Forecasting of Vannamei Shrimp Production Based on Weather Factors Using Radial Basis Function Neural Network Approach (Case Study: Lamongan District)

M U Albab¹, M I Irawan¹ and D Adzkiya¹

¹Department of Mathematics, Institut Teknologi Sepuluh Nopember Surabaya, Sukolilo-Surabaya 60111, Indonesia

Email: mululalbab06@gmail.com

Abstract. One of the economic activity sectors that can be influenced by uncertain weather conditions is aquaculture production. In Lamongan, aquaculture production especially vannamei shrimp is very dependent on the ideal weather conditions. Uncertainty of the weather conditions can cause irregular harvesting to the production of vannamei shrimp. These trend changes can have an impact on the activities of production supply chain, namely the fulfillment of the entry quota of vannamei shrimp production from agents or distributors to exporters of vannamei shrimp to meet market demand. This marketing result can increase the original income of Lamongan area. To find out the development of the trend required a forecasting process and appropriate classification based on past data using artificial neural networks. One structure of artificial neural networks that can be predicting and classifying is a radial basis function neural network (RBFNN). The structure of RBFNN is trained using K-means clustering and gradient descent method. We use average temperature, average humidity and rainfall each month starting from January 2013 until December 2017 as the actual datasets. From those datasets, the training datasets start from January 2013 until December 2016 and the remaining datasets are used as the testing datasets. Built in the Python program, the test results show that our forecasting and classification had an accuracy level with mean absolute percentage error (MAPE) 16.7%.

1. Introduction

Vannamei shrimp production is the commodity of aquaculture in Lamongan regency which has a high selling price in the market and can be harvested in a relatively fast time for 3 months. So it is very profitable for fisheries household from an economic perspective. One of the challenges faced by cultivators of vannamei shrimp is the weather parameters that exist in the neighborhood of cultivation environment. The current weather parameters have produced uncertain weather conditions that have an impact on human and potentially affect on the economic activities, especially capture fisheries sector [1] and aquaculture [2].

One of the aquaculture activities can be affected by the uncertain weather conditions that is the production of vannamei shrimp that experiences trend of upswings and downswings. These trends will have an impact on interrelated activities in the supply chain of vannamei shrimp conducted by business people, such as agents or distributors and exporters. Hence developing more realistic models to forecast the up and down trends and to classify in the vannamei shrimp production affected by
weather conditions is a big challenge for most of the aquaculture business people and professional analysts.

One of the most used methods for forecasting and classification problems is artificial neural networks (ANNs) which are the machine learning method is based on the functional structure of biological neural network. A review of the models and simulations to predict the commercial shrimp growth using the comparison between ANNs of multiple layers perceptron models and nonlinear regression models was performed in [3]. The result showed that ANNs of multiple layers perceptron models provided better descriptions of shrimp growth curves and accuracy than using nonlinear regression models. Irawan, et. al. [4] had applied a forecasting using ANN to obtain predictions on the climatological and hydrological data. The forecasting results are used to determine the minimum crop water requirements in one year and the time of cropping patterns in agriculture to maximize the benefits and minimize crop failure. The application of backpropagation neural network (BPNN) in water environment monitoring system for vannamei shrimp cultivation to study the relationship between the six factors of water environment and vannamei shrimp growth has been discussed in [5]. In contrast, the study comparing BPNN and radial basis function neural network (RBFNN) provided information about speed and accuracy test for small-signal models of active devices. From the computational time and accuracy, RBFNN is better than BPNN as shown in [6].

Sachan [7] had solved a problem of rainfall forecasting using global positioning system (GPS), meteorological data and RBFNN to get the magnitude of rainfall for the next 24 hours. The accuracy of RBFNN was more than 78%. Dubey [8] had modified and implemented RBFNN by using K-means clustering to provide the neural networks set of homogeneous data so the learning time is reduced. By using K-means clustering, the performance of RBFNN increases. Qiao, et. al. had conducted research on the application of RBFNN with K-means clustering algorithm for forecasting the hard landing problem. After the training and testing of the neural network, the results were compared to the results obtained by using support vector machine (SVM) and BPNN. The results obtained by the proposed method were found to be more accurate compared to others.

