Short-Term Forecasting of Electric Loads Using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Vector Inputs

Authors:

Jaime Buitrago, Shihab Asfour

Date Submitted: 2019-07-26

Keywords: closed-loop forecasting, artificial neural networks, nonlinear autoregressive exogenous input, short-term load forecasting

Abstract:

Short-term load forecasting is crucial for the operations planning of an electrical grid. Forecasting the next 24 h of electrical load in a grid allows operators to plan and optimize their resources. The purpose of this study is to develop a more accurate short-term load forecasting method utilizing non-linear autoregressive artificial neural networks (ANN) with exogenous multi-variable input (NARX). The proposed implementation of the network is new: the neural network is trained in open-loop using actual load and weather data, and then, the network is placed in closed-loop to generate a forecast using the predicted load as the feedback input. Unlike the existing short-term load forecasting methods using ANNs, the proposed method uses its own output as the input in order to improve the accuracy, thus effectively implementing a feedback loop for the load, making it less dependent on external data. Using the proposed framework, mean absolute percent errors in the forecast in the order of 1% have been achieved, which is a 30% improvement on the average error using feedforward ANNs, ARMAX and state space methods, which can result in large savings by avoiding commissioning of unnecessary power plants. The New England electrical load data are used to train and validate the forecast prediction.

Record Type: Published Article

Submitted To: LAPSE (Living Archive for Process Systems Engineering)

Citation (overall record, always the latest version): LAPSE:2019.0702
Citation (this specific file, latest version): LAPSE:2019.0702-1
Citation (this specific file, this version): LAPSE:2019.0702-1v1

DOI of Published Version: https://doi.org/10.3390/en10010040

energies

MDPI

Article

Short-Term Forecasting of Electric Loads Using Nonlinear Autoregressive Artificial Neural Networks with Exogenous Vector Inputs

Jaime Buitrago †,‡ and Shihab Asfour *,‡

University of Miami, Department of Industrial Engineering, 1251 Memorial Drive, 268 McArthur Engineering Building, Coral Gables, FL 33146, USA; jbuitrago@miami.edu

* Correspondence: sasfour@miami.edu; Tel.: +1-305-284-2367

† Current address: 1251 Memorial Drive, MacArthur Engineering Building, Rm 288, Coral Gables, FL 33146, USA.

‡ These authors contributed equally to this work.

Academic Editor: Chunhua Liu

Received: 14 November 2016; Accepted: 14 December 2016; Published: 1 January 2017

Abstract: Short-term load forecasting is crucial for the operations planning of an electrical grid. Forecasting the next 24 h of electrical load in a grid allows operators to plan and optimize their resources. The purpose of this study is to develop a more accurate short-term load forecasting method utilizing non-linear autoregressive artificial neural networks (ANN) with exogenous multi-variable input (NARX). The proposed implementation of the network is new: the neural network is trained in open-loop using actual load and weather data, and then, the network is placed in closed-loop to generate a forecast using the predicted load as the feedback input. Unlike the existing short-term load forecasting methods using ANNs, the proposed method uses its own output as the input in order to improve the accuracy, thus effectively implementing a feedback loop for the load, making it less dependent on external data. Using the proposed framework, mean absolute percent errors in the forecast in the order of 1% have been achieved, which is a 30% improvement on the average error using feedforward ANNs, ARMAX and state space methods, which can result in large savings by avoiding commissioning of unnecessary power plants. The New England electrical load data are used to train and validate the forecast prediction.

Keywords: short-term load forecasting; nonlinear autoregressive exogenous input; artificial neural networks; closed-loop forecasting

1. Introduction

Short-term electrical load forecasting is vital for the efficient operation of electric power systems. A power grid integrates many stake holders who can be affected by an inaccurate load forecast: power generation utilizes 24-hour or 48-hour ahead forecasts for operations planning, i.e., to determine which power sources should be allocated for the next 24 h; transmission grids need to know in advance the power transmission requirements in order to assign resources; end users utilize the forecast to calculate energy prices based on estimated demand. Contingency planning, load shedding, management strategies and commercialization strategies are all influenced by load forecasts. Forecast errors result in increased operating costs [1]: predicting a lower load than the actual load results in utilities not committing the necessary generation units and therefore incurring higher costs due to the use of peak power plants; on the other hand, predicting a higher load than actual will result in higher costs because unnecessary baseline units are started and not used. Reliable forecasting methods are essential for scheduling sources and load management [2].

Load forecasting methods can be classified into two main categories: statistical methods and computational intelligence (CI) methods. Statistical methods include all forms of autoregressive and parametric models. CI methods include artificial neural networks (ANN), fuzzy logic and expert systems. Typical load patterns are non-linear, and therefore, non-linear models have proven more effective in generating short-term forecasts. ANNs provide an accurate approach to the problem and have the advantage of not requiring the user to have a clear understanding of the underlying mathematical relationship between input and output. ANNs have proven to be effective forecasting techniques with most studies showing an improvement in forecast accuracy over their statistical counterparts [3–8]. The resulting performance of different ANN methods has been analyzed by many authors [9]. The results of this work fall into the expected forecast accuracy range with advantages both in performance and simplicity.

This paper proposes a method of obtaining accurate short-term load forecasts by using artificial neural networks in recurrent mode with weather-related variables used as exogenous inputs. The model produces a 24-hour forecast as the output of a fully-connected ANN with one hidden layer and sigmoid functions as activation functions in the hidden nodes. The output nodes use a linear activation function. Inputs are historical values of load. The input vector also includes day, month, day of the week and whether or not the day is a working day. Weather information is also used as exogenous input. The architecture of the network is such that the neural network is trained in open-loop using actual load data in order to calculate the node weights, and then, the network is used in closed-loop to generate the forecast applying the calculated forecast load as input. The results show an improvement of the forecast when compared with ANN models without exogenous inputs. The resulting network is a recursive network, and therefore, by using its own calculated output, it does not need to be retrained in order to continue providing forecasts.

This research lies in the intersection of the fields of data forecasting, electrical load forecasting, artificial neural networks and time-series forecasting. We will focus the literature review on the topic of short-term load forecasting using artificial neural networks.

1.1. Review Papers

Raza [8] wrote a review of some of the most common techniques used in STLF forecasting and included a description of the ANN model and transfer functions most commonly used. The authors included a comprehensive bibliography on the methods described in their paper.

Tzafestas [7] wrote a review paper on the computational techniques for short-term electric load forecasting. The techniques reviewed were multi-layer perceptron (MLP), fuzzy logic (FL), genetic algorithms (GA) and chaos. Then, the authors went on to review hybrid techniques: ANFIS, GARIC, fuzzy ART, KB-NN, chaos-FL and NFGA. The author found a mean absolute percent error (MAPE) for MLP of 1.7%, with some improvement by using combined techniques.

1.2. Multi-Layer Feedforward Neural Network Approach

Abdel et al. in [10] developed a univariate model for medium-term load forecast utilizing abductive and neural networks. The authors compared the performance (MAPE-based) between the abductive networks model and the neural network model. The MAPE of the neural network model was about 4%. The ANN used by the authors in this research was a multi-layer feedforward neural network (MLFFN). Since their model was univariate, only historical values of load data were used. The authors utilized six years of load data, normalized in order to remove the trend, and used a single-next month model for their iterative forecasting. The authors compare two different approaches: a single 12-month stream of data and 12 separate one-month streams of data. The single stream method proved more accurate, 3.2% vs. 3.8%. The authors claim better MAPE performance with the ANN than ARIMA and other forecasting methods.

The researchers Alfuhaid et al. [11] utilize a cascaded artificial neural network (CANN) to produce a forecast that includes peak, minimum and daily energy as additional input data for the final forecast

stage. The authors utilize two neural networks: one small network to predict the peak, maximum and minimum loads; another network that uses the output of the first network, to predict half-hourly loads for the next day. The first network cascades onto the second. Researchers used data from the network in Kuwait. The average (MAPE) forecasting error by applying CANN is 2.707% as opposed to 3.367% when using conventional ANN (FitNet).

The work by Bennett [12] describes the comparison between an ARIMAXmodel and an MLFFN ANN model to forecast low voltage next day total energy use and next day peak demand. The authors propose a hybrid model incorporating a double exponential smoothing algorithm, autoregressive terms, relative humidity and day of the week dummy variables to increase accuracy. The ARIMAX model included linear and quadratic terms for the temperature and relative humidity as a multiplying term for both. The double exponential regression term was introduced in order to account for the general trend. The ARIMAX model yields an MAPE of an average of 7% and the ANN model an MAPE of 6%. Data were taken from a transformer serving 128 residential customers. The authors found that daily average temperature explained half of the observed variance in the model. The ANN used was a feedforward classical FitNet network. The hybrid ARIMAX-ANN model showed improvement in forecast accuracy and fit.

The work done by Bilgic [13] shows an application of an ANN network for forecasting hourly load 24-hour ahead. The authors used previous research by others for comparison purposes. The research used a multilayer ANN with five years of scaled data. The best results are with one, three and five hidden neurons. They tested the algorithm for each of the nine geographical regions, with MAPE results in the order of 3%. Then, they tested the model on the aggregate of all regions with MAPE around 1.85% for the aggregate. Compared with previous results for the same dataset by other authors, the ANN model showed improvement in accuracy. Kandil [14] uses an MLFFN network trained with a backpropagation algorithm to achieve 0.981% error in forecasting 24-hour loads for Hydro-Quebec.

In [15], the authors study the energy consumption of Mauritius with different training schemes. The first scheme, used for comparison purposes, uses a supervised training MLFFN ANN. The second scheme uses a three-step hybrid process in which data clustering, by means of a Kohonen self-organizing map (SOM) clustering technique, which is used to group load by type, identifying the different day use patterns. The second step updates the information manually in order to incorporate the results of the first phase into an input model for the third step. The third step is to run an MLFFN ANN with a backpropagation learning algorithm using supervised learning. MAPE results for the first model run in the order of 4.l5%, whereas using the hybrid model MAPE is in the order of 3.9%.

The work presented by the author by Charytoniuk [16] focuses on predicting a very short-term (60–90 min) forecast based on the incremental changes of load during the day before. The author claims that incremental changes of load during a day are more predictable than actual loads, and therefore, his model focuses on incremental load forecasting. The research mentions that this method is more robust because it is less dependent on the actual weather conditions, especially when the forecast weather is different from the training weather values. The author uses an MLFFN ANN with only one day's data. The author also breaks down the forecast by day type, generating one forecast for each type and using up to 40 neural networks per day type.

Ekonomou et al. [17] present a comparison of several feedforward ANNs using combinations of two learning algorithms (gradient descent and Levenberg–Marquardt), three transfer functions (hyperbolic tangent sigmoid, logarithmic sigmoid and hard-limit) and several different structures (1–3 hidden layers with 2–30 neurons in each hidden layer). The methods were applied to actual loads in Greece for a period of one year with similar results in different combinations. Furthermore, Ekonomou [18] presents a hybrid method in which a wavelet denoising algorithm is used to clean up the input data to a feedforward ANN.

