

# RISET-33-2019-PDUPT- ICENIS2019- e3sconf\_icenis2019\_23015.pdf

*by Budi Warsito*

---

**Submission date:** 27-Jan-2020 04:01AM (UTC+0700)

**Submission ID:** 1246596410

**File name:** RISET-33-2019-PDUPT-ICENIS2019-e3sconf\_icenis2019\_23015.pdf (1.74M)

**Word count:** 3657

**Character count:** 18101

# Forecasting of Rainfall in Central Java using Hybrid GSTAR-NN-PSO Model

Hasbi Yasin<sup>1,\*</sup>, Budi Warsito<sup>1</sup>, Rukun Santoso<sup>1</sup>, and Arief Rachman Hakim<sup>1</sup>

<sup>1</sup> Department of Statistics, Faculty of Science and Mathematics, Diponegoro University, Semarang – Indonesia

**Abstract.** Forecasting of rainfall trends is essential for several fields, such as airline and ship management, flood control and agriculture. The rainfall data were recorded several time simultaneously at a number of locations and called the space-time data. Generalized Space Time Autoregressive (GSTAR) model is one of space-time models used to modeling and forecasting the rainfall. The aim of this research is to propose the nonlinear space-time model based on hybrid of GSTAR, Feed Forward Neural Network (FFNN) and Particle Swarm Optimization (PSO) and it called GSTAR-NN-PSO. In this model, input variable of the FFNN was obtained from the GSTAR model. Then use PSO to initialize the weight parameter in the FFNN model. This model is applied for forecasting monthly rainfall data in Jepara, Kudus, Pati and Grobogan, Central Java, Indonesia. The results show that the proposed model gives more accurate forecast than the linear space-time model, i.e. GSTAR and GSTAR-PSO. Moreover, further research about space-time models based on GSTAR and Neural Network is needed to improving the forecast accuracy especially the weight matrix in the GSTAR model.

Keywords: GSTAR; FFNN; PSO; Rainfall.

## 1 Introduction

Rainfall is the amount of water that falls on a flat surface during a certain period measured in units of millimeters (mm) above the horizontal surface. In other explanations rainfall can also be interpreted as the height of rainwater collected in a flat place, not evaporating, not seeping and not flowing. Rainfall forecasting trends is essential for some areas, such as flood control and agriculture, airline and ship management. The rainfall data were recorded several time simultaneously at a number of locations and called the space-time data [1]. The challenge in forecasting space-time series data is that there is an increase in error rates almost every time.

Generalized Space Time Autoregressive (GSTAR) is one of the space-time models that can be used to model and forecast data that has previous time relationships and linkages to adjacent locations. This model is a development of the Space Time Autoregressive (STAR) model which was introduced by Pfeifer and Deutsch in 1980 [2]. Several previous studies related to GSTAR were done by Irawati, et al. [3], Gusnadi, et al. [4], Diani et al. [5], and Caraka et al. [6, 7].

The implementation of artificial neural networks is developing rapidly, especially in the forecasting problem of the Spatio-temporal model. Neural networks are good methods for predicting non-linear functions [8]. This method does not require assumptions from the data being modeled. This model is very appropriate for solving problems where prior information is not easy to define. Therefore, neural networks are non-linear and non-

parametric methods in representing functional relationships in data analysis [9]. Studies on comparing the performance between non-linear spatio-temporal model, i.e. FFNN-VAR and FFNN-GSTAR, and the linear model, i.e. VAR and GSTAR has been done by Suhartono et al. [10]. The results show that the non-linear is more accurate than the linear model.

The selection method of optimization is very important to get the optimal weight parameters. The gradient-based optimization method will require that the activation function must be continuous and can be distinguished because the weight update process is carried out through calculations using the derivative function [11]. Various classical gradient-based algorithms for training neural networks have been created in recent years [12-14]. However, some of them can remain trapped in unwanted solutions. Optimization methods based on Gradient are often unable to get the global optimum and are often trapped in the local optimum at the initial stage. Therefore, another type of technique was developed for neural network training. Many heuristic algorithms are then overcome to solve the problem. Heuristic techniques utilize knowledge-specific problems to find efficient solutions [15]. In this research, the Particle Swarm Optimization (PSO) algorithm is combined with the Back Propagation algorithm to optimizing the GSTAR model to forecast the rainfall data at Central Java, Indonesia.

