

An Effective Short-Term Electrical Load Forecasting Model: A Constructive Neural Network Approach

Kazi Rafiqul Islam, Md. Monirul Kabir, Amit Shaha Surja, and Md. Shahid Iqbal

Abstract — In this paper, an Effective Electrical Load Forecasting (EELF) model has been introduced based on Feed-Forward Neural Network (FFNN) which utilizes the constructive method during training. The key aspect of this model is to automate the FFNN architecture during training phase in order to forecast the electrical load. Thus, the robustness of standard FFNN increases while forecasting the electrical load. Moreover, this proposed model can efficiently overcome the existing limitations of FFNN to successfully predict the fast load changes and also the holiday loads. The model has been named as Constructive Approach for Effective Electrical Load Forecasting (CAEELF) on a short-term basis. In order to evaluate the performance of CAEELF, Spain's daily electrical load demand data have been used. Furthermore, extensive experimental results and comparisons have been shown to validate the acceptability of proposed CAEELF for electrical load prediction over other standard FFNN models.

Keywords — CAEELF, Constructive Technique, Electricity Load Forecasting, Partial Training, Short-term, Neural Network.

I. INTRODUCTION

Prediction of the power system operation and control is severely dependent on system demand. An overly cautious or risky activity is a direct result of a big forecast inaccuracy. Therefore, cost savings and system security are increased as load forecasting accuracy improves. It is a common notion in power systems that the power dispatch needs to be scheduled earlier for the very following day. This crucial task of day-ahead Short-Term Electricity Load Forecasting (STLF) is therefore essential. On the contrary, load forecasting analysis is a vital aspect when it comes to the reliable planning and operation of the power systems. In order to ensure and maintain power plants, exchange power, and schedule tasks for both power generation and distribution facilities, daily electricity consumption on a short-term basis is crucial [1]. Accurate load forecasting enables utilities to run at the lowest possible cost, potentially saving electric power companies a lot of money. According to [2], in 1984, a 1 percent increase in forecasting error would result in an increase in operation costs of 10 million pounds annually in UK. As a result, we can conclude that a small forecasting inaccuracy could result in a considerable profit loss in the competitive electricity markets.

Electricity load forecasting can be broken down into four categories based on the length of time [1], [3], [16]: (a) long-term load forecasting, that includes a time span longer than one year; (b) mid-term load forecasting, which covers a time span of six months to one year; (c) short-term load forecasting (STELF), where it encompasses a time span of one week to six months; and (d) very short-term load forecasting, with the time of shorter than one day. This article presents a short-term energy load forecasting which serves the next day's unit commitment and reliability analysis. Varied categories of forecasting pursue different types of goals. Using a variety of techniques, including Support Vector Machines (SVMs) [5], pattern recognition [6], fuzzy time series [8], fuzzy inference model [9], fuzzy neural networks [10], etc., several writers have attempted to anticipate the amount of electricity needed in these areas. Besides, recently developed hybrid techniques also include SVM and Genetic Algorithms (GA) [11], GA and Ant Colony Optimization (ACO) [12], and others. Moreover, authors are opting modern approaches with short-term load forecasting models that hybridizes several machine learning methods, such as, Support Vector Regression (SVR), Grey Catastrophe (GC) and Random Forest (RF) modelling [13].

There are several methods presented in the literature (such as [14]-[23]) that attempt to use Neural Networks (NNs) to tackle the STELF problem. It has been demonstrated that the employment of NN in STELF constantly beats any computational analysis based on human labour in terms of accuracy and user-friendly maintenance. The rationale is that, despite daily increases in load (i.e., output), NN has a good ability to map between input and output [4]. To tackle the ELF problem for various locations with a suitable computational cost, feed forward NN (FFNN) has been employed in particular in [18]-[24]. The use of FFNNs for static mapping relationships between inputs and outputs is observed, and this finally leads to good ELF outcomes. However, FFNNs have a limited ability to predict loads during vacations and rapid load variations and require substantial amounts of previous data [24]. Recently, several attempts have been made in [15]-[18] to address the drawbacks of FFNN by using Echo State NN, Radial Basis Function NN, Recurrent NN, and Nonlinear Autoregressive NN respectively. It should be observed that, when compared to the FFNN, the aforementioned NN models perform satisfactorily in estimating the power usage although they are

Submitted on July 3, 2022.

Published on August 2, 2022.

K. R. Islam, Dhaka University of Engineering and Technology, Bangladesh.

(e-mail: rafiqul@duet.ac.bd)

M. M. Kabir, Dhaka University of Engineering and Technology, Bangladesh.

(e-mail: munir@duet.ac.bd)

A. S. Surja, Sylhet Engineering College, Bangladesh.

(e-mail: amit@sec.ac.bd)

M. S. Iqbal, Sylhet Engineering College, Bangladesh.

(e-mail: iqbal@sec.ac.bd)

computationally costly. As a result, the hardware setups take huge resources, and their maintenance requires professionals.