This paper is organized in 5 sections. Section 2 provides a brief overview of RBFNN and the details of RBFNN based on K-means clustering algorithm. The research methodologies are presented in Section 3. The simulation results obtained using RBFNN with K-means clustering algorithm to solve problems of forecasting and classification are discussed in Section 4. The last section contains the concluding remarks.

2. RBFNN Based on K-Means Clustering Algorithm

This section introduces the basic of RBFNN and its modification using K-means clustering algorithm. The radial basis function for the learning process used a Gaussian function. K-means clustering is used to estimate centers and to calculate the standard width of the Gaussian radial basis function. While gradient descent method is applied to get weights.

2.1 The Radial Basis Function Neural Network

ANNs are a computational method inspired by the behavior of working brains and nervous system. The radial basis function neural network (RBFNN) is a type of ANNs which could be used to solve the problems of supervised learning such as time series prediction, classification and pattern recognition. Keller, et. al. [9] explained that an RBFNN, is an alternative to multilayer perceptrons, which is used to find the function approximation defined by a single weighted sum. While in a multilayer perceptron, the approximation is defined by a nested set of weighted summations. The RBFNN is designed to perform a nonlinear mapping from the input space to the hidden space, followed by a linear mapping from the hidden space to the output space.

In this section, we describe the structure of RBFNN (see Figure 1). Then by using the structure, we explain the relationship between output variables $f(x)$ and input variables $x$.

The structure of RBFNN, as shown in Figure 1, has three different layers: the input layer, the hidden layer and the output layer. The input layer consists of $n$ source nodes, where $n$ is the
dimension of the input vector $x$. In the hidden layer, each node is activated by a radial basis function $\phi$. The output layer consists of a single array of computing unit called an output node $f(x)$. The size of the output layer is not restricted, but commonly it is much smaller than that of the hidden layer to follow the number of length centers. The output layer can be determined by the following equation:

$$
f_j(x) = \sum_{i=1}^{l} w_{ij} \phi(x - c_i)$$  \hspace{1cm} (1)

where $l$ is the number of centers in the hidden layer, $w_{ij}$ is the connection weights between $i^{th}$ hidden layer and $j^{th}$ output layer node, $c_i$ is the center vector for $i = 1, 2, \ldots, l$ and $\phi(x - c_i)$ is the real-valued radial basis function that depends on the distance between the input vector and the center vector. The distance is usually assumed to be the Euclidean distance $\|x\|$. The performance of RBFNN is greatly dependent on the choice of radial basis function.

In the study, the radial basis function is a Gaussian function which can be expressed by following equation [10]:

$$
\phi(x - c_i) = \exp\left(-\frac{1}{2\sigma^2}\|x - c_i\|^2\right)
$$  \hspace{1cm} (2)

where $\sigma$ is the standard width (spread) of the Gaussian radial basis function. Gupta, et. al. [10] explained that a Gaussian function, an unnormalized form of the Gaussian density function, is highly nonlinear and provides good locality for incremental learning. It has been proved that a Gaussian function has many well-defined mathematical features and can be used in the learning and control of nonlinear dynamic systems, and as some powerful schemes for modeling complex input-output mappings.

2.2 The RBFNN Modification Using K-Means Clustering Algorithm

K-means clustering algorithm is one of the clustering algorithms which has been widely used because of its simplicity and ability to produce good result. This algorithm is used in the learning process of RBFNN to group the training datasets into subgroups or clusters and further select the centers of the radial basis function according to the natural measure of the attracting centers in the sense of the
Euclidean distance. Each cluster center is associated with one of the hidden Gaussian nodes in the hidden layer. K-means clustering algorithm in this study uses convergent K-means clustering algorithm described by Anderberg and Spath in [10]. It may be used to achieve the goal that the training datasets are finally in the current clusters in the sense of the nearest distance. This task is easily carried out by adding an additional iterative process to the K-means clustering.

The learning process of RBFNN consists of two different stages. First, convergent K-means clustering algorithm was used to determine the centers and the standard width of the Gaussian radial basis function. Second, gradient descent method was applied to obtain the linear weights between the hidden layer and the output layer. The procedure of convergent K-means clustering algorithm is as follow [10]:

1. Select the number \( l \) of cluster center \( c_i \) for \( i = 1, 2, \ldots, l \) where \( l \) is smaller than the number \( m \) of the learning data.

