JOURNAL OF STATISTICAL MODELING & ANALYTICS (JOSMA)
(ISSN: 2180-3102)

UNIVERSITI MALAYA

Nonlinear Autoregressive Neural Network for Forecasting COVID-19 Confirmed Cases in Malaysia

Lee Jia Yu[1], Nur Haizum Abd Rahman[1*] & Hani Syahida Zulkafli[1]

[1]*Department of Mathematics and Statistics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor Darul Ehsan*

**Corresponding author: nurhaizum_ar@upm.edu.my*

RESEARCH ARTICLE

Received 31 May 2023
Accepted 15 September 2023
Published 25 October 2023

Abstract

A nonlinear autoregressive neural network (NARNN) model is a feedforward neural network for handling complex nonlinear time series problems. In this study, the tangent sigmoid (tansig) activation function with the different numbers of past values and different numbers of hidden neurons for the NARNN model is determined. The COVID-19 daily confirmed cases in Malaysia are collected with different amounts of samples used, which are 100, 500 and 900. Therefore, data from 100, 500 and 900 days before 21 September 2022 are extracted for the NARNN model training, validation and testing procedure. The lowest average mean squared error (MSE) becomes the best combination. The result shows that the past value is 1:10 and the number of neurons of 10 when the sample size is 100. At sample size 500, past values of 1:10 and neurons of 8 enable the model to perform at its best. Whereas for sample size 900, the network setting of 1:5 past value and five hidden neurons gives the least MSE. Multi-step ahead time series forecasting is conducted to forecast the number of confirmed COVID-19 cases in 7 days from 22 to 28 September 2022. The result shown for 7-days-ahead confirmed cases indicating Malaysia datasets, the best forecasting outcome occurs when 900 samples are inputted.

Keywords: Activation function; COVID-19; Forecasting; NARNN

1. Introduction

In December 2019, Coronavirus disease-2019 (COVID-19), a contagious disease transmitted through droplet or direct contact caused by the Severe Acute Respiratory Syndrome-Related Coronavirus 2 (SARS-CoV-2) virus, was first discovered in Wuhan, China, as a mysterious pneumonia with symptoms of fever, cough and tiredness. The disease rapidly infiltrated other countries within a few months, mainly through person-to-person transmission and human mobility. A severe public health concern was triggered when the outbreak of COVID-19 was declared as the sixth Public Health Emergency of International Concern (PHEIC) by the World Health Organisation (WHO) on 30 January 2020, whereby the previous five were H1N1 (2009), Polio (2014), Ebola in West Africa (2014), Zika (2016) and Ebola in the Democratic Republic of Congo (2019) (Wu et al., 2020). Further, on 11 March 2020, the outbreak of COVID-19 was announced as a pandemic by WHO.

According to Namasudra et al. (2021), most government authorities implemented preventive measures and policies such as social distancing, contact tracing, enforced curfew or isolation and certain cities or even nationwide lockdowns to flatten the coronavirus curve and lower the mortality rate. However, controlling and preventing the spread of COVID-19 has met challenges due to uncertain waves and peaks in the number of cases. The evolution of SARS-CoV-2, which led to the emergence of several variants, worsened the situation. Meanwhile, the policymakers must consider the issues that arise, which include readiness in healthcare systems, resource crises, economic burden and community welfare (Khankeh et al., 2021). Forecasting during the pandemic is essential to have an effective decision-making system that can relieve the impact of COVID-19, as the prediction helps identify possible changes in the future.

The COVID-19 dataset is collected as a series of time series data. Time series forecasting methods, which predict the futuristic outcomes based on historically timestamped data, are often employed to estimate the spread of an epidemic. Classical time series models such as autoregressive (AR), exponential, moving average (MA) (Rahimi et al., 2021), and vector autoregressive (VAR) models (Gomez-Cravioto et al., 2021) are used to model the COVID-19 cases in the previous studies. Other than that, the development of artificial intelligence (AI) nowadays added machine learning (ML) and deep learning (DL) techniques to prediction applications, including time series problems. Thus, research exists on COVID-19 data modelling using time series algorithms such as artificial neural networks (ANN). A neural network is a series of algorithms inspired by the operation framework of the human brain that uses interconnected nodes or neurons in a structure with layers. Machine learning enables computer programs to be trained to recognise patterns and solve complex problems, thus assessing the complexity of the COVID-19 time series data.

According to Istaiteh et al. (2020), nonlinear approaches are recommended to deal with high variability and transient time series data. COVID-19 datasets collected have a nature of nonlinearity and non-stationarity due to the unstable trend of the COVID-19 outbreak. A nonlinear dynamic neural network model suitable for the nonlinear dataset, namely the nonlinear autoregressive neural network (NARNN), is developed by combining the nonlinear autoregressive (NAR) model and the multilayer feedforward artificial neural network (ANN) (Liu et al., 2021). The ANN model can extract nonlinear relationships in the data by applying activation functions between every two layers. The NARNN model, as a variant of ANN, has been proposed in the literature as a modelling tool for the COVID-19 dataset. This technique uses historical data based on the number of lags in the time series data as the lagged input (Adedeji et al., 2019).

Benmouiza and Cheknane (2013) state that NARNN has high similarity with a Multilayer Perceptron (MLP), whereby both models are composed of neurons in an input layer, one or more hidden layers and an output layer. Both models are feedforward neural networks with backpropagation training algorithms, but the NARNN model consists of the feedback layers to approximate the nonlinear function. Therefore, the NARNN model is a widely used ANN for modelling dynamic structures and forecasting nonlinear time series. The architecture of the NARNN is represented by three layers: the input, hidden and output layers. A few network configuration parameters such as training algorithms, activation function, feedback delays or past values and the number of neurons in the hidden layer must be decided to build the NARNN model. The ultimate goal of the research is to obtain the best-performing model that suits the data with minimum error since forecasting accuracy is always a matter of concern when providing insights.

The NARNN model is a neural network with feedback connections where future values of a time series (the current output) depend on the past values (the past output). Hence, the number of past values and feedback delays differ from the current output. Patil et al. (2013), in their study to predict sea surface temperature with the NARNN model, concluded that it would be more reliable to train the model with more past input rather than a smaller data segment to improve the fitting flexibility. In the research conducted by Pawlus et al. (2013) on vehicle collision modelling using the NARNN model, it has also been found that the increasing number of feedback delays will level up the model's performance. Raturi and Sargsyan (2018), when studying oil and gas price forecasting using the NARNN model, increased the past values from 50 to 100. As a result, there is a significant improvement in validation and regression and a palpable reduction of output error nearly two times. Meanwhile, there is also a reduction of autocorrelation error by almost 2.5 times.