A recent work suggests a development of model which utilizes the conventional back-propagation neural network trained by long-short term memory dataset [25]. Others have also compared specific methods for short-term load forecasting namely, artificial neural networks (ANNs) and multiple linear regression (MLR) [26]. The use of conventional neural network for electrical load forecasting is quite popular nowadays where the learning factor is derived by satisfying the convergence condition [27].

The constructive ELF technique, which is described in this study as an efficient ELF approach employing FFNN (CAEELF). The concept that underlies this model was first presented in our prior work [29]. This method varies from earlier ones in that CAEELF chooses the ideal NN design before the ELF begins training with constructive NNs. They typically employ a fixed NN architecture with random selection of the hidden neuron in the hidden layer during training before the ELF begins, in contrast to the prior methods (such as [19]-[24]). It is generally known that the generalization capabilities of NNs are impacted by the random selection of hidden neurons. The rationale is that any NN's design has a significant impact on how well it performs [30], [31]. In order to construct learning models utilizing NNs for ELF, a novel method is thus provided by automatically calculating the numbers of hidden neurons.

The rest of this essay is structured as follows: The proposed model CAEELF is thoroughly explained in Section II, including the computational complexity of each stage. The outcomes of our experimental research are presented in Section III, together with the experimental setup, outcomes, analysis, and comparisons to other ELF models. The paper is wrapped up with a summary and a few remarks in Section IV.

II. PROPOSED MODEL (CAEELF)

A training method associated with incremental training is used in CAEELF to determine the minimal number of hidden neurons. During a training process, hidden neurons (HNs) are concurrently and constructively added one by one. HN is eliminated if it does not enhance the accuracy of NN. Fig. 1 summarizes the key CAEELF stages, which are further detailed as follows:

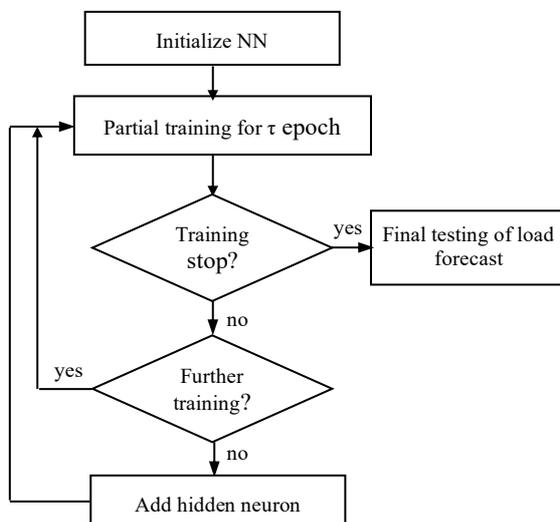


Fig. 1. Flowchart of CAEELF.

Step 1: Pick a feed-forward NN with a small size first. The total number of input variables and output load of the given ELF data samples, respectively, determine the size of the input layer and output layer, while one hidden neuron is used to initialize the hidden layer's size.

Step 2: Start the partial training of the NN using the back-propagation (BP) method on the training data sample up to epoch τ [30]. The user specifies the number of training epochs, τ . The NN is trained for a fixed number of epochs irrespective of whether it has converged under partial training, which was originally applied in association with an evolutionary algorithm [33].

Step 3: Verify the NN training termination criteria. If met, CAEELF produces the current NN architecture for the specified data sample. If not, proceed to the next action. On the validation samples, this approach calculates the average training error (E_a) [33]. In other words, the mean squared error is used here to represent the average training error (MSE). The error, E_a , is then calculated as:

$$E_a = \frac{1}{2p\tau} \sum_{c=1}^c (t_c(p) - o_c(p))^2 \quad (1)$$

where, $t_c(p)$ and $o_c(p)$, respectively, are the actual and predicted responses of the c -th output neuron for the validation pattern p . The total number of output neurons and validation patterns is represented by p and c , respectively.

Step 4: Evaluate the network training performance criteria. The network is given the responsibility of being trained further and moves on to Step 2 if the criterion is met. If not, move on to the next action.

Step 5: Add a hidden neuron to the network and return to Step 2 to repeat the partial training.

Step 6: After that, NN is tested using an unknown testing pattern. Get the current NN's forecasting of the electricity load.

The training error on samples of validation data is the only cost function that CAEELF employs. Finally, CAEELF makes an effort to use NN to create a better load forecaster. The following sections go into more detail regarding a few of the fundamental CAEELF steps.

A. Performance Criterion of NN Training

After the training epoch τ , it is presumed that the training program is running successfully if the average training error on validation samples decreases by a predefined amount ϵ . As a result, Step 2 is necessary after additional training. The decrease in training error can be characterized as follows:

$$E_a(t - \tau) - E_a(t) > \epsilon, t = \tau, 2\tau, 3\tau, \dots \quad (2)$$

where τ and t determine positive integers specified by the user.

