Comparison of backpropagation artificial neural network and SARIMA in predicting the number of railway passengers

O A Amalia* and A H M Putri
Department of Mathematics, Universitas Gadjah Mada, Indonesia

*Corresponding author's e-mail: okialmasamalia@mail.ugm.ac.id

Abstract. Trains are one of the most popular public transportations in Indonesia. The data from the Indonesian Central Bureau of Statistics show an increasing trend in the number of train passengers in Indonesia. However, the improvement of the railway network and service is needed. This study aims to forecast the number of railway passengers in Indonesia to help the government make appropriate improvements in the railway system for the future and evaluate the potential loss due to COVID-19. We use the data from the Indonesian Central Bureau of Statistics, from January 2006 to February 2020 and assume that there is no pandemic of COVID-19. The two models we use are Backpropagation Artificial Neural Network (BPANN) and Seasonal ARIMA (SARIMA). To find the best model, we observe BPANN with various parameters and the potential SARIMA models in MATLAB and R software, respectively. Our finding is that Backpropagation Artificial Neural Network of 12-5-1 with a learning rate of 0.001 has a smaller root mean squared error (RMSE) compared to SARIMA (2,1,0)(0,1,2)_{12}. Hence, it yields a more accurate forecast of the number of train passengers, which helps the railway company to improve and understand the loss due to COVID-19 accurately.

1. Introduction
Java and Sumatra islands are the most populous islands in Indonesia. According to the data from the Indonesian Central Bureau of Statistics, Java is inhabited by around 140 million people, while Sumatra has a population of approximately 51 million in 2015 [1]. There is an increasing trend in the population. This trend is followed by some societal problems where congestion is a part of it. Take Indonesia's capital, Jakarta, as an example. It is reported that the economic loss caused by congestion is 65 trillion rupiahs per year [2]. The increasing number of private vehicles flowing in and out of the city worsens the congestion in Jakarta. As the center of the economic activities, people from nearby cities travel to Jakarta for work. The traffic density increases due to commuting workers. However, the development of road construction to tackle this problem does not seem feasible anymore. Origin–destination surveys show that approximately 900 million people travel annually by rail in 2030. This motivates the government to make the improving and expanding urban rail the main strategy. Part of the strategy is to revive the abandoned rail network in Java [3].

The development plan launched by the government induces an increase in the number of train passengers in Indonesia (Java and Sumatra). The number of passengers escalated from around 21,000 in January 2014 to above 35,000 in December 2017 [3]. Following this demand, the government needs to plan its infrastructure development in supporting the railway service accordingly. In making an
appropriate development plan for the railway service, it is crucial for the government to have a forecasted number of train passengers.

The data on the number of railway passengers is a time series data that enables the Indonesian railways' company to observe the trend in the number of passengers over the years. Predicting the future number of train passengers helps the railways' company to anticipate the change of trend in the number of passengers and make appropriate adjustments. Also, by considering the situation where no pandemic of COVID-19 happens, we aim to see the potential loss suffered by the Indonesian railways' company in terms of the number of passengers due to COVID-19. This prediction is useful for the Indonesian railways' company in making the policy regarding the number of crew and trains working and other service-related procedures. In addition to it, it is helpful to calculate the potential passengers to plan or evaluate the development plan in the railway service within a year.

Forecasting is a part of data science. One of the emerging subjects in data science, including forecasting, is machine learning. One of the topics in machine learning is neural networks. During the past few decades, there have been abundant publications on forecasting using neural networks [4][5]. Backpropagation Artificial Neural Network (BPANN) nowadays is widely used to forecast data in various fields. Anjar Wanto et al. [6] use BPANN to predict the Consumer Price Index of the food sector in an Indonesian region, Pematangsiantar. The author uses min-max normalization to pre-process the data and compare the BPANN and BPANN with conjugate gradient Fletcher Reeves. The results show that the later approach dominates the earlier one with faster convergence and better accuracy.

From a statistical point of view, the Box-Jenkins method, which is widely known as ARIMA has been widely used to forecast data in numerous study fields. Widiyaningtias et al. [7] use ARIMA to predict the number of train passengers in Malang, East Java. The result suggests that ARIMA gives an accurate prediction. Meanwhile, Sartono employs Box-Jenkins, Holt-Winters, and Trend and Seasonal Linear Model (TSLM) models to predict the train passengers in Java and Sumatra, separately [8]. The author finds that overall, no model dominates the forecast and that considering the seasonal and trend effects separately will probably give a better result.