However, in some model architectures, the highest number of past values is not optimal. This scenario can be discovered in the study by Blanchard and Samanta (2020) regarding wind speed forecasting using the NARNN model. They have compared the past values by setting the values range from 2 to 72. Consequently, three out of five experiments using datasets with 60 past values as input have generated the best model performance even though 60 is not the highest number of past values. As the optimal number of past values is not necessarily high (Chi, 2021b), the study of soybeans global price time series forecasting has used 3 feedback delays with 8 hidden neurons in the NARNN model. Regarding COVID-19 forecasting, Ghazaly et al. (2020) set the number of delays from 1: 2 to 1: 10 for the NARNN model to forecast COVID-19 cases. The result reflects that the model with 1: 6 feedback delays has the lowest error.

The number of hidden neurons is one of the important parameters of neural network models. In the neural network training process, the network weight and neuron bias are adjusted iteratively to optimise the accuracy of future value prediction (Sarkar et al., 2019). According to Olney et al. (2022), the number r of neurons in the hidden layer defines the configuration of the NARNN model by carrying a weight term for each delay state and a bias term. A number of hidden neurons also accounts for the complexity of a neural network, where a network system with more neurons would be more complex. A greater number of neurons not only longer the training time but also weakens its generalisation ability. However, reducing the number of neurons would lower the model fault tolerance (Chang et al., 2022). Hence, the suitable number of hidden neurons for a neural network model affects the network performance.

In previous research regarding COVID-19 forecasting using the NARNN model, a different number of hidden neurons is set to train the model as the complexity needed for the model is different for each dataset training process. Ghazaly et al. (2020) tested the NARNN model's performance with 1 to 4 hidden neurons in conducting COVID-19 forecasting. It turns out that a network with 3 hidden neurons recorded with the lowest MAPE becomes the most appropriate configuration. During a simulation study, Saliaj and Nissi (2022) found that configuring a maximum of 2 hidden neurons provides the best NAR neural network performance. However, in real COVID-19 time series forecasting, nea work structure with five hidden neurons gives a more accurate result with lower RMSE. Thus, a model with five hidden neurons becomes the choice of parameter. Therefore, the number of hidden neurons need to be tuned to the optimum depending on different requirement of network complexity.

2. Materials and Methods

2.1 Data set

The COVID-19 dataset used in real data analysis is collected from the data source hosted by the Center of System Science and Engineering (CSSE) at Johns Hopkins University (JHU CSSE, 2020). The dataset is stored in the GitHub repository. This data source shows the number of confirmed COVID-19 cases, deaths and recoveries for all affected countries. In the study, only the COVID-19 confirmed cases will be used for forecasting the number of confirmed cases. The data extracted for investigation is from 5 April 2020 to 28 September 2022, even though the repository has aggregated COVID-19 time series data since 22 January 2020. Time series data of daily COVID-19 cases in Malaysia is extracted for data analysis. The raw dataset of COVID-19 cases will be further extracted in the sample sizes of 100, 500 and 900.

Before modelling, the time series data is divided into training, validation and testing sets based on the ratio of 70%, 15% and 15%, respectively. The training dataset is used during the neural network training stage to update the network weights and biases and determine the gradient to produce a well-generalised network model. The validation dataset ensures that the network is sufficiently trained and not overtrained. When the validation set error is at the minimum location and stops improving, the training is halted, and the network weights and biases are saved. Training neural networks must be stopped at suitable iterations to prevent overfitting and underfitting networks. The testing dataset is not involved in the training stage. The training dataset's purpose is to measure the forecasting performance of the NARNN model after all the forecasting work has been done (Zhou et al., 2016).

2.2 Nonlinear Autoregressive Neural Network (NARNN) Model

Artificial neural network (ANN) is inspired by the neural architecture of the brain system to tackle complex machine learning tasks and real-world problems. By mimicking the operation of the brain system to acquire knowledge with interconnected neurons, a neural network algorithm capable of learning based on past information is formed (Jeatrakul & Wong, 2009). The ultimate purpose of the neural network is to enable the trained model to become versatile and come out with valuable outputs based on the information extracted from the previous data. As a variant of ANN, the NARNN model is widely used for time series prediction from historical data (Dhamodharavadhani et al., 2020).

Historical data is known as past values or feedback delays in the NARNN model. The foundation supports the NARNN model as the new expected time series outcome is generated by re-feeding the lagged value of the time series to the model. The equation of the NARNN model below shows how p past values contribute to the current estimation of $y(t)$.

$$y(t) = f\big(y(t-1), y(t-2), y(t-3), \dots, y(t-p)\big) + \varepsilon(t) \tag{1}$$

where y represents the time series value, and t stands for the time period. For instance, $y(t)$ is the time series value at time t, also known as the present value, while $y(t-1)$ is the past value of the first lag. p denotes the number of past values or feedback delays, and f denotes the activation function. $\varepsilon(t)$ is the error term, which depends greatly on the learning ability of an algorithm, as noise in the data that cannot be accessed will be accumulated in this term.

The NARNN model also extends the idea of the autoregressive (AR) process (Chi, 2021a). Similarly, the AR model also uses the p past values, $y(t-1), y(t-2), y(t-3), \dots, y(t-p)$ to explain the current value of the time series, $y(t)$. However, the AR model is linear. According to Kumar and Murugan (2018), most systems in the real world are nonlinear, neural networks become a suitable tool to fit the systems. Saliaj and Nissi (2022) state that the NARNN model, which inherits the features of ANN, is said to be the AR process with nonlinear functions. Thus, NARNN is able to map the past observations, $y(t-1), y(t-2), y(t-3), \dots, y(t-p)$ to the present value, $y(t)$ in a nonlinear function through neurons in the hidden layers of neural network (Chi, 2021a). In short, the assimilation of the AR model with the artificial neural network enables the NARNN model to handle complex nonlinear datasets while following the AR process in utilising the series of past values to predict future behaviour.

Probing into the architecture of the NARNN model, it is a feed-forward neural network with multilayer perceptron (MLP). The model architecture comprises at least three layers, including a default input and output layer, along with one or more hidden layer(s), taking a number of neurons and delays. The input layer is the first layer to receive input data, while the output layer is the last layer to obtain results or solutions. Between these two layers, the data flows through the intermediate hidden layer(s) that contain a collection of neurons to detect the pattern of the data and the nonlinear relationships (Saliaj & Nissi, 2022).