B. Termination Criterion of NN Training

The training error would decrease as the training procedure of a NN proceeded since CAEELF introduces hidden neurons one by one. However, CAEELF aims to enhance the NN's capacity for generalization. This indicates that the training error may not be the best option for ending the NN's training procedure. The validation samples are a distinct set of data samples that are typically utilized for termination. Because

the validation data are not utilized to change the NN's weights, it is presumed that the validation error provides an impartial estimate.

The average training error on validation samples is used by CAEELF as its termination criterion in order to obtain strong generalization ability. After each strip of training, or τ epoch, it measures validation error. When the average training error, as measured at the conclusion of each T successive strip, increases by a predetermined amount (λ) for T successive times, training is terminated [35]. It can be inferred that such rises indicate the start of the final overfit rather than just the intermittent overfit because the average training error on validation samples increases not just once but T repeatedly. The termination requirement can be stated as follows:

$$E_a(\tau + i) - E_a(\tau) > \lambda, i = 1, 2, 3, \dots, T \quad (3)$$

Here τ and T are some positive integer numbers which are determined by the user. CAEELF model tests the termination criterion after every τ epoch of training and it necessarily stops the training when the condition explained by the Eq. (3) is satisfied. In this work, T is chosen to be 3.

C. Hidden Neuron Addition

According to Eq (4), CAEELF alters the current network architecture by introducing a hidden neuron. The existing network architecture is unable to extract all the information from the data samples, which is why expanding the network size is required. After that, the altered architecture is trained for a predetermined number of epochs τ .

$$E_a(t - \tau) - E_a(t) \leq \varepsilon, t = \tau, 2\tau, 3\tau \dots \dots \quad (4)$$

where ε is the predefined amount specified by the user.

D. Computational Complexity

An indicator of how much complexity is included in the computing process for a model is its computational complexity. Computational complexity theory, on the other hand, is a subfield of the theory of computation in theoretical computer science and mathematics that focuses on categorizing computational issues into groups based on their level of difficulty and connecting those groups. A computational problem is a task that can theoretically be completed by a computer, which is comparable to saying that the issue could be resolved by the mechanical application of mathematical operations.

Understanding the actual computing cost of an algorithm is made easier through the analysis of computational complexity. It feels motivated to determine the computational cost of our CAEELF because Kudo and Sklansky [38] demonstrated such an analysis using big-O notation. The computational complexity of CAEELF is demonstrated in the following sentences to demonstrate that adding additional approaches does not increase the computational complexity of training NNs.

(i) **Partial Training:** In this thesis, the standard back propagation (BP) algorithm [30] has been used for training. Every epoch of this BP algorithm takes a unique number of $O(W)$ computations for training one example. Here, the notation W represents the number of weights in the current

NN. Thus, training all of the examples in a training set having τ epochs require computations of

$$O(\tau \times p_\tau \times W) \quad (5)$$

computations, where p_τ denotes the number of examples in the training set.

(ii) **Termination Criterion:** CAEELF duly employs the termination criterion for stopping the training of the NN that utilizes both training and validation errors. As the training error is computed as a part of the training process, the termination criterion takes computations, $O(p_v \times W)$, Where p_v determines the number of examples in the validation set. Here $p_v < p_\tau$,

$$\therefore O(p_v \times W) < O(\tau \times p_\tau \times W) \quad (6)$$

(iii) **Further Training:** CAEELF model uses Eq. (4) to check if any (τ) further training for the added HN is necessary or not. The evaluation of Eq. (4) takes a constant computation of $O(1)$, as the error values that have been used in Eq. (4) are already evaluated during training.

(iv) **Adding a Hidden Neuron:** The computational cost due to the addition of a hidden neuron is $O(N_1 + C)$, which initializes its connection weights. Here N_1 represents the number of added input features and C denotes the number of neurons in the output layer. It can also be noted that, here $O(N_1 + C) < O(\tau \times p_\tau \times W)$. All of the afore-mentioned computation has been done for partial training purpose which consists of t epochs. Generally, CAEELF needs several, let M , such partial trainings. Therefore, the total computational cost of CAEELF in order to train a total of T epochs is,

$$T = \tau \times M = O(N^2 + p_t) + O(\tau \times M \times p_t \times W) \quad (7)$$

However, the first term, which is, $N^2 + p_t$ remains much less than the second one in practice. Thus, the total computational cost CAEELF becomes,

$$O(\tau \times M \times p_t \times W) \quad (8)$$

which is equal for training a fixed network architecture using BP algorithm [31]. So, encompassing different techniques in CAEELF does not result in increasing its computational cost.

III. EXPERIMENTAL STUDY

This section has consolidated a sample of daily load data to illustrate how well CAEELF can perform at estimating the electrical load in near future. The daily electricity demand in Spain, measured in megawatts/hour, was the source of the study's data [36], [37]. Predicted error was used to assess the CAEELF's performance. Predicted error specifically alludes to the error of an existing NN on test data samples. Moreover, this section is further divided into the following subsections for detailed explanation related to the performance evaluation of CAEELF.

