A Novel Hybrid GSTARX-RNN Model for Forecasting Space-Time Data with Calendar Variation Effect

Suhartono*, F Hikmawati, E Setyowati, N A Salehah, A Choiruddin
Department of Statistics, Institut Teknologi Sepuluh Nopember, Surabaya, 60111, Indonesia
suhartono@statistika.its.ac.id

Abstract. Recent development in space-time data forecasting includes a hybrid model. In this study, we propose a hybrid spatio-temporal model by combining Generalized Space-Time Autoregressive with exogenous variable and Recurrent Neural Network (GSTARX-RNN) for space-time data forecasting with calendar variation effect. The GSTARX model as a linear model is used to modeling space-time data with exogenous variables while the RNN model as nonlinear model is used to modeling the nonlinear patterns of the data. In particular, we employ two variants of RNNs, i.e. Elman RNN and Jordan RNN. We apply our methods on the simulation study. The results show that the proposed methods yielded more accurate forecast especially in the simulated data containing nonlinear patterns. Moreover, the GSTARX-Elman RNN as a more complex model tends to give more accurate forecast than the GSTARX-Jordan RNN.

1. Introduction
There have been rapidly advanced development in spatio-temporal forecasting in recent decades. One of the most popular models for space-time data forecasting is the Generalized Space-Time Autoregressive (GSTAR) due to its flexibility on capturing the different dependent-locations effects, for instance, compared to the Space-Time Autoregressive (STAR). The GSTAR model is an extension of the STAR model [1-3] by relaxing the assumptions that all the locations have the same effect, i.e. parameters correspond to dependent-locations are the same. To allow different dependent-locations effects, the GSTAR considers different weighting strategies in the model [3-6].

Recently, the GSTAR model has been developed in more advances setting. First, to involve exogenous variables or to combine with time series regression, the GSTARX model is proposed [7, 8]. Second, the GSTAR model is combined with methods that allow to model nonlinear patterns such as GSTAR-FFNN (FFNN: Feedforward Neural Network) [10]. It is worth noticing that GSTAR is a linear forecasting model which cannot detect nonlinear noise structures. Third, the interest is to combine the first and second developments, i.e., the GSTAR that is able to involve exogenous variables and detect nonlinear patterns. Suhartono, Dana, and Rahayu [11] consider GSTAR-FFNN and GSTAR-DLNN (DLNN: Deep Learning Neural Network) and show that hybrid models can accurately predict space-time data containing trend, seasonal, calendar variation, and linear or nonlinear noise patterns.

In this study, we develop a hybrid model combining GSTARX and RNN (Recurrent Neural Network), named GSTARX-RNN motivated by Suhartono, Dana, and Rahayu [11] to propose a GSTAR model involving exogenous variables which is able to model nonlinear noise patterns. In
particular, we consider two types of RNN model [see e.g. 14], i.e. Elman RNN and Jordan RNN. The Elman and Jordan RNN models both have similarities with a multilayer perceptron (MLP). The Elman RNN adds one or more context layers from the hidden layer while the Jordan adds single context layer from the output layer [12].

Our contribution is to consider RNN-type rather than FFNN or DLNN methods in the hybrid model to model nonlinear noise structures. In general, RNN has similarities with FFNN in terms of the number of hidden units, but it has a context layer as delay neurons, which makes a copy of the hidden layer in the previous time steps before toward the output layer [12, 13]. In the situation when the data have more complex pattern, RNN has major advantages.

The rest of the paper is organized as follows. Section 2 reviews the methodology considered in this paper. Next, we describe the simulation study scenario in section 3 and discuss its results in section 4. Discussion and concluding remarks are given by section 5.

2. Research Methods

2.1 Time Series Regression (TSR)

Time Series Regression (TSR) has the same form as a common linear regression by assuming that output or dependent series \( y_t \), for \( t = 1, 2, ..., n \) are influenced by a collection of possible inputs or independent series, where the inputs are fixed and known. This relation can be expressed as linear regression model [21].

In this research, TSR is used to model the calendar variation effect. The calendar variation model is a time series model used to forecast data based on seasonal patterns with varied periods [22]. This model has the number of the input layer which is similar to the number of independent variables, and in the output layer is a dependent variable. Fig. 1 shows the flowchart of the TSR model.

![Flowchart of Time Series Regression Model](image)

**Figure 1.** Flowchart of Time Series Regression Model

In general, the calendar variation model is based on the regression method if there are trends, seasonality, and calendar variation effect on the data, then the model can be written as

\[
Y_{t_j} = \delta t + \gamma_1 S_{t,1} + \gamma_2 S_{t,2} + \ldots + \gamma_4 S_{t,4} + \nu_1 V_{t,1} + \ldots + \nu_4 V_{t,4} + \nu_{1,1} V_{t-1,1} + \ldots + \nu_{4,4} V_{t-4,4} + \epsilon_t \tag{1}
\]
where \( \delta \) is a linear trend parameter, \( S_{1,t}, S_{2,t}, \ldots, S_{12,t} \) are dummy variables for seasonal pattern, and \( V \) are dummy variables for calendar variation effect [22].

### 2.2 Generalized Space-Time Autoregressive (GSTAR)

The Generalized Space-Time Autoregressive (GSTAR) model is a model that can overcome heterogeneous locations by setting the parameters for each location are different. Due to the STAR model can only be used for a homogeneous location, it causes the GSTAR model become more flexible than the STAR model. Moreover, GSTAR model is also the special case of the VAR model [4]. These characteristics are illustrated by providing location-weighted or spatial \((F)\).

