Journal of Physics: Conference Series

PAPER • **OPEN ACCESS**

Rainfall and Wave Height Prediction in Semarang City Using *Vector Autoregressive Neural Network* (VAR-NN) Methods

To cite this article: Sugito *et al* 2019 *J. Phys.: Conf. Ser.* **1320** 012017

IOP Conf. Series: Journal of Physics: Conf. Series **1320** (2019) 012017 doi:10.1088/1742-6596/1320/1/012017

Rainfall and Wave Height Prediction in Semarang City Using *Vector Autoregressive Neural Network* (VAR-NN) Methods

Sugito[1], Mustafid[2], D Safitri[3], D Ispriyanti[4], A R Hakim[5] and H Yasin[6]

[1, 2, 3, 4, 5, 6] Department of Statistics, Faculty of Science and Mathematics, Diponegoro University
Jl. Prof. Soedharto, SH, Tembalang, Semarang 50275, Indonesia

E-mail: arief.rachman@live.undip.ac.id

Abstract. In the past few months in Semarang City, high rainfall and high waves give negative impact because the activities of fishermen to be stopped due to the waves of water. Another disadvantage is ROB flood. This is quite detrimental in addition to disrupting transportation and economic activities, but also causing damage to both government-owned assets and personal assets. This damage is caused by flood water which is mixed with sea water containing salt, so that it can cause corrosion. Vector of Autoregressive Neural Network (VAR-NN) is a development of the Autoregressive Vector (VAR) method using the Neural Network (NN) algorithm. The VAR method does not need to distinguish between endogenous and exogenous variables, its means that all variables used in the VAR model are used as endogenous variables. The VAR model can be used to explain and make predictions from endogenous variables from past data on these variables. VAR is developed using the Neural Network (NN) algorithm, where the basic principle of NN itself is an information processing algorithm that resembles the workings of the human brain, which uses a number of neurons to perform simple tasks. VAR-NN is often used in the economic, financial fields and believed to be able to make predictions of high rainfall and wave height. In this study using VAR-NN with FeedForward algorithm or called Backpropagatian. The results of rainfall prediction and wave height in Semarang city using this method have a Mean Square Error of 12,325 with the VAR-NN model (3, 2, 4).

1. Introduction

Extreme weather is a rare natural occurrence but if the resulting impact will be quite detrimental, extreme weather can be in the form of high rainfall or wave height. In the past few months in Semarang, high rainfall and high waves caused the activities of fishermen to stop due to water waves or waves above 2 meters which exceeded the safe limit of going to sea which is 1 to 1.5 meters. The updated weather information that is always provided by BMKG regarding wave height and recommended fishing has been the basis of fishermen in carrying out fishing activities.

In addition, some losses caused by the high rainfall and sea wave height in the city of Semarang are tidal floods. This is quite detrimental in addition to disrupting transportation and economic activities, but also causing damage to both government-owned assets and personal assets. This damage is caused by flood water which is mixed with sea water containing salt, so that it can cause corrosion. Research on the prediction of high rainfall and wave height is needed to be used as initial information, so that anticipatory steps can be taken.

One method that can be used to predict rainfall and wave height is the Vector Autoregressive Neural Network (VAR-NN), which is the development of the Autoregressive Vector (VAR) method using the Neural Network (NN) algorithm. Vector Autoregressive (VAR) is an econometric method that is useful for testing between variables in a model that has a dynamic impact [1]. VAR method does not need to distinguish between endogenous and exogenous variables, its means that all variables used in the VAR model are used as endogenous variables [2]. VAR method itself is widely used in the case of economic cases as previously done by Mauludiyanto (2009) [3] and Anggraeni (2008)[4]. Then VAR was developed using the Neural Network algorithm, with the aim of getting smaller forecasting errors using nonlinear modeling. This study discusses the use of VAR-NN to make predictions of the height of rainfall and wave height.

