# Day Ahead Energy Consumption Forecasting Through Time-Series Neural Network

**R. Reshma Gopi[1], Chitra Annamalai[2]**
[1,2]School of Electrical Engineering, Vellore Institute of Technology, Vellore, India

**Article Info**

***Article history:***

Received Oct 31, 2022
Revised Feb 12, 2023
Accepted Mar 3, 2023

***Keyword:***

Demand forecasting
Artificial neural networks
Predictive models
Power demand
Recurrent neural networks

**ABSTRACT**

Demand response management through appliance scheduling can effectively decrease electricity bills. Similarly, it can decrease the peak demand of both consumers and the utility grid. However, prior knowledge of the load profile of the consumer is required for effective appliance scheduling. This work developed a novel time-series forecasting model within the shallow neural network framework to predict the load curve for optimal planning of demand response by a nonlinear autoregressive network with exogenous input network. The models were tested on a dataset including 744 samples obtained at one-hour intervals using a smart meter. The algorithms such as Levenberg-Marquard, scaled conjugate gradient, and Bayesian regularisation were applied to train the model. In particular, finding the optimal combination of the training algorithm, the hidden number of neurons, and the delay parameter was of importance. The results were compared with conventional seasonal autoregressive integrated moving average statistical model and a long short-term memory network. The results indicated that the optimum methodology is a nonlinear autoregressive network with exogenous inputs using the Bayesian regularisation algorithm, which has the lowest MSE value in the training and testing phases of 0.0031 and 0.0029, respectively. It is practical to continue designing artificial neural networks to analyse hourly load consumption in the context of the positive outcomes acquired.



***Corresponding Author:***

Chitra Annamalai,
School of Electrical Engineering,
Vellore Institute of Technology,
Vellore, India.
Email: chitra.a@vit.ac.in

## 1. INTRODUCTION

Rapid urbanisation and increased industrial production over the past decade have significantly increased the demand for electricity. Electricity generation, which has historically relied heavily on non-renewable and non-green energy sources such as coal and oil, is gradually shifting to renewable and green energy sources such as wind, solar, and nuclear. The transition to renewable and environmentally friendly energy sources for power generation also necessitates an examination of electricity usage patterns. The energy usage patterns such as the load curve illustrate the fluctuation in demand over a particular period. Accurate load forecasting that facilitates optimal decision-making is essential for assuring the safety, dependability, and economic functioning of the power system [1]. Effective operation and management of supplies by utility companies, increase in profitability and efficiency of power generation and distribution companies, capacity planning and operation to provide a steady supply of energy for all consumers, lower utility bills, reduced greenhouse gas emissions, and less impact on the climate from increased energy use are just a few benefits that may be realised through more efficient energy utilisation [2,3]. Enhancing forecasting accuracy helps conserve energy resources, assesses the effectiveness of managing the power supply, and consequently boosts the revenues of energy firms.

The technological and financial factors influence the requirement for precise electricity consumption forecasts. The four factors are the key determinants of electrical load are economic factors, time factors, weather, and random factors. Economic factors include investments in the infrastructure of the facility through the construction of new buildings, labs, and experiments that increase the demand for the electric grid. Seasonal effects, weekly-daily cycles, and holidays are the time factors that influence the electric load. Weather-sensitive loads, such as ventilation, heating, and air conditioning equipment, will have a bigger effect on smaller industrial/institutional power systems because these tend to be the system's larger loads. Humidity, solar irradiance, wind speed, barometric pressure, and precipitation are some weather variables that impact the hourly load profile for electricity. For example, to remove more moisture from the conditioned air on days with high humidity, cooling equipment will need to run for longer duty cycles. Long periods of high sun irradiation will radiatively heat buildings' interiors, requiring the cooling system to run longer and with less variety. As a result of precipitation, the cooling load is probably reduced as the air temperature drops. Random elements that affect the electrical load profile include all additional random disruptions in the load pattern that cannot be accounted for by economic, time, and meteorological considerations. The facility's load profile may be significantly impacted by other disruptions such as widespread employee absences (caused by illness, bad weather, etc.) and planned or unscheduled power system outages. These disturbances may include substantial loads with erratic operation schedules, making prediction challenging. Buildings are therefore regarded as the world's top energy consumers, with building energy consumption in the United States rising from 33.7% to 75% between 1980 and 2022.

In India, the Central Electricity Authority (CEA), a constitutional agency under the central ministry of electricity, is responsible for estimating energy demand periodically. The CEA releases a Load Generation Balance Report (LGBR) every year that includes estimates for the following year as well as zonal and state-level monthly energy demand statistics for the preceding year. CEA uses a straightforward trend-based model to predict monthly estimates of total and peak energy consumption for the upcoming twelve months. The inadequacies of the CEA model for neglecting the impacts of seasonality, non-stationarity, and uncertainty contained in power statistics were noted by researchers. These problems frequently result in an imbalance between anticipated and real energy consumption, which raises the financial risk for organisations that produce and distribute energy. Distribution businesses, for instance, are compelled to buy power from the deregulated power market at a higher price, decreasing profit margins, if projections are lower than actual demand. The forecasting methods are classified into three: statistical models, machine learning models, and multi-agent models. The moving average method, regression models, decomposition models, exponential smoothing, Autoregressive Integrated Moving Average (ARIMA) models, dynamic regression models, Generalised Autoregressive Conditional Heteroscedastic (GARCH) models, and Vector Autoregressive (VAR) models are some examples of statistical models. Support vector machines and artificial neural networks (perceptron, recurrent, fuzzy, extreme learning, or deep learning) are examples of machine learning models that are based on artificial computation techniques. Multi-agent approaches include equilibrium, game-theoretic, structural, and multi-agent simulations. Artificial Neural Network (ANN) is commonly used in electrical applications such as load forecasting and power price forecasting due to its capacity to handle non-linear relationship problems with greater precision [4]. Machine learning is categorised as a subfield of artificial intelligence. Deep learning is a subfield of machine learning, and neural networks provide the foundation of deep learning algorithms. The two main categories of machine learning approaches are supervised and unsupervised. Methods associated with the supervised learning paradigm identify objects from a pool based on a set of known features. By recognising the similarities between objects in a batch, unsupervised learning approaches generate groups and use them to classify unknowns. Reinforcement learning is comparable to supervised learning. However, unlike supervised learning, it does not rely on training samples but rather on trial and error. The ideal solution emerges from a series of successful outcomes for a particular problem [5].