Let’s assume a series of \( \{Y(t) \mid t = 0, \pm 1, \pm 2, \ldots, T\} \) that it is a multivariate time series from \( N \) locations, then the autoregressive order of GSTAR model is \( p \) and the spatial order \( 1, \lambda_2, \ldots, \lambda_p \) or defined as GSTAR \( \{p, \lambda_1, \lambda_2, \ldots, \lambda_p\} \) can be written as follows: [24]

\[
Y(t) = \sum_{k=1}^{p} \Phi_{k0} + \sum_{l=1}^{p} \Phi_{lF} Y(t-k) + e(t).
\]  

(2)

In identifying the order of the GSTAR model, the spatial order is generally limited to one order due to the higher order will be more difficult to interpret [23]. For example, the equation of the GSTAR model for both the time order and spatial order is one, particularly at four different locations can be written as

\[
Y(t) = \Phi_{10} Y(t-1) + \Phi_{1F} F Y(t-1) + e(t).
\]  

(3)

The matrix form of the equation (3) can be written as

\[
\begin{bmatrix}
Y_1(t) \\
Y_2(t) \\
Y_3(t) \\
Y_4(t)
\end{bmatrix} = 
\begin{bmatrix}
\phi_{10} & 0 & 0 & 0 \\
0 & \phi_{20} & 0 & 0 \\
0 & 0 & \phi_{30} & 0 \\
0 & 0 & 0 & \phi_{40}
\end{bmatrix}
\begin{bmatrix}
0 & F_{12} & F_{13} & F_{14} \\
0 & F_{21} & 0 & F_{23} \\
0 & 0 & F_{31} & F_{32} \\
0 & 0 & 0 & F_{41}
\end{bmatrix}
\begin{bmatrix}
Y_1(t-1) \\
Y_2(t-1) \\
Y_3(t-1) \\
Y_4(t-1)
\end{bmatrix} + 
\begin{bmatrix}
e_1(t) \\
e_2(t) \\
e_3(t) \\
e_4(t)
\end{bmatrix}.
\]  

(4)

Fig. 2 shows the flowchart of the GSTAR model with four locations and the input is lag 1 of the dependent variable.

![Figure 2. Flowchart of GSTAR Model with Four Locations](Image)
2.3 Neural Network (NN)
The most commonly used form of Neural Network (NN) architecture is Feedforward Neural Networks (FFNN). FFNN is a flexible class of nonlinear functions in statistical modeling. NN’s has several unique characteristics features such as its adaptability, nonlinearity, arbitrary function mapping ability. These characteristics make this method quite suitable and useful for forecasting tasks [24]. The mathematical expression of the FFNN model is defined as follows

\[ f(z_i, v, w) = f(z_i, \sum_{j=1}^{q} v_j f_i \left( \sum_{i=1}^{p} w_{ji} z_i \right) ) \]

where \( w \) is the weight connecting the input layer to hidden layer, \( v \) is the weight connecting the output layer, \( f(.) \) and \( \hat{f}(.) \) is the activation function as \( w_{ij} \) and \( v_{ij} \) the weights that will be modeled.

Recurrent Neural Network (RNN) model has similarities to FFNN by adding one or more context layers. RNN allows the identification of dynamical systems in form of high dimensional, nonlinear state-space models. In comparison to FFNN, they offer the particular advantage that they are able to explicitly model memory and to identify inter-temporal dependencies [25]. Two major types of RNN’s for dynamic systems are often discussed: the Elman RNN and the Jordan RNN [26-29]. The Elman RNN has information feedback from the hidden layer whereas the Jordan RNN uses feedback from the output layer. Their nature makes them suitable for representing different types of nonlinear dynamic systems.

The number of neurons in the context layer of Elman RNN model is the same as in the hidden layer. Thus, neurons in the context layer are connecting to all neurons in the hidden layer. Neurons in the context layer used as reminders of events on the previous network by storing hidden layer neuron values then returned as additional input on the network [12]. In general, the Elman RNN mathematics model with one hidden layer can be written as follows

\[ \hat{Y}_i = f^0 \left[ b^0 + \sum_{j=1}^{s} w^h_{ij} f_j \left( b^h_i + \sum_{k=1}^{p} w^h_{jk} z_{i-k} + \sum_{i=1}^{q} w^h_{ji} f_{i(i+k)} \right) \right] \]

The Jordan network is much simpler as only the delayed output signal is fed back to the input of the network. The different between Elman RNN and Jordan RNN is on the context layer obtained information from the output layer not from the hidden layer as Elman RNN [12]. Context layer contains delay neurons, and delay neuron is a neuron that saves the memory is located on the activation values in the previous step, then neurons will release the previously saved amounts to the network in the next level. In general, the equation of Jordan RNN model with one hidden layer can be written as follows:

\[ \hat{Y}_i = f^o \left[ \sum_{j=1}^{q} w^h_{ij} z_j + u^i(\hat{Y}_i) \right] \]

2.4 Hybrid Model
Zhang is the first researcher who introduced a hybrid model to improve the accuracy of the forecasting model [9]. In his study, the hybrid model was conducted in two stages. Recently, Setiawan et al. [30] proposed GSTAR model for seasonal spatio-temporal and Suhartono et al. [7] developed the GSTARX model by involving exogenous variables and this model is known as a hybrid TSR and GSTAR model. This model has two stages modeling process, i.e. the first stage is applying TSR model for handling trend, seasonal, and calendar variation effect, and the second stage is using GSTAR model for overcome spatio-temporal dependency. The inputs of the GSTAR model are the residual lags of TSR model. The flowchart of the GSTARX model is illustrated at Fig. 3.