2. Literature review

2.1. Vector Autoregressive (VAR)

Vector Autoregressive (VAR) is a method that is included in the multivariate Time Series which can be used to explain and predict endogenous variables from past data, it also used as an econometric method for testing between variables in a model that has a dynamic impact [1]. In VAR there is no priority difference between endogenous and exogenous variables, where all variables are considered as endogenous variables [2]. Rosadi (2011) [5] describes the VAR with the order p meaning that the independent variable of the model is p the value of the non-independent variable lag. In general, the VAR equation can be expressed in the form of:

$$\mathbf{Y}_t = \boldsymbol{\varphi}_0 + \sum\nolimits_{i=1}^{p} \boldsymbol{\varphi}_i \mathbf{Y}_{t-i} + \boldsymbol{\varepsilon}_t$$

The parameter estimation of the VAR method is done using the Ordinary Least Square (OLS) method, so that the following equation is obtained:

$$\mathbf{Y} = \mathbf{A}\boldsymbol{\beta} + \varepsilon$$

Where

$$\mathbf{A} = \begin{bmatrix} 1 & Y_{(1-1)} & Y_{(1-2)} & \cdots & Y_{(1-p)} \\ 1 & Y_{(2-1)} & Y_{(2-2)} & \cdots & Y_{(2-p)} \\ 1 & Y_{(3-1)} & Y_{(3-2)} & \cdots & Y_{(3-p)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & Y_{(n-1)} & Y_{(n-2)} & \cdots & Y_{(n-p)} \end{bmatrix}_{n\times(p+1)}, \mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_n \end{bmatrix}_{n\times 1}, \boldsymbol{\beta} = \begin{bmatrix} \Phi_0 \\ \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_p \end{bmatrix}, \boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_0 \\ \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}_{n\times 1}$$

2.2. Neural Network (NN)

Neural Network (NN) is a learning machine built from a number of simple processing elements called neurons or nodes [6]. The principle of neural network modelling is developed from the characteristics and workings of the human brain, in which information processing of the human brain consists of a number of neurons that perform simple tasks. The relationship between neurons causes the human brain to perform very complex processing functions. In a simple picture of a human neural network system is presented in Figure 1

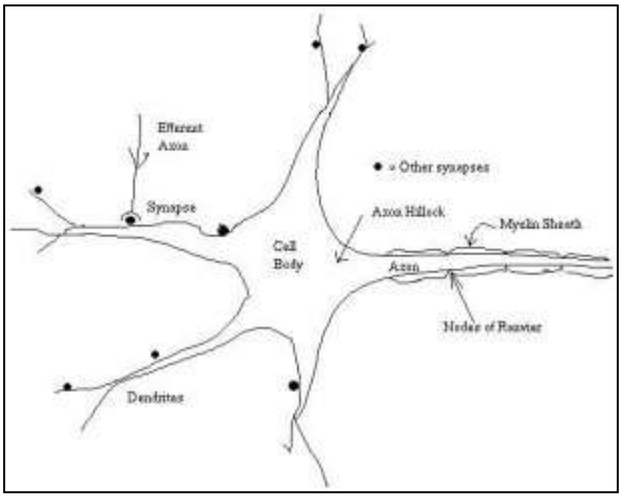

Figure 1. Human Neural Network System [7].

2.3. Backpropagation

Back propagation is an algorithm used in calculating gradient error functions related to weights on a network. There are 2 phases in the backpropagation algorithm, the first is the forward phase and the second is the backward phase. Calculation of weights in the forward phase started from the input unit to the output, using the predetermined activation function then getting the error value. Calculation in the backward phase, the error is propagated backwards from the output to the input unit, so that new weights that minimize errors (Siang, 2005) [8] are obtained. The architecture that is often used in the Back propagation Algorithm is a feed forward network (Feed Forward Neural Network / FFNN). The network architecture consists of one unit of input layer with the number of neurons / nodes p, one hidden layer with n units and one unit of output layer consisting of neurons / nodes can be written as the following neural network model:.

$$Y_t = \psi_0 \left\{ v_{bo} + \sum_{n=1}^{H} v_{no} \psi_n \left(w_{bn} + \sum_{j=1}^{p} w_{in} Y_{t-j} \right) \right\}$$

Where $(w_{bn}, w_{in}, v_{no}, v_{bo})$ is weighting parameters and (ψ_0, ψ_n) is activation function. Before training on artificial neural networks, it is necessary to scale the input and target data from certain data ranges. This is to make the data match the original activation function. This process is called Pre-Processing. After the training is complete, the data is taken to the original form (Post Processing). FFNN architecture image with 1 hidden layer, 3 neurons in the hidden layer, and 1 neuron at the output layer, as seen in Figure 2.