<sup>3</sup>  
\* Corresponding author: [hasbiyasin@live.undip.ac.id](mailto:hasbiyasin@live.undip.ac.id)

## 2 Methods

### 2.1 Generalized Space Time Autoregressive (GSTAR) Model

GSTAR is one of spatio-temporal models that can explain response variables from past data of the variables and the spatial effect of these variables. Let  $\{Z(t): t=0, \pm 1, \pm 2, \dots\}$  is a space-time data of  $N$  locations, then GSTAR with autoregressive order  $p$  and spatial order  $\lambda_s$  or GSTAR  $(p; \lambda_s)$  in matrix notation can be written as follows:

$$\mathbf{Z}(t) = \sum_{s=1}^p \left( \Phi_{s0} + \sum_{k=1}^{\lambda_s} \Phi_{sk} \mathbf{W}^{(k)} \right) \mathbf{Z}(t-s) + \mathbf{a}(t) \quad (1)$$

where  $\mathbf{a}(t)$  is residual model,  $\Phi_{s0} = \text{diag}(\phi_{s0}^1, \dots, \phi_{s0}^N)$ , and  $\Phi_{sk} = \text{diag}(\phi_{sk}^1, \dots, \phi_{sk}^N)$  [2]. For example, GSTAR model with time order one and spatial order one for four locations is as follows :

$$\mathbf{Z}(t) = \Phi_{10} \mathbf{Z}(t-1) + \Phi_{11} \mathbf{W}^{(1)} \mathbf{Z}(t-1) + \mathbf{a}(t) \quad (2)$$

and in matrix form, it can be written as follows:

$$\begin{bmatrix} Z_1(t) \\ Z_2(t) \\ Z_3(t) \\ Z_4(t) \end{bmatrix} = \begin{bmatrix} \phi_{10} & 0 & 0 & 0 \\ 0 & \phi_{20} & 0 & 0 \\ 0 & 0 & \phi_{30} & 0 \\ 0 & 0 & 0 & \phi_{40} \end{bmatrix} \begin{bmatrix} Z_1(t-1) \\ Z_2(t-1) \\ Z_3(t-1) \\ Z_4(t-1) \end{bmatrix} + \begin{bmatrix} \phi_{11} & 0 & 0 & 0 \\ 0 & \phi_{21} & 0 & 0 \\ 0 & 0 & \phi_{31} & 0 \\ 0 & 0 & 0 & \phi_{41} \end{bmatrix} \begin{bmatrix} 0 & \omega_{12} & \omega_{13} & \omega_{14} \\ \omega_{21} & 0 & \omega_{23} & \omega_{24} \\ \omega_{31} & \omega_{32} & 0 & \omega_{34} \\ \omega_{41} & \omega_{42} & \omega_{43} & 0 \end{bmatrix} \begin{bmatrix} Z_1(t-1) \\ Z_2(t-1) \\ Z_3(t-1) \\ Z_4(t-1) \end{bmatrix} + \begin{bmatrix} a_1(t) \\ a_2(t) \\ a_3(t) \\ a_4(t) \end{bmatrix}$$

Borovkova et al. [2] state that some of the spatial weight matrices normally used in the GSTAR model are uniform weights, weights based on inverse distances between locations, and weights based on normalization of cross-correlation inference. In general, the number of parameters in VAR is greater than the GSTAR model.

### 2.2 Feed Forward Neural Network (FFNN)

Normally, the Feed Forward Neural Networks (FFNN) model is developed based on the gradient descent training algorithm. On FFNN, neurons are arranged in at least three layers, i.e. input layer, hidden layer, and output layer. Based on the result of the GSTAR model as an input layer, the FFNN formula can be written as follows:

$$\mathbf{Z} = \psi_2 \left\{ v_0 + \sum_{i=1}^q v_i \psi_1 \left( w_0 + \sum_{j=1}^p w_{ij} X_j \right) \right\} \quad (3)$$

where  $(w_0, w_{ij})_{i=1,2,\dots,p}$  and  $j=1,2,\dots,q$  are the weight that connected the input layer and hidden layer,  $(v_0, v_i)$  are the that connected the hidden layer and the output layer, and  $(\psi_1, \psi_2)$  are the activation function [16]. In this research  $\psi_1$  is the 'logistic sigmoid' function, and  $\psi_2$  is the 'linear' activation function. Before the artificial neural network training process, it is needs to transform the data in a certain range. This is needed so that the data is processed according to the activation function used, or also called Pre-Processing. Then, the data is returned to its original form, after the training process is finished, or named Post Processing [9].