2. Randomly choosing the initial center vectors:
   \[
   c_i = x_i, \quad i = 1, 2, \ldots, l
   \]  

3. Assign the learning data \( x_k \) for \( k = l + 1, l + 2, \ldots, m \) to one of the clusters based on the nearest distance criterion using the Euclidean distance:
   \[
   r(x, c) = \sqrt{(x_1 - c_1)^2 + (x_2 - c_2)^2 + \cdots + (x_l - c_l)^2}
   \]
   The learning data \( x_k \) belongs to the \( i^{th} \) cluster if
   \[
   \left\| x_k - c_i^* \right\| = \min_{i'} \left\| x_k - c_i \right\|, \quad 1 \leq k \leq m \text{ and } 1 \leq i \leq l
   \]

4. Recompute the center vectors using the new mean:
   \[
   c_i = \frac{1}{s_i} \sum_{k \in C_i} x_k, \quad 1 \leq k \leq m \text{ and } 1 \leq i \leq l
   \]  
   where \( s_i \) is the number of the learning datasets belonging to the \( i^{th} \) cluster \( C_i \).

5. Repeat Step 3.

6. If at least one learning data switches to another cluster, then repeat Step 4 and go to Step 5; otherwise, stop the procedure.

Subsequently, let \( r_{\text{max}} \) be the maximum distance between the obtained \( C \) centers of convergent K-means clustering algorithm. The standard width of all the Gaussian radial basis functions can be expressed as [10]:

\[
\sigma = \frac{r_{\text{max}}}{\sqrt{2C}}
\]  

After determining the update center of the Gaussian radial basis functions and the standard width, the second stage is to apply gradient descent method to identify the linear weights of the output layer. A gradient descent method is based on supervised learning procedure. Assume that the learning
process is described by the input-output data pairs \( \{x(k), y(k)\} \), then the error signal produced at the output layer at iteration \( h \) can be determined by the following equation:

\[
e_k(h) = y_k(h) - \sum_{i=1}^{l} w_i(h) \phi(\|x - c_i\|)\tag{8}
\]

Then the instantaneous value of the cost function at the output layer can be defined as follows:

\[
E(h) = \frac{1}{2} \sum_{k=1}^{m} e_k^2(h)\tag{9}
\]

The partial derivative of equation (9) is the gradient

\[
\frac{\partial E(h)}{\partial w_i(h)} = -\sum_{k=1}^{m} e_k(h) \phi(\|x - c_i\|)\tag{10}
\]

The correction \( \Delta w_i(h) \) applied to \( w_i(h) \) is defined by the delta learning rule (Widrow-Hoff rule):

\[
\Delta w_i(h) = -\eta \frac{\partial E(h)}{\partial w_i(h)}\tag{11}
\]

where \( \eta \) is the learning rate parameter. Parameter \( \eta \) must satisfy \( 0 < \eta \leq 1 \) [12]. Accordingly, the linear weights parameter between the hidden layer and the output layer can be updated as

\[
w_i(h+1) = w_i(h) + \eta \sum_{k=1}^{m} e_k(h) \phi(\|x - c_i\|)\tag{12}
\]

for \( 1 \leq i \leq l \).

3. Research Methodology
This section introduces the datasets collected and the performance evaluation for the selected RBFNN structure. The collected datasets are devided into the training and testing datasets. The training and testing datasets are time series data in each month. Performance evaluation uses mean absolute percentage error (MAPE) and the value of \( E \) (Nash-Sutcliffe model efficient coefficient).