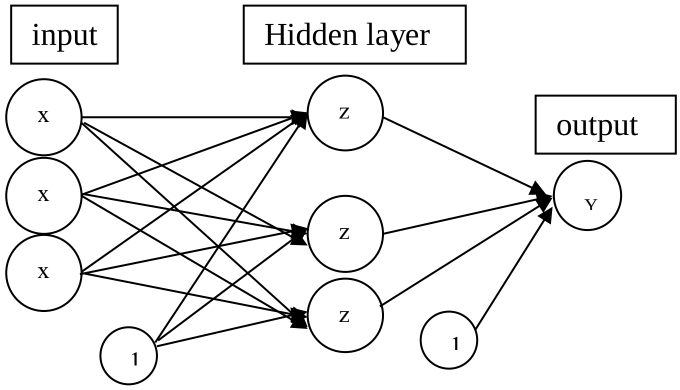

Figure 2. FFNN architecture image

2.4. Vector Autoregressive Neural Network

The VAR-NN model is a neural network model based on the input order in the VAR model. The architecture of this modelling has the number of neurons in the output layer as many as the multivariate series used.

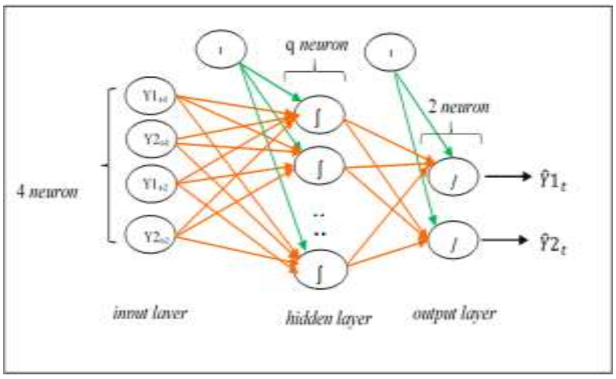

Figure 3. Architecture *VAR-NN*

As seen in Figure 3, architecture VAR-NN (2,2,q) with one hidden layer, 4 input neurons from 2 lag dependent variable, q neurons in hidden layer, and 2 neurons in output layer from 2 dependent variable, Suppose the weight of the input layer to the hidden layer in Figure 3 is denoted as $\mathbf{w}_{ij}$ for the weight of the i-neuron in the input layer to the-j neuron in the hidden layer. While the weight of the hidden layer to output is $\boldsymbol{\lambda}$ which denotes the weight of the j-neuron in the hidden layer to the neuron to k in the output layer. Then the weight matrix **w** and $\boldsymbol{\lambda}$ from Figure 3, will following as:

$$\mathbf{w} = \begin{bmatrix} w_{1,t-1,1} & w_{1,t-1,2} & \cdots & w_{1,t-1,h} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,t-p,1} & w_{1,t-p,2} & \cdots & w_{1,t-p,h} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,t-1,1} & w_{m,t-1,2} & \cdots & w_{m,t-1,h} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m,t-p,1} & w_{m,t-p,2} & \cdots & w_{m,t-p,h} \end{bmatrix} \quad \boldsymbol{\lambda} = \begin{bmatrix} \lambda_{1,1} & \lambda_{1,2} & \cdots & \lambda_{1,h} \\ \lambda_{2,1} & \lambda_{2,2} & \cdots & \lambda_{2,h} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_{m,1} & \lambda_{m,2} & \cdots & \lambda_{m,h} \end{bmatrix}$$

Example $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_h)'$ and $\boldsymbol{\beta} = (\beta_1, \beta_2, \ldots, \beta_m)'$ is a bias vector on input and hidden layer, then the output of the VAR-NN model above can be written as follows :

$$\mathbf{y}_t = \lambda F\left[(\mathbf{yw})' + \boldsymbol{\alpha}\right] + \boldsymbol{\beta} + \boldsymbol{\varepsilon}_t$$