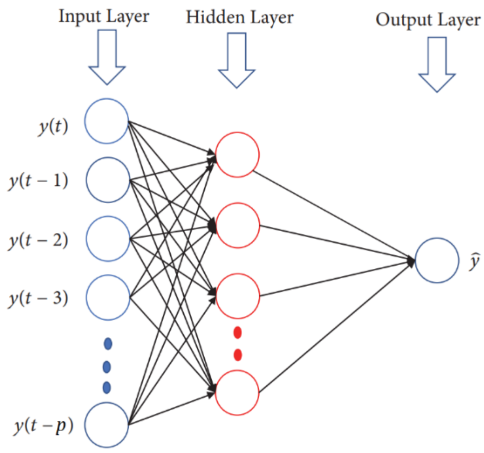


Figure 1. NARNN model architecture

The trained feedforward algorithm can be mathematically described as Equation (2) (Chi, 2021a). Each neuron has weight terms for every delay and a single bias term with a nonlinear activation function. The formulation is equivalent to the estimated output value or response after the training.

$$\hat{y}(t) = \alpha_0 + \sum_{i=1}^{I} W_i f\left(\sum_{j=1}^{p} W_{ij} y(t-j) + \beta_i\right) \quad (2)$$

where $\hat{y}(t)$ is the estimated value of output; $y(t-j)$ is input past values, $j = 1, 2, \ldots, p$; f is nonlinear activation function; W_{ij} is input-to-hidden layer weights; W_i is hidden-to-output weights; I is the total number of neurons in the hidden layer; β_i is the biases in the ith neuron in the hidden layer, and α_0 is the bias in the neurons in the output layer.

2.2.1 Data Normalisation

Data normalisation is normalising the dataset, especially the nonlinear data, into values between 0 and 1 using the minimum-maximum approach of subtracting the minimum value from the observed value and dividing by the difference between the maximum and minimum values. The normalising process is usually done before training, whereby processed data is fed into the network. This process will render the computation more convenient and more straightforward to speed up the update of weights to the minimum error as it converges. Moreover, normalised data is easier to distinguish since the distance between the values is more probable (Ghazaly et al., 2020). Equation (3) is to generate the normalised value from the original value.

$$y_{norm} = \frac{y_i - y_{min}}{y_{max} - y_{min}} \quad (5)$$

After getting the output from the trained network, the denormalisation step can be done using Equation (4).

$$y_i = y_{norm}(y_{max} - y_{min}) + y_{min} \quad (4)$$

where y_{norm} is the normalised or scaled value; y_i is the original or unscaled value; y_{min} is the minimum value, and y_{max} is the maximum value in the dataset.

2.2.2 Activation Functions

The activation function is a core building block of a neural network as it plays an important role in the learning ability of neural networks. It controls whether to activate or deactivate a neuron by computing the weighted sum and adding more bias (Kaleeswaran et al., 2020). Besides, activation functions introduce nonlinearity into a neuron's output, thus determining the output processed by the neuron. With such a nonlinear nature, the neural network model can learn and handle complex tasks.

Each activation function has mathematical functions, which will indicate how the activation functions transform input signals from neurons of previous layers to produce the informative output. Such different form of activation functions differs from the output contributed by the model.

There are many activation functions available for the NARNN model. Different activation functions will significantly affect the model's overall performance. The tansig function is commonly used for multilayer networks. It is related to a bipolar sigmoid whose output ranges from –1 to +1. Tansig is said to be mathematically equivalent to hyperbolic tangent function (tanh) (Dorofki et al., 2012). Both functions are s-shaped and zero-centered. However, it is found that tansig runs faster yet outputs result in lower numerical differences. Such compromise between speed and shape flexibility of activation function suits neural networks that prioritise speed over the exact shape of the activation function. Equation (5) is the mathematical formulation of tansig function.

$$tansig(x) = \frac{2}{1 + e^{-2x}} - 1 \tag{5}$$

According to Sarkar et al. (2019), tansig functions have stronger gradients than logsig. Hence lowering the possibility of neuron saturation. Neuron saturation is an issue that negatively affects a neural network's learning ability, whereby the neurons' output sticks to the asymptotic ends of the range of the activation function (Rakitianskaia & Engelbrecht, 2015). Moreover, the tansig function is able to train faster when working along with the backpropagation algorithm.

2.2.3 Number of Past Values

The number of past values is also called feedback delays and is often shown as 1: p, where p is the number of feedback delays (Blanchard & Samanta, 2020). Since the NARNN model uses past observations to predict the future value, the number of past values used for model building significantly affects the result. Based on the model equation of NARNN (Equation (1)), a series of past values are denoted as $y(t - 1), y(t - 2), \dots, y(t - p)$, where each of them is observations at the corresponding time $t - 1, t - 2, \dots, t - p$ respectively. They become the input data for the NARNN model to predict the output value $\hat{y}(t)$, which is the estimated value at future time t. Following the research by Zheng et al. (2022), the range of feedback delays is within the number of values in the training set.

With a different number of past values inserted, the complexity of the model with change accordingly as well. The more the number of past values, the costlier the computations (Ghazaly et al., 2020). However, an inadequate amount of past information will cause poor forecasting results. The previous study by Molino-Minero-Re et al. (2014) also found that a complex NARNN model with a high number of past values does not necessarily perform better. Like other parameters, the number of past values is optimised through trial and error, and the architecture that gives the lowest error will be the optimal one. To investigate the most appropriate number of past values for the model, the study will take the number of past values ranges between 1 to 10 as the unit of experiment.

2.2.4 Number of Hidden Neurons

In neural networks, hidden layers are located between the input and output layers. Hidden layers are where hidden neurons, also known as hidden nodes, carry weight terms and produce output through an activation function. The neurons are part of the architecture that builds up the artificial network so that the neural network can perform complex tasks, which include learning from historical data to bring insights to new data (Saliaj & Nissi, 2022). In this study, only one hidden layer exists between the input and output layers of the NARNN model investigated. The number of hidden neurons in the only hidden layer needs to be determined so that the NARNN model can be more efficient in performing its learning process.

According to research by Sheela and Deepa (2013) on approaches to fix a number of hidden neurons in neural networks, the researchers use trial and error procedures. The lower number of hidden neurons will be set in the beginning, and the performance of the neural network model will be observed. Subsequently, more hidden neurons will be added to optimise the neural network performance further. An optimal number of hidden neurons coordinates well with the complexity of the neural network task. It is neither excessive nor results in overfitting or too few and causes an underfitting problem (Liu et al., 2007). When overfitting, the excessive neuron connection is unnecessary until the neural network's problem-solving ability does not improve anymore, but the random regularity in the training patterns is captured, causing the error to increase. However, too few hidden neurons to sense the signal in a complex dataset will result in underfitting where the model is not trained sufficiently to capture the variability of the data (Jabbar & Khan, 2015). In choosing the optimal number of hidden neurons, the MSE of each training of the NARNN model with different numbers of hidden neurons ranging from 1 to 10 is repeatedly computed and observed.