Furthermore, Suhartono et al. [8] proposed a nonlinear spatio-temporal model, known as GSTAR-FFNN, and compared the forecast accuracy with the GSTAR model. The results showed that the GSTAR-FFNN model tend to give more accurate forecast than the GSTAR model both at training and
testing datasets. It is caused the GSTAR-FFNN model can handle linear and nonlinear components simultaneously. The illustration of the GSTAR-FFNN model with one hidden layer and eight inputs for forecasting at four locations data can be seen in Fig. 4.

Figure 3. Flowchart of GSTARX Model

Figure 4. Flowchart of GSTAR-FFNN Model

Hybrid models for space-time data forecasting can also be developed by combining individual linear and nonlinear space-time model as proposed by Zhang [9] for temporal data. Space-time model with linear and nonlinear components that involving exogenous variables can be modeled by GSTARX model (to handle linear component) combined with FFNN model (to overcome nonlinear component), as known as the hybrid GSTARX-FFNN model. The flowchart of the hybrid GSTARX-FFNN model with four locations and one hidden layer can be illustrated as Fig. 5.
The first step of GSTARX-FFNN is a modeling of the trend, seasonal, and calendar variation patterns using TSR. Then, the second step is modeling of the residual from the first step using GSTAR-FFNN. Due to RNN is an extension of FFNN with some advantages particularly the ability to overcome more complex pattern in the data, then the proposed hybrid GSTARX-Elman RNN and GSTARX-Jordan RNN are introduced. Flowchart of the hybrid GSTARX-Elman RNN and the hybrid GSTARX-Jordan RNN with four locations and one hidden layer are illustrated in Fig. 6 and 7, respectively.

Figure 5. Flowchart of GSTARX-FFNN Model

Figure 6. Flowchart of Hybrid GSTARX-Elman RNN Model
At Fig. 5-7, symbol $\hat{e}_{t-1}^{(i)}$ is the first lag of residual of TSR model, and $F_{t-1}^{(i)}$ is the first lag weighted of residual of TSR model. The output from Fig. 5-7 are the vector that consist of space-time data with calendar variation effect at four locations. The more detail explanation about the notations in Fig. 5-7 can be illustrated as follows:

$$
\hat{e}_{t}^{i} = \begin{bmatrix}
\hat{e}_{t-1}^{1} \\
\hat{e}_{t-1}^{2} \\
\hat{e}_{t-1}^{3} \\
\hat{e}_{t-1}^{4}
\end{bmatrix},
\begin{align*}
\varepsilon_{t-1}^{(1)*} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},
\varepsilon_{t-1}^{(2)*} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},
\varepsilon_{t-1}^{(3)*} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix},
\varepsilon_{t-1}^{(4)*} &= \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}
\end{align*}
\begin{align*}
F_{t-1}^{1} &= \begin{bmatrix}
F_{12,t-1}e_{t-1}^{1} + F_{13,t-1}e_{t-1}^{2} + F_{14,t-1}e_{t-1}^{4} \\
0 \\
0 \\
0
\end{bmatrix},
F_{t-1}^{2} &= \begin{bmatrix}
0 \\
F_{21,t-1}e_{t-1}^{1} + F_{23,t-1}e_{t-1}^{2} + F_{24,t-1}e_{t-1}^{4} \\
0 \\
0
\end{bmatrix},
F_{t-1}^{3} &= \begin{bmatrix}
0 \\
0 \\
F_{31,t-1}e_{t-1}^{1} + F_{32,t-1}e_{t-1}^{2} + F_{34,t-1}e_{t-1}^{4} \\
0
\end{bmatrix},
F_{t-1}^{4} &= \begin{bmatrix}
0 \\
0 \\
0 \\
F_{41,t-1}e_{t-1}^{1} + F_{42,t-1}e_{t-1}^{2} + F_{43,t-1}e_{t-1}^{3} \\
0
\end{bmatrix}.
\end{align*}

3. Simulation Study Setting

We conduct simulation study in two scenarios to assess the performance of the hybrid GSTARX-RNN compared to other methods. We consider dataset containing trend, seasonal, calendar variation, and linear and nonlinear noise. The detail steps are as follows.
a. Generating the data considering equation (10) 

$$Y(i) = T(i) + S(i) + V(i) + N(i)$$

(10)

where \(i = 1, 2, 3, 4\) show the locations and \(t = 1, 2, \ldots, 144\) indicating the time series. The four locations are depicted by Fig. 8. It is worth emphasizing that our simulated data are generated through four components, \(T(i), S(i), V(i),\) and \(N(i)\) representing respectively trend, seasonal, calendar variation, and noise components. Each of these components are described in more details by equations (11)-(14).

i. The trend component is 

$$T(i) = \delta(i) t$$

(11)

where \(\delta(0) = 0.3, \delta(2) = 0.2, \delta(3) = 0.23, \delta(4) = 0.16\).