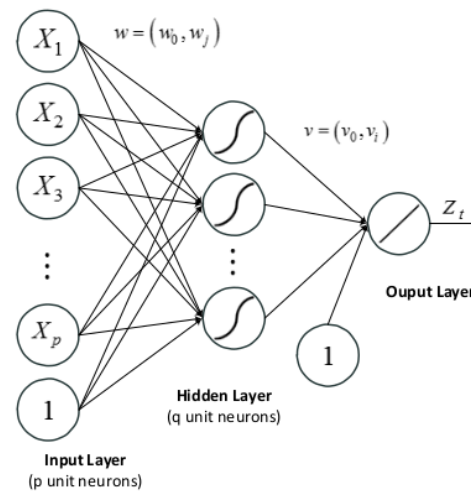


Fig. 1. The rchitecture of FFNN with p neuron in the input layer, q neuron in the hidden layer.

### 2.3 Particle Swarm Optimization (PSO)

Swarm Intelligent (SI) is a discipline that deals with natural and artificial systems that are composed of many individuals who coordinate using decentralized control and can organize themselves (self-organization) [17]. Specifically, these disciplines focus on collective behavior that results from local interactions between one individual and another individual and between individuals and their environment. Swarm Intelligent Logic (SI) then inspired most researchers to develop algorithms based on animal intelligence, especially to overcome optimization problems. Various algorithms regarding collective behavior are proposed to overcome optimization problems. There are two types of optimal values, namely global optimum and local optimum. Most optimization algorithms are created with the aim of finding the optimum global value.

One of the Intelligent Swarm (SI) algorithms is Particle Swarm Optimization (PSO) [18]. Particle Swarm Optimization (PSO) is a population optimization technique developed by James Kennedy and Russ Eberhart in 1995. PSO is inspired by social behavior in flocks of birds that flock or in groups of fish that swim in groups to form certain formations without colliding one each other even though the distance between individuals is so close [17]. Each bird is considered as a particle that represents the solution of a problem that has position ( $x$ ) and velocity ( $v$ ). Particle Swarm Optimization (PSO) has two main functions, namely the speed update and position update which is used to update the particle speed and position so that it will continue to be accelerated until it approaches the best particle position of the particle (local best/  $pBest$ ) and the best particle of the whole herd (global best/  $gBest$ ) until the minimum error condition is reached. PSO starts with a collection of particles (solutions) that are generated randomly. The following is the PSO calculation formula:

$$v_i^{t+1} = \rho \cdot v_i^t + c_1 \cdot r_1 (pBest_i^t - x_i^t) + c_2 \cdot r_2 (gBest_g^t - x_i^t)$$

$$x_i^{t+1} = x_i^t + v_i^t$$

where  $\rho$  is inertia weight,  $c_1$  and  $c_2$  are velocity coefficients,  $r_1$  and  $r_2$  are random values from uniform distribution. Symbols of  $pBest_i^t$  describe the best position of  $i^{th}$  particle at  $t^{th}$  iteration, whereas  $x_i^t$  is the particle  $i^{th}$  position at  $t^{th}$  iteration.

### 2.4 Hybrid GSTAR-NN-PSO Model

In this research, the hybrid space-time model is proposed to forecasting rainfall data in Jepara, Kudus, Pati and Grobogan, Central Java, Indonesia. This model combine the GSTAR, FFNN and the PSO algorithm to optimize the weight of GSTAR-NN. The architecture of GSTAR-NN-PSO that be used for forecasting the rainfall data could be seen in Figure 2. The algorithm of the proposed model is as follows:

**Algorithm:** Parameter Estimate of GSTAR-NN-PSO using hybrid PSO-BP

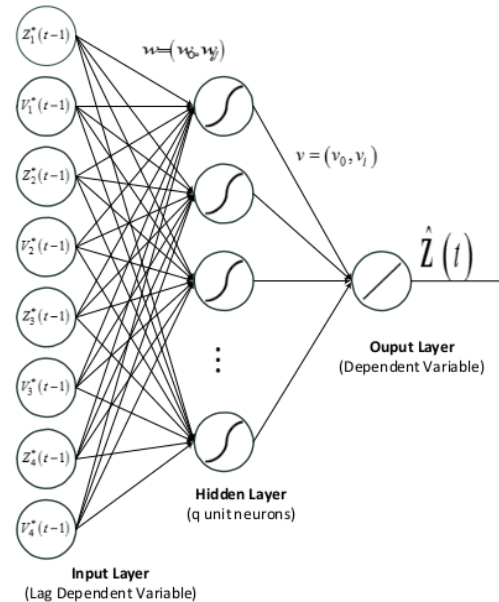
**Input:**  
 Rainfall Data at four location  
 Number of particles in PSO ( $N$ )  
 Number of neuron in hidden layer of FFNN,  
 Activation function of FFNN ( $\psi_o, \psi_k$ )

**Output:**  
 Parameter of GSTAR-NN ( $w_0, w_{ij}, v_o, v_i$ )

1. Identify the input variable using GSTAR model.
2. Construct the FFNN model based on input variable, number of neurons in hidden layer, and the activation function.
3. Pre-processing Data using the  $zscore$  transformation.
4. Initialize the weight parameter ( $w_0, w_{ij}, v_o, v_i$ ) based on the number of

input variable and the number of neuron in hidden layer as much of number of particle in PSO using the random value of (-1,1).