Many studies have been published in recent articles. Alhussein et al.[6] proposed a hybrid model of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) to observe a load curve over four periods such as overnight, breakfast, daytime, and evening. CNN layers are used to extract features from the input data, whereas LSTM layers are used to extract information about sequence patterns. A hybrid strategy for estimating short-term electricity load was put forth by Bashir et al.[7]. The use of the prophet model for training linear data and the LSTM model for training non-linear data is a major contribution of this study. Back Propagation Neural Network (BPNN) training is applied on the Prophet and LSTM forecasted data to increase the accuracy. Cross-channel-communication (C3) enabled technology was used by Saeed et al. [8]. The CNN-LSTM algorithm was used to forecast the daily electricity load. After each convolutional layer, channel communication is made possible via a C3 block. The case study used data sets from the Independent System Operator New England (ISONE) and the New York Independent System Operator (NYISO). A bidirectional long short-term memory-based sequence-to-sequence (Bi-LSTM S2S) model for the day-ahead peak load forecasting was presented by Mughees et al. [9]. The data from the past and future peak demand values are stored in the Bi LSTM layer by iteratively going through the input sequence in both forward and reverse directions. A data collection with 818-time steps was used in this work. In comparison to LSTM and Levenberg-Marquardt backpropagation artificial neural network (LMBP-ANN) models, the findings showed that the proposed model has lower values of Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE). He [10] analysed the deep learning paradigm, which contains CNN and Recurrent Neural Networks (RNN) using LSTM cells to make hourly predictions 24 hours ahead of the load. The hourly load values of North China from February $10^{th}$, 2000 to December $31^{st}$, 2012 were employed in this analysis. The outcomes showed that the proposed deep model outperformed more established techniques such as Support Vector Regression (SVR) and Linear Regression (LR). Mubashar et al. [11] proposed a novel model based on LSTM techniques. The authors used a dataset that consists of the load consumption of 10 consumers. An RMSE propagation optimiser was used to propagate the error. The results proved that LSTM gives less Mean Absolute Error (MAE) compared to Autoregressive Moving Average (ARMA) and exponential smoothing models in smart grid systems. Ozer et al. [12] presented a model based on LSTM to forecast load curves in cases such as newly built homes or newly installed meters where historical consumption data are not available. A cross-correlation-based transfer learning approach is suggested to select the most appropriate transfer learning data that is similar to the original data. Data used for transfer learning are gathered from nearby structures identical to one another.

## 2. RESEARCH METHOD

The forecast for future energy consumption enables customers to evaluate their consumption patterns and, whenever possible, move their energy usage to off-peak times. An accurate forecast of energy consumption allows energy users to correlate their present consumption pattern with their future energy consumption. By being aware of their energy consumption and future estimates, these users may be able to benefit from the forecasting algorithms and control their energy costs more effectively. The first Artificial Neural Networks have been developed in 1943 [13]. ANNs are computer programmes designed to collect information in a manner comparable to the human brain and are inspired by the way in which the human brain processes information [14]. The fundamental component of the ANN is an artificial neuron. The structure of the single-node ANN is as shown in Figure 1.

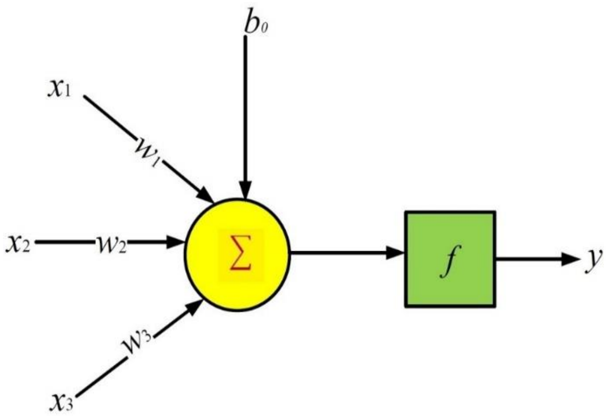

Figure 1. Composition of a single-node ANN

The neurons in a multi-layer perceptron are arranged in layers. Hidden layers are the layers between the input nodes and the output layer. The inputs that an ANN receives are weighted by variables referred to as weight (*w*). Additionally, each neuron will have a bias ($b_0$), another fundamental structural component. Bias is comparable to the intercept. The addition of bias nodes increases the adaptability of the model to the data. The bias and the sum of the weights and inputs will be added to create the input argument for the so-called activation function (*f*). A node gets inputs *x*, multiplies each input by a weight *w*, adds a bias $b_0$, and applies a transformation function *f* to get an output *y* [15]. The log-sigmoid, hyperbolic tangent sigmoid (tansig), and linear functions are some of the most commonly used transformation functions. Based on the period of forecast, the forecasting has been grouped into short-term forecasting, medium-term forecasting, and long-term forecasting. The short-term load forecast shows the expected electric load for a period of several hours to several days. The electric load projection for a period of a few months is represented by the medium-term load forecast. The long-term load forecast depicts the anticipated electric load over years [16]. The scheduling of energy flow between power sources, loads, and storage devices uses short-term forecasting. Price determination, load dispatch, and maintenance planning are all governed by the medium-term and long-term forecasting.

Before ANN, statistical models such as ARIMA and Seasonal Autoregressive Integrated Moving Average (SARIMA) were used to forecast energy consumption. In this study, a two-layer feedforward neural network is presented to model the load curve using the energy consumption dataset. The main contribution of the paper is as follows: An RNN-based model is developed to forecast the energy consumption. The knowledge of dynamic parameters such as random inputs and the impact of unidentified interfering factors is not needed in the proposed method. The model utilises the most relevant components of the input patterns for training. For this research work, algorithms such as Levenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG), and Bayesian regularisation (BR) are applied to train the model. The methodology of the SARIMA, LSTM, and Nonlinear Autoregressive network with exogenous input (NARX) models are summarised in the coming sections.

### 2.1. SARIMA model

Three components comprise the ARIMA model: autoregressive (AR), integral, and moving average (MA). AR is the practise of using lagging inputs to predict future data. Integral refers to the necessary differentiation to make the data steady. MA is similar to AR in that it uses past (lagged) errors to predict future errors [17]. ARIMA(*p*,*d*,*q*) model indicates that the data is differentiated *d* times, *p* lagged points are utilised in AR and *q* lagged errors are utilised in MA. There is no differentiation involved in an ARIMA (*p*,*0*,*q*) model, hence it can alternatively be referred to as an ARMA (*p*,*q*) model. The following definition applies to the regression with ARMA errors $y_t$:

$$y_t = B_{x_t} + n_t \tag{1}$$

$$n_t = \emptyset_1 n_{t-1} + \cdots + \emptyset_p n_{t-p} - \theta_1 z_{t-1} - \cdots - \theta_q z_{t-q} + z_t \tag{2}$$

Where $n_t$ is the residual from the fit, $z_t$ is the white noise, $x_t$ is the external regressor at time *t*, *B* is the coefficient for the external regressor, $\emptyset$ is the vector of MA coefficients, and $\theta$ is the vector of AR coefficients. Maximum Likelihood Estimation can be used to solve the equations' coefficients. Box and Jenkins [18] proposed the SARIMA model as a stochastic method for regression and forecasting of time series displaying seasonal periodicity. SARIMA is a specific example of ARIMA that manages the behaviour of seasonal events, such as hourly or monthly behaviour, using a period variable [19]. SARIMA is widely used in various industries, such as solar radiation, the stock market, energy use, and gas use, and so on. By analysing the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF), the number of terms for each component (stationary and seasonal) is estimated. $(p, d, q)(P, D, Q)$ can be used to depict the SARIMA model where $p, d, q, P, D,$ and $Q$ are the number of AR parameters, differences, MA parameters, seasonal AR parameters, seasonal differences, and seasonal MA parameters, respectively.