A. Description of Data

The daily power demand in megawatts/hour for Spain from

January 1, 1993, to June 30, 1998, for a total of 2007 days, was the data used in the experimental study of this paper. Table 1 displays a portion of the data sheet to provide further information about the data. Using this insight, Fig. 2 also depicts the design of a feed-forward NN.

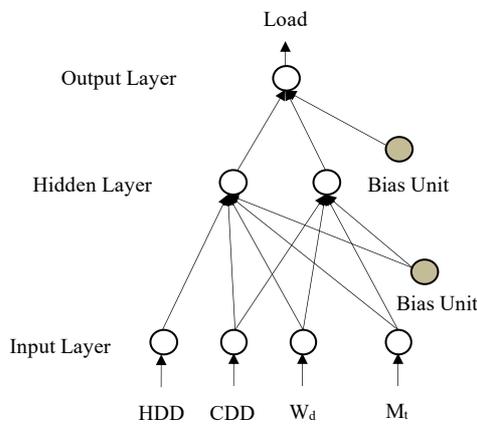


Fig. 2. Model of FFNN for forecasting the electrical load.

Here, Heating Degree Days and Cooling Degree Days, respectively, are referred to as HDD and CDD, the exogenous variables of degree days. The dummy variables W_d and M_t , on the other hand, stand in for all the weekly and monthly seasonalities, respectively. For more information about these input variables can be found in [23].

B. Experimental Setup

The 1826 days of data from January 1, 1993 to December 31, 1997 were utilized to train the NN model in CAEELF, whereas the 184 samples from July 1, 1998 to December 31, 1998 were used to validate the NN during training. These samples were specifically referred to as "in-sample" data since they were employed in the NN model for training. On the other hand, the data sample period of 120 days from January 1, 1999 to April 30, 1999, that were utilized to test the performance of forecasting by comparing model output (i.e., expected load) with actual load. Due to the fact that these data samples were not used in the NN training, they are referred to as "out-of-sample."

In each experiment, the hidden layer and output layer were each coupled to a single bias unit with a fixed input of 1. For NN training, 0.05–0.1 and 0.4–0.7 were selected as the learning rate and momentum term, respectively. An NN's initial connection weights were selected at random between -1.0 and 1.0. The activation function was a sigmoid function.

C. Experimental Results

Comparing actual values to model outputs for the same period allowed to evaluate how well CAEELF performed in anticipating out-of-sample data. Additionally, among the numerous predicting accuracy criteria, the mean absolute percentage error (MAPE) for the best relative accuracy metric was measured [12]. However, MAPE was calculated by:

$$MAPE = \frac{1}{N} \sum_{n=1}^N \left| \frac{P_n - \bar{P}_n}{P_n} \right| \times 100 \quad (9)$$

Here, P_n and \bar{P}_n represent the actual and predicted electricity load, respectively and N denotes the total number of available samples. From this context, Fig. 3, Fig. 5 and Fig. 6 depict the forecasting analysis for the duration of 120 days (including errors in percentage), 30 days, and 7 days, respectively.

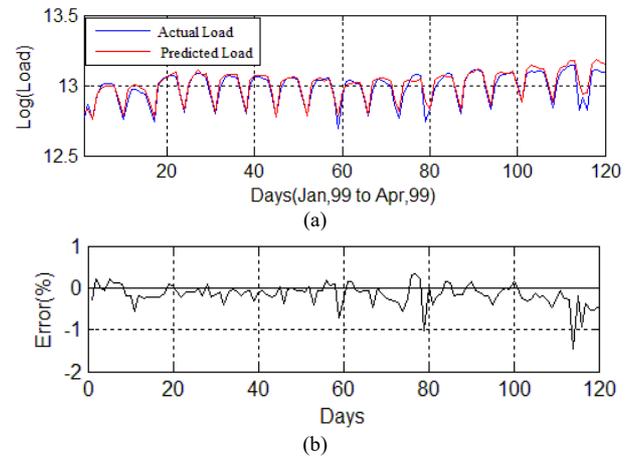


Fig. 3(a). Comparison between the actual load and the predicted load obtained from CAEELF used by constructive feed forward neural network and (b) corresponding errors in percentage.

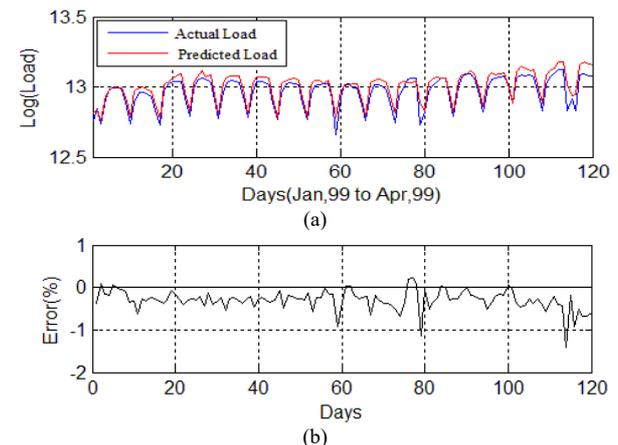


Fig. 4(a). Comparison between the actual load and the predicted load for 120 days obtained from standard model (SELF) used by feed forward neural network and (b) corresponding errors in percentage.