2.3 Performance Evaluation Measures

The model's accuracy is determined by comparing the actual and estimated values, such as computation in mean absolute error (MAE) and mean square error (MSE). The smaller the difference between the real observed value and forecast value, the closer the error value is to zero. Therefore, the better the performance of the model. Lower values of these two performance evaluation indicators reflect better forecasting results from the model.

MSE and MAE are computed to check the performance accuracy. MSE measures the variability of errors. The smaller the MSE, the better the prediction performance. The formula is shown as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2 \tag{6}$$

However, the MSE penalises extreme errors while forecasting and is sensitive to data transformations and scale change. On the other hand, MAE does not penalise extreme errors. The formula of MAE is written as:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{7}$$

where y_t is the observation at time t and $\hat{y}_t$ is the predicted values. MAE measures the average absolute deviation of forecasted values from original values. Like the MSE, the smallest MAE has the better forecast performance.

3. Results and Discussion

The raw data of COVID-19 daily confirmed cases in Malaysia is collected from the COVID-19 data repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. The data in the repository has been aggregated since the outbreak of COVID-19 on 22 January 2020 and is still actively updated. In this study, the sample used will be 100, 500 and 900. Therefore, data from 100, 500 and 900 days before 21 September 2022 are extracted for the NARNN model training, validation and testing procedure. On the other hand, the last 7 days of data from 22 September 2022 to 28 September 2022 is utilised as the out-sample to evaluate the performance of multi-step prediction. Hence, a total of 907 days of samples were gathered for this study. The illustrations of Figure 2 show the time series plot of daily confirmed cases extracted in different sample sizes.

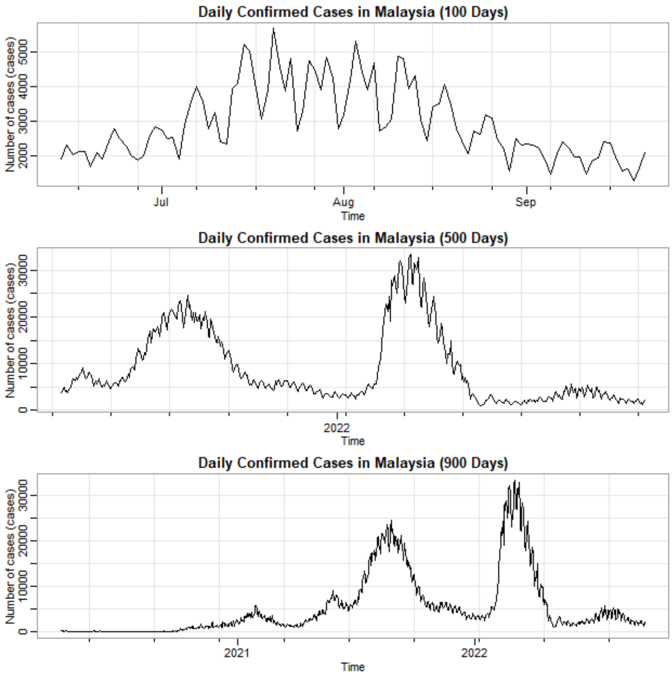


Figure 2. Time series plot of daily confirmed cases extracted in sample sizes 100, 500 and 900

Based on Figure 2, the COVID-19 trend in Malaysia can be observed the cases exhibit a steady increase from the end of 2020 until January 2021. Thus, a small peak is portrayed in the time series plot during this period. In the third quarter of 2021, the country experienced the second-highest peak of COVID-19 cases. The record of the maximum number of COVID-19 cases in Malaysia, on 5 March 2022, with 33406 cases.

The data are normalised before being fed into the neural network model to enhance the efficiency of time series forecasting. The scaled data in the same range of values speeds up the learning process and is appropriate to be fed into any training algorithm. In this study, min-max normalisation is implemented. The data are scaled to a range of [0,1] by subtracting the original value from a minimum number of confirmed cases and then dividing it by the difference between the maximum and minimum number of confirmed cases. The sample sizes are split into the training, validation and testing subsets based on the ratio 70:15:15. The information about each subset of data is tabulated in Table 1.

Table 1. Training, validation, and testing subsets

Sample Size	Data Subset	Ratio	Date	Number of Data
100	Training	70	14 June 2022 to 22 August 2022	70
	Validation	15	23 August 2022 to 6 September 2022	15
	Testing	15	7 September 2022 to 21 September 2022	15
500	Training	70	10 May 2021 to 20 March 2022	350
	Validation	15	21 March 2022 to 8 July 2022	75
	Testing	15	9 July 2022 to 21 September 2022	75
900	Training	70	5 April 2020 to 25 December 2021	630
	Validation	15	26 December 2021 to 9 May 2022	135
	Testing	15	10 May 2022 to 21 September 2022	135

The processed data are imported to the MATLAB 2022b software to be fitted and analysed in the NARNN model. The NARNN model is set to possess three layers and train with Levenberg Marquardt (LM) training algorithm. The activation function for the NARNN model is specified as tansig. The number of past values and hidden neurons is being tuned and adjusted to the optimal so that the NARNN model developed could have the most optimal setting. Both the number of past values and hidden neurons are chosen in the range from 1 to 10. The model's performance is evaluated as the past value, and a fixed number of 10 hidden neurons are manipulated. After selecting the past value, the number of hidden neurons and the corresponding optimal setting are tuned.

For each training, the number of maximum epochs is regulated at 1000 epochs, and the training will come to a halt when either the maximum epoch is achieved or the validation error increases consecutively for six iterations. 100 runs are carried out for each setting to obtain significant results. The outcome with the least average overall MSE is determined as the fixation of a number of past values and hidden neurons. The series of Table 2 shows the mean MSE obtained for each combination of parameters after training the COVID-19 datasets of Malaysia. In contrast, the charts of Figure 3 portray the combined information from the tables with bar graph representing the performance of the NARNN model manipulated with different numbers of past values and a line graph plotting the average MSE of the NARNN model with changing number of hidden neurons when the most appropriate past value is fixed.