### 2.2. LSTM

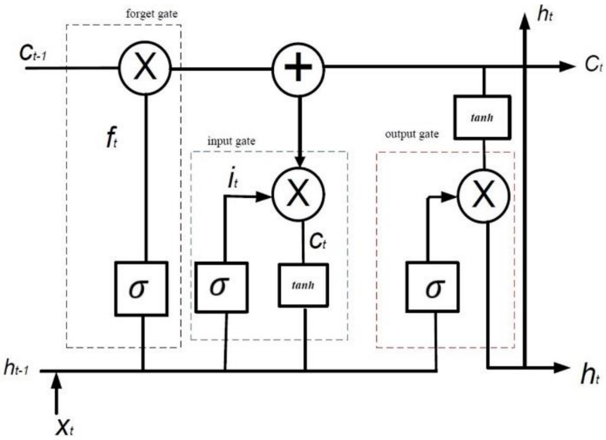


Figure 2. Structure of the LSTM unit

An LSTM network is a type of recurrent neural network that has memory blocks that may be trained to recognise long-term dependencies. It is made up of a collection of subnetworks with repeated connections that act as memory chips. The network can learn the contextual information necessary to forecast the following sequence in a time series thanks to these memory cells [20]. The input gate, forget gate, and output gate are the three gates of an LSTM. Typical RNNS is ineffective for longer sequences because long-term gradients tend to vanish in early layers. The LSTM structure resolves this by including a 'forget gate' that determines whether a piece of data should be discarded, preserved, or integrated with relevant content. All information entered as an input is controlled by the input gate. The contents of the cell are managed by the forget gate. The output gate estimates the activation function. The structure of the LSTM unit is as in Figure 2.

The hidden state $h_t^j$ of LSTM unit is defined as:

$$h_t^j = o_t^j tanh\left(c_t^j\right) \tag{3}$$

Where $o_t^j$ is output gate. The gates are evaluated by equations:

$$i_t^j = \sigma(W_i x_t + U_i h_{t-1} + V_i c_{t-1})^j \tag{4}$$

$$f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)^j \tag{5}$$

$$o_t^j = \sigma(W_o x_t + U_o h_{t-1} + V_o c_{t-1})^j \tag{6}$$

where $i_t^j$ is the input gate, $f_t^j$ is the forget gate, $\sigma$ is the logistic sigmoid function, $V$ is a diagonal matrix [21].

### 2.3. NARX

NARX is a dynamic RNN that utilises the linear autoregressive with exogenous inputs (ARX) model. The NARX model makes prediction values of a time series $y(t)$ based on the values of $y(t)$ in the past as well as the values of another external series $u(t)$. The mathematical formulation of the NARX model is as shown in Equation 7.

$$y(t) = f\left(y(t-1), \dots, y(t-n_y); u(t), \dots, u(t-n_u)\right) + \varepsilon(t) \tag{7}$$

Where $y(t)$ and $u(t)$ are the past and present exogenous inputs of the model at a discrete time step, $t$. $n_y$ and $n_u$ are the input memory and output memory orders, and $f$ is a nonlinear mapping function [22]. NARX is a form of ANN with feedback loops that are dynamically guided and recurrent. NARX networks have a superior descending gradient and they are capable of faster convergence and excellent generalisation throughout the learning process than other neural networks. The NARX model comes in two main configurations: series-parallel and parallel. A series-parallel design is used to predict values up to one step in the future, whereas a parallel architecture is used to predict more than one step in the future. The weights were estimated by the backpropagation method at each epoch.

$$y = f(x - t) \tag{8}$$

Where $f(\,)$ is the mapping function and $y(t-1)$ is the past values of the time series for the time $t-1$. A sample NARX structure is as shown in Figure 3.

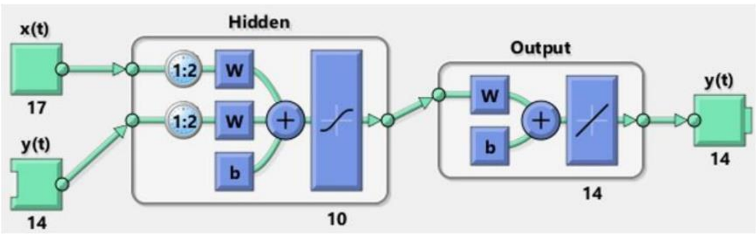


Figure 3. Architecture of NARX model

The most popular training approach for neural networks is backpropagation training. It is developed from the steepest descent approach, which looks for the largest error solution while minimising an objective function. In this study, the NARX model was trained and evaluated using three distinct backpropagation algorithms from the neural network toolbox of MATLAB-R2021b, including LM, BR, and SCG.

The Levenberg - Marquardt algorithm was developed in two phases: the first by Levenberg in 1944 and the second by Marquard in 1963. The Levenberg-Marquardt algorithm is frequently used in Artificial Neural Networks as a training algorithm since many authors believe it to be the fastest approach. The Levenberg-Marquardt (LM) is a combination of the gradient descend method and the Gauss-Newton method. When the weights and biases are not close to their best fit value, the amendment of weights and biases is performed along the steepest descend direction to reduce the goodness-of-fit parameters along that path. When it is near, the updating of weights and biases is performed by assuming that the least-square function is locally quadratic, as in the Gauss-Newton method. Jacobian matrix estimation and error evaluation can be used to design the LM algorithm, as shown by the following equations:

$$H = J^T J \tag{9}$$

$$x(k+1) = x(k) - [J^T J + \mu I]^{-1} J^T \epsilon \tag{10}$$

Where $J$ is the Jacobian matrix, $\mu$ is a scalar operator constant, $\epsilon$ is residual error, $x(k+1)$ is Hessian matrix. $x(k)$ is weight vector, and $I$ is identity matrix [23].

The BR training algorithm is a model that transforms a non-linear regression to a statistical model, with the benefit of not using validation and its robustness. The network weights are incorporated into the objective function. In each iteration, it behaves in the same manner as that of the LM algorithms. Simultaneously, weights and biases are adjusted in a direction to reduce both objective function and goodness-of-fit parameters. The objective function and error equations in Bayesian regularisation are depicted by the following equations:

$$F = \beta E_d + \alpha E_w \tag{11}$$

$$E_w = \frac{1}{2} \sum_{k=1} (w_i)^2 \tag{12}$$

Where $E_d$ and $E_w$ are the sum of squared errors and the sum of squared weights, respectively. $\alpha$ and $\beta$ are objective function regularisation parameters and $F$ is the objective function. By minimising the squared errors and weights in the network, this technique discovers the ideal combination to create a network that generalises well.

The SCG algorithm is a learning model that is frequently used to train ANN. This algorithm operates according to the fundamentals of optimisation, having an objective function and convergent to the target, but the control of step size makes the same targeting more effective for second-order information. The weights and biases are modified along a conjugate gradient track to lower the goodness-of-fit parameters. Compared to the direction of the steepest descent, this offers a quick convergence. The search direction and step size are chosen with the use of information from the second-order approximation.

$$y(k+1) = y(k) + \alpha_k d_k \tag{13}$$

Where $y(k+1)$ is the subsequent iteration, $\alpha_k$ is the step size and $d_k$ is the search direction. The Hessian Matrix to the error function is given by:

$$x(k+1) = \frac{E'\,(y_k + \alpha_k d_k) - E'\,(y_k)}{d_k \alpha_k} \tag{14}$$

Where $E'(\,)$ is gradient to the function.

Mean Square Error (MSE), MAPE, and RMSE are the most commonly used metrics in evaluating forecasting technique performance. The degree of linearity between the observed and predicted data is assessed using the coefficient of determination ($R^2$).

$$R^2 = 1 - \frac{\sum_{j=1}^{n} \left(y_j - x_j\right)^2}{\sum_{j=1}^{n} \left(y_j - \bar{y}\right)^2} \tag{15}$$

Mean square error is defined as the average deviation between the predicted and observed values. Better models have low MSE values. If the MSE is 0.00, it indicates that the prediction model contains no error values [24].

$$MSE = \frac{1}{n}\sum_{j=1}^{n}\left(y_j - x_j\right)^2 \quad (16)$$

The absolute mean value of the percentage error between the actual and predicted values is calculated using MAPE [25].

$$MAPE = \frac{1}{n}\sum_{j=1}^{n}\left|\frac{(x_j - y_j)}{y_j}\right| * 100 \quad (17)$$

Where $y_j$ is the predicted value within interval $j$, $x_j$ is the actual value at interval $j$, and $\bar{y}$ is the mean of the $y_j$ values.