ii. We fix the seasonal component by 

$$S(i) = \gamma(1)iS_{1} + \gamma(2)iS_{2} + \ldots + \gamma(12)iS_{12}$$

(12)

where, 

$$S_{1} = 16S_{1} + 19S_{2} + 21S_{3} + 19S_{4} + 16S_{5} + 12S_{6} + 7S_{7} + 4S_{8} + 3S_{9} + 4S_{10} + 7S_{11} + 12S_{12}$$

$$S_{2} = 14S_{1} + 15S_{2} + 16S_{3} + 15S_{4} + 14S_{5} + 12S_{6} + 10S_{7} + 8S_{8} + 8S_{9} + 10S_{10} + 12S_{11} + 12S_{12}$$

$$S_{3} = 15S_{1} + 18S_{2} + 19S_{3} + 18S_{4} + 15S_{5} + 12S_{6} + 8S_{7} + 5S_{8} + 5S_{9} + 8S_{10} + 8S_{11} + 12S_{12}$$

$$S_{4} = 14S_{1} + 16S_{2} + 17S_{3} + 16S_{4} + 14S_{5} + 12S_{6} + 9S_{7} + 7S_{8} + 7S_{9} + 7S_{10} + 9S_{11} + 12S_{12}$$

iii. The calendar variation component is generated considering the influence of certain events, e.g. the Eid al-Fitr 

$$V(i) = \tau(1)iV_{1} + \ldots + \tau(4)iV_{4} + \nu(1)iV_{1,i-1} + \ldots + \nu(4)iV_{4,i-1}$$

(13)

where, 

$$V_{1} = 8V_{1} + 47V_{2} + 66V_{3} + 82V_{4} + 62V_{5,i-1} + 48V_{6,i-1} + 33V_{7,i-1} + 32V_{8,i-1}$$

$$V_{2} = 3V_{1} + 11V_{2} + 22V_{3} + 29V_{4} + 19V_{5,i-1} + 13V_{6,i-1} + 11V_{7,i-1} + 9V_{8,i-1}$$

$$V_{3} = 3V_{1} + 15V_{2} + 33V_{3} + 36V_{4} + 28V_{5,i-1} + 20V_{6,i-1} + 18V_{7,i-1} + 16V_{8,i-1}$$

$$V_{4} = 2V_{1} + 8V_{2} + 14V_{3} + 19V_{4} + 14V_{5,i-1} + 11V_{6,i-1} + 10V_{7,i-1} + 9V_{8,i-1}$$

iv. We use both linear and nonlinear patterns for the noise component. For linear noise patterns, we follow the GSTAR(11) 

$$N(i) = \phi(1)iN_{1,i-1} + \phi(2)iN_{2,i-1} + \phi(3)iN_{3,i-1} + \phi(4)iN_{4,i-1} + \alpha_{i}$$

(14)

where, 

$$N_{1} = 0.41N_{1,i-1} + 0.19N_{2,i-1} + 0.19N_{3,i-1} + 0.19N_{4,i-1} + \alpha_{i}$$

$$N_{2} = 0.21N_{1,i-1} + 0.38N_{2,i-1} + 0.21N_{3,i-1} + 0.21N_{4,i-1} + \alpha_{i}$$

$$N_{3} = 0.22N_{1,i-1} + 0.22N_{2,i-1} + 0.36N_{3,i-1} + 0.22N_{4,i-1} + \alpha_{i}$$

$$N_{4} = 0.2N_{1,i-1} + 0.2N_{2,i-1} + 0.2N_{3,i-1} + 0.32N_{4,i-1} + \alpha_{i}$$

and for the nonlinear noise structures are obtained by
\[ N_t^{(i)} = 3.5N_{t-1} \times \exp\left(-0.2N_{t-1}^2\right)N_{t-1} + 0.9N_{2,t-1} \times \exp\left(-0.2N_{2,t-1}^2\right)N_{2,t-1} + 1.5N_{3,t-1} \times \exp\left(-0.2N_{3,t-1}^2\right)N_{3,t-1} + 1.8N_{4,t-1} \times \exp\left(-0.2N_{4,t-1}^2\right)N_{4,t-1} + a_t^{(i)} \]

\[ N_t^{(2)} = 1.8N_{t-1} \times \exp\left(-0.2N_{t-1}^2\right)N_{t-1} + 4N_{2,t-1} \times \exp\left(-0.2N_{2,t-1}^2\right)N_{2,t-1} + 1.1N_{3,t-1} \times \exp\left(-0.2N_{3,t-1}^2\right)N_{3,t-1} + a_t^{(2)} \]

\[ N_t^{(3)} = 1.2N_{t-1} \times \exp\left(-0.2N_{t-1}^2\right)N_{t-1} + 2.3N_{t-1} \times \exp\left(-0.2N_{2,t-1}^2\right)N_{2,t-1} + 4.5N_{t-1} \times \exp\left(-0.2N_{3,t-1}^2\right)N_{3,t-1} + a_t^{(3)} \]

\[ N_t^{(4)} = 1.9N_{t-1} \times \exp\left(-0.2N_{t-1}^2\right)N_{t-1} + 2.1N_{t-1} \times \exp\left(-0.2N_{2,t-1}^2\right)N_{2,t-1} + 1.3N_{t-1} \times \exp\left(-0.2N_{3,t-1}^2\right)N_{3,t-1} + a_t^{(4)} \]

where \( a_t \sim N(0, \Sigma) \) and

\[ \Sigma = \begin{bmatrix} 1 & 0.6 & 0.5 & 0.6 \\ 0.6 & 1 & 0.6 & 0.6 \\ 0.5 & 0.6 & 1 & 0.5 \\ 0.6 & 0.6 & 0.5 & 1 \end{bmatrix} \]

b. By step (a), we obtain \( Y_t^{(i)} \), \( i = 1, 2, 3, 4 \) and \( t = 1, 2, \ldots, 144 \). We evaluate our methods in two scenarios, i.e.,

i. Scenario I, generated data obtained by (10) but involving linear noise pattern, meaning that the term \( N_t^{(i)} \) follows equation (14).

ii. Scenario II, the same scenario as in scenario 1 except that the noise structure is nonlinear. See step (a-iv).