2. Methods
We use the monthly data on the number of train passengers in Indonesia (Java-Sumatra Islands) provided by the Indonesian Central Bureau of Statistics shown in table 1 [3]. The data covers a period of 173 months, from January 2006 to May 2020. Our models do not consider the pandemic's effect. Therefore, we use the 170 months historical data and leave out the ones from the last three months, where pandemic already affected society. The data of the last three months are used to observe the number of potential passengers that the Indonesian Railways Company (PT. KAI) lost due to the pandemic. Further, from our prediction, we would be able to see the forecasted number of passengers after the society slowly returns to normal activities, as encouraged by the Indonesian government in June 2020.

| Year | Jan   | Feb   | Mar   | April | May   | Jun   | July  | Aug   | Sep   | Oct   | Nov   | Dec   |
|------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 2006 | 11828 | 11931 | 13314 | 12909 | 13575 | 13203 | 14433 | 13255 | 13436 | 14290 | 13631 | 13614 |
| 2007 | 13960 | 10969 | 13409 | 14415 | 15232 | 15104 | 16454 | 15419 | 15033 | 15866 | 14391 | 15084 |
| 2008 | 15027 | 14378 | 16071 | 15711 | 16363 | 17010 | 17887 | 17108 | 15879 | 17337 | 15973 | 15332 |
| 2009 | 14494 | 13869 | 17132 | 16775 | 17824 | 18143 | 18385 | 17527 | 17281 | 17281 | 16778 | 17581 |
| 2019 | 35122 | 31899 | 35751 | 35809 | 35102 | 35090 | 39035 | 35189 | 35221 | 36448 | 35877 | 37463 |
| 2020 | 34134 | 32286 | 23425 | 5890  | 5484  |       |       |       |       |       |       |       |

Table 1. The number of passengers in Indonesia.
We use two models in forecasting the railway passengers. One is from a machine learning perspective, i.e., Backpropagation Neural Network (BPANN), and the other is from the statistical perspective, i.e., Seasonal ARIMA (SARIMA).

2.1. Backpropagation neural network
Before training, testing, and using the network to make a prediction, it is important to understand the data we are working with. It is convenient to get a glimpse of the data by looking at its plot. The graph representing our data is given in figure 1. It is obvious from the plot in figure 1 that our data is non-linear and has an increasing trend. This observation is then used in the data pre-processing step.

In the next parts, the data we put into the network is called the input data. The target data is the desired outcome that the network should learn. The prediction or results of the network is called output. Before training the neural network, we first pre-process the data. The details of the data pre-processing are given in the next part.

![Number of Railway Passengers in Indonesia](image)

**Figure 1.** Plot of the original data.

2.1.1. Data pre-processing. First, we arrange 158 data patterns out of the available data. In these patterns, we use the first 12 months data as the input and the data on the next month as the target [6]. For example, the first data pattern consists of January-December 2006 data as the input and January 2007 data as the target. Next, the second pattern consists of February-December 2006, January 2007 data as the input, and February 2007 data as the target. The data arranged with this pattern are shown in table 2 and used to train the network using backpropagation artificial neural network (BPANN).

BPANN includes activation functions in each neuron and weights between neurons. The activation function is attached to each neuron and decides whether or not it is significant for the prediction. Choosing the right activation function is important as it affects the output of the network and computational efficiency. In our model, we use tansig (hyperbolic tangent) and purelin as the activation functions for the hidden layers and output layer, respectively. The tansig is a sigmoid function which takes a value within the range of $[-1, 1]$. It introduces the non-linearity property of the data to the network. We choose tansig because it often converges faster than the standard logistic function [9]. The purelin activation function is then used in the output layer. Purelin is a linear activation function. Having a linear
activation function enables the output to have any value, while a sigmoid activation function restricts
the output to lie in a small range [10].