The forecasted data are used by the consumers to schedule their appliances and to reduce their electricity bills. The forecasting device has a two-way communication capability that enables it to send forecasted load demand to consumers and receive previous daily load curve patterns from the consumers. Chinnathambi et al. [26] analysed different datasets such as one-week, two-weeks, three-weeks, one month, 45 days, 60 days, 75 days and 90 days and observed the MAPE variation do not follow a decreasing pattern as sample size increases.

The daily power consumption of a serviced apartment in Kayathar during December 2019 has been taken as input data. The study area is located in Thoothukudi district, Tamil Nadu state, India, at a latitude and longitude of 08°57'44.27"N, 77°43'10.80"E. The load profile for 1st January 2020 has been forecasted by various models. In this study, MATLAB-R2021b has been used for forecasting in the models. Conventional models such as ARIMA and SARIMA models are developed by the econometric modeler. The data are fitted into a model by estimating the model. Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS) tests are employed to determine whether a trend is stationary and whether the data is stationary. The Gaussian distribution serves as the distribution function for SARIMA studies. Autocorrelation, partial autocorrelation, and coefficients of the SARIMA equation are used for statistical performance analysis. The flow chart of the SARIMA model is shown in Figure 4.

In the LSTM, the architecture begins with input feature management and processing. The input features have values from the 1st of December 2019 to 31st of December 2019. Then, the value of the time steps is 744. The input features are provided to the sequence input layer as a matrix after normalisation. Next, LSTM layers construct a deep network and dropout layers are used to avoid overfitting and generalise the proposed projected model. Finally, the fully connected and regression layers collaborate to transfer the input attributes to the output of the demand forecast. The LSTM layers' hidden unit count is set at 5. The dropout probability for the dropout layer is set to 0.005. The LSTM layer's learnable parameters are set to recommended default values. The maximum epochs and mini-batch sizes are both set at 60. The gradient threshold is set at 2 to prevent gradient explosion. The regression output layer's loss function is set to mean squared error (MSE), and training is terminated when both MSE and RMSE are at their lowest values. A sigmoid transfer function in the hidden layer and a linear transfer function in the output layer are used for the NARX model. The flowchart of the NARX model is shown in Figure 5.

The output is a 24 x 1 vector that contains the forecast for the following 24 hours. The past load values are used as endogenous input. The hour of the day has been used as exogenous input. The future load has been taken as a target vector. The NARX model divides the training phase into three subgroups: training, validation, and testing. Using the dataset from the training phase, the NARX model is created. The model created is used to forecast future load. After the testing, the produced model is validated using train data from the training phase. The error histogram, error, and the relationship between actual and forecasted values are used for statistical performance analysis. Furthermore, the $R^2$ and error functions are used to test both models utilising test data from the test phase.

## 3. RESULTS AND DISCUSSION

Data on hourly demand load from 1st December to 31st December 2019, are taken from the consumer. All the data collected before the forecast day is used to create the training and validation sets. The same dataset has been used for all forecasting models. In NARX, the total number of samples that are used was 744, the size of the hidden layer is 10, and the feedback delays are 1:2. 70% of the total data samples are put towards the training process, while the remaining 30% are split evenly between the validation and testing phases. The training utilises algorithms such as LM, BR, and SCG. Figure 6 displays the developed neural network's closed-loop training form. The performance of the neural network at different iterations is shown in Figure 7. At the second iteration, the best performance is observed. Until the highlighted epoch, both training and validation errors decrease, as shown in the graph. There has not been any overfitting since the validation error does not go up before this epoch.

Gradient and validation checks are presented in Figure 8, and it is seen that as the number of iterations increased, the gradient decreased. However, the validity check is increased. As a secondary check to make sure the network has generalised correctly without interfering with the training, test vectors are used. The error convergence of the network is affected by the momentum "mu", which is a control parameter.