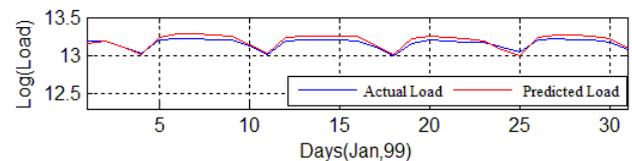


Fig. 5. Comparison between the actual load and the predicted load for 30 days obtained from CAEELF used by constructive feed forward neural network.

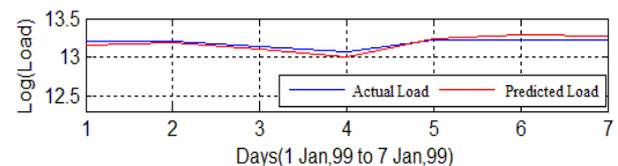


Fig. 6. Comparison between the actual load and the predicted load for 7 days obtained from CAEELF used by constructive feed forward neural network

In particular, Fig. 3(a) used CAEELF to compare the actual load to the expected load (i.e., predicting load). MAPE between these two loads was determined to be 0.2132. Fig. 5 and 6, where the forecasting of electrical load was done in

between 30 days and 7 days, respectively, provide additional analytical forecasting results of CAEELF. In a thorough analysis of this data, it was discovered that CAEELF forecasting of the electricity load is effective since the anticipated load curve closely overlaps the actual load curve.

Any type of model will have a hard time accurately anticipating the load demand over the holidays. The consumers spend their time in various locations and in varied ways, which is why the nonlinear fluctuation of the electricity usage. Fig. 7 shows the forecasting findings, particularly for the holidays, between the actual load curve and forecasted load curve from a 30-day experiment that CAEELF did to observe this issue. It has been observed that, with a few exceptions, load variations are satisfactory. Additionally, Table 3 demonstrates that there is relatively little variation between forecasted and real load.

D. Comparison with Other Works

Three electricity load forecasting models, including (i) standard electricity load forecasting (SELF), (ii) NNELF-1 [18], and (iii) NARx-2 [18], have been used to compare the predicting outcomes of CAEELF using daily electricity load data from Spain.

TABLE I: COMPARISONS WITH OTHER MODELS FOR THE ELF PROBLEM IN TERMS OF THE NEXT 120 DAYS

Models	Mean	SD	Max	Min
NNELF-1[16]	0.6426	0.0551	0.7686	0.4946
NARx-2 [16]	0.3852	0.1132	0.6455	0.2367
SELF	0.3151	0.0009	0.3168	0.3142
CAEELF	0.2132	0.0008	0.2136	0.2110

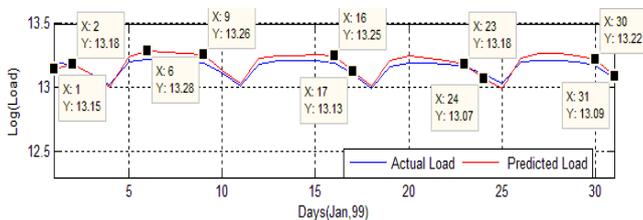


Fig. 7. Comparison between the actual load and the predicted load for Holidays of January 99 obtained from CAEELF used by constructive feed-forward neural network.

The first two models employed conventional feed-forward NN for predicting electrical load, considering a fixed number of hidden neurons in the NN's hidden layer and a fixed number of training iterations. The third one is a nonlinear autoregressive model, which consists of two components: first, the genuine available output is given as an input to train the NN; and second, the resulting network has a completely feed forward architecture, and BP method is used for training. Here, MAPE was the only parameter used for comparisons.

With the exception of the constructive approach and partial training, the entire CAEELF setup was utilized in SELF. 200 rounds of the NN training process and 5 hidden neurons were taken into account in this instance. The predicted results for SELF after running it ten times for comparisons were averaged. In contrast, the forecasting outcomes for the models NNELF-1 [18] and NARx-2 [18] were averaged across 20 separate runs and used 10 hidden neurons in the hidden layer. The comparison between these four models, including CAEELF, is shown in Table I. It has been discovered that, among the other models, the value of MAPE for CAEELF is the one that has been reduced the greatest. On

the other side, the robustness of CAEELF is indicated by the smallest value of SD.

IV. ANALYSIS

To assess the generalization capacity, a thorough investigation of CAEELF's experimental performance is conducted in several aspects. Three artificial time series data samples were used in this regard.