With,

$$f((Yw)' + a) = \frac{1}{1 + \exp(-((Yw)' + a))}$$

3. Research Methods

The data used in this study is the daily rainfall intensity data and the daily wave height of the city of Semarang with the time span of August 2015 - August 2017. The steps taken in this study the first to divide data into in sample and out sample data, determine of lag dependent variable, define the number neuron in hidden layer, Initialize all weight in both of hidden layer and output layer, Calculate outputs gained from neurons in the hidden layer with logistic sigmoid activation function and from the neurons in the logistic sigmoid output layer. Calculate error gradient for the neurons and weight correction for the output layer and then update all weights. Calculate error gradient for the neurons and weight correction in hidden layer. For validation of predict data in sample and out sample using MSE, predict with the best model.

4. Result

This paper uses daily data of rainfall and wave height in Semarang city, then the data is divided into two parts, which are 90% divided into in sample data and 10% as sample data out used for data validation. The best VAR-NN model obtained using trial and error techniques is (3, 2, 4), then the performance evaluation of the model is based on the Mean Square Error (MSE) value of the sample out testing data that is equal to 12,325. VAR-NN model (3, 2, 4) with 3 dependent variable lags and 4 hidden layer neurons. Output can be seen in Figures 4, 5 and 6:

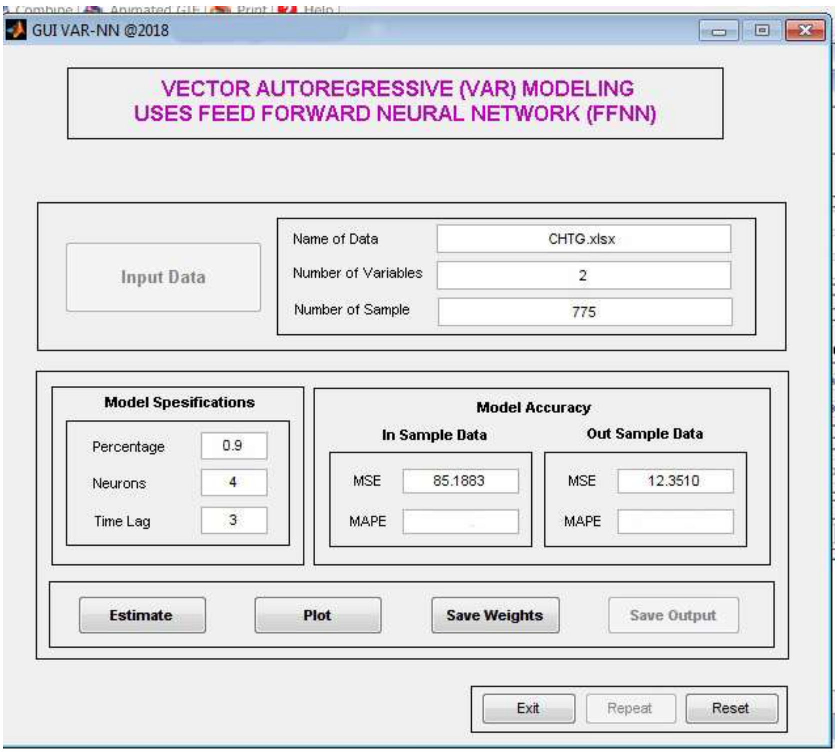

Fig 4. Output VAR NN

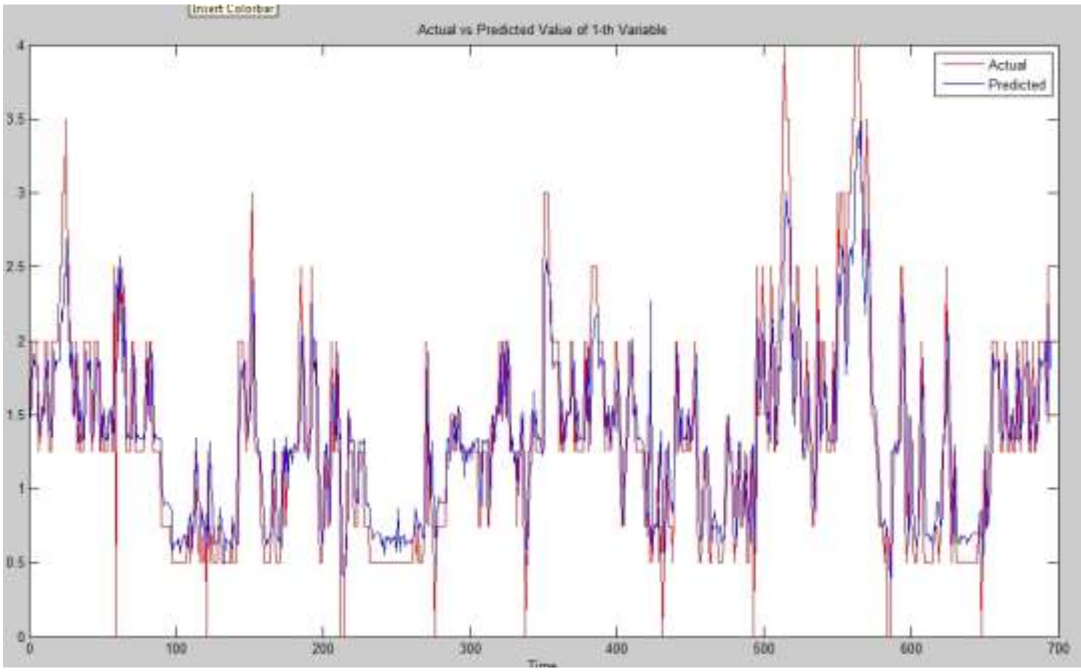

Figure 5. Time Series Plot of Actual vs Predicted Value of Daily Rainfall

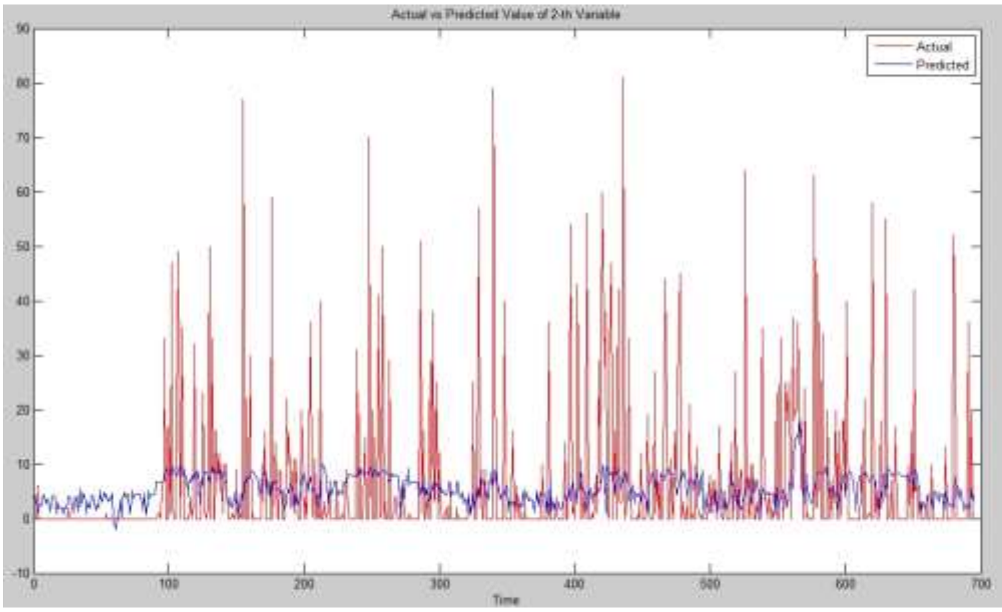

Figure 6. Time Series Plot of Actual vs Predicted Value of Daily Height Wave

From figures 5 and 6 it can be seen that the prediction results with testing data using VAR-NN for rainfall variables are quite accurate, this is sufficient to be seen in figure 5 with a plot that almost matches between testing data and predictive testing data. But for wave height predictions with testing data the results are inaccurate because it is quite far from the plot difference between wave height testing data with wave height prediction.