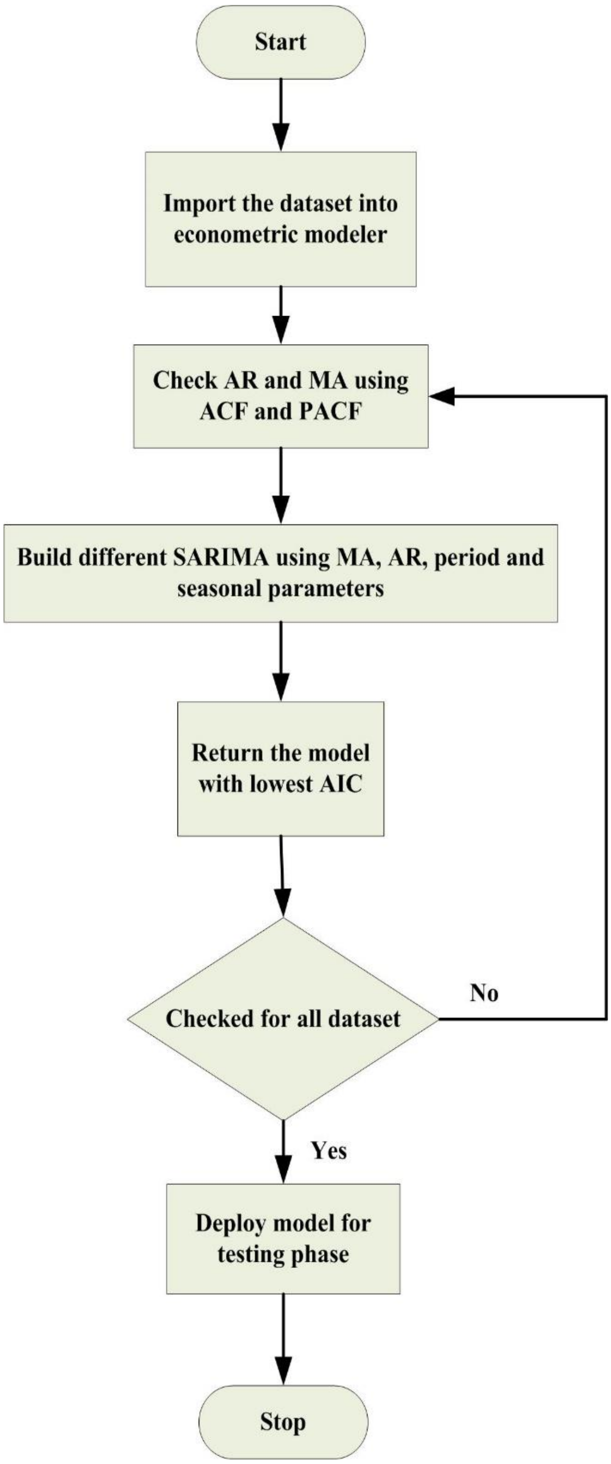


Figure 4. Steps used in SARIMA model

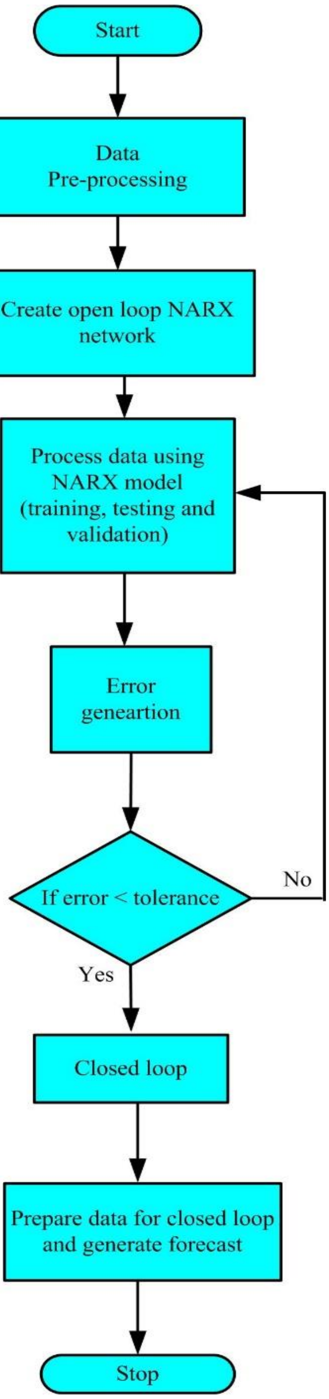


Figure 5. Flowchart of NARX model

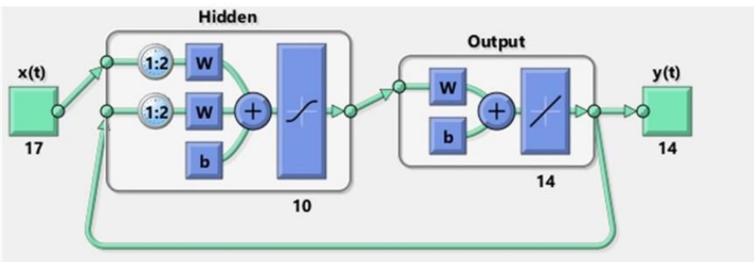


Figure 6. Architecture of NARX model in closed-loop

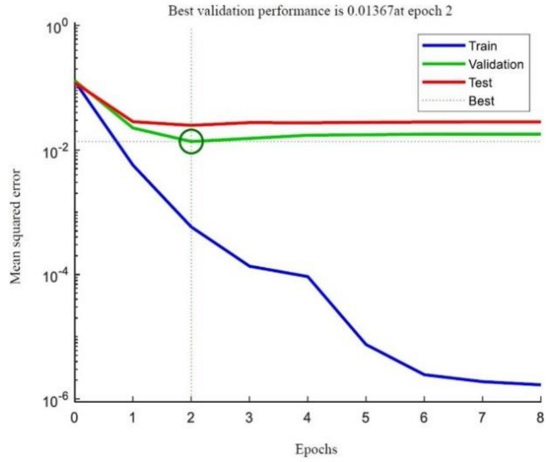


Figure 7. Mean square error vs. epochs

The error histogram and regression graphs using LM are shown in Figures 9 and 10, respectively. As per the error histogram, in this instance, the distribution of residuals between target and network outputs reveals that most mistakes range between -0.0196 and 0.0995. Plots are made by dividing the error range into 20 equal increments. The regression plot, which depicts the linear relationship between the desired targets and the predicted outputs, is an additional important indicator of network performance. Both outputs and targets should be positioned horizontally on the 45-degree line during optimal training conditions. With an R-value of roughly 0.8153, the data fit appears appropriately good for training, validation, testing, and entire data sets.

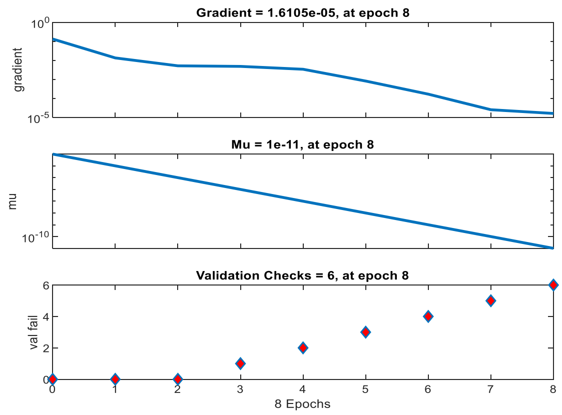


Figure 8. Gradient and validation check

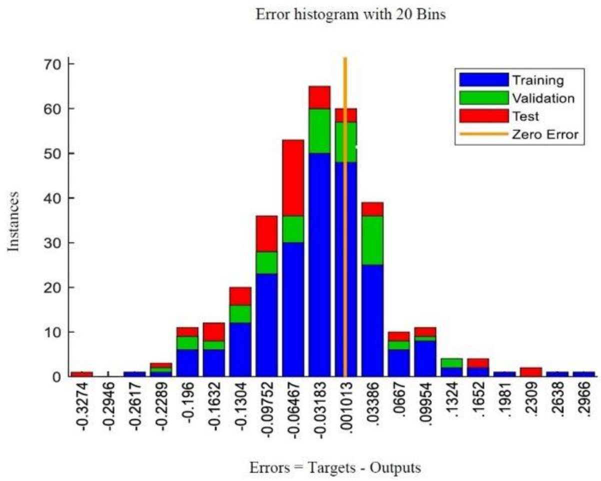


Figure 9. Error histogram

Figure 11 illustrates the output responses provided by the NARX network. The data for training, testing, and validation are displayed along with the associated targets and outputs. The discrepancy between targets and outputs (error), which is less than 95% of the confidence interval, enables highly accurate predictions of future data Figure 12 presents the output responses, given out by the BR algorithm in MATLAB. It shows the data points for training, testing, and validation in terms of targets and outputs.