\[
\text{Table 2. Input and target data for BPANN.}
\]

| Data | Input | Target |
|------|-------|--------|
| 1 | 11828 11931 13314 12909 13575 13203 14433 13255 13436 14290 13631 13614 13960 | 13960 |
| 2 | 11931 13314 12909 13575 13203 14433 13255 13436 14290 13631 13614 13960 10969 | 10969 |
| 3 | 13314 12909 13575 13203 14433 13255 13436 14290 13631 13614 13960 10969 13409 | 13409 |
| 4 | 12909 13575 13203 14433 13255 13436 14290 13631 13614 13960 10969 13409 14415 | 15232 |
| 5 | 13575 13203 14433 13255 13436 14290 13631 13614 13960 10969 13409 14415 15232 | 15232 |
| 6 | 13203 14433 13255 13436 14290 13631 13614 13960 10969 13409 14415 15232 15104 | 15104 |
| | | | 156 37965 35122 31899 35751 35809 35102 35090 39035 35189 35221 36448 35877 37463 | 37463 |
| | | | 157 35122 31899 35751 35809 35102 35090 39035 35189 35221 36448 35877 37463 34134 | 34134 |
| | | | 158 31899 35751 35809 35102 35090 39035 35189 35221 36448 35877 37463 34134 32286 | 32286 |

Considering the tansig transfer function, our data should lie within the range that can be defined by
the function. We transform our data so that they lie within the range of [0, 1]. To achieve this, we use
the min-max normalization, formulated as in equation (1).

\[
v' = \frac{v - \min(v)}{\max(v) - \min(v)} \left( \max(v') - \min(v') \right) + \min(v') \quad (1)
\]

Where \( v \) is original data, and \( v' \) is the normalized data. Recall that our data should lie within [0, 1].
Hence, we have \( \min(v') = 0 \) and \( \max(v') - \min(v') = 1 \). Substituting to the equation (1) results in the
min-max normalization formula given by equation (2).

\[
v' = \frac{v - \min(v)}{\max(v) - \min(v)} \quad (2)
\]

Using equation (2) to normalize the data on table 2 using MATLAB, we get the normalized input and
target data in table 3. The normalized data makes the network converge faster and provide outputs
that minimize errors [11]. In the next part, this data is used in the network. The output of the network is
also in the normalized form. Hence, to compare with the raw data, we need to de-normalize the outputs
using equation (3).

\[
v = v' \left( \max(v) - \min(v) \right) + \min(v) \quad (3)
\]

\[
\text{Table 3. Normalized input and target data for BPANN.}
\]

| Data | Input | Target |
|------|-------|--------|
| 1 | 0.0318 0.0356 0.0869 0.0719 0.0965 0.0796 0.1234 0.0815 0.0879 0.1183 0.0948 0.0942 0.1066 | 1.0000 0.9875 0.9750 0.9625 0.9500 0.9375 0.9250 0.9125 0.9000 0.8875 0.8750 0.8625 0.8500 0.8375 |
| 2 | 0.0356 0.0869 0.0719 0.0965 0.0828 0.1234 0.0815 0.0879 0.1183 0.0948 0.0942 0.1066 0.0000 |
| 3 | 0.0869 0.0719 0.0965 0.0828 0.1283 0.0815 0.0879 0.1183 0.0948 0.0942 0.1066 0.0000 0.0000 |
| 4 | 0.0719 0.0965 0.0828 0.1283 0.0847 0.0879 0.1183 0.0948 0.0942 0.1066 0.0000 0.0000 |
| 5 | 0.0965 0.0828 0.1283 0.0847 0.0914 0.1183 0.0948 0.0942 0.1066 0.0869 0.1228 0.1519 |
| 6 | 0.0828 0.1283 0.0847 0.0914 0.123 0.0948 0.0942 0.1066 0.0869 0.1228 0.1519 0.1776 0.2043 0.2310 0.2577 0.2844 0.3111 0.3378 0.3645 |
| | | | 156 0.8947 0.7753 0.918 0.9201 0.8599 0.8594 0.863 0.8641 0.9078 0.8875 0.944 0.8254 |
| | | | 157 0.8947 0.7753 0.918 0.9201 0.8599 0.8594 0.863 0.8641 0.9078 0.8875 0.944 0.8254 |
| | | | 158 0.7753 0.918 0.9201 0.8939 0.8935 0.1 0.863 0.8641 0.9078 0.8875 0.944 0.8254 0.7595 |
2.1.2. Network architecture and parameters. The neural network in our research has an architecture that includes input, hidden, and output layers, as shown in figure 2. Observe that we have 12 input nodes and one output node as there are 12 inputs and one target in each data pattern. BPANN calculates the error from the output to identify the errors in the hidden layers. The algorithm then aims to minimize the error. A detailed explanation on backpropagation can be found in a book and recent paper [11][12]. Our goal is to find the best network architecture, in terms of the number of hidden layers, and parameters. We use MATLAB to simulate the network.