![Figure 8. Map of four locations considered in the simulated data](image)

Fig. 9-10 give the time series plot of the generated data obtained from (10). In general, Fig. 9-10 show increasing trend for all locations in each scenario and seasonal pattern. In addition, calendar variation is clearly visible shown by very high values in certain months. The main difference between Fig. 9 and Fig. 10 is that the patterns (trend, seasonal, and calendar variation) is more obvious for Fig. 1 due to lack of nonlinearity structures of the noise.
4. Results and Discussion

4.1 Time Series Regression (TSR) Model

Independent variables in TSR model are trend, seasonal, and calendar variation. The TSR model from scenario I at the first replication that follow equation (1) is:

\[
\begin{bmatrix}
Y_{t}^{(1)} \\
Y_{t}^{(2)} \\
Y_{t}^{(3)} \\
Y_{t}^{(4)}
\end{bmatrix} =
\begin{bmatrix}
0.3r + 25.6S_{1,t} + \ldots + 20.4S_{12,t} + 60.6V_{1,t} + \ldots + 84.8V_{4,t-1} \\
0.2r + 18.4S_{1,t} + \ldots + 13.3S_{12,t} + 21.2V_{1,t} + \ldots + 31.2V_{4,t-1} \\
0.2r + 14.3S_{1,t} + \ldots + 10.1S_{12,t} + 24.9V_{1,t} + \ldots + 31.7V_{4,t-1} \\
0.2r + 12.8S_{1,t} + \ldots + 9.7S_{12,t} + 15.9V_{1,t} + \ldots + 18.1V_{4,t-1}
\end{bmatrix} +
\begin{bmatrix}
\epsilon_{t}^{(1)} \\
\epsilon_{t}^{(2)} \\
\epsilon_{t}^{(3)} \\
\epsilon_{t}^{(4)}
\end{bmatrix}.
\]

\[(15)\]

Figure 9 Time series plot of the simulated data considered in the scenario I

Figure 10. Time series plot of the simulated data considered in the scenario II
The estimated parameter of TSR based on equation (15) is statistically not difference with the parameter values of data generating model. It shows that the TSR estimator is unbiased. Fig. 11 presents the bar-chart of the RMSE value of TSR model from both scenarios. These results show that RMSE of scenario II is higher than RMSE of scenario I. It is caused by the TSR model can not capture nonlinear patterns at the noise component. Otherwise, the RMSE value for both scenarios is greater than 1 (as simulation setting) and indicate that the modeling process not yet finish or other method is needed to achieve a smaller error accuracy.

\[
\begin{bmatrix}
Y_t^{(1)} \\
Y_t^{(2)} \\
Y_t^{(3)} \\
Y_t^{(4)}
\end{bmatrix} =
\begin{bmatrix}
-0.6 & 0.2 & 0.2 & 0.2 \\
-0.03 & -0.4 & -0.03 & -0.03 \\
-0.03 & -0.03 & -0.4 & -0.03 \\
-0.01 & -0.01 & -0.01 & -0.5
\end{bmatrix}
\begin{bmatrix}
Y_{t-1}^{(1)} \\
Y_{t-1}^{(2)} \\
Y_{t-1}^{(3)} \\
Y_{t-1}^{(4)}
\end{bmatrix} +
\begin{bmatrix}
e_t^{(1)} \\
e_t^{(2)} \\
e_t^{(3)} \\
e_t^{(4)}
\end{bmatrix},
\]

(16)

The results of the forecast accuracy of the GSTAR(1,1) are illustrated in Fig. 12. It shows that the RMSE at the out-of-sample of the GSTAR(1,1) at four locations is very high and greater than 1. It means that the GSTAR(1,1) can not forecast well the data due to this model can not capture the calendar variation effect. Fig. 12 also shows that the average RMSE at the out-of-sample dataset on each location is higher than an in-sample dataset.

4.3 RNN Model

The RNN models, both Elman RNN and Jordan RNN, used lag 1 of the dependent variables as inputs of GSTAR model. The hyperbolic tangent function was employed at the activation function in the hidden layer and linear function in the output layer. The number of neurons in the hidden layer was determined from 1 to 5, 10, and 15. The best model was selected based on the smallest RMSE at out-of-sample dataset.
The Elman RNN model in the scenario I at the first replication with four neurons is expressed as follows:

$$\hat{Y}_i = 1.8 f_1^h(z) - 1.5 f_2^h(z) - 0.8 f_3^h(z) - 1.4 f_4^h(z)$$  \hspace{1cm} (17)

where

$$f_1^h(z) = \tanh \left( 7Y_{i-1}^{(1)v} + \cdots + 2Y_{i-1}^{(4)v} + 2.5 f_1^h(z_{i-1}) + \cdots + 1.4 f_4^h(z_{i-1}) \right)$$

$$f_2^h(z) = \tanh \left( -3.5 Y_{i-1}^{(1)v} + \cdots + 3.6 Y_{i-1}^{(4)v} - 2.3 f_1^h(z_{i-1}) + \cdots + 0.7 f_4^h(z_{i-1}) \right)$$

$$f_3^h(z) = \tanh \left( 7Y_{i-1}^{(1)v} + \cdots + 1.3 Y_{i-1}^{(4)v} - 4.4 f_1^h(z_{i-1}) + \cdots - 11 f_4^h(z_{i-1}) \right)$$

$$f_4^h(z) = \tanh \left( -1Y_{i-1}^{(1)v} + \cdots - 4.7 Y_{i-1}^{(4)v} + 6.4 f_1^h(z_{i-1}) + \cdots - 4.9 f_4^h(z_{i-1}) \right)$$