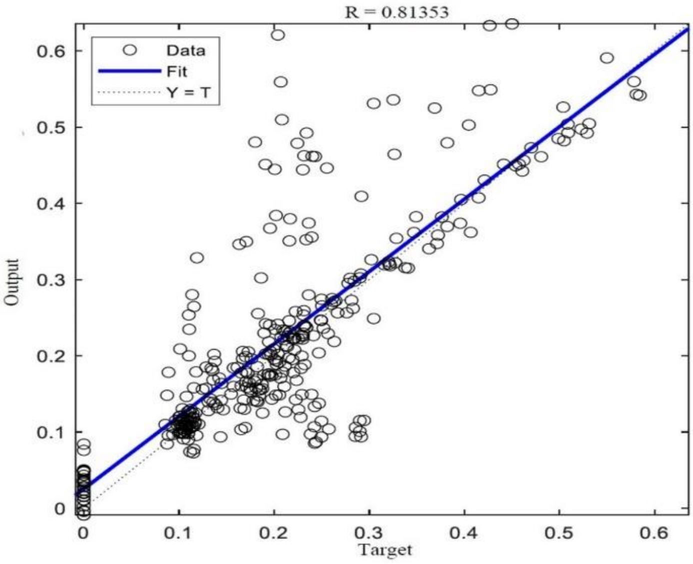


Figure 10. Regression plot between targets and outputs

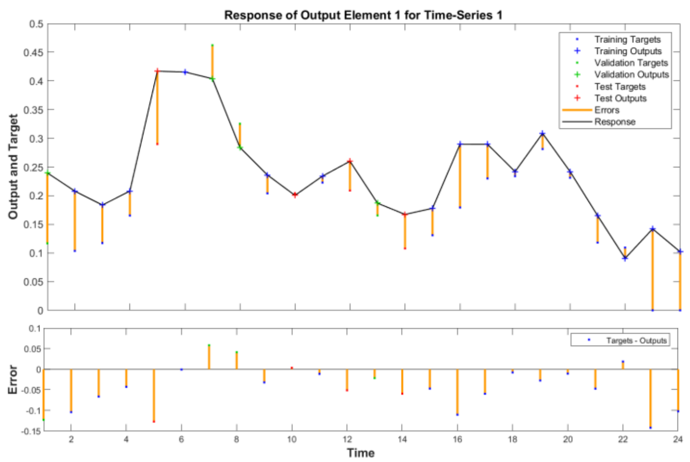


Figure 11. Output response by NARX network

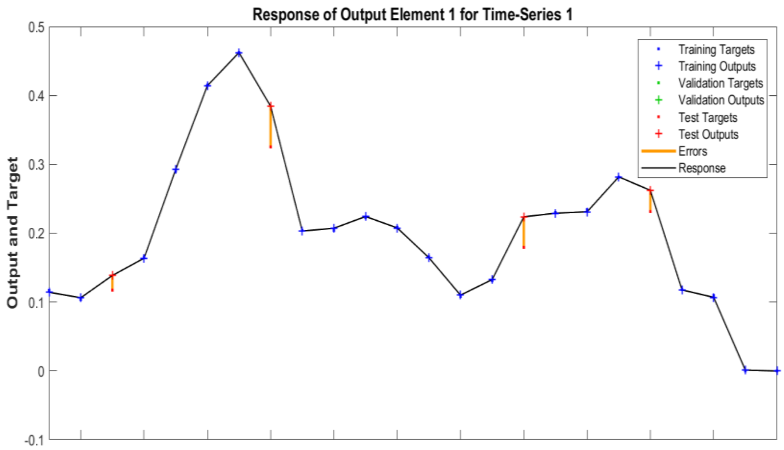


Figure 12. Forecasted load profile by BR network

Figure 13 illustrates the autocorrelation of error for various lags. The NARX network's functionality is validated using the error autocorrelation function. This function describes the connection between prediction errors and time. Based on the sample size of the time series data, the MATLAB autocorrelation function produces a set of confidence intervals around zero, and the forecast errors all fall within those intervals.

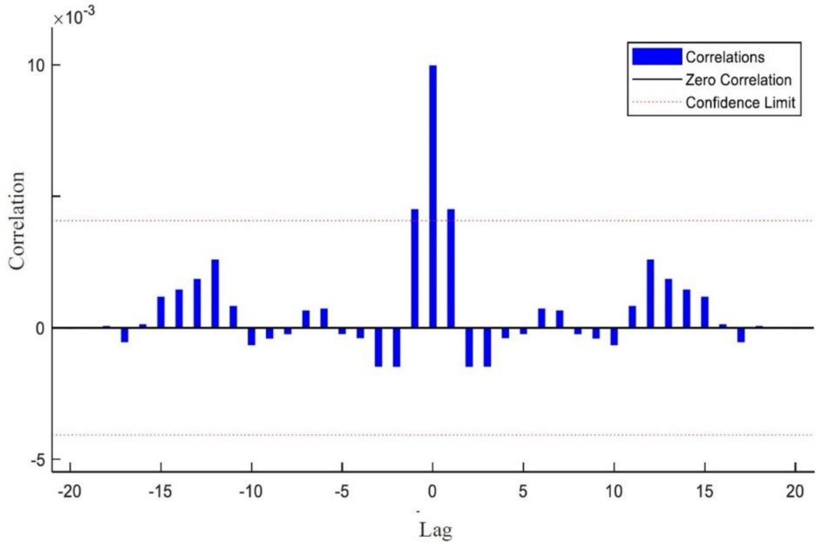


Figure 13. Autocorrelation function of error

As can be seen from the plot in Figure 13, values of the ACF for lags greater than zero are within the confidence interval. This suggests that there is no autocorrelation in the data. This demonstrates that the network training has been effective.

Table 1 shows the values of MSE and R in the NARX model with LM, BR, and SCG. The simulations and results showed that the Bayesian regularisation training model performed well and achieved great accuracy, with correlation coefficient L(R) of 0.9998 and an MSE of 0.0031 during the training phase. As a result, Bayesian Regularisation is performed significantly better than LM and SCG.

Table 1. Error analysis by NARX algorithms

| Error | LM | BR | SCG |
|---|---|---|---|
| MSE (Training) | 0.0032 | 0.0031 | 0.0051 |
| R (Training) | 0.9382 | 0.9998 | 0.9412 |
| MSE (Testing) | 0.0035 | 0.0029 | 0.0036 |
| R (Testing) | 0.8804 | 0.8962 | 0.8753 |
| MSE (Validation) | 0.0059 | 0.0017 | 0.0067 |
| R (Validation) | 0.3392 | 0.5106 | 0.7468 |

A comparison study with a conventional time-series prediction model (SARIMA) and LSTM is being conducted to further demonstrate NARX's high efficiency. Forecasting by SARIMA methods has been done using an econometric modeller in MATLAB. The forecasted daily load curve obtained by the SARIMA model is shown in Figure 14. The performance of these methods is assessed using MAPE is shown in Table 2. The Bayesian regularisation training model performs best and achieves excellent accuracy, with R of 0.9998 and MAPE of 0.003. The findings demonstrated that, compared to SARIMA models and LSTM, the NARX forecaster is more effective in predicting daily load consumption. Figure 15 shows the results of the forecasting models along with the raw data. The more detailed forecasting results are plotted in Figure 16 along with sample data for one day input. The forecasted values of daily power consumption using NARX, LSTM, and SARIMA are listed in Table 3. Table 4 summarised the related study results in activation function, sample size, the prediction method and MAPE values. The results show that the present model gives lesser MAPE compared to other models in related research.

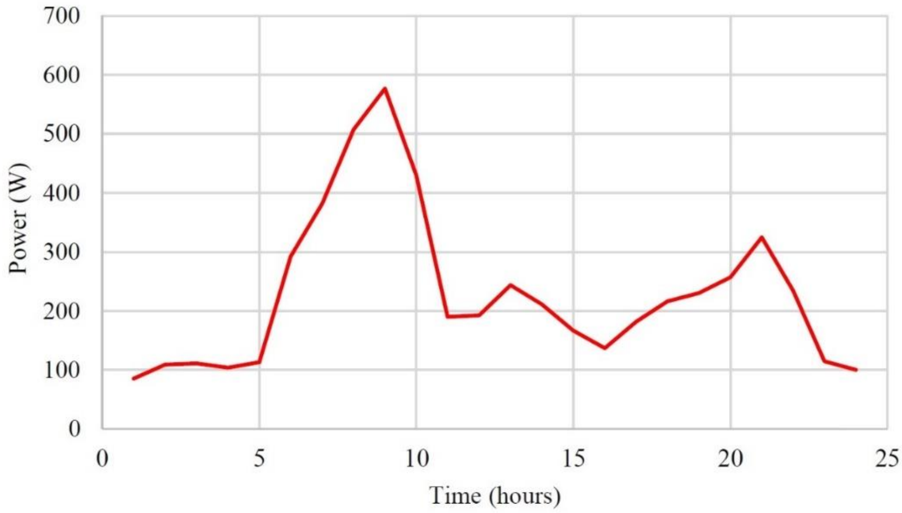

Figure 14. Forecasted load profile from SARIMA

Table 2. Performance parameters on training dataset

| S.no | Algorithm | R | MAPE |
|---|---|---|---|
| 1 | LM | 0.9382 | 0.00362 |
| 2 | BR | 0.9998 | 0.003 |
| 3 | SCG | 0.9412 | 0.0037 |
| 4 | LSTM | 0.9762 | 0.00367 |
| 5 | SARIMA | 0.9620 | 0.0931 |