The work by Gooi [19] reveals an ANN network using peak and valley load with a single hidden layer. Inputs include load and predicted weather data. The output predicts either a peak or a valley load depending on whether the model is trained to forecast the peak or the valley load. Once the peak

and valley load are predicted, the hourly forecast can be obtained from a linear combination of both. Five years of historical data were used. Statistical methods are used to classify the inputs according to patterns. Ad hoc groups are created based on their similarity, with some groups, like weekend or holiday, added and some days grouped into similar day types. The model parameters are varied manually in order to improve the forecast. The parameters are the number of input nodes, the number of hidden nodes, momentum rates and learning rates. Once the model is trained, it is put online in order to use the latest one-hour load data. MAPE errors obtained are in the range of 1.43%–4.14% for days of the week and 5.8% for special days.

Tee [20] proposes a model using a model using an MLFFN with 51 inputs, 16 hidden neurons and one output layer. The inputs include 24 dummy variables for time of day, past hour load for 24 h, one dummy variable for weekend or weekday, temperature at the hour before and month of the year. The author supports the choice of just temperature as a sole weather variable, citing [3,14] that wind speed, humidity and cloud cover have little influence on the performance of the ANN. The training method is a Levenberg–Marquardt back propagation algorithm. The MAPE achieved by this study is 0.439% with a maximum MAPE of 7.986% for the month of December.

The work by Harun [21] shows a feedforward ANN, which has two data preprocessing schemes for comparison: one group does not use differencing of the data, and the other uses first order differencing in order to achieve stationarity. The authors run different simulations using lags of 24, 48 and 72 h, with the latter giving the best results. Furthermore, the stationary model gave better results for most forecast results.

Hernandez [22] uses an MLFFN in order to forecast load for a region in Spain. The author's approach is a three-layer MLFFN (input, hidden, output) with input variables day of week, month and total day load. The model uses 16 neurons in the hidden layer. A heuristic method is used to select the number of hidden neurons. Training was done via a Bayesian regulation backpropagation training function. The mean error MAPE was 2.4037%. Special days had errors in the order of 4%. In a different research work, Hernandez [23] tests two different models of MLFFN in order to forecast total load for a city. The first model is a traditional MLFFN, and the second one is a two-stage modified MLFFN in which a first MLFFN model is run to predict peaks and valleys in load for the next day, and the output of that model is used as input for the regular MLFFN model. The variables considered were precipitation, air temperature, average wind speed, average wind direction, relative humidity, atmospheric pressure and solar radiation. The MAPE for the dual-stage MLFFN model was 1.62% and 2.47% for the MLFFN.

The work by Kalaitzakis [24] is a comparison of the performance of nine different methods, ordered from the highest error to the lowest error: the autoregressive (stochastic AR) method; an MLFFN trained with backpropagation; an adaptive learning rate with back propagation; an MLFFN with Gaussian encoding; a random activation weight neural network; a weight matrix random activation weight neural network; a zero-order regularization radial basis function neural network; an MLFFN using a real-time recurrent learning algorithm; and an autoregressive recurrent neural network (ARNN). The authors found that the best performance was obtained by an ARNN, with a relative error of 1.22%.

Khotanzad [25] presents an approach with two different generations of the same MLFFN ANN. The first generation broke down the model into three separate ANNs: hourly (yesterday's and two days ago data at this hour), daily (yesterday's data for 24 h) and weekly (24-hour data for the last week on the same day). The second generation broke down the model into different hours of the day (early morning, mid-morning, afternoon peak and late night). Both models utilize an adaptive combiner in order to provide a forecast based on the output of the individual models. The team achieved 2.19% MAPE for the second generation model, compared with 2.52% MAPE for the first generation for a day-ahead forecast.

Kiartzis [26,27] presents an MLFFN structure capable of achieving 2.52% MAPE by using a three-layer MLFFN and backpropagation training. The model uses 64 inputs: loads for one and

two days ago (24 h), maximum temperature for one and two days ago, square of the temperature deviation for both days, maximum temperature difference for the past two days, day of the year as a sine wave and a cosine wave and day of the week. No comparison data are provided.

Matsumoto [28] presents a short-term load forecasting technique for summer loads, using a two-part predictor. The first part is an MLFFN that uses data from the same year only; the second part is another MLFFN that uses data from consecutive years. The forecast produced by the first part is adjusted by the trend found from the second module. This way, the model can accommodate variations in the trend from one year to another. The first ANN is used to classify inputs using the norm as the classifier, then grouping the data in order to select the forecast load to input the second module. The authors found that they can generate forecasts with an MAPE of 2.52%.

Moharari [29] shows the implementation of an MLFFN ANN for forecasting short-term loads considering special days. The input to the MLFFN network includes day variables, such as weekends and holidays, as well as weather variables, such as minimum and maximum forecast temperatures, and historical loads for the past 15 days, for a total of 23 inputs. The implementation results in a forecast with an MAPE of 1.43%.

Raza [8] presents a model using an MLFFN trained with a gradient descent algorithm. The inputs to the network include day of week, working day, hour of day, dew point, dry bulb temperature and loads for current day, day before and week before. Twenty neurons were used in the hidden layer. The forecast accuracy achieved was separated by season, and it varied from 3.81% in the spring to 4.59% during the summer. The analysis included a statistical insight into the resulting MAPE error with confidence level intervals for the error.

Papalexopoulos [30] shows a model with an MLFFN trained with 77 inputs. The architecture is a feedforward network (FitNet) that uses seasonal input variables in the shape of sine and cosine functions of a period equal to one year in order to accommodate seasonal variations; temperature for today and tomorrow (forecast), indirect temperature variables (average maximum temperature forecast, average minimum temperature forecast) for tomorrow and last week; temperature trend variables, such as the difference in temperature between today and average maximum, etc.; cooling and heating degrees by day; and historical load variables (averages, trends, peaks). The model achieved an MAPE of 1.783%.

Reinschmidt [31] describes a model with two modules: the first module is a feedforward neural network that produces an ARMA model of the load data; the second module, a recurrent ANN, uses the output from the first module, in addition to other weather variables in order to train the network to produce a load forecast. One subnet is developed for each of the 24 h of the day, and then, their results are combined to produce the complete forecast. The author claims that traditional models utilizing only ANN networks do not have the capability of adapting to sudden changes in weather conditions. The authors do not discuss the accuracy of their forecast.

Santos et al. [32] discuss a univariate ANN model that uses minimal load data, only current day, past week and past two weeks' load in order to make a forecast using a feedforward ANN. The activation function is a hyperbolic tangent function. One hidden layer was used. In order to incorporate the effects of temperature, one additional load variable is added to the input, which is the average expected load for the next day, which is calculated offline by another method. The MAPE obtained by this arrangement is 1.71%. Furthermore, Santos [33] includes reactive power in the input of an ANN model in order to reduce the forecast error. The reason the authors included this variable in their study is that the test data were taken in an area with a mild climate, and therefore, the correlation between load and weather variables was very low. The approach mentioned in the paper includes pre-processing the data in order to influence the composition of the input vector in such a way as to reduce the margin of discretion in its definition. The results yield MAPE values in the range of 3.27%–4.96%.

Shimakura [34] describes a two-step method for generating a forecast. The idea is based on the fact that there can be a trend factor in the load from one year to another in the same comparable season.

In order to remove the problems associated with ANN learning time-series data that contain trends, the author proposes a system in which the trend is first calculated and removed from the data by means of a data compensation process, and then, the data are processed in the regular manner by an MLFFN. The trend extracted by the preprocessing algorithm is then added to the forecast of the MLFFN to generate the final forecast values. The author also describes a technique in which some of the weights are not allowed to change (restricted change) in order to minimize overfitting. The MAPE obtained by this model is in the range of 2.1%–2.7%.

Zhang [35] proposes an MLFFN model that uses three layers on the ANN (input, hidden, output). The input includes information on weather, load and whether the day is a working day, weekend or holiday. Even though the network is not recurrent by design, past values of the load are used in the input vector. The network is trained with a backpropagation algorithm. The MAPE obtained by this method is in the range of 1.87%–3.051%.

Sinha [36] presents research on an MLFFN that is broken into six subnetworks in order to improve training time. Each subnetwork processes 4 h of the day. The ANN is trained using a backpropagation algorithm. The error reported was in the order of 3%.

1.3. Input Selection Methods

Da Silva [37] focuses his work on the development of the correct set of inputs to the model. The author explains that neural network models are very accurate as long as they do not overfit the data. Neural network models must match the data regularity to the model structure in order to be accurate. Current input selection methods rely on linear correlation analysis. Higher order statistical information is necessary to optimize the model. The author proposes and compares two models: a filtering model and a wrapping model. The first method looks for statistical information relevant to the inputs; the second method uses inference to estimate the relevant error caused by an input and selects those that reduce the error. The ANN selected varies from 4–8 neurons. The resulting MAPE is 2.5%.

The work by Ferreira [38] describes a similar approach as Da Silva [37], but in this case, support vector machines (SVMs) are used. SVMs are used as classifiers. The point on which the maximum margin is obtained yields the support vectors. Here, the margin is the difference between the training value and the test accuracy. A support vector regression (SVR) is performed on the inputs in order to classify the inputs. The best results were obtained with 84 inputs: 24 dummy variables for the hour of the day, lags, load, temperature and temperature squared, maximum forecast temperature value and its square and their lags by one hour. The assumption is that forecast temperatures are perfect. The idea was to present the model with a large number of inputs to identify those that are more significant. The objective was to create a robust model that could select the best input vectors for each forecasting period. Bayesian inference was applied for clustering load dynamics to feed different SVM load forecasting models and to estimate the SVM learning parameters. The latter model seems to be more promising for STLF.

Santos [39] studies the selection of the input vector to the ANN model. In order to avoid discretionary selection of the input vector, the author proposes the use of an entropy analysis to measure the level of complexity of the finite length time series of the active power. As a result, a minimal number of input variables is used, including the day of the forecast load data and two preceding weeks of corresponding values. The authors first do a data preprocessing, in order to fill the gaps left by the data acquisition method, then a correlation analysis is made on the different factors, then an entropy analysis (SampEn) on the load time series, then an autocorrelation analysis of the time series to identify the data sequences that contribute the most, then the forecast is done using an MLFFN ANN. Based on the entropy results, only the values with the higher entropy are selected, and then, a correlation analysis is done on the inputs. The forecast MAPE obtained by the method was in the range of 1.38%–2.53% for both substations under study.

Arahal [40] presents a comparison of two methods for input selection: step-wise inclusion of variables and index-based selection. The authors propose a modification of the index-based selection by doing a step-wise index-based selection.