Otherwise, the equation of Jordan RNN model from the scenario I at the first replication with one neuron is

$$\hat{Y}_i = 0.99 \tanh \left[ (14.6 + 3.4) Y_{i-1}^{(1)v} + (1.8 + 3.4) Y_{i-1}^{(2)v} + (3.5 + 3.4) Y_{i-1}^{(3)v} + (2.2 + 3.4) Y_{i-1}^{(4)v} - 3.4 a_{i-1} \right].$$ \hspace{1cm} (18)

Moreover, the best Elman RNN model of the scenario II at the first replication with two neurons is as follows:

$$\hat{Y}_i = 0.99 f_1^h(z) - 1.5 f_2^h(z)$$  \hspace{1cm} (19)

where

$$f_1^h(z) = -1.1 Y_{i-1}^{(1)v} + 0.5 Y_{i-1}^{(2)v} + 0.3 Y_{i-1}^{(3)v} - 1.4 Y_{i-1}^{(4)v} + 4.3 f_1^h(z_{i-1}) - 6.2 f_2^h(z_{i-1})$$

$$f_2^h(z) = -3.2 Y_{i-1}^{(1)v} + 3.3 Y_{i-1}^{(2)v} - 3.3 Y_{i-1}^{(3)v} - 4.1 Y_{i-1}^{(4)v} + 0.7 f_1^h(z_{i-1}) - 4.4 f_2^h(z_{i-1})$$

and the best Jordan RNN model of the scenario II at the first replication with two neurons is

$$\hat{Y}_i = -0.727 f_1^h(z) + 1.31 f_2^h(z)$$  \hspace{1cm} (20)

where

$$f_1^h(z) = \tanh \left[ (0.29 + 0.299) Y_{i-1}^{(1)v} + (0.012 + 0.299) Y_{i-1}^{(2)v} + (0.089 + 0.299) Y_{i-1}^{(3)v} + (-0.134 + 0.299) Y_{i-1}^{(4)v} - 0.299 a_{i-1} \right]$$

$$f_2^h(z) = \tanh \left[ (8.13 - 1.34) Y_{i-1}^{(1)v} + (5.309 - 1.34) Y_{i-1}^{(2)v} + (8.29 - 1.34) Y_{i-1}^{(3)v} + (4.5 - 1.34) Y_{i-1}^{(4)v} + 1.34 a_{i-1} \right]$$

$Y_{i-1}^{(i)v}$ is standardize of $Y_{i-1}$ in the $i^{th}$ locations, and
These equations show that the best number of neurons of Elman RNN and Jordan RNN are not always the same even though using the same initial inputs. Due to the additional inputs (context layer) of Elman RNN and Jordan RNN are different, both models yield different smallest RMSE. The results about RMSE of both Elman RNN and Jordan RNN are illustrated in Table 1.

### Table 1. The Average of RMSE from RNN Model on Scenario I and II

| Location | RMSE in Scenario I | RMSE in Scenario II |
|----------|-------------------|---------------------|
|          | Elman RNN | Jordan RNN | Elman RNN | Jordan RNN |
| 1        | 17.081     | 21.182     | 19.268     | 22.336     |
| 2        | 8.370      | 8.075      | 9.813      | 9.085      |
| 3        | 8.119      | 10.499     | 8.771      | 11.515     |
| 4        | 6.172      | 5.828      | 6.772      | 6.874      |

Based on Table 1, it can be seen that the RMSE in both methods, Elman RNN and Jordan RNN, is greater than 1 or not accurate forecast. It is caused both model can not capture the calendar variation pattern by involving only lag 1 as the input variable.

**4.4 GSTARX Model**

The first stage of GSTARX model is fitting TSR to capture trend, seasonal, and calendar variation patterns in both scenarios. The second stage is fitting GSTAR model from the residual of TSR. Thus, the result of TSR model at the first stage at the first replication in the scenario I is as equation (15). Then, the residual of the TSR model is modelled by GSTAR(1) and the result is as follows:

\[
\begin{bmatrix}
ed(t) \\
ed(t) \\
ed(t) \\
ed(t)
\end{bmatrix} = \begin{bmatrix}
0.3 & 0.2 & 0.2 & 0.2 \\
0.2 & 0.15 & 0.2 & 0.2 \\
0.1 & 0.1 & 0.4 & 0.1 \\
0.15 & 0.15 & 0.15 & 0.3
\end{bmatrix} + \begin{bmatrix}
ed(t) \\
ed(t) \\
ed(t) \\
ed(t)
\end{bmatrix}.
\]  

Furthermore, Fig. 13 summaries the results of RMSE of GSTARX model in both scenarios at each location. The results show that the RMSE of the scenario II is greater than the scenario I. It means that the GSTARX model, as a hybrid linear model, can predict well the data at the scenario I due to only consist of linear data pattern. Otherwise, the GSTARX model can not fit well the data at the scenario II because of nonlinearity pattern in the data.