![Figure 2. The BPANN architecture](image)

The parameters we observe in training the network include learning rates, the number of hidden layers, and the division of the data for training, validating, testing processes. We observe the network performance by varying the number of hidden layers from 1 to 30 and learning rates of 0.001, 0.01, and 0.1. In dividing the data for training, validating, and testing steps, we use more than 50% of the data for training to make accurate network outputs. In table 4, we show the partition of the data used in our analysis.

| Division | Training | Validating | Testing |
|----------|----------|------------|---------|
| 1        | 60%      | 35%        | 15%     |
| 2        | 70%      | 15%        | 15%     |

Table 4. Data partition for the analysis (in the percentage out of the total data).

We program the BPANN in MATLAB to analyze the network performances. We choose the Levenberg-Marquardt (LM) algorithm in training the network because it converges faster than other algorithms in MATLAB [10]. It also results in a smaller mean squared error compared to other algorithms [10]. The drawback of the LM algorithm is that it needs a larger memory.

Recall that the neural network has weights between neurons or nodes. This weight shows the importance of the nodes to the output. MATLAB generates a randomized weight each time we run the neural network code. Therefore, to capture the network performance, we run the simulation ten times for each parameter and calculate the mean performance. In addition to the output, MATLAB generates values of the correlation coefficient, R, for the training, validating, testing, and the whole data. The value of R shows the correlation between BPANN output and the target data. The R-value that exceeds 0.95 or 95% shows a good accuracy. In addition to the coefficient of correlation, R, the performance parameter that we choose is mean squared error (MSE), which is widely used in forecasting. We also calculate the square root of MSE (RMSE) to compare the results of BPANN and SARIMA.
2.1.3. Observation of neural network training. In doing the analysis, we look at the RMSE (root mean square error) and R (autocorrelation coefficient) from the whole data to get the network that works well for the whole dataset. We also observe the network performance for each data partition after finding the best overall RMSE and R. We aim to find the network architecture which has the lowest RMSE and the highest value of R with the least number of hidden layers.

We code the neural network in MATLAB with the learning rate of 0.001 to compare the network performance for the networks' different numbers of hidden layers and data partitions. We use two kinds of data partitions given in table 4. In comparing the network performance, we look at the RMSE and R of the whole data, because the overall RMSE and R cover the performance of the network in training, validating, and testing processes. We do the same steps for the network with a learning rate of 0.01 and 0.1. We aim to find the network which has the smallest RMSE and largest R from these observations.

The MATLAB simulation for the learning rate of 0.001 results in the best BPANN of 12-5-1 (12 input neurons, 5 hidden layers, 1 output neurons) where the RMSE is 0.03697. In this network, 70% of data is used for training, and 15% of data is each for validating and testing. The R-value of BPANN 12-5-1 (12 input neurons, 5 hidden layers, 1 output neurons) is 0.99122 or approximately 99%. It can be concluded that the prediction from the 12-5-1 network is accurate.

For the network with a learning rate of 0.01, the lowest RMSE of 0.03938 is given by the network with 7 hidden layers where 60%, 25%, and 15% of the data is used for training, validating, and testing, respectively. The coefficient of correlation, R, for this 12-7-1 network is 0.9887, which shows a significant relationship between the prediction and the raw data.

The best architecture for the network with a learning rate of 0.1 is 12-6-1 (70% training data, 15% validating data, 15% testing data) with an RMSE value of 0.03828. The R-value of this network, 98.925%, shows a significant relationship between the prediction and the raw data. Table 5 summarizes the performance of the best network, each for the learning rate of 0.001, 0.01, and 0.1.

