vikas Sindhwani, Ha Quang Minh, and Aur´elie C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and Granger causality. In Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, 2013.

[9] Alex Graves. Supervised sequence labelling. In Supervised Sequence Labelling with Recurrent Neural Networks. Springer, 2012.

[10] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2006.

[11] W. B. Nicholson, J. Bien, and D. S. Matteson. Hierarchical Vector Autoregression. ArXive-prints, 2014.

[12] Hailei Jiang, Rohit Patwardhan, Sirish L. Shah. Root cause diagnosis of plant-wide oscillations using the concept of adjacency matrix.Journal of Process Control, 2009:1347-1354.

[13] Ping Duan,Tongwen Chen,Sirish L.Shah, Methods for Root Cause Diagnosis of Plant-Wide Oscillations: AIcHE Journal,60(4):2019-1034,2014.

[14] Nina F. Thornhill, John W. Cox, Michael A. Paulonis,Diagnosis of plant-wide oscillation through data-driven analysis and process understanding:Control Engineering Practice,2003:1481-1490.