Short-Term Load Forecasting in Power Systems Using Deep Learning

Sanjeeb Humagain, Nga Nguyen
Electrical Engineering and Computer Science, University of Wyoming, Laramie, WY 82071
{shumagai, nga.nguyen}@uwyo.edu

***Abstract*—Accurate load forecasting is crucial in ensuring efficient, reliable, and cost-effective planning and execution of the power grids. The uncertain characteristic of consumers as well as the increasing distributed generation units makes precise load forecasting an increasingly challenging task. Additionally, due to the complex nonlinear relationship of electric load with weather factors, advanced load forecasting models are necessary to accommodate the fast-changing nature of power grids. This research proposes an improved model for short-term load forecasting using deep learning. Important factors like electricity price, dew point, dry bulb, wet bulb, humidity, day of month, day of week, year, and time of day will be included in the model. First, a deep neural network is implemented with default hyperparameters. Then, hyperparameter optimization is applied, during which the model searches for the optimal combination of hyperparameters within the defined range. Finally, the model is run with the optimized hyperparameter values to predict the load. The proposed method is implemented on the Electrical Reliability Council of Texas system to validate its accuracy. The results show a significant improvement in system performance after hyperparameter optimization while considering diverse impact factors.**

***Index Terms*—Artificial neural network, deep learning, hyperparameter optimization, load forecasting.**

I. INTRODUCTION

Load forecasting (LF) is pivotal in power systems due to its impact on efficient energy management and planning [1]. There are three types of LF: long-term load forecasting (LTLF), mid-term load forecasting (MTLF) and short-term load forecasting (STLF). STLF covers LF from one hour to a few weeks. It is utilized in fields like real-time control, economic dispatch, and load management [2]. MTLF is generally applied for a time frame from 1 month to 5 years while LTLF is used for a time frame of 5 years to 20 years. LTLF is applied for planning the generation by capacity and type to satisfy future demand and cost efficiency [3]. Accuracy of LF has a considerable effect on system expenses [4]. The advantages of accurate LF include the reduction of operation as well as maintenance costs, enhancement of power system reliability, and improvement in planning and decision making for power dispatch. Overestimation of load causes an increase in the reserve capacity, which might not be necessary, thereby increasing operation and installation costs. Similarly, underestimation of electrical load causes a failure to supply the needed spinning and standby reserve, potentially leading to stability issues on voltage and frequency, and even collapse of the power system in extreme cases [5].

To cope with the energy transition to a carbon-neutral system, advanced techniques need to be developed to keep the grid stable with reliable energy supply [6]. However, due to this transition, additional challenges regarding voltage and frequency control, intermittency, and variability have emerged, thereby reducing the system stability and complicating the working mechanism of the power grid. Therefore, it is important to have a precise load forecast to help make the right decisions on the usage of electricity which in turn improves the system stability and reliability.

Deep learning (DL) has been getting a lot of attention recently in electrical LF. Due to the ability to handle complicated and non-linear equations, DL is being implemented widely in power system operation and planning [7]. Compared to artificial neural networks (ANNs), DL systems have more than one hidden layer making them capable of learning complex patterns and skills. However, as DL models are composed of more layers than ANN, more data is needed to train the DL model to acquire better results [8]. DL has been used in STLF to increase the accuracy of load forecasts [1].

According to [9], various factors affect the load demand like weather conditions, electricity price, day of month (DoM), day of week (DoW), and time of day (ToD). To improve accuracy of load forecast, all factors affecting the load demand needs to be considered while training the DL model.