Table 5. Summary of the best networks.

| Learning Rate | Architecture | RMSE   | R      | Data Partition         |
|---------------|--------------|--------|--------|------------------------|
| 0.001         | 12-5-1       | 0.03697| 0.99122| Training = 70%         |
|               |              |        |        | Validating = 15%       |
|               |              |        |        | Testing = 15%          |
| 0.01          | 12-7-1       | 0.03938| 0.98870| Training = 60%         |
|               |              |        |        | Validating = 25%       |
|               |              |        |        | Testing = 15%          |
| 0.1           | 12-6-1       | 0.03828| 0.98925| Training = 70%         |
|               |              |        |        | Validating = 15%       |
|               |              |        |        | Testing = 15%          |

The lowest RMSE of 0.03697 and the highest R of 0.99122 show that the best network architecture is 12-5-1 with the learning rate of 0.001. The data partition used is 70%, 15%, 15% each for training, validating, and testing, respectively. This means that the 12-5-1 network (12 input neurons, 5 hidden layers, 1 output neurons) with the parameters mentioned before gives the most accurate prediction of the railway passengers in Indonesia out of the parameters that we consider. This prediction provides the best approximation to the real data, which is helpful for the railway company to calculate the loss in terms of number of passengers during the pandemic of COVID-19. It also provides the expected number of passengers in normal situations (without COVID-19). We use this network to predict the number of passengers 1-year ahead in section 3. In section 3, we also compare the performance of the 12-5-1 BPANN to the performance of the best SARIMA model.
2.2. Seasonal ARIMA (SARIMA)
Seasonal ARIMA is an ARIMA (Autoregressive Integrated Moving Average) model that supports time-series data with seasonal factors. The Seasonal ARIMA model contains regular autoregressive and moving average terms for the correlation in the early lags. The seasonal autoregressive and moving average terms of SARIMA shows the correlation at the seasonal lags. For nonstationary seasonal series, an additional seasonal difference is often required to completely specify the model [13]. Seasonal ARIMA models are generated with the same iterative modeling procedure used for nonseasonal data: identification, estimation, diagnostic checking, forecasting, which is shown in figure 3.

![Flow diagram for the seasonal ARIMA model-building strategy.](image)

The Seasonal ARIMA model consists of two types i.e., seasonal ARIMA only or ARIMA \((P,D,Q)^S\) and multiplicative ARIMA models (seasonal and nonseasonal) or ARIMA \((p,d,q)(P,D,Q)^S\), where S is the seasonal period. The lower-case letters \((p,d,q)\) indicate the nonseasonal, whww \(p,q,d\) is AR (autoregressive) orders, MA (moving average) orders, and differencing of nonseasonal models. The upper-case letters \((P,D,Q)\) denote the seasonal orders, where \(P,Q,D\) is AR orders, MA orders, and differencing of seasonal models. If a series is an ARIMA \((P,D,Q)^S\), then it can be expressed as in equation (4) [14].

\[
1 - \phi_1 B^S - \phi_2 B^{2S} - \cdots - \phi_p B^{pS} (1 - B^S)^D b_t = 1 - \theta_1 B^S - \theta_2 B^{2S} - \cdots - \theta_q B^{qS} a_t \quad [4]
\]

Furthermore, the mathematical form of the model ARIMA \((p,d,q)(P,D,Q)^S\) can be written as follows:
\[ \Phi_p(B^S) \phi_p(B) (1 - B)^d (1 - B^S)^D Z_t = \theta_q(B) \theta_q(B^S) a_t \]

where \((1 - B)^d\) or \((1 - B^S)^D\) operation is mathematical operations of nonseasonal differencing and seasonal differencing. Now, we begin to generate the models that suit our data.

2.2.1. **SARIMA models identification.** The first step in identifying the model is making the data plot and checking whether the data contains trend factors, cyclical fluctuation, seasonal variation, irregular or random influences [15]. Based on figure 1, railway passenger data has a trend of increasing. This means that the data is not stationary in mean and variance. The non-stationarity in variance can be identified using the lambda value from the Box-Cox test in R, given by figure 4. The lambda value is far from 1, i.e., -0.4646465. Thus, the data is nonstationary in variance.

---

**Figure 4.** The Box-Cox Plot.

The non-stationarity in the mean is shown by the ACF (Auto-Correlation Function) and the Partial ACF plots of the original data. The ACF plot has decreased slowly, indicating that the raw data is not stationary in the mean. The non-stationarity in the mean can also be determined using the Augmented Dickey-Fuller (ADF) test, which result is shown in figure 5.