Table 2. Average MSE of NARNN model with different numbers of past values and hidden neurons ranging from 1 to 10

Sample Size	Number of past values	MSE	Number of hidden neurons	MSE
100	1:1	0.0004008	1	0.0002816
	1:2	0.0004020	2	0.0003251
	1:3	0.0003258	3	0.0002666
	1:4	0.0002495	4	0.0002092
	1:5	0.0002228	5	0.0002000
	1:6	0.0002254	6	0.0001848
	1:7	0.0003027	7	0.0001778
	1:8	0.0002465	8	0.0001837
	1:9	0.0002011	9	0.0001768
	1:10	**0.0001762**	10	**0.0001765**
500	1:1	0.01210	1	0.02349
	1:2	0.009206	2	0.03783
	1:3	0.01013	3	0.02078

	1:4	0.01163	4	0.01944
	1:5	0.008861	5	0.01383
	1:6	0.01838	6	0.01617
	1:7	0.01079	7	0.01293
	1:8	0.01246	8	**0.007324**
	1:9	0.01161	9	0.01160
	1:10	**0.006378**	10	0.01220
900	1:1	0.002023	1	0.004460
	1:2	0.001270	2	0.002997
	1:3	0.003962	3	0.007671
	1:4	0.001632	4	0.003804
	1:5	**0.001231**	5	**0.001403**
	1:6	0.003711	6	0.005019
	1:7	0.002122	7	0.002755
	1:8	0.001842	8	0.003631
	1:9	0.002937	9	0.003085
	1:10	0.001839	10	0.002592

Based on the result, the lowest average MSE exists at the past value of 1:10 and the number of neurons of 10 when the sample size is 100. At sample size 500, past values of 1:10 and number of neurons of 8 enable the model to perform at its best. Whereas for sample size 900, the network setting of 1:5 past value and 5 hidden neurons gives the least MSE.

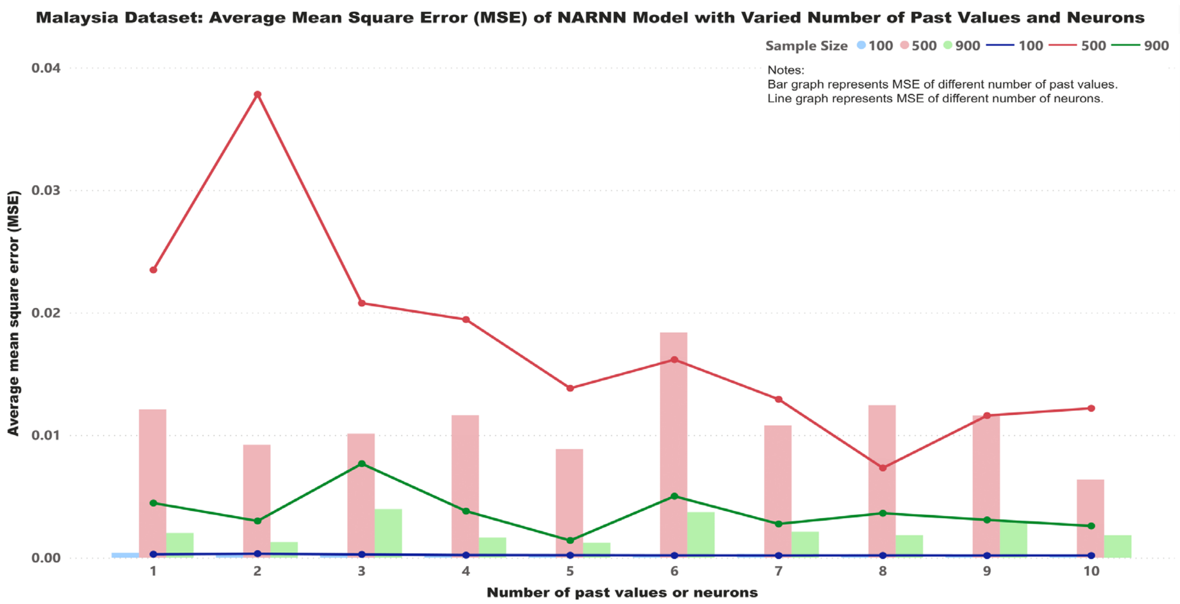
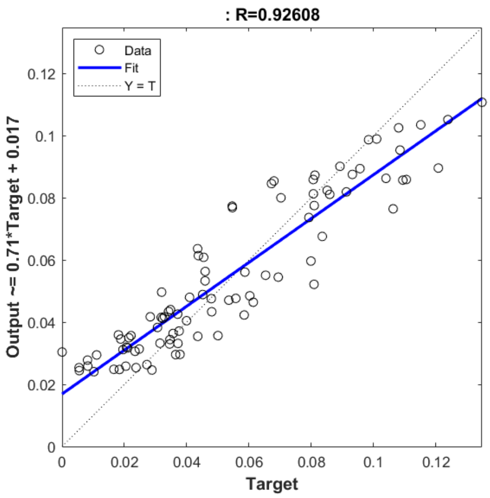


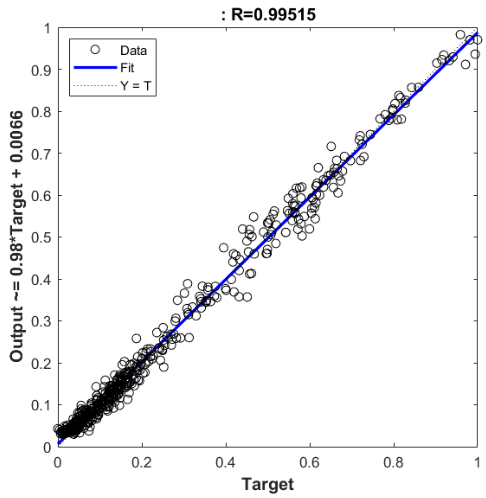
Figure 3. Average mean square error (MSE) of the NARNN model with different numbers of past values and hidden neurons ranging from 1 to 10

The white noise condition needs to be ensured before the result of training is accepted. The error autocorrelation plot illustrates the relationships between the prediction errors over time. The x-axis of the plot is the lag values, whereas the y-axis is plotted with the correlation values. When most of the autocorrelation values at different lags fall within the confidence limits indicated by the horizontal red dotted line, it can be deduced that the prediction errors over time are uncorrelated, and the condition of white noise is achieved. Hence, the model is adequate to be used as a prediction model (Namasudra et al., 2021). The only nonzero autocorrelation value that exceeds confidence limits is at zero lag.

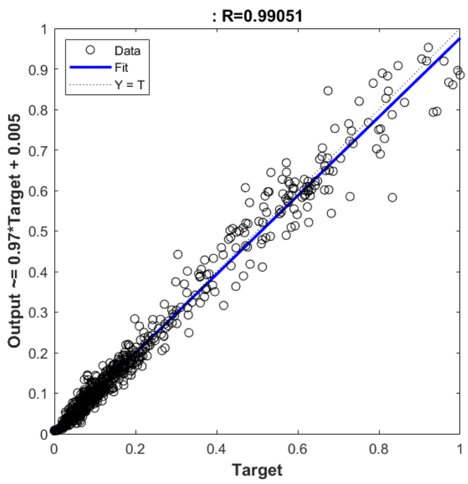
To obtain an adequate model appropriate for generating predicted values, the error autocorrelation plot is checked to see whether the white noise condition is achieved. If the model is inadequate, it is retrained until the desired condition is reached. The retraining process could vary the initial network weights and biases, improving the neural network's reliability. After the model achieves white noise, the performance of the neural network model is evaluated. The regression plot and time series response plot are produced. In regression plots, the targeted values are plotted against the output values. Data scattered around the 45-degree straight line of regression plots indicate a good fit for the model (Kumar & Murugan, 2013). On the other hand, time series response plots display the deviation between target and output values against time. The yellow lines indicate the intensity of errors. The results are shown in Figure 4 and Figure 5.