A. Mackey-Glass Time Series

A common benchmark for evaluating the generalization potential of various approaches is the Mackey-Glass series, which is based on the Mackey-Glass differential equation.

TABLE II: COMPARISONS BETWEEN THE ACTUAL AND PREDICTED LOAD FOR HOLIDAYS OF JANUARY 99

Holidays	Predicted	Actual	Differences
Jan ¹ st	13.1888	13.1482	0.0406
2 nd	13.187	13.1809	0.0061
6 th	13.2189	13.2833	-0.0644
9 th	13.1974	13.2569	-0.0595
15 th	13.2077	13.2582	-0.0505
16 th	13.1921	13.2542	-0.0621
23 rd	13.1645	13.1842	-0.0197
24 th	13.1078	13.0741	0.0337
30 th	13.1744	13.2209	-0.0465
31 st	13.0745	13.0928	-0.0183

The following time-delay ordinary differential equation produced the chaotic time series that makes up this series: The following differential equation is referred to as a Mackey-Glass time series:

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^{10}} - bx(t) \quad (10)$$

With discrete, evenly spaced time steps, it can be numerically solved, for instance, using the Runge-Kutta technique for the 4th order:

$$x(t + \Delta t) = \text{mackeyglass_rk4}(x(t), x(t - \tau), \Delta t, a, b) \quad (11)$$

where, the function *mackeyglass_rk4* numerically solves the Mackey-Glass delayed differential equation using the 4th order Runge-Kutta. The generated data sample are presented in the following Fig. 8, where the value of $\tau = 17$, $a = 0.2$, $b = 0.1$.

where, the Mackey-Glass delayed differential equation is numerically solved using the Runge-Kutta of fourth order by the function *mackeyglass rk4*. The obtained data sample is shown in the following Fig. 8, where the value of $\tau = 17$, $a = 0.2$ and $b = 0.1$.

B. Forecasting Results

Some samples that served as training and testing data samples were generated using the Mackey-glass time series model. The testing samples were applied to the trained NN model after CAEELF had been trained using the training data samples to determine the forecasting outcome. The results are shown in Fig. 9 and Table III and were obtained using samples of Mackey-glass time series data.

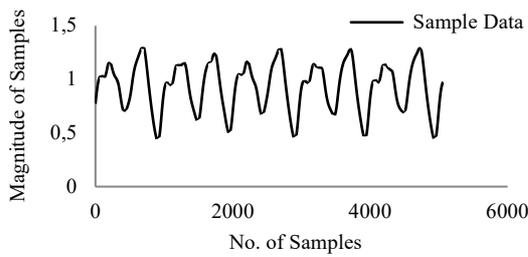


Fig. 8. Sample data of the Mackey-glass time series.

According to Fig. 9, it has been discovered that the two curves—the expected and actual data curves—are closely overlapped. The explanation is because the MAPE in this instance, which is 0.10823, and the value of SD, which is shown in Table IV, are both relatively low.

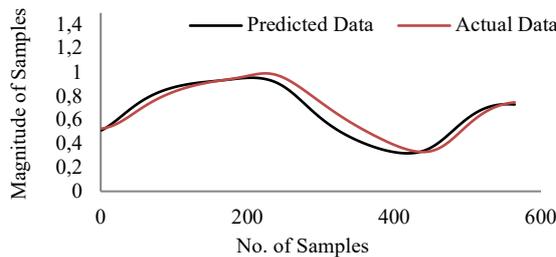


Fig. 9. Comparison between the actual data and the predicted data for Mackey-glass data samples obtained from CAEELF.

TABLE III: THE VALUE OF MAPE OF CAEELF ON MACKEY-GLASS DATA

	Mean	SD	Max ^m	Min ^m
MAPE	0.108234	0.014332	0.127487	0.087631

It may therefore be concluded that CAEELF model is reliable and effective at estimating the solution to the Mackey-Glass time series problem.

C. Lorenz Time Series

Edward Lorenz was the first to study the Lorenz system, which is a set of ordinary differential equations. For specific parameter values and initial conditions, it is noteworthy for having chaotic solutions. The Lorenz attractor, in particular, is a collection of chaotic Lorenz system solutions that, when plotted, look like a butterfly or figure eight. A streamlined mathematical model for atmospheric convection was created by Edward Lorenz in 1963. The Lorenz equations, a set of three ordinary differential equations, make up the model.

$$\begin{aligned}
 (i) \quad & \frac{dx}{dt} = \sigma(y - x) \\
 (ii) \quad & \frac{dy}{dt} = x(\rho - z) - y \\
 (iii) \quad & \frac{dz}{dt} = xy - \beta z
 \end{aligned} \tag{12}$$

Here, x , y and z form the system state, t denotes time and σ, ρ, β , represent the system parameters. The Lorenz system is nonlinear, three-dimensional, and deterministic from a technical perspective.