**Figure 5.** Augmented Dickey-Fuller test's result.

The ADF test shows that the data is not stationary in mean, because the p-value of 0.6986 is larger than the significance value \(\alpha = 0.05\). So, the railway passenger data is not stationary in variance and mean. Therefore, the data must be transformed and differenced using R software to be stationary data. We use log transformation and differencing 1 time to make the data stationary. Log transformation equalizes the standard deviation, meaning stabilizes the variance [16]. Next, we construct the ACF and Partial ACF plots of the stationary data, as shown in figure 6 below.
We identify the nonseasonal ARIMA \((p,d,q)\) models based on the ACF and Partial ACF plots. In figure 6, the Partial ACF plot cuts off (decreases drastically) after lag 1 and lag 2. Therefore, the possible ARIMA models are ARIMA \((1,1,0)\) and ARIMA \((2,1,0)\), with differencing \(d = 1\).

The next step is to determine the seasonal ARIMA \((P,D,Q)\) model. The ACF and Partial ACF plots are given in figure 6 indicate a monthly seasonality, \(S = 12\), as the ACF and Partial ACF patterns at lags 12, 24, 36 are significant. Next, we do seasonal differencing 12 times \((D = 1)\) to the data and construct the ACF and Partial ACF plots in figure 7.

We look at the ACF and Partial ACF values at lags 12, 24, 36, and so on. The ACF plot in figure 7 shows cut-offs after lag 12 and 24. The Partial ACF plot in seasonal lags (lag 12, 24, and 36) tends to die down. Based on these identifications, there are four Seasonal ARIMA models that are suitable for railway passenger data:

1. Seasonal ARIMA \((2,1,0)\) \((0,1,2)\)\(^{12}\)
2. Seasonal ARIMA \((1,1,0)\) \((0,1,2)\)\(^{12}\)
3. Seasonal ARIMA \((2,1,0)\) \((0,1,1)\)\(^{12}\)
4. Seasonal ARIMA \((1,1,0)\) \((0,1,1)\)\(^{12}\)

### 2.2.2. Parameters estimation and diagnostic checking of SARIMA models

Once the four models are established, the parameters and the corresponding standard errors can be estimated using statistical techniques, such as Maximum Likelihood (ML). In the Maximum Likelihood Estimation (MLE) method, the Akaike Information Criterion (AIC) was used to compare the models and find the best fitted Seasonal ARIMA model. The best model is the model that has the smallest AIC and RMSE value [7].

A statistically adequate model satisfies the hypothesis that random shocks are independent. This hypothesis can be tested using ACF residual plots. If the residual is independent, then the random shocks are independent [17]. We use the test statistic of Ljung-Box to examine the independence of the residuals. The Ljung-Box test with \(n\) observations and \(k\) lags is given by the \(Q_k\) in equation (5).

\[
Q_k = n(n + 2)\sum_{m=1}^{k} \frac{r_m^2}{n-m}
\]  

\((5)\)
Where \( r_k \) is the autocorrelation coefficient value in lag \( k \). If \( Q_k > \chi^2_{\alpha} \) with \( db = k - p \) or p-value < 0.05, then the residue meets the white noise process.

Furthermore, the Ljung-Box test of the four Seasonal ARIMA models using R software. The standardized residual plots for the four models indicate that there are no residual trends and no outliers. The ACF of the residuals from SARIMA (2,1,0) (0,1,2)\(^{12} \) is given in figure 8.

According to the residual assumption test, which is based on \( \alpha = 0.05 \), SARIMA (2,1,0) (0,1,2)\(^{12} \) meets the white noise requirement as all p-value > 0.05 (figure 9). This means that there is no significant autocorrelation between residuals at different lags.

![ACF of Residuals](image)

**Figure 8.** ACF of the residuals from SARIMA (2,1,0) (0,1,2)\(^{12} \).

![p-values for Ljung-Box Test of the Residuals](image)

**Figure 9.** The p-values from Ljung-box test of the residuals from SARIMA (2,1,0) (0,1,2)\(^{12} \).

2.2.3. **Selecting the best SARIMA model.** The best Seasonal ARIMA model in a time series data can be determined by looking at the estimated price. A summary of estimated prices from the Root Mean Square Error (RMSE), Akaike’s Information Criterion (AIC), and Bayesian Information Criteria (BIC) for the four Seasonal ARIMA models above is given in table 6.