1.4. NARX ANN Networks and ARX Knowledge-Based Models

The study by De Andrade [41] presents an implementation of a dynamic recurrent NARX ANN for load forecasting. The application is an electric substation, and the prediction forecast is for very short-term load forecasting (5 min) in order to feed an automatic generation control (AGC) in order to maintain the balance between the demand and supply of electricity. The network is used in open-loop, i.e., it is trained in parallel identification mode. The researcher used cross-validation in order to determine the structural parameters and the training of the NARX-neural network. The authors divided the data into five days for training, one day for validation and one day for testing. The study was done using one-step regressive terms. The number of neurons was fixed at five in the hidden layer. five in the hidden layer.

The research by Chen [42] presents an implementation of traditional approaches by performing knowledge-based weather segmentation and utilizing multiple autoregression with exogenous variables (ARX) models. The authors utilize load and lagged load variables, dummy variables for special days and past and forecast values for weather variables, but in addition, they classify weather pattern variables into four types of days: normal days, abnormal days, extreme days and transition days. These latter variables are the knowledge-based portion of the model. This model is site-dependent and requires that the load forecasters have knowledge of and experience with the weather patterns.

1.5. Other Computational Intelligence Methods

The research by Fattaheian [43] uses support vector regression (SVR) and the radial basis function (RBF) on an ANN network trained with a backpropagation algorithm. First, the authors applied an SVR as a regression mechanism. Then, four kernels are tried on the results: linear, polynomial, sigmoid and RBF, selecting the best of the four in terms of the MAPE of the fit. Then, an optimization problem is solved with an objective function, which is the same as the objective of the SVR model. A combinatorial model is run in order to obtain the model with the lowest MAPE. That model is the one used for forecasting. A correlation analysis is used on the input variables in order to determine the best subset of inputs.

Hayati [44] compared the performance of three different ANN structures: (a) MLFFN with one hidden layer and various configurations for number of hidden neurons; (b) Elman recurrent neural network (ERNN), which is a modified MLFFN with feedback from the hidden layer output to the input layer, also with a varying number of hidden neurons; and (c) RBFN, where the hidden layer clusters the inputs based on a radial basis function, and the output layer performs a linear transformation of the hidden layer to generate the output. The author found that the RBFN network yields the smallest error. The number of centers (neurons in the input layer) was selected manually by the user. The MAPE results were 0.17% for RBFN, 0.38% for MLFFN and 0.76% for ERNN.

Hernandez [45] tested a three-stage model with a Kohonen SOM as a previous stage, then a K-means clustering algorithm and, finally, an MLFFN for post-processing. The groupings found by the SOM were: (a) working days and holidays; (b) seasonal months; (c) weekends and weekdays. The clusters used by the K-means were: (i) working days of January, February, March, April, November, December and 15–31 October; (ii) Saturdays, Sundays and holidays; and (iii) working days of May, June, July, August, September and 1–14 October. The MAPE found by this arrangement was in the order of 2.76%. The work by Hsu [46,47] is very similar: using an SOM to group the demand patterns and predict the peaks and valleys for the next 24 h; and the second phase uses an MLFFN to produce the 24-h load as the output by means of a linear transformation. The MAPE for the combined system is in the range of 1.14%–1.25%. Similarly, Marin [48] describes a model that uses an SOM

for pre-classification of the input into similar load profiles, then uses a recurrent ANN to generate forecast load; then, the model is put online in a recall phase with a simplified model, and retraining of the forecasting model is done once a year for adjusting the weights and biases. Their Kohonen SOM uses 15 classes (patterns) of load, e.g., "Saturdays in August", which demonstrate similarities. The authors create one Elman ANN (ERNN) for each class, and the results are presented for each network. The average MAPE ranges from 1.03% for Class 14 (Tuesday–Friday of the third and fourth weeks of June and July) to 1.71% for Class 8 (Mondays from October–March).

Another method used for comparison is fuzzy logic. In [49], Badri compares a feedforward neural network (MLFFN) with a fuzzy logic (FL) algorithm for 24-hour ahead forecast. The FL algorithm starts from a linear regression on the available data. The fitted data provide a predicted peak load and maximum daily temperature. The authors did not provide information on the length of time for the data used for the MLFFN model. Their results were an accuracy of 0.5% MAPE for the MLFFN and 4.91% for the FL model.

Khosravi [50] utilizes an interval Type-2 fuzzy logic system (FLS). The idea is to utilize three dimensional fuzzy membership functions in order to accommodate uncertainty in the data. By using this FLS system, the researchers accomplished an improvement over the interval Type-1 FLS. The RMSE found was in the order of 0.170.

Kim [51] demonstrates an implementation of an MLFFN network, which produces a provisional forecast, which is later adjusted for weather and holiday behavior by a fuzzy logic algorithm to produce a final forecast. The MAPE achieved in this model is 1.3%. The MLFFN utilizes load data only. The training algorithm is backpropagation.

Mahmoud [52] presents a post-processing module that takes the output from the traditional forecasting module and, using some of the inputs to the models, adjusts their forecast output by means of a fuzzy logic algorithm. The idea is to make a forecasting system more robust to operational scenarios not reflected in the training data. The goal is to increase forecasting intelligence in order to optimize parameter selection and cover any missing knowledge in the model. The fuzzy algorithm optimizes the error by means of converging to an acceptable range. The system works similarly to a PI controller in which two gains need to be tuned in order to achieve the optimum error range. The fuzzy logic mechanism finds the optimal K_p and K_i. The technique is applied on SVM, MLFFN and FBTFSwith improvement in the forecast accuracy of 4.9%, 4.2% and 4.6%, respectively.

Santos [53] and Rafael [54] developed a model that uses a neuro-fuzzy approach utilizing Gaussian membership functions to group signals into specific days based on fuzzy logic. The fuzzy logic parameters were adjusted by using backpropagation. The number of parameters for the model were determined using trial and error. The result is a 20-member function model that yields a forecast accuracy of 3.64%.

Srinivasan [55] proposes a genetic algorithm (GA) to deal with some shortcomings of the ANNs: dependence on initial parameters, long training time, lack of a problem-independent way to choose appropriate network topology and the incomprehensive (black box) nature of ANNs. The GA is used to evolve an MLFFN and connecting weights in order to improve its forecasting accuracy. One network is developed for each day of the week, and Mondays are lumped together with days after holidays. The achieved MAPE of the evolved ANN ranges between 0.8% and 1.01%, compared with the MAPE of the comparable statistical model used by the same utility of 1.28%–1.89%, respectively.

Subbaraj [56] presents an approach using two modules: evolutionary programming (EP) and particle swarm optimization (PSO). The modules do linear combinations between the results of different MLFFN networks: instead of selecting the best fit ANN, it selects an optimal combination of ANNs to produce the forecast. The input consists of current and time-delayed values for load, temperature, relative humidity and forecast values of temperature and relative humidity for the forecast period. The EP algorithm searches for and finds the optimal solution by evolving a population of candidate solutions over a number of iterations. The PSO algorithm is a stochastic global optimization technique in which all of the solutions tend to follow the optimal one. The technique results are as follows:

the unprocessed best ANN result by an MLFFN gives an MAPE of 2.95%. By using EP, the solution improves to an MAPE of 2.24%; by using PSO, the solution improves to an MAPE of 2.27%. Therefore, EP gives a better forecast error.

Sun [57] describes a method that uses two different models for different days: a fuzzy logic (FL) support vector (SVM) method to forecast low load days, such as weekends and Mondays, and a linear extrapolation method for the rest of the weekdays. The linear extrapolation method is adapted to include weather variables. The method reduces the MAPE from 2.32% down to 1.63%

Yang [58] proposes a forecasting scheme where the input is partitioned by an FL algorithm into different groups (fuzzy sets), and an ARMAX forecast is done on each one of the groups. The input variables are pre-filtered based on their correlation with the system load. The ANN used for comparison, an MLFFN, used six inputs, 10 hidden layer neurons and one output neuron. The result is a model with an MAPE of 1.98% compared with 2.31% for MLFFN and 2.22% for ARMAX run in SAS.

Carpinteiro [59] presents a model with two SOM networks, one on top of the other. Seven inputs are used in the representation, five loads and a sine/cosine function for the hour on a 24-hour period. The comparing ANN is an MLFFN with univariate input of hourly loads (2160 instances). The SOM networks utilize feedback in between them. The results of the SOM networks show an MAPE of 2.33% vs. 2.64% for the MLFFN for a Friday and 2.03% vs. 5.92% for a Sunday.

Crone [60] proposes an empirical comparison between MLFFN and SVR models using the radial basis function (RBF) and linear kernel functions, by analyzing their forecasting power on five time series. Their results show that RBF SVR models have problems in extrapolating trends, while MLFFN and linear SVR models without data preprocessing provide robust accuracy across all patterns and clearly outperform the commonly-used RBF SVR on trended time series. The MAPE results are: 1.391% for MLFFN, 1.632% for linear SVR and 1.590% for RBF SVR.

ANNs are used in combination with other techniques in order to address specific deficiencies. Ghayekhloo [9] uses a hybrid time series and regression analysis to select the best sets on inputs and, then, a genetic algorithm to process the weights. Satish [1] uses a multi-ANN method to address the day-of-week problem, which affects most statistical methods. Buitrago [61] and also Abdulaal et al. [62] propose a framework combining parameter estimation, clustering and ANNs to group load patterns. Lopez [63] uses self-organizing maps (SOM) for addressing the groupings of load patterns. Liang [64] uses differential dynamic programming together with ANNs trained with supervised and unsupervised learning algorithms. Zhang [6] uses wavelet decomposition and an adaptive ANN model for pricing purposes. The motivation of this research is that no attempt has been found in the references to model short-term loads using non-linear autoregressive ANNs. A different implementation using a feedforward neural network can be found in [22,23].

2. Proposed Framework

2.1. NARX Model

The dynamics of the ANN using NARX can be described by its input-output relationship [65]:

$$y(t) = F[x(t), x(t-\Delta t), \ldots, x(t-n\Delta t), y(t), y(t-\Delta t), \ldots, y(t-m\Delta t)] \quad (1)$$

where n is the number of time delay steps in the input, m is the number of time delays on the feedback (output) and F is typically a nonlinear function. Notice that in Equation (1), in addition to the exogenous variables x, we incorporate the lagged output y. The exogenous and endogenous (time-lagged) inputs are shown in Table 1. The time and weather variables are exogenous inputs $x - t$, and the load y_t is the endogenous input. The network is trained using the actual values of the load y_i and then used in closed-loop by feeding back each one-hour step prediction of the load y_i to produce the next 24-hour forecast. The network is trained using the 365 previous consecutive days of data in open-loop using a Levenberg–Marquardt backpropagation algorithm. When the loop is closed, the network is fed as input the last 24 h of data and produces one-hour incremental outputs

feeding back the calculated load. During the closed-loop phase, the exogenous variables that have to do with time (month, day, hour, day of the week, working day) are used as inputs in the regular manner because they are well known. However, the weather variables (dew point and dry bulb temperatures) are taken from the weather forecast for the next day (the day of the calculated load forecast). In future work, these variables will be fed back from the multi-variable output, which will include both temperatures.