(a) $n = 100$

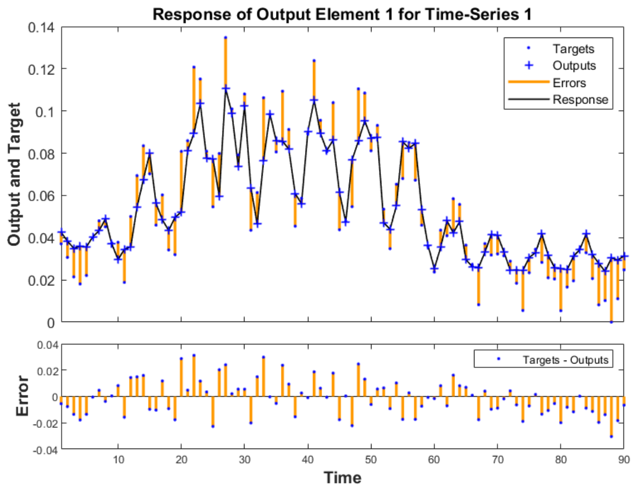


(b) $n = 500$

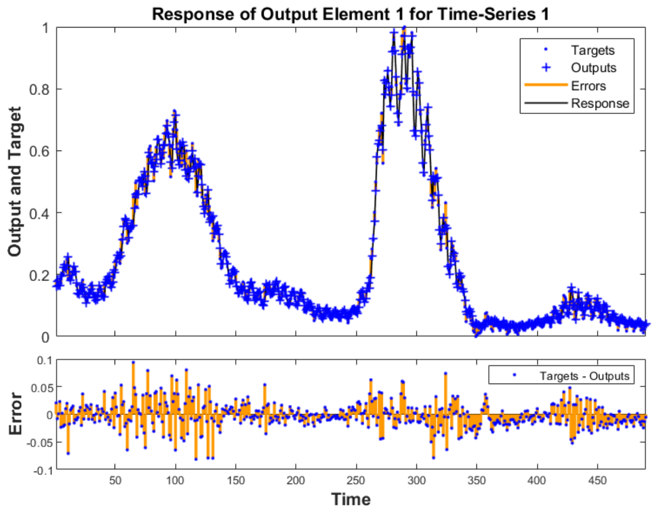


(c) $n = 900$

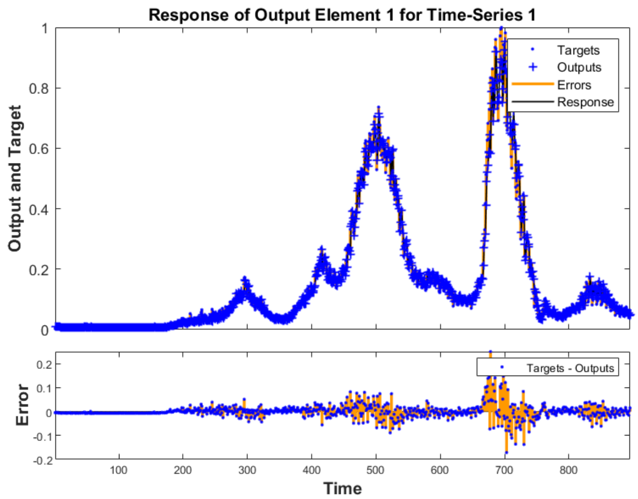
Figure 4. The adequacy of the model by using R-value (a) $n = 100$, (b) $n = 500$, (c) $n = 900$



(a) $n = 100$



(b) $n = 500$



(c) $n = 900$

Figure 5. Time series response plot (a) $n = 100$, (b) $n = 500$, (c) $n = 900$

Observing the time series response plot in Figure 5, some yellow lines show the deviation of output from the targeted value, thus producing some errors and the errors will be represented in MSE values. When the overall training, validation and testing MSE are computed, the values are observed to be small. These results indicate that the model could come out with output that is quite close to the normalised actual value. The R-value (Figure 4) and the mean square error (MSE) of the overall training, validation and testing steps are tabulated in Table 3. A lower MSE indicates better performance, while R-value close to 1 represents good overall performance (Saba & Elsheikh, 2020). From the plots in Figure 4, the R-value of each dataset is more than 0.8 and is very close to the value 1.

Table 3. R-value and mean square error (MSE) of overall training, validation and testing steps

Sample Size	MSE				R
	Overall	Training	Validation	Testing	
100	0.0001807	0.0001910	0.00009250	0.0002178	0.9261
500	0.0005705	0.0006566	0.0002943	0.0004411	0.9952
900	0.0008332	0.0003589	0.003556	0.0003303	0.9905