5. **Repeat:**
  1. Evaluate the fitness value of the particle in PSO using the Root Mean Squared Error (RMSE).
  2. Update the  $pBest$  (best position) and  $gBest$  (global optimal).
  3. Update the velocity of a particle.
6. **Until** fitness Convergence
7. Use the weight parameter ( $w_0, w_{ij}, v_o, v_i$ ) as an initial value of the weight in FFNN model using Backpropagation algorithm.
8. Post-processing Data
9. **Return** ( $w_0, w_{ij}, v_o, v_i$ )



**Fig. 2.** The architecture of GSTAR-NN with 8 input neurons as input layers,  $q$  neurons in the hidden layer, and one neuron in the output layer from one dependent variable.

The Figure 2 shows that GSTAR-NN uses only one dependent variable  $\hat{Z}(t)$ . In general, the GSTAR-NN model with  $p$  neurons in input layer,  $q$  neurons in hidden layer can be written as follows:

$$\hat{Z}_t = \psi_2 \left\{ v_0 + \sum_{i=1}^q v_i \psi_1 \left( w_0 + \sum_{j=1}^p w_{ij} X_{ij} \right) \right\} \quad (5)$$

where  $X_t = \begin{Bmatrix} Z_1^*(t-1), Z_2^*(t-1), Z_3^*(t-1), Z_4^*(t-1), \\ V_1^*(t-1), V_2^*(t-1), V_3^*(t-1), V_4^*(t-1) \end{Bmatrix}$  are the input variable of GSTAR,

$(w_0, w_{ij}) i=1,2,\dots,p$  and  $j=1,2,\dots,q$  are the weight that connected the input layer and hidden layer,  $(v_0, v_i)$  are the that connected the hidden layer and the output layer, and  $(\psi_1, \psi_2)$  are the activation function. In this research  $\psi_1$  is the 'logistic sigmoid' function, and  $\psi_2$  is the 'linear' activation function.

This output as a vector that consists of rainfall data at four locations i.e. Jepara, Kudus, Pati and Grobogan in Central Java, Indonesia. The explanation about notation in figure 2 can be written as follows:

$$\hat{Z}_t = \begin{bmatrix} \hat{Z}_1(t) \\ \hat{Z}_2(t) \\ \hat{Z}_3(t) \\ \hat{Z}_4(t) \end{bmatrix},$$

$$Z_1^*(t-1) = \begin{bmatrix} Z_1(t-1) \\ 0 \\ 0 \\ 0 \end{bmatrix}, Z_2^*(t-1) = \begin{bmatrix} 0 \\ Z_2(t-1) \\ 0 \\ 0 \end{bmatrix},$$

$$Z_3^*(t-1) = \begin{bmatrix} 0 \\ 0 \\ Z_3(t-1) \\ 0 \end{bmatrix}, Z_4^*(t-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ Z_4(t-1) \end{bmatrix},$$

$$V_1^*(t-1) = \begin{bmatrix} V_1(t-1) \\ 0 \\ 0 \\ 0 \end{bmatrix}, V_2^*(t-1) = \begin{bmatrix} 0 \\ V_2(t-1) \\ 0 \\ 0 \end{bmatrix},$$

$$V_3^*(t-1) = \begin{bmatrix} 0 \\ 0 \\ V_3(t-1) \\ 0 \end{bmatrix}, V_4^*(t-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ V_4(t-1) \end{bmatrix},$$

$$V_1(t-1) = \omega_{12}Z_2(t-1) + \omega_{13}Z_3(t-1) + \omega_{14}Z_4(t-1)$$

$$V_2(t-1) = \omega_{21}Z_1(t-1) + \omega_{23}Z_3(t-1) + \omega_{24}Z_4(t-1)$$

$$V_3(t-1) = \omega_{31}Z_1(t-1) + \omega_{32}Z_2(t-1) + \omega_{34}Z_4(t-1)$$

$$V_4(t-1) = \omega_{41}Z_1(t-1) + \omega_{42}Z_2(t-1) + \omega_{43}Z_3(t-1)$$

The training process to estimate the optimal weights are done by the hybrid PSO and Back Propagation algorithm. In this research, Matlab Code is developed and used for optimization to find the optimal weights in GSTAR-NN.