After selecting the VAR-NN model that is VAR-NN model (3, 2, 4), then the model can then be used to predict rainfall and wave height. Prediction results using VAR-NN using out-sample / testing data are presented in table 1

Table 1. Prediction results using VAR-NN using out-sample / testing data

No	Prediction		No	Prediction		No	Prediction	
	Rainfall	Height wave		Rainfall	Height wave		Rainfall	Height wave
1	1.805	5.093	28	1.297	3.925	55	1.835	1.440
2	1.838	1.702	29	1.318	3.273	56	2.233	-0.073
3	1.832	1.597	30	1.426	3.335	57	2.581	-0.912
4	1.453	2.901	31	2.470	-0.086	58	1.851	3.997
5	1.477	2.570	32	2.587	-0.240	59	1.842	1.968
6	1.293	2.771	33	2.244	2.656	60	1.835	1.440
7	2.215	2.174	34	2.230	1.251	61	1.835	1.440
8	2.169	2.027	35	1.833	2.430	62	2.233	-0.073
9	2.233	0.553	36	1.468	3.244	63	2.217	0.560
10	1.847	3.326	37	1.373	2.071	64	2.224	0.887
11	1.836	2.437	38	1.845	6.345	65	1.830	2.544
12	1.533	0.671	39	1.743	5.497	66	1.459	3.167
13	1.394	8.131	40	1.835	1.440	67	1.477	2.570
14	1.392	6.096	41	1.835	1.440	68	1.273	3.316
15	1.415	3.189	42	1.453	2.901	69	1.470	2.814
16	0.715	5.020	43	1.308	2.916	70	1.419	3.070
17	1.391	4.744	44	1.865	2.364	71	1.453	2.709
18	1.605	4.725	45	2.167	0.823	72	1.273	3.316
19	2.178	0.269	46	1.822	2.189	73	1.470	2.814
20	1.822	2.189	47	1.839	1.679	74	1.242	3.622
21	1.839	1.679	48	1.835	1.440	75	1.283	3.427
22	1.453	2.901	49	1.835	1.440	76	1.426	3.335
23	1.477	2.570	50	1.835	1.440	77	1.419	3.070
24	1.453	2.709	51	1.835	1.440			
25	1.453	2.709	52	1.453	2.901			
26	1.273	3.316	53	1.880	1.099			
27	2.238	0.312	54	1.817	1.390			

5. Conclusion

In this study, the VAR-NN model (3, 2, 4) was obtained by trying and error, MSE obtained by 12,325. However, if you see MSE from the sample, it is quite large at 85,188. The VAR-NN model with FFNN algorithm is not appropriate to be used in this research case, for future research it can also use alternatives such as the Neural Network Radial Base Function (RBFNN) algorithm.

Reference

[1] O. Ozcelebi, *International Jurnal of Economics and Finance* **3**, 130-139 (2011)
[2] Gujarati, D.N. 2004. *Basic Ekonometrics, Fourth Edition.* New York: The McGraw-Hill.
[3] Mauludiyanto, A., Hendrantoro, G., Hery, M., and Suhartono. 2009. *Pemodelan VARIMA Dengan Efek Deteksi Outlier Terhadap Data Curah Hujan*.

[4] Anggraeni, W and Dewi, L.K. 2008. *Peramalan menggunakan MetodeVector Autoregressive Moving Average* (VARMA). Vol 7, No 2, July 2008 : 99–106.
[5] Rosadi, D. 2011. *Analisis Ekonometrik & Runtun Waktu Terapan dengan R*. Yogyakarta: Andi
[6] Fausett, L. 1994. *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*. London: Prentice Hall, Inc.
[7] Warsito, B. 2009. Kapita Selekta Statistika Neural Network. Semarang. BP Undip Semarang
[8] Siang, J.J. 2005. *Jaringan Syaraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: Andi.