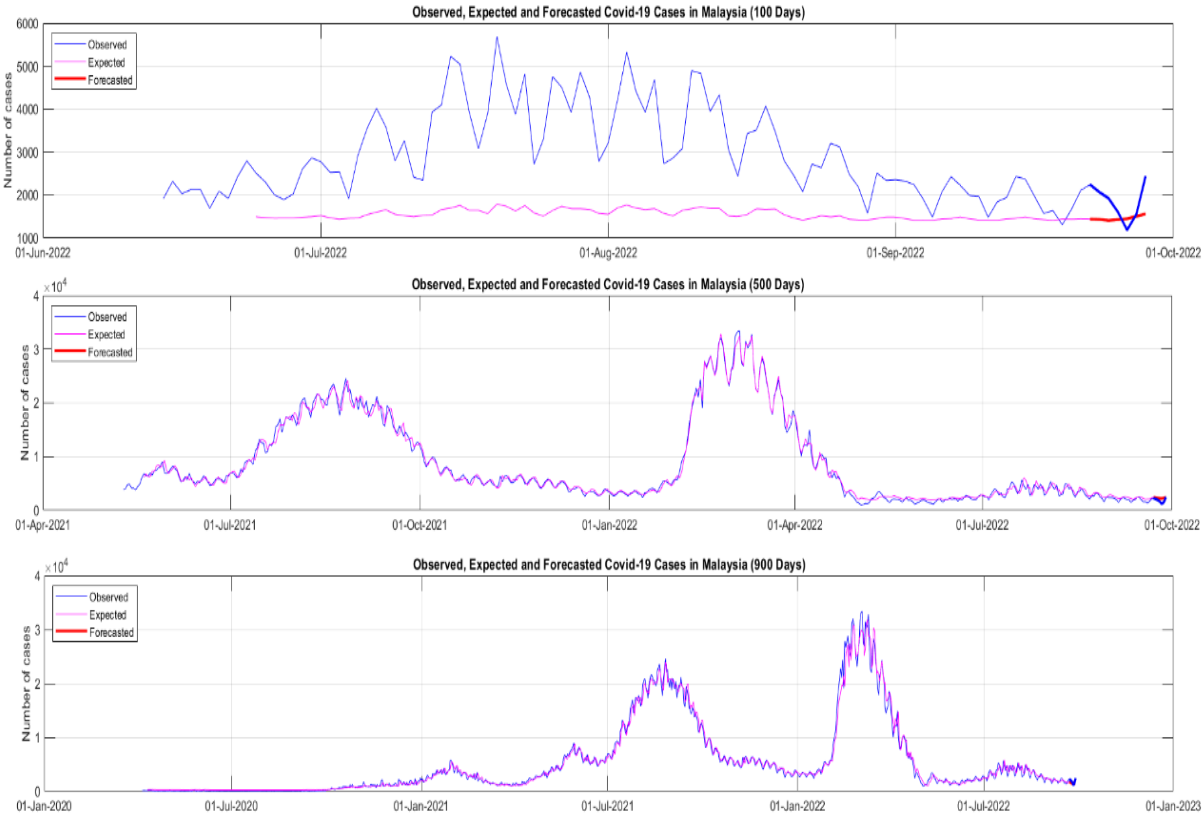


Figure 6. Observed, expected and forecasted COVID-19 cases

Table 4. MSE and MAE of 7-days-ahead forecasting by NARNN model.

Sample Size	MSE	MAE
100	315046.7549	476.0192
500	311607.06	460.43
900	**252147.79**	**409.83**

Multi-step ahead time series forecasting is conducted to forecast the number of COVID-19 confirmed cases in 7 days after 21 September 2022, from 22 September 2022 to 28 September 2022. The 7-day predicted values of COVID-19 confirmed cases in Malaysia are plotted along with previous forecasted values by the same NARNN model in Figure 6. In each graph, the 7-days ahead predicted values, also known as the output as a result of a close loop, the output of the current time step is used as an input for the next time step. NARNN models are plotted and connected with a bold red line. The magenta lines, labelled as predicted values, are produced by the NARNN model as a result of the training, validation and testing. The observed number of cases in blue lines are input values used for the training, validation and testing process of the NARNN model, except that the last 7 observed values in bolded blue lines are out samples to compare with the forecasted values. It is observed that most of the output values adhere to the actual values closely, with $n = 500$ and $n = 900$.

The MSE and MAE of the 7-day ahead forecast are also computed and tabulated in Table 4. Based on the result, it is observed that the best forecasting outcome for Malaysia datasets occurs when 900 samples are analysed.

4. Conclusion

The NARNN time series model is a feedforward neural network that can deal with complex time series problems, such as forecasting COVID-19 cases in different countries. However, the appropriate set of network configurations must be determined before a neural network can produce accurate and reliable results. The parameters of the NARNN model include activation function, training algorithm, number of past values, and number of hidden neurons. In this study, the training algorithm of the NARNN model is fixed as Levenberg-Marquardt (LM) algorithm since it is recognised as an efficient algorithm for the NARNN model in previous research. The activation function, tansig with the number of past values and hidden neurons, will be chosen from 1 to 10. A grid search procedure is conducted to determine the most optimal combination of parameters for NARNN models.

Real data of COVID-19 confirmed cases in Malaysia is collected and analysed. The raw data are preprocessed and extracted into different sample sizes (100, 500 and 900). Each prepared dataset is utilised as the input to train, validate and test the NARNN model with tansig as the activation function. The training, validation and testing of the built NARNN model is done until white noise is achieved by monitoring the autocorrelation and regression plots. From the output, it is noticed that the NARNN model with a tansig activation function and selected optimal past values, as well as hidden neurons, can provide prediction values close to actual values with low MSE and R-values near 1. In addition, the number of past values and hidden neurons affects the performance of the NARNN time series model, whereby tuning the number of past values and hidden neurons to the optimal value between 1 and 10 based on the complexity of respective datasets enables NARNN time series model to perform at lower MSE and provide more accurate results.

The data analysis proceeded with 7 days-ahead COVID-19 forecasting. Finally, the result shown with more data provides the best result in forecasting 7 days-ahead COVID-19 confirmed cases. As future work, the study can be further extended to investigate more parameters of the NARNN model in order to improve the forecasting performance of the NARNN model. Different activation functions can be applied in this context for the hidden and output layers since this study uses the same activation function. Besides, the number of past values and hidden neurons is tuned in the range of 1 to 10 in this research. The future study could work on a more comprehensive range of a number of past values and hidden neurons to observe the effect on NARNN time series model performance.

5. References

Adedeji, P. A., Akinlabi, S., Ajayi, O., & Madushele, N. (2019). Non-linear Autoregressive Neural Network (Narnet) with SSA Filtering for a University Energy Consumption Forecast. *Procedia Manufacturing*, 33:176–183.

Benmouiza, K. & Cheknane, A. (2013). Forecasting Hourly Global Solar Radiation Using Hybrid k-means and Nonlinear Autoregressive Neural Network Models. *Energy Conversion and Management*, 75:561–569.

Blanchard, T. & Samanta, B. (2020). Wind Speed Forecasting Using Neural Networks. *Wind Engineering*, 44(1):33–48.

Chang, T.-J., Cheng, S.-J., Hsu, C.-H., Miao, J.-M., & Chen, S.-F. (2022). Prognostics For Remaining Useful Life Estimation in Proton Exchange Membrane Fuel Cell by Dynamic Recurrent Neural Networks. *Energy Reports*, 8:9441–9452.

Chi, Y. N. (2021a). Modeling And Forecasting Long-Term Records Of Mean Sea Level At Grand Isle, Louisiana: SARIMA, NARNN, and Mixed SARIMA-NARNN Models. *Journal of Applied Data Sciences*, 2(2).

Chi, Y. N. (2021b). Time Series Forecasting of Global Price of Soybeans Using a Hybrid SARIMA and NARNN Model: Time Series Forecasting of Global Price Of Soybeans. *Data Science: Journal of Computing and Applied Informatics*, 5(2):85–101.

Dhamodharavadhani, S., Rathipriya, R., & Chatterjee, J. M. (2020). COVID-19 Mortality Rate Prediction for India Using Statistical Neural Network Models. *Frontiers in Public Health*, 8:441.