**Table 6.** Optional models and the related standards values.

| Models               | AIC    | BIC    | RMSE  |
|----------------------|--------|--------|-------|
| Seasonal ARIMA (2,1,0) (0,1,2)\(^{12} \) | -519.52 | -503.8402 | 0.04083115 |
| Seasonal ARIMA (1,1,0) (0,1,2)\(^{12} \) | -518.79 | -506.2507 | 0.04107667 |
| Seasonal ARIMA (2,1,0) (0,1,1)\(^{12} \) | -520.47 | -507.9257 | 0.04143228 |
| Seasonal ARIMA (1,1,0) (0,1,1)\(^{12} \) | -519.44 | -510.0299 | 0.04185866 |

Based on the summary on table 6, Seasonal ARIMA (2,1,0) (0,1,2)\(^{12} \) models have the minimum BIC and RMSE values. Thus, the Seasonal ARIMA (2,1,0) (0,1,2)\(^{12} \) model is a good model for the forecasting process

3. **Results and discussion**

The results from section 2 established the best BPANN and SARIMA of 12-5-1 network with learning rate = 0.001 and SARIMA (2,1,0) (0,1,2)\(^{12} \) respectively. In this section, we compare the RMSE from the network and SARIMA in table 7. Table 7 shows that BPANN has a lower RMSE, meaning it provides a better prediction compared to the SARIMA model.
Table 7. The RMSE of the best BPANN and SARIMA.

| Model                | RMSE   |
|----------------------|--------|
| 12-5-1 BPANN         | 0.03697|
| SARIMA (2,1,0) (0,1,2) | 0.04083|

We present the forecasted data from each model in table 8 and 9. In table 8, we also show the residual, i.e., the difference between the actual and predicted data. The residual indicates not only the model accuracy but also the potential passengers that the Indonesian Railways Company (PT.KAI) lost due to the pandemic of COVID-19.

Table 8. The actual and prediction data from the best BPANN (12-5-1).

| Month | Actual Data | Prediction | Residual |
|-------|-------------|------------|----------|
| Mar-20| 23425       | 36572      | 13147    |
| Apr-20| 5890        | 34839      | 28949    |
| May-20| 5484        | 34387      | 28903    |
| Jun-20| -           | 34759      | -        |
| Jul-20| -           | 34905      | -        |
| Aug-20| -           | 35022      | -        |
| Sep-20| -           | 35004      | -        |
| Oct-20| -           | 35192      | -        |
| Nov-20| -           | 35048      | -        |
| Dec-20| -           | 34994      | -        |
| Jan-21| -           | 35111      | -        |
| Feb-21| -           | 34902      | -        |

Figure 10. The data and prediction up to 1 year ahead resulted from the 12-5-1 BPANN.

Assuming that the COVID-19 spread from March 2020, based on the BPANN results, the company lost 13,147; 28,949, and 28,903 passengers in March, April, and May, respectively. The plot of actual data and the prediction up to February 2021 are given in figure 10.
In this research, testing was conducted to predict the number of train passengers over the next 12 months. Prediction results using the Seasonal ARIMA (2,1,0) (0,1,2)\textsuperscript{12} model is shown in table 9.

**Table 9.** Prediction result of railway passengers.

| Month | Actual | Prediction | Lower |
|-------|--------|------------|-------|
| Jan-20 | 34134 | | |
| Feb-20 | 32286 | | |
| Mar-20 | 36554 | 33625 | |
| Apr-20 | 36117 | 32707 | |
| May-20 | 37300 | 33332 | |
| Jun-20 | 36918 | 32540 | |
| Jul-20 | 39262 | 34201 | |
| Aug-20 | 37169 | 32028 | |
| Sep-20 | 37132 | 31672 | |
| Oct-20 | 38567 | 32584 | |
| Nov-20 | 37309 | 31237 | |
| Dec-20 | 39254 | 32585 | |
| Jan-21 | 37159 | 30593 | |
| Feb-21 | 34356 | 28064 | |

**Figure 11.** The prediction result of railway passengers.

Prediction results can also be seen in the black graph line in figure 11. The red graph line shows the actual data of the railway passenger number. Furthermore, the blue graph line shows lower and upper prediction. Prediction results begin from March 2020 to February 2021. The highest number of train passengers occurred in July 2020. On the other hand, the lowest number of train passengers occurred in February 2021. Compared to the backpropagation neural network, the advantage of using SARIMA has the upper and lower prediction. This range helps as there is uncertainty in the precision of the exact approximation. The upper and lower prediction takes into account the individual error of the prediction.
By looking at the potential loss in March-May 2020 and the predicted number of passengers in the future, the government can make an appropriate decision to adjust its policy in building railway infrastructure and railway service.