The steps of estimation GSTAR-NN-PSO model in this research are as follows:

1. Specify the input variable using GSTAR model
2. Specify the number of neurons in hidden layers
3. Initialize all weights in all layers using PSO
4. Calculate the output value in the hidden layer with logistic sigmoid activation function
5. Update all weights until convergence
6. Determine the GSTAR-NN-PSO prediction accuracy using RMSE to obtain the best model

## 7. Forecast rainfall using the best model

To compute the GSTAR-NN model, the Matlab code is developed to train the network and find the optimal weights.

## 3 Results

### 3.1 Description of Rainfall Data

In this paper, the monthly rainfall data at Jepara, Kudus, Pati, and Grobogan are used to develop the GSTAR-NN. The data consist of 96 observation at every location. Rainfall data were collected from January 2010 until December 2017. The data is divided into two parts, i.e. 84 for in sample dataset and 12 for out sample dataset. The time series of the data is shown in Figure 3.

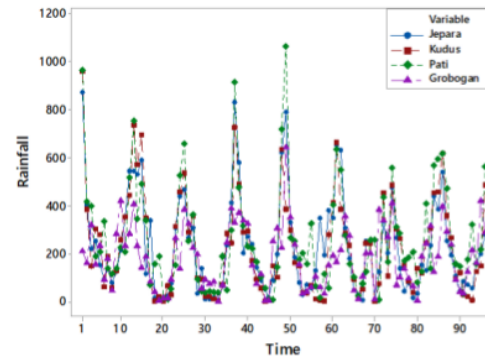


Fig. 3. Time Series Plot of Rainfall Data at Jepara, Kudus, Pati, and Grobogan.

### 3.2 Identification the Input Variable in GSTAR

Figure 3 shows that in general, the rainfall patterns of the four locations are relatively the same. This allows the tendency of the effects of interrelationships between locations one to another. Based on the MACF and MPACF plots, it can be seen that the data contains seasonal patterns. Therefore it is necessary to do differencing on lag 12. Then based on the AIC value it is obtained that the best GSTAR model is GSTAR(1)<sub>I</sub>(1)<sup>12</sup>. Therefore, the model is as follows:

$$\dot{Z}(t) = \Phi_{10}\dot{Z}(t-1) + \Phi_{11}W^{(1)}\dot{Z}(t-1) + a(t) \quad (6)$$

where  $\dot{Z}(t) = Z(t) - Z(t-12)$ , and  $W^{(1)}$  is the weight matrix, i.e.:

if  $W^{(1)}$  is Uniform Weight,

$$W^{(1)} = \begin{bmatrix} 0 & 0.33 & 0.33 & 0.33 \\ 0.33 & 0 & 0.33 & 0.33 \\ 0.33 & 0.33 & 0 & 0.33 \\ 0.33 & 0.33 & 0.33 & 0 \end{bmatrix};$$

if  $W^{(1)}$  is Invers Distance Weight,

$$\mathbf{W}^{(1)} = \begin{bmatrix} 0 & 0.35 & 0.31 & 0.34 \\ 0.35 & 0 & 0.31 & 0.34 \\ 0.34 & 0.35 & 0 & 0.31 \\ 0.35 & 0.36 & 0.29 & 0 \end{bmatrix};$$

if  $\mathbf{W}^{(1)}$  is Normalized Cross-Correlation Weight,

$$\mathbf{W}^{(1)} = \begin{bmatrix} 0 & 0.41 & 0.30 & 0.29 \\ 0.36 & 0 & 0.26 & 0.38 \\ 0.32 & 0.41 & 0 & 0.27 \\ 0.35 & 0.34 & 0.31 & 0 \end{bmatrix}.$$

Using the 84 observations as in-sample dataset, the best GSATR model is the GSTAR model with a normalized cross-correlation weighting matrix because it has a minimum RMSE value.