The construction of the proposed neural network starts with the structure of a feedforward perceptron network in order to learn the behavior of the output (target) y at time t (y_t), by using inputs x_t, and modeled as a nonlinear functional form of a regressional model for y (output layer):

$$y_t = \Phi[\beta_0 + \sum_{i=1}^{q} \beta_i h_{it}] \tag{2}$$

where (hidden layer)

$$h_{it} = \Psi[\gamma_{i0} + \sum_{j=1}^{n} \gamma_{ij} x_{jt}] \tag{3}$$

Φ is the activation function for the output, which is $\Phi(x) = x$, the linear function; Ψ is the activation function for the hidden neurons; in our case, the logistic function of the form:

$$\Psi(t) = \frac{1}{1 + e^{-t}} \tag{4}$$

which is used to flatten or limit the neural weights; β_0 is the output bias; β_i are the output layer weights; γ_{i0} is the input bias; and γ_{ij} are the weights of the input layer. i is the subindex of the q neurons, and j is the subindex of the n inputs. Combining Equations (2) and (3), we have:

$$y_t = \Phi\{\beta_0 + \sum_{i=1}^{q} \beta_i \Psi[\gamma_{i0} + \sum_{j=1}^{n} \gamma_{ij} x_{jt}]\} \tag{5}$$

Then, we add the dynamic term, an autoregression on the output in order to describe a recurrent network where the hidden layers are described by:

$$h_{it} = \Psi[\gamma_{i0} + \sum_{j=1}^{n} \gamma_{ij} x_{jt} + \sum_{r=1}^{q} \delta_{ir} h_{r,t-1}] \tag{6}$$

where δ_{ir} is the weight of the delayed $h_{r,t-1}$ feedback term. By replacing (6) into (2), we obtain:

$$y_t = \Phi\{\beta_0 + \sum_{i=1}^{q} \beta_i \Psi[\gamma_{i0} + \sum_{j=1}^{n} \gamma_{ij} x_{jt} + \sum_{r=1}^{q} \delta_{ir} h_{r,t-1}]\} \tag{7}$$

Equation (7) accounts for network dynamics: past values of the output and multiple inputs. However, our model so far only accounts for one hidden neural layer. We must extend the description to N layers by adding index l and the multi-dimensional nature of the τ outputs by adding index k to yield:

$$y_t^k = \Phi\{\beta_0^k + \sum_{l=1}^{N} \sum_{i=1}^{q} \beta_i^l \Psi[\gamma_{i0}^l + \sum_{j=1}^{n} \gamma_{ij}^l x_{jt} + \sum_{r=1}^{q} \delta_{ir} h_{r,t-1}]\}$$
$$k = 1 \ldots \tau \tag{8}$$

Equation (8) describes the implemented NARX neural network in this research. The open-loop and closed-loop networks are identical, except where the value of the delayed output is obtained.

The open-loop network obtains the value of y from known past values of the output, and therefore, it is a regular input to the network; and the closed-loop network obtains the value from the predicted value of the output. An example of the NARX implementation can be found in [12].

Neural Network Description

An exhaustive description of artificial neural networks features and how each one affects the performance of the forecast is beyond the scope of this paper. The most important features that need to be defined in order to construct a well-performing ANN are:

A. Feedforward or recurrent:

The ANN used in the proposed framework is first trained in open-loop, i.e., the training set includes all of the historical data for weather variables and also the historical data for the hourly loads. In this sense, it is a feedforward network. Once the node weights are found in open-loop, the network is used to calculate the output, which is then fed as input to the ANN, making it a recurrent network (feedback). Figure 1 shows the proposed structure. Notice the position of the switch: the switch is moved to the open-loop position for training, when the actual values of the load are known, and moved to the closed-loop position for forecasting, when we need to provide future values of the load and reuse the calculated values for predicting the next value. The reason why the closed-loop is used is that during training, the actual output $y(t)$ is available, and it is fed to the neural network in order to determine the network weights; however, during the forecasting phase, the actual output is not available, and therefore, the predicted delayed output is used to produce a forecast.

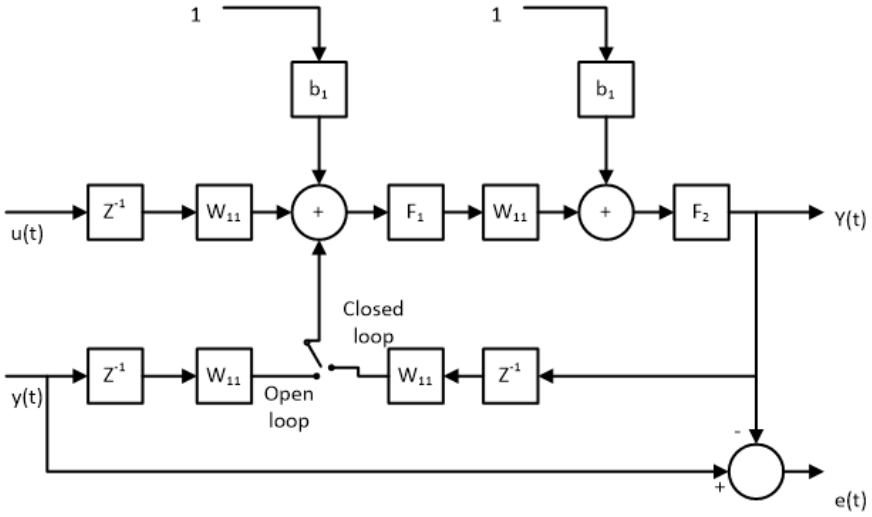


Figure 1. NARX artificial neural network structure.

B. Input size:

In order to generate a parsimonious model, i.e., a model that can provide sufficient forecasting accuracy with the minimum set of inputs, a statistical analysis of the correlation between input and output was carried out for each variable with respect to estimated load. Table 2 below shows the subset of parameters that yield the best performance with the smallest input set. The flowchart in Figure 2 shows the procedure for preparing and processing the data through the ANN.

C. Data normalization:

Data are normalized in pre-processing in the range [0, 1] by using feature scaling, i.e., the equation:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (9)$$

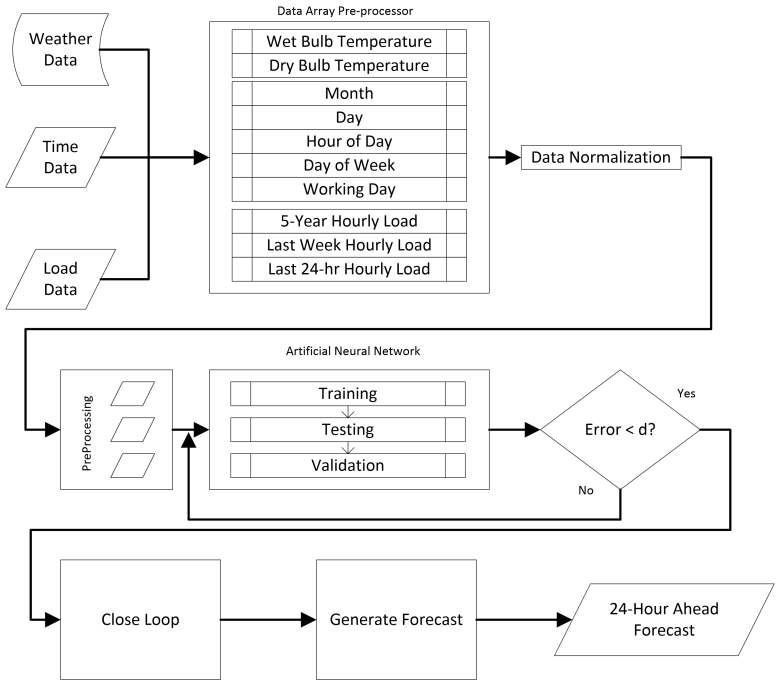


Figure 2. Proposed flowchart.

Table 1 summarizes the ANN model input.

Table 1. ANN model input.

Exogenous	Endogenous	Time Lag
Month	Load	Last 24 h
Day		
Hour of Day		
Day of Week		
Working Day		
Wet Bulb Temperature		
Dew Point		

Table 2 summarizes the ANN architecture.

Table 2. ANN architecture.

ANN Architecture	
Input Nodes	One node for each input variable
Hidden Layers	1
Hidden Nodes	Equal to number of input nodes
Output Nodes	Equal to size of forecasting horizon (1 node)
Interconnection	Full
Activation Function	Sigmoid Function
	$S(t) = \frac{1}{1+e^{-t}}$ (10)
Learning Algorithm	Levenberg–Marquardt backpropagation

D. Training and testing sample size:

The method utilizes 70% of the available data for training, 15% for testing and 15% for validation.

E. Sample size:

We used varied sample sizes from two years to five years. It is important to go over a period that covers more than one year in order to have the network learn the seasonal characteristics of the load.

F. Performance measurements:

The measure of performance used in this work is the mean absolute percent error defined as:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|E_i| \times 100\% \tag{11}$$

where the error E_i is given by:

$$E_i = \frac{L_i^{actual} - L_i^{forecast}}{L_i^{actual}} \tag{12}$$

The error has been calculated in two instances: in open-loop, the error corresponds to the fit error, i.e., the difference between the actual value and the output value of the open-loop network for the validation set. This error is used to select the best network, which is the one that will be used for forecasting. Once the forecast is produced in closed-loop, the error is calculated between the actual value of the load, which is known to us because we are using past data for validation, and the closed-loop forecast output. This error is the error reported in the results as forecast error.

G. Operation procedure:

Starting with the dataset, including year, month, date, day, hour, day of the week, whether the day is a working day or a holiday, dew point temperature and dry bulb temperature, as well as hourly load, for five years, we first segment the data into one-year groups. Each one-year group is used for testing. The data are also aggregated with the last week load data (as a redundant input) and last day (24 h) also as a redundant input. The data are then prepared so that the input size matches the number of input neurons. The initial weights are assigned in order to guarantee stability, as discussed in Section 5. The resulting dataset is split into training, validation and testing sets. The training set undergoes Levenberg–Marquardt backpropagation in order to obtain the open-loop weights. Testing and validation are done on the dataset. This process is repeated 10 times each for 5–30 neurons in increments of five. At the end, the best fitting open-loop network is selected, based on the minimum MAPE criterion. The process does not guarantee an optimal solution, just the selection of the best solution available from within many different parameters. The best open-loop network is selected; the loop is closed; and the date and weather data forecast is fed to it in increments of one hour, in order to produce one-hour forecast at a time for the next 24 h. Now, the model utilizes its own output load forecast as an input to generate the next hour forecast. This is how the 24-hour forecast is obtained.