3.1 The Study Datasets
The datasets take the case study in Lamongan district. Astronomically, Lamongan is located between 6°51'54" up to 7°23'6" South latitude, between 112°4'41" up to 112°33'12" East longitude and 12 miles above sea level. Aquaculture in Lamongan especially vannamei shrimp production are classified into several types of culture such as minapadi cultivation, brackish or fresh water pond and fish breeding in paddy fields. Vannamei shrimp production is aquaculture production with the second highest production amount after the production of milkfish for the kind of fish in Lamongan district . Lamongan has two seasons that are rainy and dry season. The high rainy season in Lamongan happens from November until April. In the month of May and June, the rainy season starts decreasing rainfall. While the dry season starts from July until October.
The datasets collected consist of three weather variables which are monthly average temperature, average humidity, rainfall datasets as the input nodes and vannamei shrimp production as the output node from January 2013 until December 2017. Average temperature datasets are obtained from Perak 1 Meteorology Station of Surabaya. Average humidity and rainfall datasets respectively were collected from the satellite data at the website https://www.worldweatheronline.com/lamongan-weather-averages/east-java/id.aspx and the Central Bureau of Statistics joined within the Department of Irrigation Lamongan district. For the output node, vannamei shrimp production is acquired from the Department of Fisheries Lamongan district.

For the five years in the Lamongan district, average number of highest and lowest cases in the study datasets each month is shown in Table 1. The input datasets are normalized to \([0,1]\) using min-max normalization as follows:

\[
d_k = \frac{x_k - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{13}
\]

where \(d_k\) is the normalized result of the \(k\)th input node, \(x_k\) is the actual value of the \(k\)th input node, \(x_{\text{min}}\) is the minimum value of the input nodes set and \(x_{\text{max}}\) is the maximum value of the input nodes set.

### Table 1. Average number of highest and lowest cases in the datasets each month for five years

| Datasets          | Highest Average | Month  | Lowest Average | Month |
|-------------------|-----------------|--------|----------------|-------|
| Average Temperature| 29.86 °C        | October| 27.79 °C       | July  |
| Average Humidity  | 79.4 %          | January| 60 %           | February|
| Rainfall          | 294.6 mm        | December| 9.8 mm        | August |
| Vannamei Shrimp   | 2499,2524 ton   | May    | 196,1686 ton   | December|

Furthermore, the normalized datasets are divided into training and testing datasets. From each input node or weather variable, 80% samples are January 2013 until December 2016 used as the training datasets and remaining 20% samples are used as the testing datasets. When the input nodes were processed in the RBFNN structure and resulted the approximated value \(f(x)\) is normalized so it had to be denormalized using the following equation:

\[
d'_k = f_{\text{min}} + y_k \left( f_{\text{max}} - f_{\text{min}} \right) \tag{14}
\]

where \(d'_k\) is the denormalized result of the \(k\)th approximation value, \(y_k\) is the value of the \(k\)th actual target data, \(f_{\text{min}}\) is the minimum value of the approximation datasets and \(f_{\text{max}}\) is the maximum value of the approximation datasets.

### 3.2 Performance Evaluation

The performance evaluation of RBFNN approach method uses MAPE equation and Nash-Sutcliff model efficient coefficient (E). MAPE and E equation were given as follow:
\[ MAPE = \frac{\sum_{t=1}^{N} |Y_t - F_t| / Y_t}{N} \times 100\% \]  

(15)

\[ E = 1 - \frac{\sum_{t=1}^{N} (Y_t - F_t)^2}{\sum_{t=1}^{N} (Y_t - \bar{Y}_0)^2} \]  

(16)

where \( Y_t \) is the value of the actual target data at time \( t \), \( F_t \) is the approximation (forecasting) value on the actual target data at time \( t \), \( N \) is the number of datasets and \( \bar{Y}_0 \) is the mean of rainfall over a given period \( N \).

The value of \( E \) (Nash-Sutcliffe model efficient coefficient) shows an indication about the ability of forecasting model with actual values. The value of \( E \) can vary from \(-\infty \) to \( 1 \). The closer the value of \( E \) to \( 1 \), the better the RBFNN structure is selected to be able to forecast the input node especially the rainfall [8]. Meanwhile MAPE shows how big the error value of forecasting within actual datasets. The smaller the value of MAPE, the better the RBFNN structure is selected to be able to forecast.

4. Results and Discussions

This section describes a method to create the best RBFNN structure, the forecasting of each weather as input nodes and the forecasting of vannamei shrimp production. In the creating of the RBFNN structure, RBFNN parameters are obtained by convergent K-means clustering algorithm and gradient descent method. The number of input nodes in the forecasting of weather variables commonly is defined based on the cultivation period of vannamei shrimp conducted by fisheries in Lamongan district. The forecasting result of weather parameters will be used to forecast the production of vannamei shrimp.