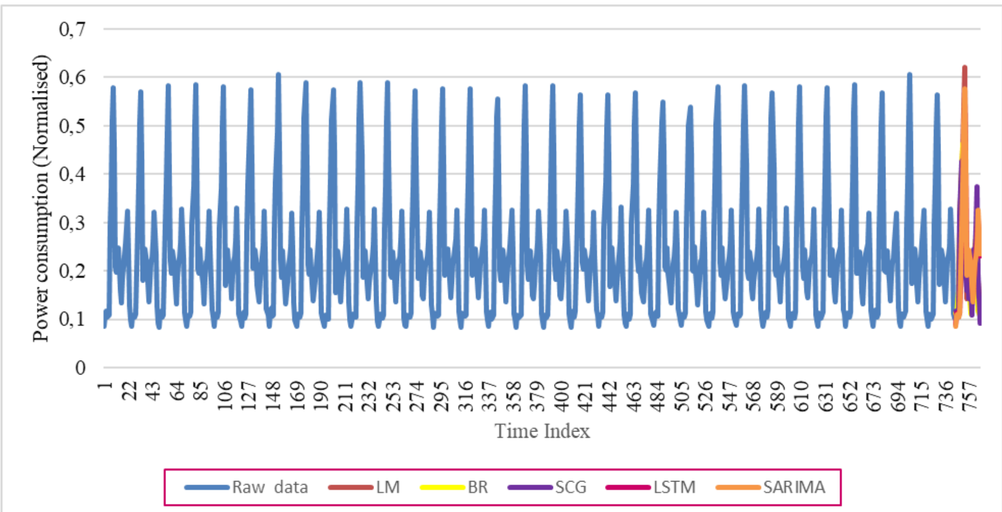

Figure 15. In-sample and out-of-sample comparison for the power consumption data set.

Table 3. Forecasted values of load demand by NARX, LSTM and SARIMA models

| Hours | Load demand predicted by LM algorithm (kW) | Load demand predicted by BR algorithm (kW) | Load demand predicted by SCG algorithm (kW) | Load demand predicted by LSTM (kW) | Load demand predicted by SARIMA (kW) |
|---|---|---|---|---|---|
| 01:00 | 0.113701 | 0.114148 | 0.117098 | 0.089329 | 0.085361 |
| 02:00 | 0.10365 | 0.106207 | 0.115703 | 0.112462 | 0.109135 |
| 03:00 | 0.12507 | 0.13861 | 0.110738 | 0.112913 | 0.110925 |
| 04:00 | 0.103272 | 0.163631 | 0.124131 | 0.107556 | 0.103666 |
| 05:00 | 0.302295 | 0.292551 | 0.316827 | 0.113109 | 0.113057 |
| 06:00 | 0.40733 | 0.414063 | 0.427805 | 0.268435 | 0.292107 |
| 07:00 | 0.456258 | 0.462258 | 0.428836 | 0.353513 | 0.381726 |
| 08:00 | 0.535789 | 0.387146 | 0.314954 | 0.469438 | 0.507584 |
| 09:00 | 0.620881 | 0.203079 | 0.263189 | 0.545238 | 0.576717 |
| 10:00 | 0.241345 | 0.207153 | 0.184751 | 0.399556 | 0.430177 |
| 11:00 | 0.142473 | 0.224181 | 0.159038 | 0.210754 | 0.189909 |
| 12:00 | 0.178599 | 0.207606 | 0.207601 | 0.203987 | 0.192359 |
| 13:00 | 0.177539 | 0.16444 | 0.141542 | 0.229047 | 0.243875 |
| 14:00 | 0.146547 | 0.109975 | 0.156391 | 0.209247 | 0.211204 |
| 15:00 | 0.128167 | 0.132659 | 0.109333 | 0.170305 | 0.166928 |
| 16:00 | 0.16658 | 0.223739 | 0.18967 | 0.142775 | 0.136877 |
| 17:00 | 0.222078 | 0.229008 | 0.244561 | 0.182483 | 0.181782 |
| 18:00 | 0.237148 | 0.231044 | 0.223061 | 0.223181 | 0.216109 |
| 19:00 | 0.272824 | 0.281799 | 0.291368 | 0.239953 | 0.230092 |
| 20:00 | 0.259269 | 0.262088 | 0.373715 | 0.271221 | 0.257028 |
| 21:00 | 0.179018 | 0.117563 | 0.192108 | 0.30352 | 0.324933 |
| 22:00 | 0.126944 | 0.106788 | 0.090836 | 0.230933 | 0.234346 |
| 23:00 | 0.028059 | 0.001246 | 0.081879 | 0.13045 | 0.11486 |
| 24:00 | 0.011047 | 0.0009 | 0.011073 | 0.104996 | 0.099983 |

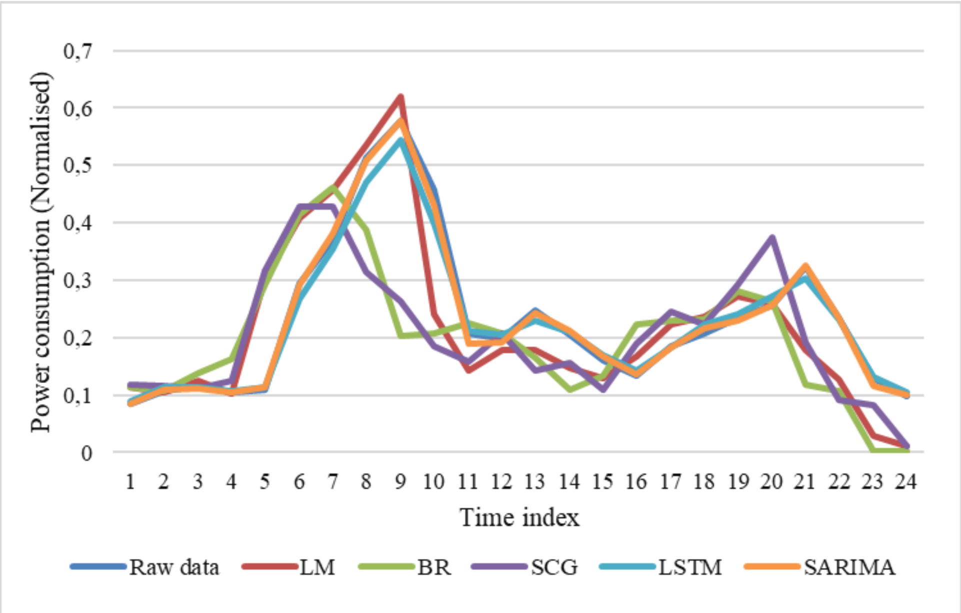


Figure 16. Forecasting results of NARX, LSTM and SARIMA models with imput sample.