**4.5 Hybrid GSTARX-RNN Model**

The first stage of hybrid GSTARX-RNN model is TSR as in GSTARX model and the results as in equation (15). Next, the second stage is modeling the residuals of TSR by using Elman RNN and Jordan RNN. Fig. 14 shows the best architecture of GSTARX-Elman RNN and GSTARX-Jordan RNN model in the scenario I at the first replication.
Figure 13. Boxplot of RMSE from GSTARX Model on Data Scenario I and II

Based on the architecture in Fig. 14, the GSTARX-Elman RNN model is as follows:

\[
\hat{e}_i = -1.16 \times \tanh \left( -2.33e_{i-1}^{(1)} - 1.79F_{i-1}^{(1)} - 1.35e_{i-1}^{(2)} + 3.35F_{i-1}^{(2)} - 2.10e_{i-1}^{(3)} - 2.14F_{i-1}^{(3)} - 0.82e_{i-1}^{(4)} + 3.62F_{i-1}^{(4)} - 0.62f_i^z(z) \right)
\]

and the GSTARX-Jordan RNN model is

\[
\hat{e}_i = 1.14 \times \tanh \left( -0.02 - 0.62e_{i-1}^{(1)} + (7.73 - 0.62)F_{i-1}^{(1)} + \cdots + (3.59 - 0.62)e_{i-1}^{(4)} + (4.33 - 0.62)F_{i-1}^{(4)} + 0.62a_{i-1} \right)
\]

where the \( \hat{e}_i \) and \( F_i \) are as equation (8) and (9). The number of inputs in GSTARX-RNN on the scenario I is eight, i.e. consist of residual lag-1 of TSR from each location ( \( \hat{e}_{i-1} \)) and four \( F_i \) series as multiplication of residual lag-1 by uniform weight. Furthermore, the architecture of GSTARX-Elman RNN and GSTARX-Jordan RNN on scenario II at the first replication given in Fig. 15.

The RMSE from GSTARX-Elman RNN and GSTARX-Jordan RNN are shown in Fig. 16 and 17, respectively. The results show that the GSTARX-Elman RNN and GSTARX-Jordan RNN yield more accurate forecast than other previous models. It means that both hybrid GSTARX-RNN models can capture well all the pattern in the data, both linear and nonlinear patterns.
4.6 Best Model Selection
The best model is chosen based on the RMSE ratio criteria. In this study, time series regression (TSR) will be used as the benchmark model. If the value of the RMSE ratio is less than 1, it shows that the
The 5th International Conference on Basic Sciences
IOP Conf. Series: Journal of Physics: Conf. Series 1463 (2020) 012037
doi:10.1088/1742-6596/1463/1/012037

method can reduce forecast errors compared to the benchmark model. The RMSE ratio of each model is shown in Table 2.

| Methods      | Scenario I | Scenario II |
|--------------|------------|-------------|
|              | 1 2 3 4 1 2 3 4 |
| TSR          | 1 1 1 1 1 1 1 1 |
| GSTAR        | 20.307 5.562 7.397 3.843 14.135 4.522 5.445 3.639 |
| Elman RNN    | 11.547 5.546 6.235 4.330 6.291 2.914 2.802 2.125 |
| Jordan RNN   | 9.428 3.217 4.016 2.229 6.270 2.339 2.737 1.780 |
| GSTARX       | 0.738 0.730 0.880 0.861 0.767 0.779 0.761 0.763 |
| GSTARX-FFNN  | 0.832 0.747 0.900 0.956 0.016 0.016 0.016 0.018 |
| GSTARX-DLNN  | 0.954 0.964 1.028 0.785 0.021 0.016 0.021 0.021 |
| GSTARX-Elman RNN | 0.812 0.789 0.954 0.856 0.012 0.034 0.020 0.021 |
| GSTARX-Jordan RNN | 0.535 0.494 0.512 0.540 0.025 0.024 0.024 0.029 |

Based on the results, in general, the hybrid GSTARX and GSTARX-RNN methods yield RMSE ratio which smaller than 1. It shows that these methods can effectively improve the forecast accuracy rather than using the TSR model. Hybrid GSTARX-FFNN, GSTARX-DLNN, and GSTARX-Elman RNN and GSTARX-Jordan RNN performs similarly to the GSTARX model when handling data with a linear case. Nonetheless, the hybrid GSTARX model combines with neural network yields more accurate forecast when modelling the data containing nonlinear patterns than GSTARX. The proposed methods, i.e. GSTARX-Elman RNN and GSTAR-Jordan RNN, have similar results with GSTARX-FFNN and GSTARX-DLNN, especially in nonlinear data patterns. According to the average RMSE ratio from scenario II in the second location, the hybrid GSTARX-FFNN and hybrid GSTARX-DLNN models can reduce the RMSE of time series regression models by 98.4%. While the hybrid GSTARX-Elman RNN and GSTARX-Jordan RNN models can reduce it by 96.6% and 97.6%.

5 Discussion and Concluding Remark

In this study, we develop a combined method between GSTARX and two types of RNN (Elman and Jordan RNNs) to model a complex simulated spatio-temporal data. We test our method in the simulation study from generated data with trend, seasonal, calendar variation, and (linear and nonlinear) noise structures. The results show that the hybrid methods (GSTARX-RNN, GSTARX-FFNN, and GSTARX-DLNN) outperform the other methods, especially in the situation where nonlinear noise patterns are present. Furthermore, GSTARX-Elman RNN gives slightly better results as an expense of considering a more complex model. These current findings further support the idea of combining statistical and machine learning approaches in forecasting space-time data which correspond to the forecast accuracy improvement [31]. In the case where the generated data has linear noise structures, the GSTARX-RNN performs similarly to the the GSTARX. Further study consists of investigating the performance of the methods in the real dataset. One example is to apply our methods to forecast space-time air quality data as it has strong evidence of trend, seasonal, calendar variation and nonlinear patterns.

Acknowledgments

This research was supported by DRPM under the scheme of “Penelitian Dasar Unggulan Perguruan Tinggi”, project No. 938/PKS/ITS/2019. The authors thank the General Director of DIKTI for funding and anonymous referees for their useful suggestions.
References

[1] Pfeifer P E and Deutsch S J A 1980 Three stage iterative procedure for space-time modeling. Technometrics 22 35-47.