DL application in LF has been developed intensively with diverse algorithms. The authors in [10] present the LF implementing long short-term memory (LSTM) algorithm in two architectures: standard LSTM and LSTM based sequence to sequence algorithm. LSTM algorithm overcomes the issue of diminishing gradient existing in recurrent neural network (RNN) by regulating the flow of gradient through the gates and the memory cells. RNN is generally trained using backpropagation or a real-time recurrent algorithm. Despite resolving the problem of vanishing gradient, the LSTM model was trained using only 120 hours of data, which may be insufficient to adequately capture the full range of input features. For instance, there will be less variation in temperature and other factors in such a short amount of time. Therefore, the model may not perform well when the weather factors are extreme [11]. STLF using deep neural network (DNN) is compared with five different ML algorithms: Multi-Layer Perceptron (MLP) and LSTM, Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM), to show its superiority in [12]. The proposed method implements LSTM architecture using a convolutional neural network (CNN) algorithm. Mean absolute percentage error (MAPE) for proposed DNN method is asserted to be better than the other five algorithms. However, this work does not consider hyperparameter optimization (HPO) and the performance of different ML algorithms with sub-optimal solutions can not be compared. The reason is that the results may not be reliable: an algorithm that appears superior under sub-optimal conditions might perform worse than other methods when hyperparameters are properly optimized. Therefore, hyperparameter optimization should be implemented before comparing the performance of different ML algorithms. In [13], bagged ANN and boosted ANN is used for STLF. This reduces the bias and variance thereby improving the forecasting accuracy. The method uses resampled subsets of the training data which is fed into several ANN models simultaneously. Then, the evaluated loads are averaged for the final predicted load. However, the time required for the execution would be significantly higher when running several ANN algorithms simultaneously thereby reducing the time efficiency. A modified mutual information (MMI) for pre-processing of the dataset and feature selection was used in LF in [14]. The module used to train and forecast relies on factored conditional restricted Boltzmann machine (FCRBM) model (DL model). The module used to optimize relies on a genetic wind-driven (GWDO) algorithm. The proposed hybrid model uses the advantages of every model thereby enhancing system accuracy. Even though the accuracy is better than the four benchmark models, the important factors like price is not considered. A new probabilistic LF approach that integrates quantile regression (QR) and hybrid model thereby enhancing the reliability of smart grids is used in [15]. A combined probabilistic forecasting model (CPFM) is also applied. The performance is better than a single forecast model. However, computational complexity increased significantly. Additionally, the data distribution information is not considered. Moreover, weather parameters, energy market parameters, and time are not taken into account.

The work in this paper overcomes the drawback of the previous works by the following contribution:

- Developing a DNN model for LF that implements HPO in a broad range of hyperparameters.
- Considering the electricity price, weather factors, and time factors that affect load demand in the proposed DNN model to improve the LF accuracy.

DL algorithm will be implemented in Python. Electricity price, humidity, wet bulb, dew point, dry bulb, DoM, DoW, month, year, and ToD are used as input features, and system load as the target value or the output of the model. Additionally, HPO is also implemented which significantly improves the performance of forecasting. The data from the Electrical Reliability Council of Texas (ERCOT) is used to validate the proposed approach [16].

This paper includes five sections. Section II details load forecasting using deep learning. Section III explains load forecasting methodology. Section IV presents the simulation and result. Finally, section V addresses the conclusion.

II. Load Forecasting Using Deep Learning

A DNN architecture is depicted in Fig. 1. The DNN has d input features and c output values. The neurons in each hidden layer can be varied as needed. In this research, the DL-assisted LF model has one neuron at the output layer as we are forecasting the system load only. The four key elements

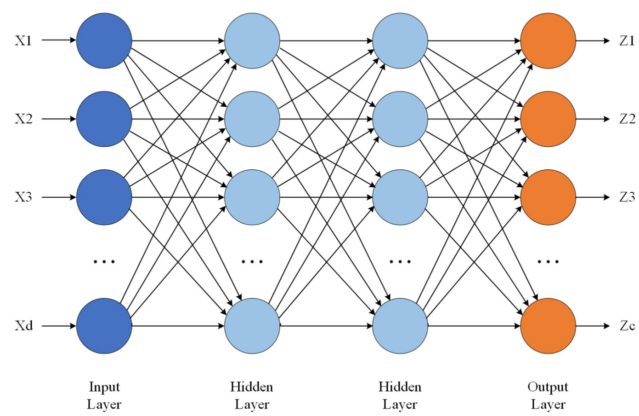

Fig. 1: A general architecture of DNN with two hidden layers.

of DL include:

A. Neural Networks

NNs are composed of interconnected nodes organized in layers. Every node implements a mathematical computation using the inputs and provides the outcome to next layer. The input layer of NN will take the input features. Number of neurons on input layer will be equal to the dimension of the input feature. The variables that affect the electrical load will be fed through the input layer. In the proposed model, the variables include electricity price, humidity, wet bulb, dew point, dry bulb, DoM, DoW, month, year, and ToD. The output layer, particularly in this work, will have a neuron which will provide a forecasted electrical load as output.