D. Forecasting Results

Here authors tried to create some data samples that served as training and testing data samples using the Lorenz time series model. The testing samples were applied to the trained NN model after CAEELF had been trained using the training

data samples to determine the forecasting outcome. The results are shown in Fig. 10 and Table V and were obtained utilizing Lorenz time series data samples after that.

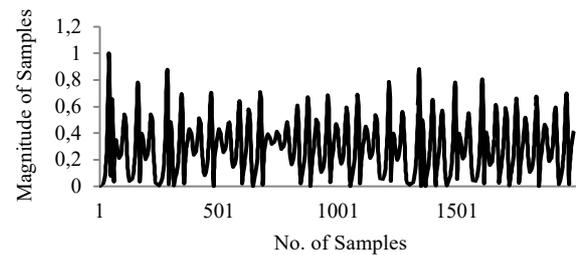


Fig. 10. Sample data of the Lorenz time series.

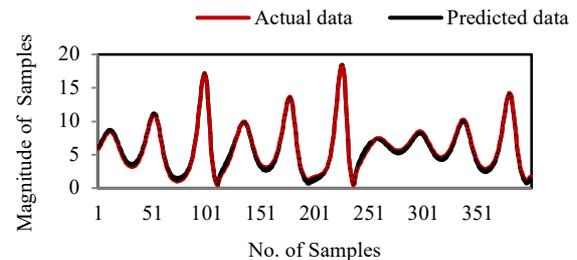


Fig. 11. Comparison between the actual data and the predicted data for Lorenz data obtained from CAEELF.

TABLE IV: THE VALUE OF MAPE OF CAEELF ON LORENTZ DATA

	Mean	SD	Max ^m	Min ^m
MAPE	0.916056	0.015022	0.93433	0.89491

It has been discovered that, in line with Fig. 11, the two curves — that is, the predicted and actual data curves are nearly overlapped with one another. The explanation is that the MAPE in this instance, which is 0.916056, and the value of SD, which is shown in Table IV, are both relatively low. So, we may conclude that our CAEELF model is reliable and effective at forecasting the outcome of the Lorenz time series problem.

V. CONCLUSION

The constructive technique in feed-forward NN training is used in this research to develop CAEELF, an efficient model for forecasting short-term electrical load. Therefore, based on the ELF issue domain, the FFNN automatically decides the size of the hidden layer during training.

On a short-term basis, experimental results shown in Fig. 3, 5, and 6 demonstrate that CAEELF operates effectively in ELF difficulties (e.g., 120 days, 30 days, and 7 days). The forecasting curve between expected load and actual load (Fig. 3(a)) is close to similar, which is the cause. Additionally, the majority of the error spots on the error curve in Fig. 4(b) are near to zero. Fig. 3 and 4 demonstrate how our model's outcome is significantly superior to that of SELF when compared to the results of the other models. In addition to the MAPE idea, Table I demonstrates that our model outperforms models like SELF, NNELF-1 [18], and NARx-2 [18].

Additionally, in CAEELF, the forecasting results in the final section of the out-of-sample reported in Fig. 3 are not as promising (a). The nonlinearity of the load fluctuations in the data samples may be the cause. It is left to future research to add more heuristic strategies to CAEELF in order to lessen these restrictions.