4.1 The Best RBFNN Structure

The RBFNN structure is composed of input nodes, hidden nodes and output node which are respectively in the input layer, hidden layer and output layer. The number of nodes in each layer can be different. In the RBFNN structure, there are the significant RBFNN parameters such as centers \( c \), the standard width \( \sigma \), radial basis activation function \( \phi \) and update weights \( w \) which determine the accuracy of approximation results. The process to obtain these parameters that is called the RBFNN training. The parameters are used to obtain the approximation result \( f \) in the RBFNN testing. Patterns of RBFNN structure have the number of hidden neurons (centers) different using Gaussian radial basis activation function as shown in Table 2. The update weights of these patterns are obtained by giving the number of epochs 5000 within different standard width and learning rate values. The RBFNN training (see Table 2) was performed starting from datasets January 2013 until December 2016 while the remaining datasets for the RBFNN testing.

| Patterns of RBFNN Structure | Number of Centers | Standard width | Learning Rate | Training MAPE | Testing MAPE | Result |
|-----------------------------|------------------|----------------|--------------|---------------|--------------|--------|
| 3-1-1                        | 2                | 0.30618        | 0.01         | 43.68%        | 53.64%       | Rejected |
| 3-1-1                        | 30               | 0.16055        | 0.01         | 25.01%        | 35.13%       | Rejected |
| 3-1-1                        | 40               | 0.16101        | 0.01         | 19.61%        | 26.77%       | Rejected |
Table 2. Simulation results from training and testing RBFNN

| Patterns of RBFNN Structure | Number of Centers | Standard width | Learning Rate | Training MAPE | Testing MAPE | Result    |
|-----------------------------|-------------------|----------------|---------------|---------------|-------------|-----------|
| 3-1-1                       | 47                | 0.14853        | 0.01          | 13.31%        | 19.1%       | Rejected  |
| 3-1-1                       | 47                | 0.14853        | 0.02          | 10.94%        | 18.02%      | Rejected  |
| 3-1-1                       | 47                | 0.14853        | 0.03          | 9.74%         | 17.1%       | Rejected  |
| **3-1-1**                   | **47**            | **0.14853**    | **0.04**      | **9.44%**     | **16.7%**   | **Accepted** |
| 3-1-1                       | 47                | 0.14853        | 0.05          | 8.51%         | 17.89%      | Rejected  |
| 3-1-1                       | 47                | 0.14853        | 0.06          | 8.12%         | 16.98%      | Rejected  |
| 3-1-1                       | 47                | 0.14853        | 0.07          | 7.87%         | 17.4%       | Rejected  |

During testing on the RBFNN structure for 3 input nodes in input layer, 1 hidden layer with 2 hidden nodes and 1 output layer with standard width value and learning rate respectively 0.30618 and 0.01, the testing MAPE was obtained 53.64% and it was not smallest. Then the number of nodes in hidden layer was determined 30 hidden nodes with standard width value and learning rate respectively 0.16055 and 0.01 and retesting was performed on the RBFNN structure but again the obtained testing MAPE was still big. The number of nodes in hidden layer and learning rate were being changed again and again until the testing MAPE was smallest for 47 hidden nodes in hidden layer and learning rate 0.04. Further the RBFNN structure was tested for bigger learning rate parameter but the testing MAPE was not smallest.

Table 2 shows that the best RBFNN structure for training and testing is the pattern of RBFNN structure 3 nodes (average temperature, average humidity, rainfall datasets) taken in input layer, 1 hidden layer with 47 nodes and 1 output layer with standard width value and learning rate respectively 0.14853 and 0.04. This best structure will be used as the structure of RBFNN training to forecast vannamei shrimp production. In this study, the datasets of vannamei shrimp production as the actual-output node was classified in three classes which were shown in Table 3.

Table 3. Classification vannamei shrimp production

| Class | Range       |
|-------|-------------|
| 1     | ≤ 650 ton   |
| 2     | 651-999 ton |
| 3     | ≥ 1000 ton  |

4.2 Forecasting of Weather Variables

Forecasting of weather variables is used as input nodes in the RBFNN testing to forecast vannamei shrimp production for the next 12 months. In obtaining the new datasets from each weather variabel, the number of nodes in the input layer is 3 input nodes based on the harvesting time conducted by fisheries of vannamei shrimp cultivation Lamongan district each 3 months. The number of nodes in hidden layer is determined 3 centers for average temperature and average humidity variables. While rainfall is determined with 20 centers. The number of nodes in output layer for each weather variabel is 1 output node. Table 4 shows the forecasting results of each weather variable.