B. Activation Functions

Activation functions introduce non-linearities into the network, enabling it to acquire complicated structures in the data. They include several types like rectified linear unit (ReLU), hyperbolic tangent function (tanh), and sigmoid function. In this research, sigmoid activation function is selected due to its differentiability and bounded output, which helps with convergence during training. Additionally, the sigmoid function introduces non-linearity to the model, enabling the NN to learn complicated patterns present in various input factors such as electricity price, humidity, wet bulb, dew point, dry bulb, DoM, DoW, month, year, and ToD, along with output system load. This helps to enhance the correctness of the LF. The graph of the sigmoid function is depicted in Fig. 2 and the mathematical expression is shown below.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{1}$$

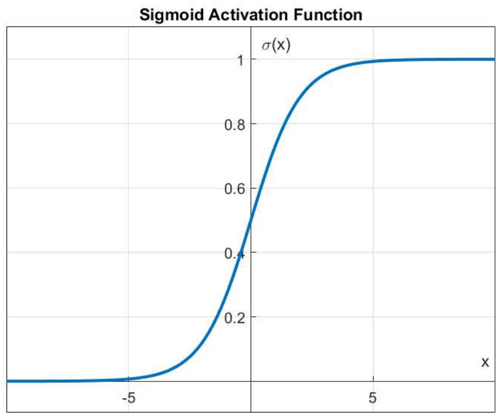

Fig. 2: Plot of sigmoid activation function.

C. Loss Functions

Loss functions calculate difference between evaluated system load and actual system load, leading the learning mechanism during training process. Loss functions or objective functions such as mean squared error (MSE), mean absolute error (MAE), and R^2 score are commonly used for regression problems. The R^2 score is a metric for measuring how well the evaluated values from a model match the actual values [17]. These functions are mathematically represented in the following expressions.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{3}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4}$$

where, n represents the count of total samples, y_i indicates actual value of target variable for i^{th} data, $\hat{y}_i$ indicates evaluated value of target variable for i^{th} data, and $\bar{y}$ denotes mean of actual values.

The loss function is used to measure how close the forecasted system load is to the actual system load. MSE, MAE, and R^2 score are evaluated before and after HPO and the values are presented. The R^2 score value is used for comparison and interpretation because of its simplicity.

D. Optimization Algorithms

Optimization algorithms adjust the parameters of the NN to minimize the loss function, typically using techniques like stochastic gradient descent (SGD) and Adam optimizer. Adam optimizer is used for optimization because it tends to converge faster than traditional SGD with momentum. Additionally, Adam adjusts the learning rate individually for every parameter using estimates of 1^{st} and 2^{nd} moments of gradients. This capability allows Adam to automatically adjust the learning rate during training, leading to more stable and efficient convergence, thereby enhancing the performance of the load forecast.

The next section will explain the load forecasting methodology used in this research.

III. Load Forecasting Methodology

Flowchart of proposed DL approach is represented in Fig. 3. Initially, ERCOT system data [16] was imported and exploratory data analysis (EDA) was performed. EDA helps to improve the accuracy of load forecast by helping to remove outliers. Then, data preprocessing was implemented to standardize the dataset. This is important to ensure all features such as electricity price, humidity, wet bulb, dew point, dry bulb, DoM, DoW, month, year, and ToD contribute equally to the LF model. Therefore, data preprocessing is pivotal to enhance the accuracy of LF. Then, dataset is separated into training and test sets. The dataset is then converted to a tensor. Furthermore, the training dataset is fed to DL model with default hyperparameters and performance evaluation was performed in test data. HPO is performed and model evaluation is performed after running the DL model. Finally, the result of the electrical load forecasted by the proposed DL method before and after HPO are compared. The process is elaborated further in the subsequent sections.