H. Model parameters:

The parameters of the model are the inputs, the lag of the inputs, the number of hidden layers, the number of neurons in each layer and the connectivity between neurons. The selection of the parameters was done as follows: For the inputs, a multiple correlation and partial correlation analysis was performed between a set of available variables, and the subset with the highest correlation was selected based on a stepwise selection. The lag of the inputs was tested from 1 h–168 h, obtaining a good trade off between performance and error for 24 h. The number of hidden layers was tested from 1–5, with significantly better results for one hidden layer. The number of neurons in the hidden

layers was tested from 5–30 neurons, and it was proven that a number between 10 and 15 neurons worked best. A fully-connected network following a NARX definition was used as the selected model. Again, the parameters selected do not guarantee an optimal result, but the best result available from within the tested solutions.

2.2. Simulation Environment

The NARX and FitNet ANN models were simulated using MATLAB 2016 Neural Networks Toolbox. The ARMAX and state space models were simulated using IBM's SPSS statistical package.

Two sets of simulations were run in MATLAB: one set for a feedforward neural network, which is used for performance comparison, as it is the de facto standard for neural network data forecasting, and a second set for an NARX neural network as described in this paper. Both sets were run with the same dataset and the same number of iterations on the following variables:

- Number of data groups: 10 different data periods were used;
- Number of neural networks: five different runs for each iteration in order to separate the outcome from the initial conditions;
- Number of neurons in the hidden layer: simulations were run from 5–30 neurons in increments of five.

In addition, an ARMAX and a state space estimation model were developed for comparison using IBM's SPSS statistical package. The best ARIMA model produced in SPSS was an ARIMA (2, 1, 8), which is using a second order autoregressive order, one degree of differencing and an eighth degree moving average. The ARIMA model was selected from SPSS output utilizing Expert Modeler in order to minimize the fit error. The resulting ARIMA model was transformed into a state space model to obtain the comparison. The best NARX network based on the MAPE fit of the existing data in open loop was selected in order to provide a forecast. The NARX neural network was trained in open-loop using historical data and then used for forecasting in closed-loop using the calculated load as the input for the next step. The MATLAB network configuration is shown in Figure 3.

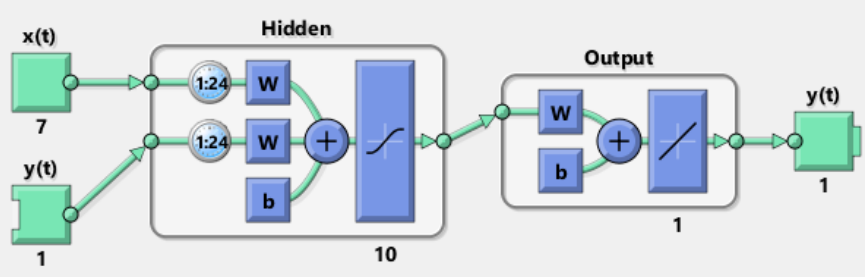
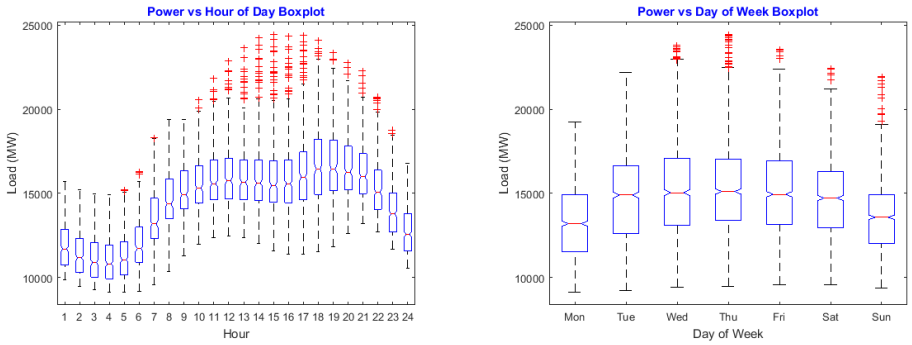


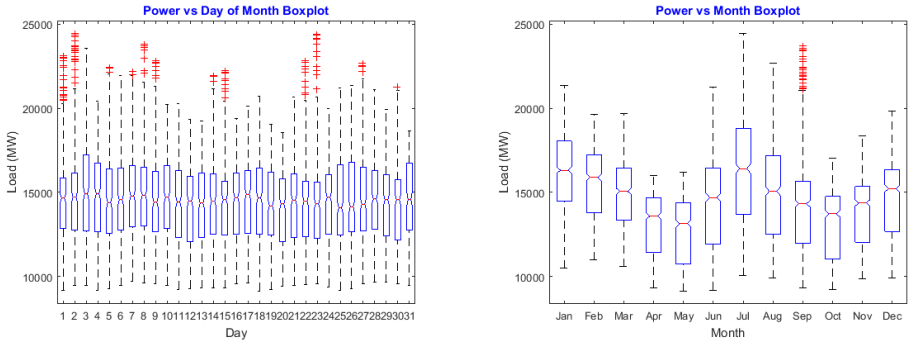
Figure 3. MATLAB NARX network configuration.

3. Data

Hourly real-time system demand load data were obtained from the ISO-NEgrid operator for New England from 2005–2015 [66]. The statistical characteristics of the load data are shown in Figure 4. Notice the seasonal characteristic of the load, i.e., variability month to month, the particular reduction in the load on weekends and the variation in the load according to the hour of the day. These features of the data are very appropriate for using ANNs, since all of the distinct features are identified by the parameters in the network. A correlation analysis was carried out from a set of candidate independent variables, and the result of this study is the input vector selected for the proposed framework shown in Table 2.



(**a**) Hour of Day (**b**) Day of Week



(**c**) Day of the Month (**d**) Month of the Year

Figure 4. Statistical characteristics of electrical load data, (**a**) Hour of Day; (**b**) Day of Week; (**c**) Day of Month; (**d**) Month of Year.

4. Results

Table 3 shows the results of the simulations for a 24-hour-ahead load forecast and the corresponding error performance comparison between the feedforward neural (FitNet) network, the ARMAX model, the state space estimation and the NARX neural network. The NARX forecast was generated in closed-loop, i.e., the network was trained in open-loop by using known values of the load; then, the first hourly load forecast value is calculated with the trained network, and it is fed back to the input in order to obtain the second value, and so on. The open-loop fit of the NARX model is shown in Figure 5.

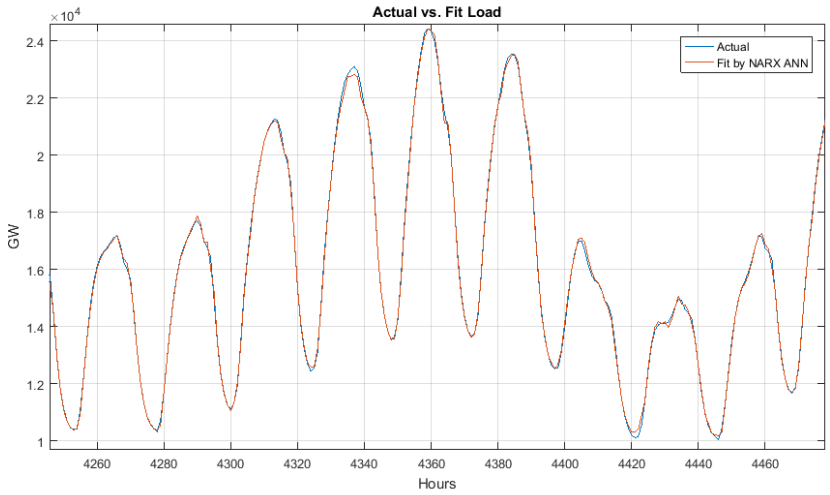


Figure 5. Open-loop fit of the NARX ANN model for one week.

Table 3. Results comparison for FitNet, ARMAX, State Space and NARXNet.

		FitNet		ARMAX		State Space		NARXNet	
Hour	**Actual**	**Forecast**	**AE%**	**Forecast**	**AE%**	**Forecast**	**AE%**	**Forecast**	**AE%**
1	12,115	12,165	0.41	12,248	1.1	12,269	1.27	12,141	0.21
2	11,576	11,713	1.19	11,749	1.49	11,553	0.2	11,657	0.7
3	11,254	11,349	0.85	11,278	0.21	10,679	5.11	11,155	0.88
4	11,073	11,074	0.01	10,933	1.26	10,090	8.87	11,229	1.41
5	11,084	10,963	1.09	10,886	1.79	10,901	1.65	11,134	0.45
6	11,326	11,123	1.79	11,191	1.19	11,246	0.7	11,449	1.08
7	11,780	11,649	1.11	11,624	1.32	12,067	2.44	11,790	0.09
8	12,338	12,456	0.96	12,054	2.3	13,132	6.44	12,270	0.55
9	13,193	13,402	1.59	12,932	1.98	13,532	2.57	12,960	1.77
10	13,752	13,757	0.04	13,694	0.42	13,452	2.18	13,790	0.28
11	13,978	13,894	0.6	14,023	0.32	13,502	3.41	13,997	0.14
12	14,018	13,867	1.07	14,077	0.42	13,825	1.38	14,022	0.03
13	13,980	13,708	1.94	14,076	0.69	14,098	0.85	14,052	0.52
14	13,900	13,660	1.73	14,041	1.02	14,141	1.74	13,842	0.42
15	13,905	13,777	0.92	14,051	1.05	14,260	2.55	13,928	0.16
16	14,298	14,053	1.71	14,554	1.79	14,782	3.38	14,153	1.01
17	15,634	14,726	5.81	15,916	1.8	15,509	0.8	15,452	1.16
18	16,275	15,280	6.11	16,452	1.09	15,931	2.11	16,725	2.77
19	16,144	15,500	3.99	16,257	0.7	15,855	1.79	15,935	1.3
20	15,633	15,412	1.41	15,769	0.87	15,506	0.81	15,872	1.53
21	15,020	14,879	0.94	15,227	1.38	15,048	0.19	15,241	1.47
22	13,927	13,974	0.34	14,069	1.02	14,319	2.81	14,141	1.54
23	12,656	12,811	1.22	12,679	0.18	13,179	4.13	12,573	0.66
24	11,509	11,553	0.38	11,584	0.65	11,885	3.27	11,546	0.32
Maximum			6.11		2.30		8.87		2.77
MAPE%			1.55		1.09		2.53		0.85

As seen from the results Table 3, the MAPE for the NARX neural network is 0.85%, for the feedforward network is 1.55%, for the ARMAX model is 1.09% and for the state space estimation is 2.53%, which shows a great improvement accuracy offered by the NARX network. The maximum absolute percent errors are also shown in the table. A 30% improvement is achieved in forecasting performance.