REFERENCES

- [1] Hippert HS, Pedreira CE, Souza RC. Neural Networks for Short-Term Load Forecasting: A Review and Evaluation, *IEEE Transactions on Power Systems*, 2001; 16(1): 44-55.
- [2] Bunn DW, Forecasting loads and prices in competitive power markets, *Proceedings of the IEEE*, 2000;88(2):163-169.
- [3] Sheikhanand M, Mohammadi N. Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection. *Neural Computation & Application*, 2011.
- [4] Hippert HS, Pedreira CE, Zareipour RCS. Neural Networks for Short-Term Load Forecasting: A Review and Evaluation. *IEEE Transactions on Power Systems*, 2001;16:44-55.
- [5] Pai PF, Hong WC. Support vector machines with simulated annealing algorithms in electricity load forecasting. *Energy Conversion and Management*, 2005;46:2669-2688.
- [6] Dehdashti AS, Tudor JR, Smith MC. Forecasting of hourly load by pattern recognition a deterministic approach. *IEEE Trans on Power Apparatus and Systems*, 1982;3290-3294.
- [7] Padhye J, Firoiu V, Towsley D. A stochastic model of TCP Reno congestion avoidance and control. Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 1999, 99-02.
- [8] Yang HT, Huang CM. A new short-term load forecasting approach using self organizing fuzzy ARMAX models. *IEEE Transaction on Power Systems*, 1998;13:217-25.
- [9] Mamlook R, Badran O, Abdulhadi E. A fuzzy inference model for short-term load forecasting. *Energy Policy*, 2009;37:1239-48.
- [10] Ying LC, Pan MC. Using adaptive network based fuzzy inference system to forecast regional electricity loads. *Energy Conversion and Management*, 2008;49:205-211.
- [11] Paia PF, Hongb WC. Forecasting regional electricity load based on recurrent support vector machines with genetic algorithms. *Electric Power System Research*, 2005;74:417-425.
- [12] Sheikhanand M, Mohammadi N. Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection, *Neural Computation and Application*, 2011. DOI: 10.1007/s00521-011-0599-1.
- [13] Fan GF, Yu M, Dong SQ, Yeh YH, Hong WC. Forecasting short-term electricity load using hybrid support vector regression with grey catastrophe and random forest modeling, *Utilities Policy*, December 2021;73(C):101-294.
- [14] Methaprayoon K, Lee WJ, Rasmiddatta S, Liao JR, Ross RJ. Multistage artificial neural network short-term load forecasting engine with front-end weather forecast. *IEEE Transactions on Industry Applications*, 2007;43:1410-1416.
- [15] Deihimi A, Showkati H. Application of echo state networks in short-term electric load forecasting, *Energy*, 2012;39:327-340.
- [16] Xia C, Wang J, McMenemy K, Short, medium- and long-term load forecasting model and virtual load forecaster based on radial basis function neural networks, *International Journal of Electricity Power & Energy Systems*, 2010;32:743-750.
- [17] Vermaak J, Botha EC, Recurrent neural networks for short-term load forecasting. *IEEE Transactions on Power Systems*, 1998;13:126-132.
- [18] Elias RS, Fang L, Wahab MIM. Electricity load forecasting based on weather variables and seasonalities: A neural network approach, *8th International conference on service systems and service management*, 2011;Canada.
- [19] Malki A, Karayiannis NB, Balasubramanian M. Short-term electric power load forecasting using feedforward neural network. *Expert systems*, 2004;21:157-167.
- [20] Arroyo DO, Skov MK, Huynh Q. Accurate electricity load forecasting with artificial neural networks, *International conference on computational intelligence for modeling, control, and automation-International conference on intelligence agents, web technologies, and internet commerce (CIMCA-IAWTIC)*, 2005.
- [21] Baklrlzls AG, Petrldls V, Klartzls SJ, Alexlads MC. A neural network short term load forecasting model for the greek power system. *IEEE Transactions on Power Systems*, 1996;11:858-863.
- [22] Srinivasan D, Liew AC, Chang CS. A neural network short-term load forecaster. *Electric Power Systems Research*, 1994;28:227-234.
- [23] Hsu CC, Chen CY. Regional load forecasting in Taiwan-applications of artificial neural networks. *Energy Conversion and Management*, 2003;44:1941-1949.
- [24] Zhang G, Patuwo BE, Hu MY. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 1998;14:35-62.
- [25] Bashir T, Haoyong C, Tahir MF, Liqiang Z. Short term electricity load forecasting using hybrid prophet-LSTM model optimized by BPNN, *Energy Reports*, November 2022;8:1678-1686.
- [26] Govender S, Folly K. Short-Term Load Forecasting using Artificial Neural Networks and Multiple Linear Regression. *IEEE PES/IAS PowerAfrica*, 2019.
- [27] Li H. Short Term Load Forecasting by Adaptive Neural Network, *IOP Conf. Series: Materials Science and Engineering*, vol.449, 2018.
- [28] Chen Y, Luh PB, Guan C, Zhao Y, Michel LD, Coolbeth MA et al. Short-term load forecasting: similar day-based wavelet neural networks. *IEEE Transactions on Power Systems*, 2010;25:322-330.
- [29] Islam KR, Kabir MM, Murase K. Short-term electricity load forecasting using constructive feed-forward neural network, *2nd International Conference on Machine Learning and Computer Science (IMLCS)*, pp. 40-44, August 25-26, Malaysia, 2013.
- [30] Islam MM, Murase K. A new algorithm to design compact two hidden-layer artificial neural networks. *Neural Network*, 2001;14(1001):1265-1278.
- [31] Kwok TY, Yeung DY. Constructive algorithms for structure learning in feed-forward neural networks for regression problems. *IEEE Transactions on Neural Networks*, 1997;8:630-645.
- [32] Rumelhart DE, McClelland J. *Parallel Distributed Processing*, MIT Press, 1986.
- [33] Yao X, Liu Y. A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, 1997;8:694-713.
- [34] Kabir MM, Islam MM, Murase K. A new wrapper feature selection approach using neural network. *Neurocomputing*, 2010;73:3273-3283.
- [35] Prechelt L. PROBEN1-A set of neural network benchmark problems and benchmarking rules, *Technical Report 21/94*, Faculty of Informatics, University of Karlsruhe, 1994.
- [36] Elias RS, Fang L, Wahab MIM. Electricity load forecasting based on weather variables and seasonalities: A neural network approach. *8th International conference on service systems and service management*, Canada, 2011.
- [37] Pardo A, Meneu V, Valor E. Temperature and seasonality influences on Spanish electricity load, *Energy Economics*, 2002;24:50-70.
- [38] Kudo M, Sklansky J. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 2000;33:25-41