Table 4. Forecasting results of weather variables

| Month | Actual Average Temperature (°C) 2017 | Approximated Average Temperature (°C) 2018 | Actual Average Humidity (%) 2017 | Approximated Average Humidity (%) 2018 | Actual Rainfall (mm) 2017 | Approximated Rainfall (mm) 2018 |
|-------|--------------------------------------|---------------------------------------------|----------------------------------|---------------------------------------|---------------------------|----------------------------------|
| January| 28.09                                | 28.89                                       | 77                               | 66.95                                 | 183                       | 133.47                           |
| February| 27.94                               | 29.1                                        | 78                               | 68.85                                 | 259                       | 83.87                            |
Table 4. Forecasting results of weather variables

| Month  | Actual Average Temperature (°C) 2017 | Approximated Average Temperature (°C) 2018 | Actual Average Humidity (%) 2017 | Approximated Average Humidity (%) 2018 | Actual Rainfall (mm) 2017 | Approximated Rainfall (mm) 2018 |
|--------|--------------------------------------|--------------------------------------------|----------------------------------|----------------------------------------|--------------------------|----------------------------------|
| March  | 28.38                                | 29.07                                      | 77                               | 70.76                                  | 220                      | 94.03                            |
| April  | 28.61                                | 28.75                                      | 77                               | 71.01                                  | 173                      | 81.34                            |
| May    | 28.76                                | 28.23                                      | 72                               | 72.85                                  | 79                       | 24.57                            |
| June   | 28.02                                | 28.26                                      | 70                               | 68.28                                  | 92                       | 4.4                              |
| July   | 27.73                                | 28.23                                      | 65                               | 66.11                                  | 26                       | 18.92                            |
| August | 28.02                                | 29.23                                      | 59                               | 65.65                                  | 4                        | 105.56                           |
| September | 29.13                              | 28.37                                      | 56                               | 66.14                                  | 34                       | 176.16                           |
| October | 30.39                               | 27.98                                      | 57                               | 64.74                                  | 106                      | 155                              |
| November | 28.93                               | 28.03                                      | 63                               | 67.45                                  | 208                      | 211.99                           |
| December | 28.55                               | 28.22                                      | 71                               | 68.41                                  | 249                      | 107.55                           |

Forecasting of new average temperature datasets has performance evaluation with training and testing MAPE respectively 2.32% and 2.79%. For the forecasting of new average humidity datasets, the training and testing MAPE are 9.87% and 8.3%. The forecasting of new rainfall datasets has the value of E for training and testing respectively 0.8201 and -0.33487.

4.3 Forecasting of Vannamei Shrimp Production

Forecasting of vannamei shrimp production is conducted through several stages, namely determining new input datasets using respective forecasting for average temperature, average humidity and rainfall, forecasting training of new datasets, forecasting testing and performance evaluation from the forecasting result. The forecasting training of vannamei shrimp production uses the RBFNN structure that is 3-1-1 with 47 hidden nodes has the testing MAPE was smallest. The plot of training result with RBFNN structure 3-1-1 with 47 hidden nodes and learning rate value 0.04 as shown in Figure 2a. Training MAPE is obtained 9.44%.

The forecasting testing of vannamei shrimp production uses RBFNN parameters which are obtained form the RBFNN structure 3-1-1 with 47 hidden nodes and learning rate value 0.04. The testing result of vannamei shrimp production has the testing MAPE that is obtained 16.7%. The plot of testing result with RBFNN is as shown in Figure 2b.

![Figure 2](image_url)

**Figure 2.** (a) Training plot of vannamei shrimp production, (b) Testing plot of vannamei shrimp production

Figure 2(b) shows that vannamei shrimp production Lamongan district on September 2018 had increased compared to September 2017. Whereas, vannamei shrimp production Lamongan district on
February, March and July 2018 had decreased compared to these three months 2017. Table 5 shows the forecasting result of vannamei shrimp production.