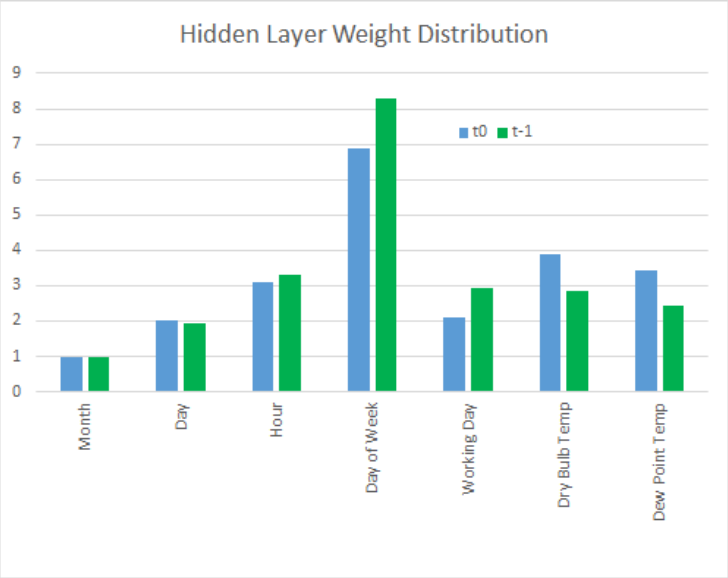


Figure 6. Synaptical weight distribution.

The distribution of the hidden layer synaptic weights is shown in Figure 6. No silent synapses were found in the model, and the highest weight was given to the input associated with the day of the week, followed by the dry bulb temperature.

The training, by means of a Levenberg–Marquardt backpropagation algorithm, converged after fewer than 50 epochs, and it showed stability (no increase after converging) and no overshoot (no increase before converging), as shown in Figure 7.

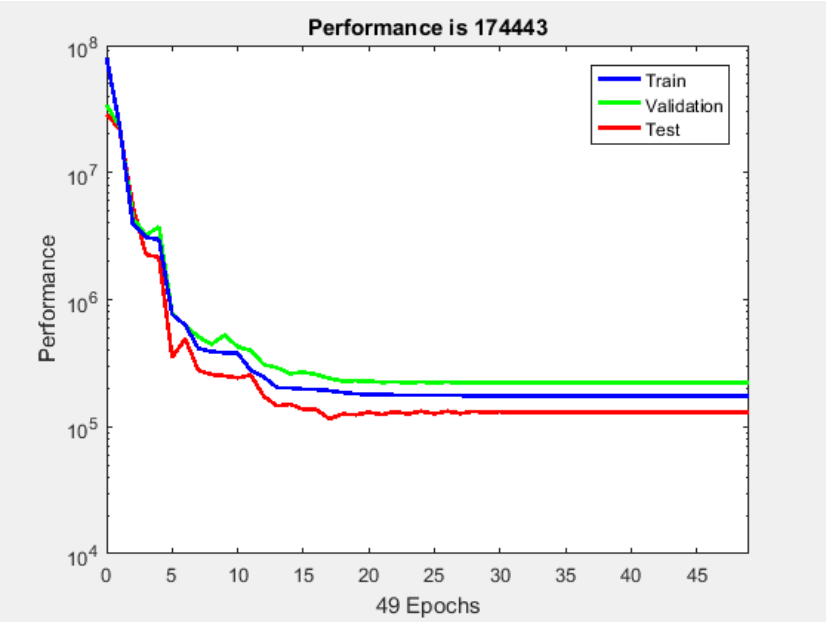


Figure 7. Training performance graph.

Figure 8 shows the forecast tracking for each of the methods used. In addition, the error histogram for the fit set is shown in Figure 9. The regression on the error between forecast and actual is a measure of performance, and it is plotted in Figure 10.

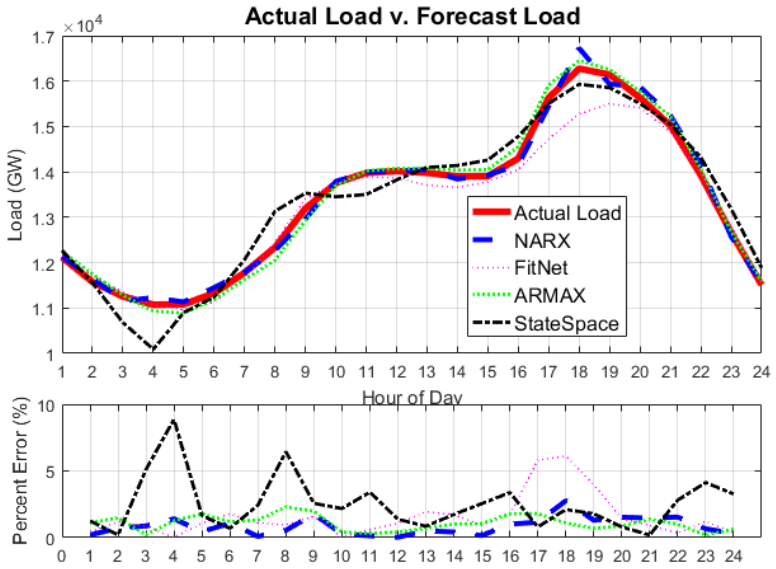


Figure 8. Twenty four-hour actual vs. forecast comparison.

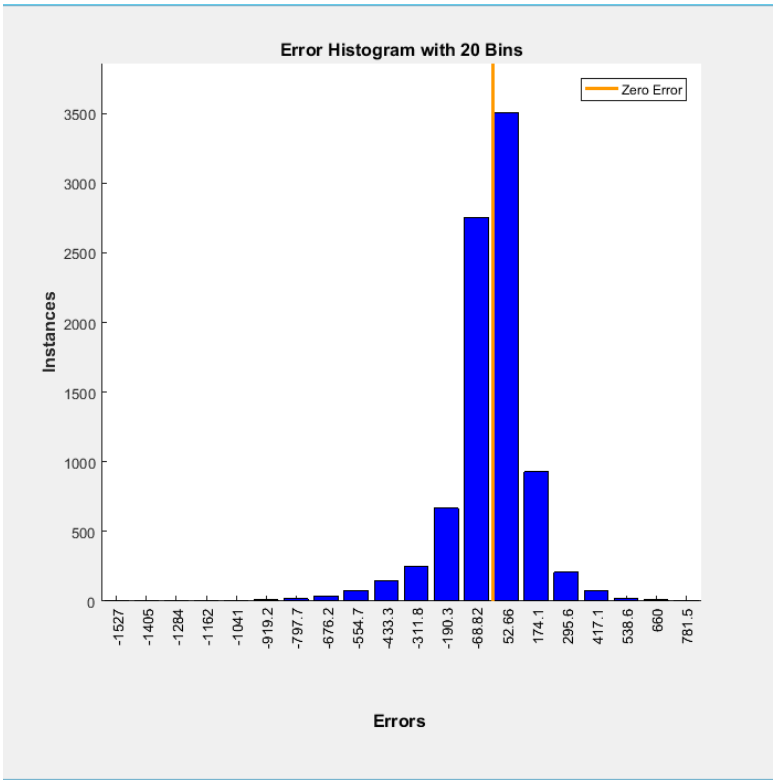


Figure 9. Error histogram for the fit set.

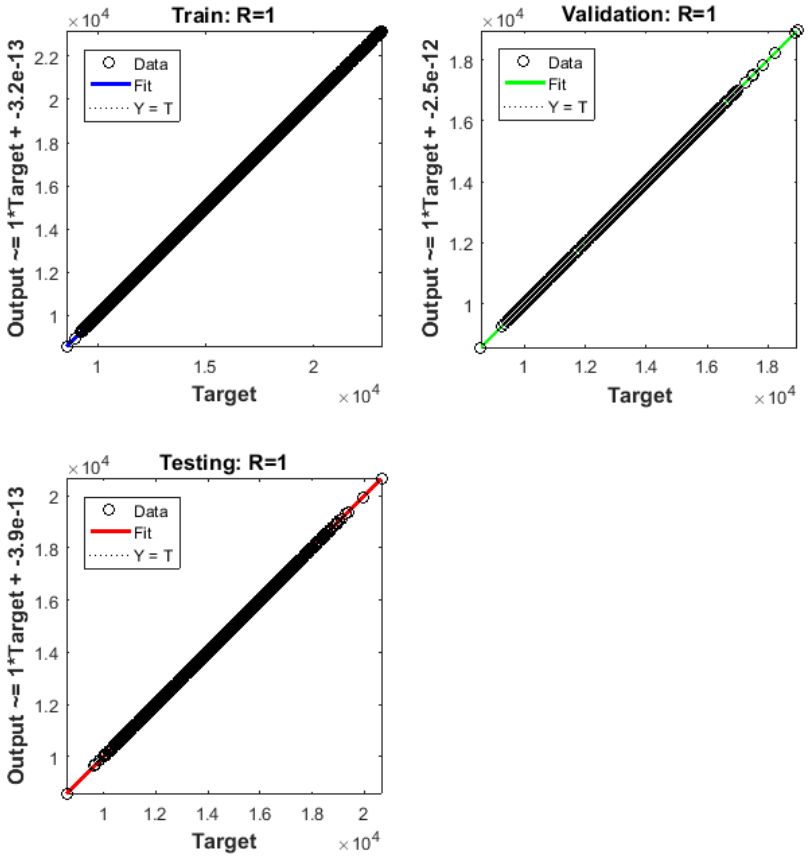


Figure 10. Error regression on forecast vs. actual.

5. Discussion

The algorithm was run under many different conditions and configurations. The following items have been taken into consideration:

- Different input periods (both in time and length): this is to include and exclude the effects of holidays, working and non-working days, seasons, etc. ANN NARX works well when sufficient cyclical data are available for training.
- The number of neurons in the hidden layer in the model: simulations were run from 1–30 neurons in the hidden layer. Typically, the forecast accuracy peaks at around 20 neurons.
- The number of delayed inputs and feedback: simulations ranging from 1 h–48 h delays were performed. A 24-hour time delay for inputs worked best.
- The redundancy on the input data: reinforcement of training data with the most recent available data. It was found that feeding redundant data for the last week and the last 24 h worked best.

An important aspect of the simulation results, which can be seen in Figure 8 is that there is a slight overshoot of the forecast calculated by the NARX model. The overshoot (on Hour 18) corresponds to a 2.77% error in the forecast with respect with the actual value. The maximum point error by the ARMAX model was 2.3%. Typically forecast overshoots of this nature are the result of overfitting, in the case of statistical models, or having too many hidden neurons in the case of ANNs. However, as the results show, the mean of the error is smaller in the NARX model than it is in the ARMAX model. Overall, the NARX model gives a much closer forecast across the time series. Further exploration needs to be done in order to analyze the effect of the number of hidden neurons on both the mean error and the point error.