Based on Equation 6 it can be obtained that the input variables used are:

$$\begin{aligned} \hat{\mathbf{Z}}(t) &= \Phi_{10} \hat{\mathbf{Z}}(t-1) + \Phi_{11} \mathbf{W}^{(1)} \hat{\mathbf{Z}}(t-1) + \mathbf{a}(t) \\ \mathbf{Z}(t) - \mathbf{Z}(t-12) &= \Phi_{10} (\mathbf{Z}(t-1) - \mathbf{Z}(t-13)) \\ &\quad + \Phi_{11} \mathbf{W}^{(1)} (\mathbf{Z}(t-1) - \mathbf{Z}(t-13)) + \mathbf{a}(t) \\ \mathbf{Z}(t) &= \mathbf{Z}(t-12) + \Phi_{10} (\mathbf{Z}(t-1) - \mathbf{Z}(t-13)) \\ &\quad + \Phi_{11} \mathbf{W}^{(1)} (\mathbf{Z}(t-1) - \mathbf{Z}(t-13)) + \mathbf{a}(t) \end{aligned}$$

Therefore, the input variables used in the GSTAR-NN model are:

$$\begin{aligned} \mathbf{Z}^*(t-12) &= \begin{bmatrix} \mathbf{Z}_1^*(t-12) \\ \mathbf{Z}_2^*(t-12) \\ \mathbf{Z}_3^*(t-12) \\ \mathbf{Z}_4^*(t-12) \end{bmatrix}^T, \\ \mathbf{Z}^*(t-1) &= \begin{bmatrix} \mathbf{Z}_1^*(t-1) \\ \mathbf{Z}_2^*(t-1) \\ \mathbf{Z}_3^*(t-1) \\ \mathbf{Z}_4^*(t-1) \end{bmatrix}^T, \mathbf{Z}^*(t-13) = \begin{bmatrix} \mathbf{Z}_1^*(t-13) \\ \mathbf{Z}_2^*(t-13) \\ \mathbf{Z}_3^*(t-13) \\ \mathbf{Z}_4^*(t-13) \end{bmatrix}^T, \\ \mathbf{V}^*(t-1) &= \begin{bmatrix} \mathbf{V}_1^*(t-1) \\ \mathbf{V}_2^*(t-1) \\ \mathbf{V}_3^*(t-1) \\ \mathbf{V}_4^*(t-1) \end{bmatrix}^T \text{ and } \mathbf{V}^*(t-13) = \begin{bmatrix} \mathbf{V}_1^*(t-13) \\ \mathbf{V}_2^*(t-13) \\ \mathbf{V}_3^*(t-13) \\ \mathbf{V}_4^*(t-13) \end{bmatrix}^T \end{aligned}$$

### 3.3 GSTAR-NN Model

Based on input variable of GSTAR model in Equation 6 the GSTAR-NN in this data can be written as follows:

$$\hat{\mathbf{Z}}_t = \psi_2 \left\{ v_0 + \sum_{i=1}^q v_i \psi_1 \left( w_0 + \sum_{j=1}^p w_{ij} \mathbf{X}_{tj} \right) \right\}$$

$$\mathbf{X}_t = \{ \mathbf{Z}^*(t-1), \mathbf{Z}^*(t-12), \mathbf{Z}^*(t-13), \mathbf{V}^*(t-1), \mathbf{V}^*(t-13) \}$$

The network architecture of this model to develop base on the GSTAR-NN at Figure 2. In this study, their weighting functions are used to find the best GSTAR-NN model. The selection of the best model is based on the minimum value of RMSE.

In each simulation, the training process is carried out for a maximum of 1000 iterations in the PSO and in each iteration we generated 20 particles. Then the best particle of the PSO algorithm is used for the initials weights parameter in Backpropagation. Using the in sample dataset, the results of this experiment can be seen in Table 1.

**Table 1.** The Comparison Results of Forecast Accuracy between GSTAR, GSTAR-PSO, and GSTAR-NN-PSO at Rainfall Data

Model	RMSE at in sample dataset		
	Uniform	Inverse of Distance	normalization of cross-correlation
GSTAR	129.954	129.954	129.997
GSTAR-PSO	129.954	129.954	129.997
GSTAR-NN-PSO	124.736	126.677	<b>120.555</b>