[2] Pfeifer P E and Deutsch S J A 1980 Identification and interpretation of first order space-time ARMA Models. Technometrics 22 397-408.

[3] Ruchjana B N 2002 Modeling of the petroleum production curve using the model generalisas S-TAR\(^1\) (Pemodelan kurva produksi minyak bumi menggunakan model generalisasi S-TAR\(^3\)) Forum Statistika dan Komputasi 1-6.

[4] Borovkova S A, Lopuhaa H P and Ruchjana BN 2002 Generalized STAR models with experimental weights Proceedings of the 17th International Workshop on Statistical Modelling 143-152.

[5] Suhartono and Atok RM 2006 The choice of optimal location weights on the GSTAR model (Pemilihan bobot lokasi yang optimal pada model GSTAR) Prosiding Konferensi Nasional Matematika XIII, Universitas Negeri Semarang. 2006.

[6] Suhartono and Subanar 2006 The optimal determination of space weight in GSTAR model by using cross-correlation inference J. Quantitative Methods 2 45-53.

[7] Suhartono, Wahyuningrum S R Setiawan and Akbar M S 2017 GSTARX-GLS model for spatio-temporal data forecasting Malaysian J Mathematical Sci 10 91-103.

[8] Suhartono, Gazali M M and Prastyo D D 2018 VARX and GSTARX models for forecasting currency inflow and outflow with multiple calendar variations effect MATHEMATIKA 34 57-72.

[9] Zhang G P 2003 Time series forecasting using a hybrid ARIMA and neural network model Neurocomputing 50 159-175.

[10] Suhartono, Prastyo D D, Kuswanto H and Lee M H 2018 Comparison between VAR, GSTAR, FFNN-VAR and FFNN-GSTAR models for forecasting oil production MATEMATIKA 34 103-111.

[11] Suhartono, Dana I and Rahayu S P 2019 Hybrid model for forecasting space-time data with calendar variation effects Telkomnika (Telecommunication Computing Electronics and Control) 17 118-130.

[12] Lewis N D 2016 Deep learning made easy with R South Carolina: Createspace Independent Publishing Platform.

[13] Chandra R 2015 Competition and collaboration in cooperative coevolution of elman recurrent neural networks for time-series prediction IEEE transactions on neural networks and learning systems 26 3123-3136.

[14] Choi E, Schuetz A, Stewart W and Sun J 2016 Using recurrent neural network models for early detection of heart failure onset J. Am. Med. Inform. Assoc. 24 361-370.

[15] Rahman A, Srikumar V and Smith A 2018 Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks Applied Energy 212 372-385.

[16] Suhartono and Endharta A J 2009 Forecasting short-term electricity consumption with seasonal arima and elman-recurrent neural network (Peramalan konsumsi listrik jangka pendek dengan arima musiman dan elman-recurrent neural network) Sci. J Information and Technology 7 183-190.

[17] Ruiz L G B, Rueda R, Cuéllar M P and Pegalajar M C 2018 Energy consumption forecasting based on elman neural networks with evolutive optimization 2018 Expert Systems with Applications 92 380–389.

[18] Liu H, Mi X and Li Y 2018 Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network Energy Conversion and Management 156 498-514.

[19] Wysocki A and Lawrynczuk M 2015 Jordan neural network for modeling and predictive control of dynamic systems 20th International Conference on Methods and Models in Automation and Robotics (MMAR) 145-150.

[20] Carcano E, Bartolini P, Muselli M and Piroddi 2008 L Jordan recurrent neural network versus
IHACRES in modeling daily streamflows *J. Hydrology* **362** 291-307.

[21] Shumway R H and Stoffer D S 2006 *Time series analysis and its applications with R examples* 2nd edition, Berlin: Springer.

[22] Suhartono, Lee M H and Hamzah N A 2010 Calendar variation model based on Time Series Regression for sales forecasts: The Ramadhan effects *Proceedings of the Regional Conference on Statistical Sciences* **10** 30–41.

[23] Wutsqa D, Suhartono and Sutijo B 2010 Generalized space-time autoregressive modelling *Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and its Applications (ICMSA2010)* Kuala Lumpur Malaysia: Universiti Tunku Abdul Rahman 752–761.

[24] Zhang P G, Patuwo E and Hu, M Y 1998 Forecasting with artificial neural networks: the state of the art *International Journal of Forecasting* **14** 35-62.

[25] Zimmermann H G, Grothmann R, Schaefer A M and Tietz C 2006 Identification and forecasting of large dynamical systems by dynamical consistent neural networks, in Haykin, S., Principe, J., Sejnowski, T., & McWhirter, J. (eds.), New Directions in Statistical Signal Processing: From Systems to Brain, MIT Press 203–242.

[26] Jang J S R, Sun C T and Mizutani E 1997 Neuro-fuzzy and soft computing-a computational approach to learning and machine intelligence, Prentice Hall.

[27] Haykin S 1999 Neural networks-a comprehensive foundation, 2nd Edition, Prentice Hall, 1999.

[28] Jordan M I 1998 Attractor dynamics and parallelism in a connectionist sequential machine. *The Eighth Annual Conf. of the Cognitive Science Society* 531-546.

[29] Elman J L 1990 Finding structure in time, *Cognitive science* **14** 179-211.

[30] Setiawan, Suhartono and Prastuti 2016 M. S-GSTAR-SUR model for seasonal spatio temporal data forecasting. *Malaysian Journal of Mathematical Sciences* **10** 53-65.

[31] Makridakis S, Spiliotis E and Assimakopoulos V 2018 The M4 competition: results, findings, conclusion and way forward *International J. Forecasting* **34** 802-808.