A comparison of the results for regular days (Tuesday to Friday) versus special days (Mondays, weekends and holidays) is shown in Figure 11. The results are very similar for both groups of days.

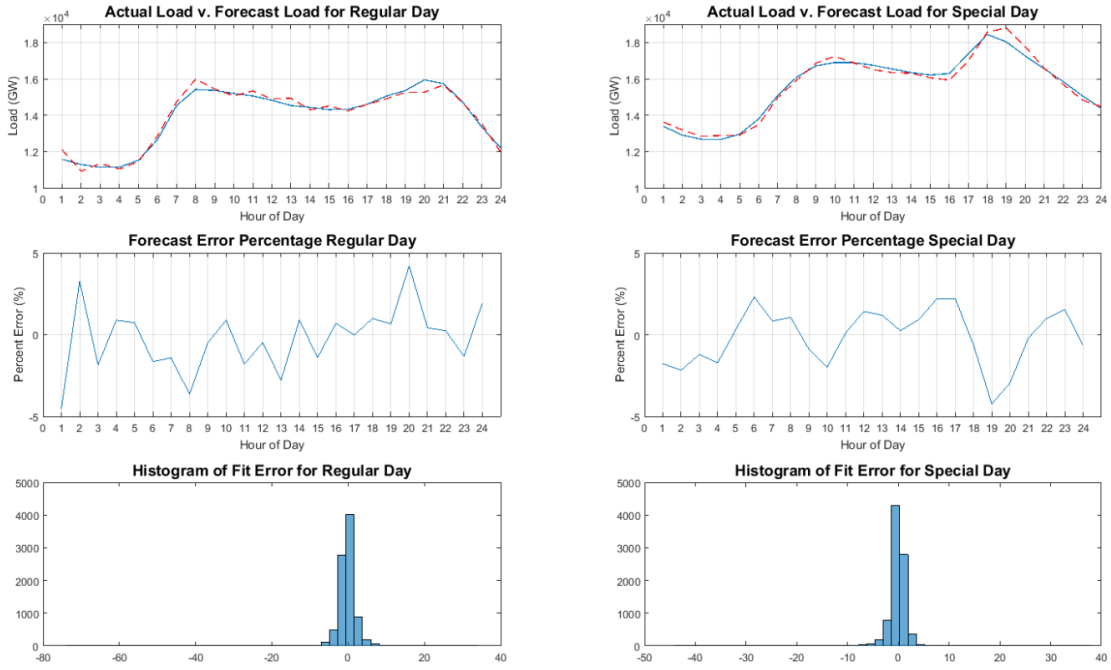


Figure 11. Comparison between results for regular vs. special days.

Stability

An important issue to discuss about closed-loop systems is stability. Many researchers have studied the problem of closed-loop stability and have demonstrated that an adequate selection of initial weights is essential for the convergence of the solutions during the learning process.

Furthermore, Irigoyen et al. [67] provide a framework for determining initial weights for NARX networks. Their results show that a sufficient condition for an NARX network in closed-loop, with two layers and K sigmoid neurons in the output layer and J in the hidden one, to be stable is that the initial weights w_{kl}^{m} have a modulus that meets the requirement

$$|w_{kl}^{m}| \leq \frac{4}{\sqrt{JK}} \tag{13}$$

In the case of the proposed model $J = 10$ and $K = 1$, and therefore, we must have initial weights that meet:

$$|w_{kl}^{m}| \leq \frac{4}{\sqrt{10}} \approx 1.26 \tag{14}$$

The initial weights were started in the range $[-1.26, 1.26]$ to ensure stability. Furthermore the convergence curves shown for training show no signs of divergence of the error.

6. Conclusions

A novel implementation of a nonlinear autoregressive neural network with exogenous input (NARX) has been presented. The approach shows that an ANN can be trained in open-loop by using all of the available endogenous and exogenous inputs. Electric load has been used as the endogenous variable, and time and weather (dry bulb and dew point temperatures) are used as exogenous inputs. The network is a recursive ANN with connections between the output, hidden and input layers. The network was trained using a Levenberg–Marquardt backpropagation algorithm. Once the neural weights are calculated, the load input is disconnected, and the predicted (forecast) value of the output is fed back to the input. This isolates the network and removes the requirement for retraining for generating each instance of the output (predicted load). The accuracy of the traditional ANN network is improved to a forecast MAPE error of less than 1%. The authors have compared the proposed method with traditional statistical models ARMAX and state space, as well as the forecast calculated by a feedforward ANN with a multilayer perceptron architecture. In all three cases, the proposed NARX architecture provides a lower MAPE error. In practical terms, this translates into savings because the forecast load is used to commit power plants for power availability, and the more accurate the forecast is, the better the operations performance achieved, thus saving energy and cost. The accuracy in the forecast is very important, especially for borderline cases in which a plant start up may be delayed for an hour or even avoided altogether given a reliable forecast. Impact on a network, such as ISO New England, where the data for this research was obtained, a network with a 19.3-GW available capacity and typical demand varying from 10 GW–18 GW, could represent savings of 0.5% of the installed capacity, i.e., 100 MW for periods of time ranging between 1 and 4 h per day. These are the typical overshoot times for forecasted loads, and their predictability offers an opportunity for real cost savings.

Acknowledgments: This research has been partially funded by a grant from the U.S. Department of Energy for the University of Miami Industrial Assessment Center.

Author Contributions: Shihab Asfour helped develop the statistical model for comparison. Jaime Buitrago developed the Artificial Neural Network models and the state space model.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
ARMA	Autoregressive and moving average model
ARMAX	Autoregressive and moving average with exogenous input model
CI	Computational intelligence

MAPE	Mean absolute percent error
NARX	Nonlinear autoregressive model with exogenous input
STLF	Short-term load forecasting

References

1. Satish, B.; Swarup, K.S.; Srinivas, S.; Rao, A.H. Effect of temperature on short term load forecasting using an integrated ANN. *Electr. Power Syst. Res.* **2004**, *72*, 95–101.
2. Nengling, T.; Stenzel, J.; Hongxiao, W. Techniques of applying wavelet transform into combined model for short-term load forecasting. *Electr. Power Syst. Res.* **2006**, *76*, 525–533.
3. Hippert, H.S.; Pedreira, C.E.; Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.* **2001**, *16*, 44–55.
4. Hahn, H.; Meyer-Nieberg, S.; Pickl, S. Electric load forecasting methods: Tools for decision making. *Eur. J. Oper. Res.* **2009**, *199*, 902–907.
5. Metaxiotis, K.; Kagiannas, A.; Askounis, D.; Psarras, J. Artificial intelligence in short term electric load forecasting: A state-of-the-art survey for the researcher. *Energy Convers. Manag.* **2003**, *44*, 1525–1534.
6. Zhang, G.; Patuwo, E.B.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62.
7. Tzafestas, S.; Tzafestas, E. Computational intelligence techniques for short-term electric load forecasting. *J. Intell. Robot. Syst. Theory Appl.* **2001**, *31*, 7–68.
8. Raza, M.; Khosravi, A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew. Sustain. Energy Rev.* **2015**, *50*, 1352–1372.
9. Ghayekhloo, M.; Menhaj, M.B.; Ghofrani, M. A hybrid short-term load forecasting with a new data preprocessing framework. *Electr. Power Syst. Res.* **2015**, *119*, 138–148.
10. Abdel-Aal, R.E. Univariate modeling and forecasting of monthly energy demand time series using abductive and neural networks. *Comput. Ind. Eng.* **2008**, *54*, 903–917.
11. Alfuhaid, A.S.; El-Sayed, M.A.; Mahmoud, M.S. Cascaded artificial neural networks for short-term load forecasting. *IEEE Trans. Power Syst.* **1997**, *12*, 1524–1529.
12. Bennett, C.; Stewart, R.A.; Lu, J. Autoregressive with exogenous variables and neural network short-term load forecast models for residential low voltage distribution networks. *Energies* **2014**, *7*, 2938–2960.
13. Bilgic, M.; Girep, C.; Aslanoglu, S.; Aydinalp-Koksal, M. Forecasting Turkey's short term hourly load with artificial neural networks. In Proceedings of the 10th IET International Conference on Developments in Power System Protection (DPSP 2010), Managing the Change, Manchester, UK, 29 March–1 April 2010; pp. 1–5.
14. Kandil, N.; Wamkeue, R.; Saad, M.; Georges, S. An efficient approach for short term load forecasting using artificial neural networks. *Int. J. Electr. Power Energy Syst.* **2006**, *28*, 525–530.
15. Bugwan, T.; King, R.T.A. Short term electrical load forecasting for mauritius using Artificial Neural Networks. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Singapore, 12–15 October 2008; pp. 3668–3673.
16. Charytoniuk, W.; Chen, M.S. Very short-term load forecasting using artificial neural networks. *IEEE Trans. Power Syst.* **2000**, *15*, 263–268.
17. Ekonomou, L.; Oikonomou, D. Application and comparison of several artificial neural networks for forecasting the Hellenic daily electricity demand load. In Proceedings of the 7th WSEAS International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases, Cairo, Egypt, 29–31 December 2008; pp. 67–71.
18. Ekonomou, L.; Christodoulou, C.; Mladenov, V. A Short-Term Load Forecasting Method Using Artificial Neural Networks and Wavelet Analysis. *Int. J. Power Syst.* **2016**, *1*, 64–68.
19. Gooi, H.; Teo, C.; Chin, L.; Ang, S.; Khor, E. Adaptive short-term load forecasting using artificial neural networks. In Proceedings of the 1993 IEEE Region 10 Conference on Computer, Communication, Control and Power Engineering (TENCON '93), Beijing, China, 19–21 October 1993.
20. Tee, C.Y.; Cardell, J.B.; Ellis, G.W. Short-term load forecasting using artificial neural networks. In Proceedings of the North American Power Symposium (NAPS), Starkville, MI, USA, 5–7 October 2009; pp. 1-6.