Table 5. Forecasting result of vannamei shrimp production

| Month    | Class (2017) | Range (ton) | Output (2018) | Class (2018) | Range (ton) |
|----------|--------------|-------------|---------------|--------------|-------------|
| January  | 2            | 651-999     | 1.8332        | 2            | 651-999     |
| February | 3            | ≥1000       | 2.0162        | 2            | 651-999     |
| March    | 3            | ≥1000       | 2.3717        | 2            | 651-999     |
| April    | 3            | ≥1000       | 3.0751        | 3            | ≥1000       |
| May      | 3            | ≥1000       | 2.9943        | 3            | ≥1000       |
| June     | 3            | ≥1000       | 2.9713        | 3            | ≥1000       |
| July     | 3            | ≥1000       | 1.9739        | 2            | 651-999     |
| August   | 2            | 651-999     | 2.3063        | 2            | 651-999     |
| September| 1            | ≤650        | 1.6803        | 2            | 651-999     |
| October  | 1            | ≤650        | 0.9812        | 1            | ≤650        |
| November | 1            | ≤650        | 1.3505        | 1            | ≤650        |
| December | 1            | ≤650        | 1.2128        | 1            | ≤650        |

Table 5 shows vannamei shrimp production is forecasted to produce a total production more than or equal to 1000 ton occurring in months, namely April, May and June. Vannamei shrimp production with a range of 651-999 ton occurred in months, namely January, February, March, July, August and September. While the remaining months, vannamei shrimp production decreased that is less than or equal to 650 ton.

5. Conclusions
This study proposes RBFNN with the Gaussian radial basis function using K-means clustering algorithm and gradient descent method to forecast vannamei shrimp production based weather factors (such as average temperature, average humidity and rainfall). The following conclusion is drawn (1) The RBFNN structure to forecast average temperature and average humidity is 3 input nodes in input layer, 1 hidden layer with 3 centers or hidden nodes and 1 output layer. While rainfall is determined with 3 input nodes, 1 hidden layer with 20 hidden nodes and 1 output layer. (2) The best RBFNN structure to forecast vannamei shrimp production is 3 input nodes (average temperature, average humidity and rainfall variables), 1 hidden layer with 47 hidden nodes and 1 output layer with the standard standard width and learning rate values respectively 0.14853 and 0.04. (3) The accuracy level showed that the error value between the approximation (forecasting) and actual target data, mean absolute percentage error (MAPE) is 16.7%. Based on its MAPE value, the chosen RBFNN structure showed a good performance to forecast vannamei shrimp production.

6. References
[1] Daw T, Adger W N, Brown K and Badjeck M C. 2009 FAO Fisheries and Aquaculture Technical Paper 530 107-150.
[2] De Silva S S and Soto D. 2009 FAO Fisheries and Aquaculture Technical Paper 530 151-212.
[3] Yu R, Leung P S and Bienfang. 2006 Aquacultural Engineering 34 26-32.
[4] Irawan M I, Syaharuddin, Utomo D B and Rukmi A M. 2013 J. of Theo. and Appl. Infor. Technol. 58 657-671.
[5] Shen X, Chen M and Yu J. 2009 Int. Conf. on Artificial Intelligence and Computational Intelligence 427-431
[6] Langoni D, Weatherspoon M H, Foo S,Y. and Martinez H.A. 2006 Engineering Appl. of Artificial Intelligence 19 883-890.
[7] Sachan A 2014 Int. Conf. for Convergence of Technol (India) (USA:IEEE)
[8] Dubey A D. 2015 IEEE International Conference on Trends in Automation, Communication and Computing Technologies (IEEE I-TACT) (India) (USA:IEEE)

[9] Keller J M, Liu D and Fogel D B. 2016 Fundamentals of Computational Intelligence: Neural Network, Fuzzy Systems and Evolutionary Computation (New Jersey: John Willy and Sons)

[10] Gupta M M, Jin L and Homma N 2003 Static and Dynamic Neural Network from Fundamentals to Advanced Theory (New Jersey: John Willy and Sons)

[11] Fausset L 1994 Fundamentals of Neural Networks: Architecture, Algorithm and Applications (New Jersey: Prentice Hall)