Dorofki, M., Elshafie, A. H., Jaafar, O., Karim, O. A., & Mastura, S. (2012). Comparison of Artificial Neural Network Transfer Functions Abilities to Simulate Extreme Runoff Data. *International Proceedings of Chemical, Biological and Environmental Engineering*, 33:39–44.

Ghazaly, N. M., Abdel-Fattah, M. A., & Abd El-Aziz, A. (2020). Novel Coronavirus Forecasting Model Using Nonlinear Autoregressive Artificial Neural Network. *Journal of Advanced Science*. 29. 1831-1849.

Gomez-Cravioto, D. A., Diaz-Ramos, R. E., Cantu-Ortiz, F. J., & Ceballos, H. G. (2021). Data Analysis and Forecasting of the COVID-19 Spread: A Comparison of Recurrent Neural Networks and Time Series Models. *Cognitive Computation*, pages 1–12.

Istaiteh, O., Owais, T., Al-Madi, N., & Abu-Soud, S. (2020). Machine Learning Approaches for COVID-19 Forecasting. *In 2020 International Conference on Intelligent Data Science Technologies and Applications (IDSTA)*, pages 50–57. IEEE.

Jabbar, H. & Khan, R. Z. (2015). Methods to Avoid Over-Fitting and Under-Fitting in Supervised Machine Learning (Comparative Study). *Computer Science, Communication and Instrumentation Devices*, p. 70.

Jeatrakul, P. & Wong, K. W. (2009). Comparing the performance of different neural networks for binary classification problems. *In 2009 Eighth International Symposium on Natural Language Processing*, pages 111–115. IEEE.

JHU CSSE (2020). COVID-19 data repository by the center for systems science and engineering (csse) at johns hopkins university. https://github.com/CSSEGISandData/COVID-19. Retrieved on 27 May, 2022.

Kaleeswaran, V., Dhamodharavadhani, S., & Rathipriya, R. (2020). A Comparative Study of Activation Functions and Training Algorithm of Nar Neural Network for Crop Prediction. *In 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1073–1077. IEEE.

Khankeh, H., Farrokhi, M., Roudini, J., Pourvakhshoori, N., Ahmadi, S., Abbasabadi- Arab, M., Bajerge, N. M., Farzinnia, B., Kolivand, P., Delshad, V., et al. (2021). Challenges to Manage

Pandemic of Coronavirus Disease (COVID-19) in Iran with A Special Situation: A Qualitative Multi-Method Study. *BMC Public Health*, 21(1):1–9.

Kumar, D. A. & Murugan, S. (2018). Performance Analysis of NARX Neural Network Backpropagation Algorithm by Various Training Functions for Time Series Data. *International Journal of Data Science*, 3(4):308–325.

Liu, Y., Starzyk, J. A., & Zhu, Z. (2007). Optimizing Number of Hidden Neurons in Neural Networks. *EeC*, 1(1):6.

Liu, Z., Zuo, J., Lv, R., Liu, S., & Wang, W. (2021). Coronavirus Epidemic (COVID-19) Prediction and Trend Analysis Based on Time Series. *In 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID)*, pages 35–38. IEEE.

Molino-Minero-Re, E., Cardoso-Mohedano, J. G., Ruiz-Fern´andez, A. C., & Sanchez-Cabeza, J.-A. (2014). Comparison Of Artificial Neural Networks and Harmonic Analysis for Sea Level Forecasting (Urias Coastal Lagoon, Mazatlan, Mexico). *Ciencias Marinas*, 40(4):251–261.

Namasudra, S., Dhamodharavadhani, S., & Rathipriya, R. (2021). Nonlinear Neural Network Based Forecasting Model for Predicting COVID-19 Cases. *Neural Processing Letters*, pages 1–21.

Olney, B., Mahmud, S., & Karam, R. (2022). Efficient Nonlinear Autoregressive Neural Network Architecture for Real-Time Biomedical Applications. *In 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS),* pages 411–414. IEEE.

Patil, K., Deo, M., Ghosh, S., & Ravichandran, M. (2013). Predicting Sea Surface Temperatures in The North Indian Ocean with Nonlinear Autoregressive Neural Networks. *International Journal of Oceanography*, 2013.

Pawlus, W., Karimi, H. R., & Robbersmyr, K. G. (2013). Data-Based Modeling of Vehicle Collisions By Nonlinear Autoregressive Model And Feedforward Neural Network. *Information Sciences*, 235:65–79.

Rahimi, I., Chen, F., & Gandomi, A. H. (2021). A Review on COVID-19 Forecasting Models. *Neural Computing and Applications,* pages 1–11.

Rakitianskaia, A. & Engelbrecht, A. (2015). Measuring Saturation in Neural Networks. *In 2015 IEEE Symposium Series on Computational Intelligence*, pages 1423–1430. IEEE.

Raturi, R. & Sargsyan, H. (2018). A Nonlinear Autoregressive Scheme for Time Series Prediction Via Artificial Neural Networks. *Journal of Computer and Communications*, 6(9):14–23.

Saba, A. I. & Elsheikh, A. H. (2020). Forecasting The Prevalence of COVID-19 Outbreak in Egypt Using Nonlinear Autoregressive Artificial Neural Networks. *Process Safety and Environmental Protection*, 141:1–8.

Saliaj, L. & Nissi, E. (2022). Artificial Neural Networks for COVID-19 Time Series Forecasting. *Open Journal of Statistics*, 12(2):277–290.

Sarkar, R., Julai, S., Hossain, S., Chong, W. T., & Rahman, M. (2019). A Comparative Study of Activation Functions of NAR And NARX Neural Network for Long-Term Wind Speed Forecasting in Malaysia. *Mathematical Problems in Engineering*, 2019.

Sheela, K. G. & Deepa, S. N. (2013). Review On Methods to Fix Number of Hidden Neurons in Neural Networks. *Mathematical Problems in Engineering*, 2013.

Wu, Y.-C., Chen, C.-S., and Chan, Y.-J. (2020). The Outbreak of COVID-19: An Overview. *Journal of the Chinese Medical Association*, 83(3):217.

Zheng, Y., Zhang, W., Xie, J., & Liu, Q. (2022). A Water Consumption Forecasting Model by Using a Nonlinear Autoregressive Network with Exogenous Inputs Based on Rough Attributes. *Water*, 14(3):329.

Zhou, L., Xia, J., Yu, L., Wang, Y., Shi, Y., Cai, S., & Nie, S. (2016). Using A Hybrid Model to Forecast the Prevalence of Schistosomiasis in Humans. *International Journal of Environmental Research and Public Health*, 13(4):355.