21. Harun, M.H.H.; Othman, M.M.; Musirin, I. Short term load forecasting (STLF) using artificial neural network based multiple lags and stationary time series. In Proceedings of the 4th International Power Engineering and Optimization Conference, Shah Alam, Malaysia, 23–24 June 2010; pp. 363–370.
22. Hernandez, L.; Baladron Zorita, C.; Aguiar Perez, J.M.; Carro Martinez, B.; Sanchez Esguevillas, A.; Lloret, J. Short-Term Load Forecasting for Microgrids Based on Artificial Neural Networks. *Energies* **2013**, *6*, 1385–1408.
23. Hernandez, L.; Baladron, C.; Aguiar, J.M.; Calavia, L.; Carro, B.; Sanchez-Esguevillas, A.; Sanjuan, J.; Gonzalez, L.; Lloret, J. Improved short-term load forecasting based on two-stage predictions with artificial neural networks in a microgrid environment. *Energies* **2013**, *6*, 4489–4507.
24. Kalaitzakis, K.; Stavrakakis, G.S.; Anagnostakis, E.M. Short-term load forecasting based on artificial neural networks parallel implementation. *Electr. Power Syst. Res.* **2002**, *63*, 185–196.
25. Khotanzad, A.; Afkhami-Rohani, R.; Lu, T.L.; Abaye, A.; Davis, M.; Maratukulam, D.J. ANNSTLF—A neural-network-based electric load forecasting system. *IEEE Trans. Neural Netw.* **1997**, *8*, 835.
26. Kiartzis, S.; Zoumas, C.; Bakirtzis, A.; Petridis, V. Data pre-processing for short-term load forecasting in an autonomous power system using artificial neural networks. In Proceedings of the Third IEEE International Conference on Electronics, Circuits, and Systems (ICECS'96), Rodos, Greece, 13–16 October 1996; Volume 2, pp. 1021–1024.
27. Kiartzis, S.J.; Bakirtzis, A.G.; Petridis, V. Short-term load forecasting using neural networks. *Electr. Power Syst. Res.* **1995**, *33*, 1–6.
28. Matsumoto, T.; Kitamura, S.; Ueki, Y.; Matsui, T. Short-term load forecasting by artificial neural networks using individual and collective data of preceding years. In Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems (ANNPS '93), Yokohama, Japan, 19–22 April 1993; pp. 245–250.
29. Moharari, N.S.; Debs, A.S. An artificial neural network based short term load forecasting with special tuning for weekends and seasonal changes. In Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems (ANNPS'93), Yokohama, Japan, 19–22 April 1993; pp. 279–283.
30. Papalexopoulos, A.; Hao, S.; Peng, T.M. Short-term system load forecasting using an artificial neural network. In Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems (ANNPS'93), Yokohama, Japan, 19–22 April 1993; pp. 239–244.
31. Reinschmidt, K.; Ling, B. Artificial neural networks in short term load forecasting. In Proceedings of the 4th IEEE Conference on Control Applications, Columbus, OH, USA, 28–29 September 1995; pp. 209–214.
32. Santos, J.P.; Gomes, M.A.; Pires, A.J. Next hour load forecast in medium voltage electricity distribution. *Int. J. Energy Sect. Manag.* **2008**, *2*, 439–448.
33. Santos, P.J.; Martins, A.G.; Pires, A.J. On the use of reactive power as an endogenous variable in short-term load forecasting. *Int. J. Energy Res.* **2003**, *27*, 513–529.
34. Shimakura, Y.; Fujisawa, Y.; Maeda, Y.; Makino, R.; Kishi, Y.; Ono, M.; Fann, J.Y.; Fukusima, N. Short-term load forecasting using an artificial neural network. In Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems (ANNPS '93), Yokohama, Japan, 19–22 April 1993; pp. 233–238.
35. Zhang, S.; Lian, J.; Zhao, Z.; Xu, H.; Liu, J. Grouping model application on artificial neural networks for short-term load forecasting. In Proceedings of the 7th World Congress on Intelligent Control and Automation (WCICA), Chongqing, China, 25–27 June 2008; pp. 6203–6206.
36. Sinha, A. Short term load forecasting using artificial neural networks. In Proceedings of the IEEE International Conference on Industrial Technology, Goa, India, 19–22 January 2000; Volume 1, pp. 548–553.
37. Da Silva, A.P.A.; Ferreira, V.H.; Velasquez, R.M.G. Input space to neural network based load forecasters. *Int. J. Forecast.* **2008**, *24*, 616–629.
38. Ferreira, V.H.; Alves da Silva, A.P. Toward estimating autonomous neural network-based electric load forecasters. *IEEE Trans. Power Syst.* **2007**, *22*, 1554–1562.
39. Santos, P.J.; Martins, A.G.; Pires, A.J. Designing the input vector to ANN-based models for short-term load forecast in electricity distribution systems. *Int. J. Electr. Power Energy Syst.* **2007**, *29*, 338–347.
40. Arahal, M.R.; Cepeda, A.; Camacho, F. Input variable Selection for Forcasting Models. In Proceedings of the 15th IFAC World Congress on Automatic Control, Barcelona, Spain, 21–26 July 2002.

41. De Andrade, L.C.M.; Oleskovicz, M.; Santos, A.Q.; Coury, D.V.; Fernandes, R.A.S. Very short-term load forecasting based on NARX recurrent neural networks. In Proceedings of the IEEE PES General Meeting Conference and Exposition, National Harbor, MD, USA, 27–31 July 2014; pp.1–5.
42. Chen, H.; Du, Y.; Jiang, J.N. Weather sensitive short-term load forecasting using knowledge-based ARX models. In Proceedings of the IEEE Power Engineering Society General Meeting, San Francisco, CA, USA, 12–16 June 2005; Volume 1, pp. 190–196.
43. Fattaheian, S.; Fereidunian, A.; Gholami, H.; Lesani, H. Hour-ahead demand forecasting in smart grid using support vector regression (SVR). *Int. Trans. Electr. Energy Syst.* **2014**, *24*, 1650–1663.
44. Hayati, M. Short term load forecasting using artificial neural networks for the west of Iran. *J. Appl. Sci.* **2007**, *7*, 1582–1588.
45. Hernandez, L.; Baladron, C.; Aguiar, J.M.; Carro, B.; Sanchez-Esguevillas, A.; Lloret, J. Artificial neural networks for short-term load forecasting in microgrids environment. *Energy* **2014**, *75*, 252–264.
46. Hsu, Y.Y.; Yang, C.C. Design of artificial neural networks for short-term load forecasting. Part II. Multilayer feedforward networks for peak load and valley load forecasting. *IEE Proc. C Gener. Transm. Distrib.* **1991**, *138*, 414–418.
47. Hsu, Y.Y.; Yang, C.C. Design of artificial neural networks for short-term load forecasting. Part I. Self-organising feature maps for day type identification. *IEE Proc. C Gener. Transm. Distrib.* **1991**, *138*, 407–413.
48. Marin, F.J.; Garcia-Lagos, F.; Joya, G.; Sandoval, F. Global model for short-term load forecasting using artificial neural networks. *IEE Proc. C Gener. Transm. Distrib.* **2002**, *149*, 121–125.
49. Badri, A.; Ameli, Z.; Birjandi, A.M. Application of Artificial Neural Networks and Fuzzy logic Methods for Short Term Load Forecasting. *Energy Procedia* **2012**, *14*, 1883–1888.
50. Khosravi, A.; Nahavandi, S.; Creighton, D.; Srinivasan, D. Interval type-2 fuzzy logic systems for load forecasting: A comparative study. *IEEE Trans. Power Syst.* **2012**, *27*, 1274–1282.
51. Kim, K.H. Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems. *IEEE Trans. Power Syst.* **1995**, *10*, 1534–1539.
52. Mahmoud, T.S.; Habibi, D.; Hassan, M.Y.; Bass, O. Modelling self-optimised short term load forecasting for medium voltage loads using tunning fuzzy systems and Artificial Neural Networks. *Energy Convers. Manag.* **2015**, *106*, 1396–1408.
53. Santos, P.; Rafael, S.; Lobato, P.; Pires, A. A STLF in distribution systems—A short comparative study between ANFIS Neuro-Fuzzy and ANN approaches—Part I. In Proceedings of the International Conference on Power Engineering, Energy and Electrical Drives (POWERENG), Lisbon, Portugal, 18–20 March 2009; pp. 661–665.
54. Rafael, S.; Santos, P.; Lobato, P.; Pires, A. A STLF in distribution systems—A short comparative study between ANFIS Neuro-Fuzzy and ANN approaches—Part II. In Proceedings of the International Conference on Power Engineering, Energy and Electrical Drives (POWERENG), Lisbon, Portugal, 18–20 March 2009.
55. Srinivasan, D. Evolving artificial neural networks for short term load forecasting. *Neurocomputing* **1998**, *23*, 265–276.
56. Subbaraj, P.; Rajasekaran, V. Short Term Hourly Load Forecasting Using Combined Artificial Neural Networks. In Proceedings of the International Conference on Conference on Computational Intelligence and Multimedia Applications, Sivakasi, India, 13–15 December 2007; Volume 1, pp. 155–163.
57. Sun, C.; Song, J.; Li, L.; Ju, P. Implementation of hybrid short-term load forecasting system with analysis of temperature sensitivities. *Soft Comput.* **2008**, *12*, 633–638.
58. Yang, H.T.; Huang, C.M. New short-term load forecasting approach using self-organizing fuzzy ARMAX models. *IEEE Trans. Power Syst.* **1998**, *13*, 217–225.
59. Carpinteiro, O.A.S.; Reis, A.J.R.; da Silva, A.P.A. A hierarchical neural model in short-term load forecasting. *Appl. Soft Comput.* **2004**, *4*, 405–412.
60. Crone, S.F.; Guajardo, J.; Weber, R. A study on the ability of support vector regression and neural networks to forecast basic time series patterns. In Proceedings of the 4th IFIP International Conference on Artificial Intelligence in Theory and Practice, Perth, Australia, 6–7 October 2006; pp. 149–158.
61. Buitrago, J.; Abdulaal, A.; Asfour, S. Electric load pattern classification using parameter estimation, clustering and artificial neural networks. *Int. J. Power Energy Syst.* **2015**, *35*, 167–174.

62. Abdulaal, A.; Buitrago, J.; Asfour, S. Electric Load Pattern Classification for Demand-side Management Planning: A Hybrid Approach. In *Software Engineering and Applications: Advances in Power and Energy Systems*; Press, A., Ed.; ACTA Press: Marina del Rey, CA, USA, 2015.
63. Lopez, M.; Valero, S.; Senabre, C.; Aparicio, J.; Gabaldon, A. Application of SOM neural networks to short-term load forecasting: The Spanish electricity market case study. *Electr. Power Syst. Res.* **2012**, *91*, 18–27.
64. Liang, R.H.; Hsu, Y.Y. A hybrid artificial neural network—Differential dynamic programming approach for short-term hydro scheduling. *Electr. Power Syst. Res.* **1995**, *33*, 77–86.
65. Venturini, M. Simulation of compressor transient behavior through recurrent neural network models. *J. Turbomach. Trans. ASME* **2006**, *128*, 444–454.
66. England, I.N. *Energy, Load and Demand Reports*; ISO New England: Holyoke, MA, USA, 2015.
67. Irigoyen, E.; Pinzolas, M. Numerical bounds to assure initial local stability of NARX multilayer perceptrons and radial basis functions. *Neurocomputing* **2008**, *72*, 539–547.