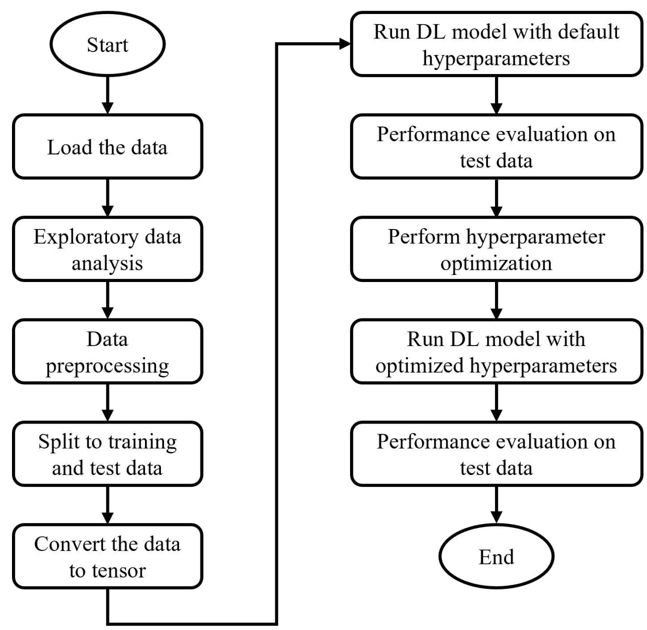

Fig. 3: Block diagram of the proposed DL approach.

A. Test and training data preprocessing

The data used for this research underwent exploratory data analysis (EDA), which involved handling the missing values and data visualization to check the outliers. It is then divided into train and test datasets in a proportion of $8/2$. Furthermore, standardization was applied to both training and test sets. Standardization helped to remove the mean and scale the data to unit variance. As a result, the standardized dataset will have 0 as mean and 1 as variance, thereby ensuring that all features contribute uniformly to the LF model.

B. Model architecture design for ANN

ANN with three hidden layers was used for this research. Ten neurons are used at the input layer as the dataset has ten features. The sigmoid activation function was applied in hidden layers. One output neuron present at output layer gives the forecasted output in each iteration. To augment the performance of the proposed method, the Bayesian hyperparameter optimization technique was implemented to optimize the epoch, learning rate, and number of neurons in three hidden layers.

C. Training ANN model

Batch size indicates the number of samples used for every gradient update. A batch size of 64 is chosen in this research for its balance between memory efficiency, training stability, and computation speed. 80% of total dataset is used while training the LF model. During training through backpropagation, the weights associated with each neuron and biases are updated after every iteration. After completion of the training process, the model runs on forward pass during which the weight and biases do not change. The learning curve in Fig. 4 depicts the MSE loss during training and validation process.

D. Model evaluation on the test dataset

The remaining 20% of the data split during preprocessing step is utilized as a test dataset for evaluating the model's performance. The Fig. 5 presents the scatter diagram of the actual system load values versus forecasted system load values before HPO. The Fig. 6 depicts the scatter plot of actual system load values versus forecasted system load values after HPO. In the scatter plot after HPO, the average distance from data points to the dotted red line has decreased compared to the plot before HPO. Therefore, HPO has increased the accuracy of the prediction. However, the plot depicts that not all the points are exactly on the red dotted line. This signifies that there are some errors in the prediction. If all the forecasted values are exactly equal to the target value, the R^2 score would be equal to 1.

The Fig. 7 shows the plot of 100 samples of actual system load values and predicted system load values before HPO and Fig. 8 depicts 100 samples of actual system load values and predicted system load values after HPO. There are 17530 test data values but only first 100 samples among those values were plotted for a better visualization.

IV. Simulation and Result

A. Dataset Description

The dataset contains eleven features such as electricity price, humidity, wet bulb, dew point, dry bulb, DoM, DoW, month, year, ToD, and system load. The first ten features are used as input features and the system load is used as the target variable which is to be predicted in this research. These features are utilized as inputs to the proposed model for forecasting system load. The dataset contains 87648 samples of data and the target system load from 2006 to 2010 [16]. The statistics of the dataset is listed in Tables I and II.

TABLE I: Statistics of Environmental Variables

	DryBulb	DewPnt	WetBulb	Humidity	ElecPrice
Count	87648	87648	87648	87648	87648
Mean	18.26	11.92	14.88	68.9	42.4
Std	4.89	5.47	4.29	16.86	215.64
Min	3.7	-8.4	2.5	7	-264.31
25%	14.7	8	11.6	58	21.8
50%	18.5	12.45	15.1	70	25.81
75%	21.8	16.35	18.4	82.5	36.94
Max	43.8	24.2	26.3	100	10000

TABLE II: Statistics of Time and Load Variables

	Day	DoM	DoW	Month	Year	ToD	Load
Count	87648	87648	87648	87648	87648	87648	87648
Mean	913.52	15.73	3	6.52	2008	705	8894
Std	527.12	8.8	2	3.45	1.41	415.6	1409.05
Min	1	1	0	1	2006	0	5498.36
25%	457	8	1	4	2007	352.5	7879.67
50%	914	16	3	7	2008	705	8992.58
75%	1370	23	5	10	2009	1057.5	9832.85
Max	1827	31	6	12	2011	1410	14274.15

B. Experimental Setup

Python is used for programming due to the availability of crucial libraries, for instance, numpy, pandas, sklearn, and matplotlib. The dataset was imported from a csv file and divided into train and test datasets. DNN with three hidden layers is used in this research to reduce the model complexity and time of execution. The model performance was calculated utilizing three metrics MSE, MAE and, R^2 score.