Table 4. Comparative table of related studies.

| Reference | Activation function | Sample size | Location | Prediction method | Prediction Error (MAPE) (%) |
|---|---|---|---|---|---|
| [27] | ELU | 168 | Seoul, South Korea | DNN | 7.02 |
| [28] | logsigmoid | 120 | Ontario, Canada | Bayesian Optimization Algorithm based NARX | 3 |
| [29] | logsigmoid | 72 | Abuja, Nigeria | Multiple Linear Regressions | 0.4545 |
| [30] | logsigmoid | 168 | Bidart, France | NARX-LM | 14.1 (winter)<br>12 (summer) |
| Present | logsigmoid | 744 | Kayathar, India | NARX-BR | 0.3 |

## 4. CONCLUSION

Recently, demand response management has gained increased attention since it can provide economic operations on the consumer side. Demand response management includes load forecasting as a key component. Forecasting improves occupant comfort while lowering energy use and the electricity bill. In the above section, forecasting by NARX, LSTM, and SARIMA models is discussed in detail. Day-ahead load curve at the consumer end is predicted using neural network-based forecasting. The network is trained using the Levenberg-Marquardt (LM), scaled conjugate gradient, and Bayesian regularisation techniques using the data collected from Kayathar, Thoothukudi district, Tamil Nadu state, India. The model provides a novel approach to the accurate prediction of daily power consumption profiles in real-time. The performance of three Neural Network learning algorithms, i.e., LM, BR, and SCG, is compared with the SARIMA and LSTM models. It is concluded that the NARX model with the BR algorithm is a good choice to predict the daily load consumption. In this study, because the amount of data available in this investigation is short, it is impossible to determine if the model was overfit. In future studies, additional dates will be secured to conduct research on the construction of more accurate performance prediction models.

## REFERENCES


[1] L. Peng, S.X. Lv, L. Wang, and Z. Y. Wang, "Effective electricity load forecasting using enhanced double-reservoir echo state network," *Engineering Applications of Artificial Intelligence*, vol. 99, p.104132, 2021.

[2] N. Son, S.Yang, and J. Na, "Deep neural network and long short-term memory for electric power load forecasting, " *Applied Sciences*, vol. 10, no. 18, p. 6489, 2020.

[3] G. Zhang and J. Guo, "A novel ensemble method for hourly residential electricity consumption forecasting by imaging time series, " *Energy*, vol. 203, p.117858, 2020.

[4] R. Lu, S. H. Hong, and M. Yu, 2019, "Demand response for home energy management using reinforcement learning and artificial neural network, " *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6629–6639, 2019.

[5] S. Ungureanu, V. Topa, and A. C. Cziker, "Deep Learning for Short-Term Load Forecasting—Industrial Consumer Case Study," *Applied Sciences*, vol. 11, no. 21, pp. 10126, 2021.

[6] M. Alhussein , K. Aurangzeb, and S. I. Haider, "Hybrid CNN-LSTM model for short-term individual household load forecasting, ," *IEEE Access*, vol. 8, pp. 180544–57, 2020.

[7] T. Bashir, C. Haoyong, MF. Tahir, and Z. Liqiang, "Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN, " *Energy Reports*, vol. 8, pp. 1678–86, 2022.

[8] F. Saeed, A. Paul, and H. Seo, H., 2022, "A Hybrid Channel-Communication-Enabled CNN-LSTM Model for Electricity Load Forecasting, " *Energies*, vol. 15, no.6, p.2263,2022.

[9] N. Mughees, S. A. Mohsin, A. Mughees, and A. Mughees, " Deep sequence to sequence Bi-LSTM neural networks for day-ahead peak load forecasting," *Expert Systems with Applications*, vol. 175, p.114844, 2021.

[10] W. He, " Load forecasting via deep neural networks," *Procedia Computer Science,* vol. 122, pp. 308–314,2017.

[11] R. Mubashar, M. J. Awan, M. Ahsan, A. Yasin, and V. P. Singh, "Efficient residential load forecasting using deep learning approach, " *International Journal of Computer Applications in Technology*, vol. 68, no. 3, pp.205-214, 2022.

[12] I. Ozer, S. B. Efe, and H. Ozbay, "A combined deep learning application for short term load forecasting;" *Alexandria Engineering Journal*, vol. 60, no. 4, pp.3807-3818, 2021.

[13] W. S. McCulloch, and W. Pitts, "A logical calculus of the ideas immanent in nervous activity." *The bulletin of mathematical biophysics*, vol.5, no. 4, pp.115-133, 1943.

[14] C. Martinez-Castillo, G. Astray, and J.C. Mejuto, "Modelling and prediction of monthly global irradiation using different prediction models," *Energies*, vol. 14, no.8, p.2332, 2022.

[15] S. B. Priyatno, T. Prakoso, and M. A. Riyadi,“ Classification of motor imagery brain wave for bionic hand movement using multilayer perceptron,” *Sinergi*, vol. 26, no.1, pp.57–64, 2022.

[16] S. H. Rafi, S. R. Deeba, and E. Hossain, “A short-term load forecasting method using integrated CNN and LSTM network,” *IEEE Access*, vol. 9, 32436–32448, 2021.

[17] O. A. Karabiber, and G. Xydis, “Electricity price forecasting in the Danish day-ahead market using the TBATS, ANN and ARIMA methods,” *Energies*, vol. 12, no. 5, p.928, 2019.

[18] G. E. Box, *Time series analysis: forecasting and control*, John Wiley & Sons, 2015.

[19] S. Chaturvedi, E. Rajasekar, S. Natarajan, and N. McCullen, “A comparative assessment of SARIMA, LSTM RNN and Fb Prophet models to forecast total and peak monthly energy demand for India,” *Energy Policy*, vol. 168, p.113097,2022.

[20] T. A. Nakabi, and P. Toivanen, “An ANN-based model for learning individual customer behavior in response to electricity prices,” *Sustainable Energy, Grids and Networks*, vol. 18, p.100212, 2019.

[21] U. Ugurlu, I. Oksuz and O. Tas, “ . Electricity price forecasting using recurrent neural networks,” *Energies*, vol. 11, no. 5, p.1255, 2018.

[22] Y. Raptodimos, and I. Lazakis, “Application of NARX neural network for predicting marine engine performance parameters,” Ships and Offshore Structures, vol. 15, no. 4, pp.443-452, 2020.

[23] D. Selvamuthu, V. Kumar, and A. Mishra, “Indian stock market prediction using artificial neural networks on tick data,” *Financial Innovation*, vol. 5, no. 1, pp.1-12, 2019.

[24] S. Namasudra, S. Dhamodharavadhani, and R. Rathipriya, “Nonlinear neural network based forecasting model for predicting COVID-19 cases,” *Neural processing letters*, pp.1-21, 2021.

[25] S. Atef, K. Nakata, K. and A. B. Eltawil, “A deep bi-directional long-short term memory neural network-based methodology to enhance short-term electricity load forecasting for residential applications,” *Computers & Industrial Engineering*, vol. 170, p.108364, 2022.

[26] A.Chinnathambi, Radhakrishnan, A. Mukherjee, M. Campion, H. Salehfar, T.M. Hansen, J. Lin, and P. Ranganathan, “A Multi-Stage Price Forecasting Model for Day-Ahead Electricity Markets,” *Forecasting*, vol. 1, pp.26-46, 2018.

[27] S. M. Jung, S. Park, S.W. Jung, and E. Hwang, “Monthly Electric Load Forecasting Using Transfer Learning for Smart Cities,” Sustainability, vol. 12, p.6364, 2020.

[28] N. Sultana, S.Z. Hossain, S.H. Almuhaini, and D. Düştegör, “A Multi-Stage Price Forecasting Model for Day-Ahead Electricity Markets,” *Energies*, vol. 15, p.3425, 2022.

[29] E.C. Ashigwuike, A. R.A. Aluya, J.E.C. Emechebe, and S. A. Benson, “Medium term electrical load forecast of Abuja Municipal Area council using artificial neural network method,” *Nigerian Journal of Technology*, vol. 39, pp.860-870, 2020.

[30] I. Zapirain, G. Etxegarai, J. Hernández, Z. Boussaada, N. Aginako, and H. Camblong, “Short-term electricity consumption forecasting with NARX, LSTM, and SVR for a single building: small data set approach,” *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects*, vol. 44, pp.6898-6908, 2022.