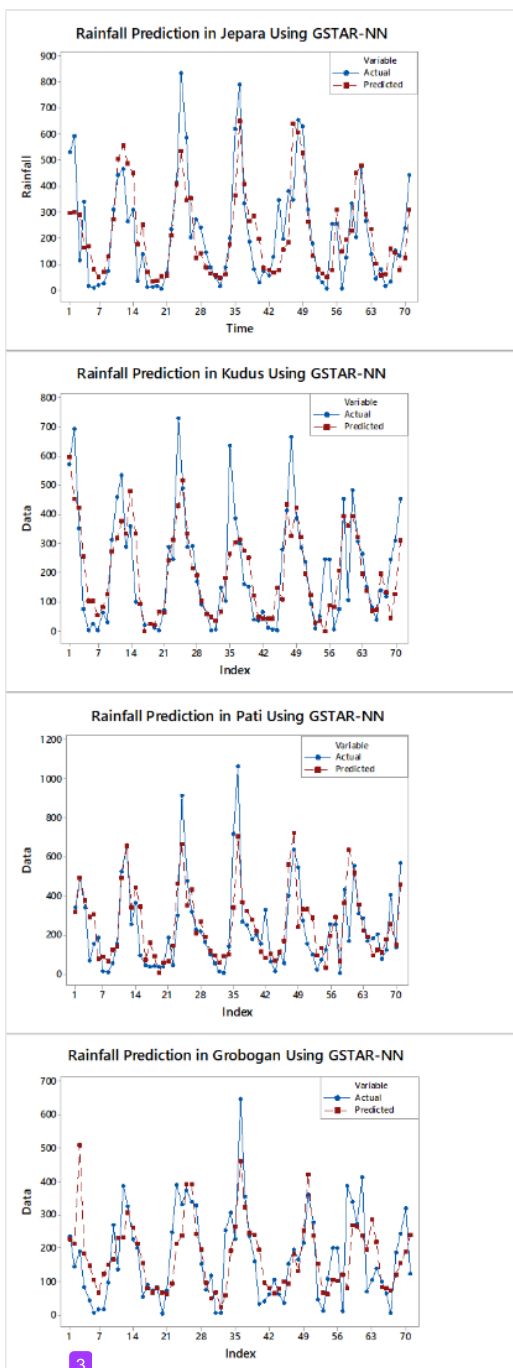
Furthermore, it could be concluded that in general GSTAR-NN-PSO as nonlinear space-time models give more accurate forecast than GSTAR and GSTAR-PSO as linear space-time models. These results are supported by the forecast of GSTAR-NN-PSO with the normalization cross-correlation weight scheme that tends to yield more accurate forecast at in sample dataset. Hence, the results also give new opportunities to do further research about NN and PSO for forecasting space time datasets.

The forecast results of out-sample dataset using the best model, i.e. GSTAR-NN-PSO with Normalization Cross-Correlation Weight, can be seen at figure 5. Based on this model, the RMSE of out-sample dataset is 174.608.

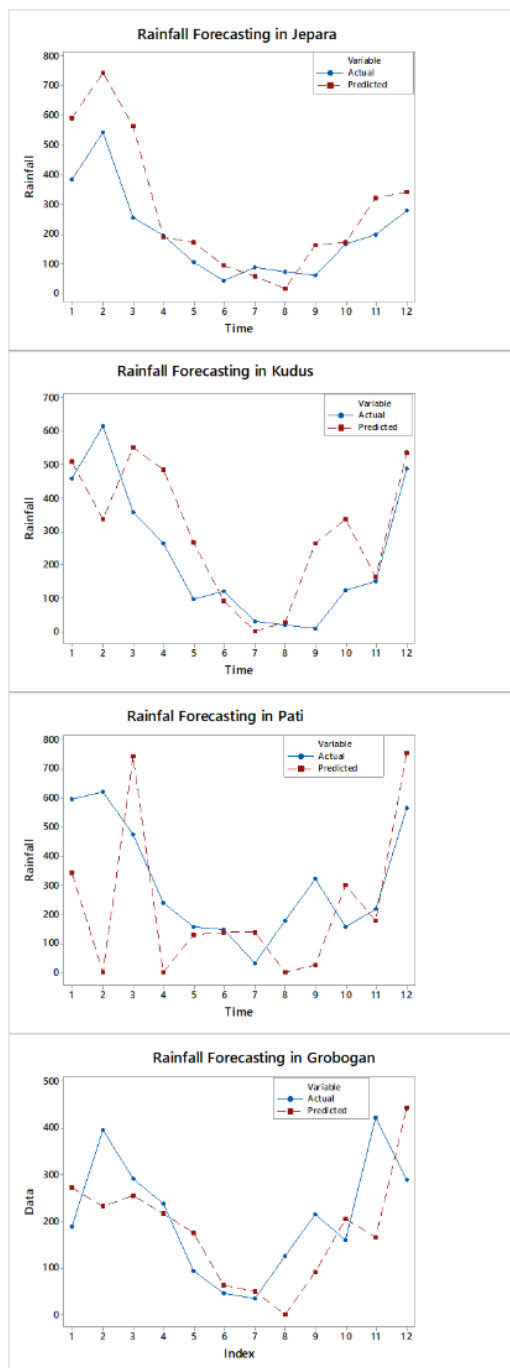
## 4 Conclusion

The present study was proposed nonlinear space-time model, i.e. GSTAR-NN-PSO, for forecasting rainfall data at Jepara, Kudus, Pati and Grobogan, Central Java, Indonesia. The empirical study concluded that this model could work well for estimating rainfall data. In addition, the results also show that GSTAR-NN-PSO tends to be more accurate estimates than GSTAR to forecast the rainfall data at Central Java. Future research is needed in determining the weighting function in GSTAR in order to obtain a more accurate space-time series model.