4. Conclusion
The best network architecture in the backpropagation neural network (BPANN) is 12-5-1 with the learning rate = 0.001, where 70%, 15%, and 15% of the data is for training, validating, and testing, respectively. It has RMSE of 0.03697 and R of 0.99122. The best SARIMA model is SARIMA (2,1,0)(0,1,2)\text{12} with RMSE of 0.04083 and BIC of -503.8402. The results indicate that BPANN outperforms SARIMA.

From the BPANN prediction, the potential number of passengers that the Indonesian Railways Company lost due to the pandemic of COVID-19 is 13,147; 28,949; and 28,903 passengers in March, April, and May, respectively. It also predicts that from July 2020 to February 2021, there are no more than 36,000 passengers each month. Looking at these results, we recommend the government to put off the planned development of the railway network and service.

For future research, it is better to use Bayes optimization to find the optimal parameters when we consider some parameters at the same time [18]. We can also use the genetic algorithm for the same purpose. Another improvement is using differencing to deseasonalized and detrend the data in the pre-processing part. For the research which aims to forecast using the network with an optimal learning rate, it is suggested to change the learning rate adaptively during the training [19]. Regarding the SARIMA model, further research may be executed to reveal the trend and seasonal specifically. Our prediction looks good as we only forecast for 1 season (12 months). However, the model may not yield an accurate prediction when we cover a long period of time as it becomes increasingly over-optimistic. Other methods considering seasonal effects may give a different view or even strengthen the result in this project.

References
[1] Badan Pusat Statistik (BPS) 2020 Jumlah Penduduk Hasil Sensus Penduduk (SP) dan Survei Penduduk antar Sensus (SUPAS) Menurut Provinsi Tahun 1971-2015 (Jakarta: BPS)
[2] Harmadi S H 2015 J. Indones. Econ. Bus. 30 (3) 220-38
[3] Badan Pusat Statistik (BPS) 2020 Jumlah Penumpang Kereta Api Tahun 2006-2020 (Ribu Orang) (Jakarta: BPS)
[4] Yu Y, Wang Y, Gao S and Tang Z 2017 Comput. Intell. Neurosci. 2017 1-9
[5] Tkáč M and Verner R 2016 Appl. Soft. Comput. 38 788-804
[6] Wanto A, Zarlis M and Hartama D 2017 J. Phys.:Conf. Ser. 930 012018
[7] Widiyaningtyas T and Qonita Int. Conf. Artif. Intell. Inform., Technol. (ICAIIT) 359-64
[8] Sartono 2017 J. Phys.: Conf. Ser. 824 12032
[9] LeCun Y A, Bottou L, Orr G B and Müller K R 2012 Neural Networks: Tricks of the Trade (Berlin: Springer)
[10] Demuth H, Beale M and Hagan M 2010 Neural Network Toolbox User's Guide MathWorks (USA: MathWorks)
[11] Rojas R 2013 Neural Networks: A Systematic Introduction (Berlin: Springer Science & Business Media)
[12] Pasini A 2015 J. Thorac. Dis. 7 953-60
[13] Hanke J E and Wichern D W 2005 Business Forecasting (8th Edition) (New Jersey: Prentice Hall)
[14] Wei S and William W S 2006 Time Series Analysis: Univariate and Multivariate Methods (2th Edition) (Boston: Pearson Addison Wesley)
[15] Astuti S W and Jamaludin 2018 IOP Conf. Ser.: Mater. Sci. Eng. 407 012105
[16] Curran-Everett D 2018 Adv. Physiol. Edu. 42 343-7
[17] Pankratz A 2009 Forecasting with Univariate Box-Jenkins Models: Concepts and Cases (Canada:
John Wiley & Sons)

[18] Brochu E, Cora V M and De Freitas N A 2010 A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning Preprint arXiv:1012.2599

[19] Park J, Yi D and Ji S 2020 Symmetry 12 660