This research was funded by DRPM-DIKTI with the 2019 PDUPT scheme. The authors thank the Directorate General of Higher Education for its funding and support.



**Fig. 4.** Time Series Plot of Rainfall Data (Actual vs Predicted value) at Jepara, Kudus, Pati and Grobogan using GSTAR-NN-PSO with Normalization Cross Correlation Weight.



**Fig. 5.** Rainfall forecasting (Actual vs Predicted value) using GSTAR-NN-PSO with Normalization Cross Correlation Weight.

## References

1. [BMKG Semarang] Badan Meteorologi Klimatologi dan Geofisika Stasiun Klimatologi Semarang (2019)
2. S.A. Borovkova, H.P. Lopuhaa, and B.N. Ruchjana, Proceedings of the 17<sup>th</sup> International Workshop on Statistical Modelling, (2002)
3. L. Irawati, Tarno, H. Yasin, Jurnal Gaussian, **4**, 3, 553–562 (2015)
4. R. Gusnadi, R. Rahmawati, A. Prahutama, Jurnal Gaussian, **4**, 4, 1017-1026 (2015)
5. K.A.N. Diani, Setiawan, Suhartono, Jurnal Sains dan Seni POMITS **2**, 1 (2013)
6. R.E. Caraka et al., Proceedings of the Indonesian Association for Pattern Recognition Conference, (2018)
7. R.E. Caraka et al., Proceedings of the International Conference on Information Management and Technology (ICIMTech), (2018)
8. R.E. Caraka, S.A. Bakar, M. Tahmid, H. Yasin, I.D. Kurniawan, *TELKOMNIKA* **17**, 4, 1818-1827 (2019)
9. B. Warsito, *Kapita Selektta Statistika Neural Network*, (BP Undip Semarang, 2009)
10. Suhartono, D.D. Prastyo, H. Kuswanto, M.H. Lee, *MATEMATIKA* **34**, 1, 103-111 (2018)
11. B. Warsito, H. Yasin, A. Prahutama J. Phys.: Conf. Ser., **1217**, 012101 (2019)
12. B.A. Garro, R.A. Vázquez, *Computational intelligence and neuroscience*, **61** (2015)
13. C.B. Khadse, M.A. Chaudhari, V.B. Borghate International Journal of Electrical Power & Energy Systems, **82** pp.197-206 (2016)
14. S. Chattopadhyay, G. Chattopadhyay Journal of Atmospheric and Solar-Terrestrial Physics, **179** pp. 202-205 (2018)
15. G.P.P. Potdar, R.C. Thool Journal of Artificial Intelligence & Applications (IJAA), **5**, 4, pp 63-74 (2014)
16. S. Kusumadewi, *Membangun Jaringan Syaraf Tiruan Menggunakan Matlab dan Excel Link*, (Graha Ilmu, Yogyakarta, 2004)
17. Suyanto, *Algoritma Optimasi Deterministik atau Probabilistik*, (Graha Ilmu Yogyakarta, 2014)
18. B. Santosa, T.J. Ai *Pengantar Metaheuristik Implementasi dengan Matlab*, (ITS Tekno Sains Surabaya, 2017)



ORIGINALITY REPORT

---

<b>11</b> %	<b>6</b> %	<b>11</b> %	<b>4</b> %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

---

PRIMARY SOURCES

---

<b>1</b>	<b>B Warsito, H Yasin, A Prahutama. "Particle swarm optimization versus gradient based methods in optimizing neural network", Journal of Physics: Conference Series, 2019</b> Publication	<b>4</b> %
<b>2</b>	<b>Submitted to Universitas Brawijaya</b> Student Paper	<b>4</b> %
<b>3</b>	<b>Hasbi Yasin, Budi Warsito, Rukun Santoso, Suparti. "Soft Computation Vector Autoregressive Neural Network (VAR-NN) GUI-Based", E3S Web of Conferences, 2018</b> Publication	<b>3</b> %

---

Exclude quotes  On  
Exclude bibliography  On

Exclude matches  < 3%