In the program, data is separated into the training and test data in proportion of $8/2$ which works as an outer loop for model evaluation. Hyperparameter tuning is performed on training folds, whereas the validation fold is used to calculate the accuracy of the DNN model with the selected hyperparameters. A total 100 number of iterations were performed for Bayesian Optimization. We have defined a function called DeepNeuralNetwork, which takes the learning rate, hidden layer one size, hidden layer two size, hidden layer three size, and epochs as the arguments. We have provided 0.001, 128, 64, 32, and 20 as default values before implementing the Bayesian optimization process to check the model's performance before HPO.

C. Results

The MSE, MAE and, R^2 score were evaluated before and after implementing the HPO to access the model performance. The MSE, MAE and, R^2 score before and after using HPO are listed in Table III. There has been a considerable increase in the model's performance after HPO which is indicated by the performance metrics.

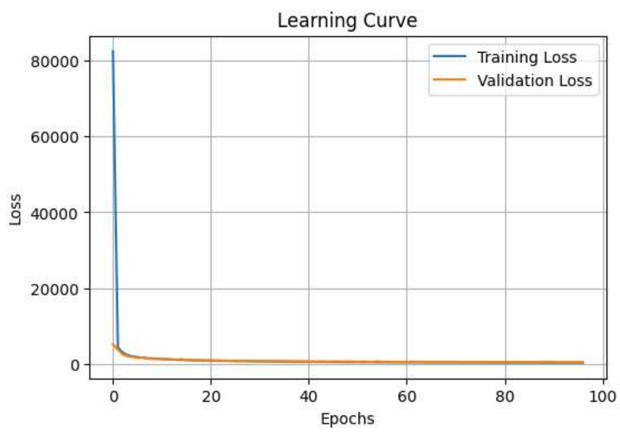

Fig. 4: Learning Curve.

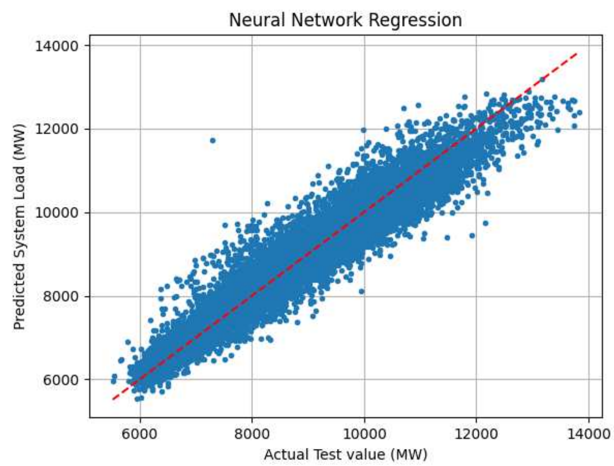

Fig. 5: Scatter plot of real versus forecasted system load before HPO.

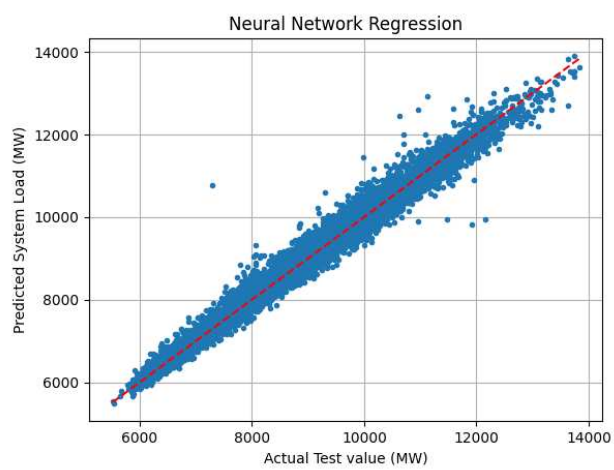

Fig. 6: Scatter plot of real versus forecasted system load after HPO.

Table IV contains the hyperparameters range defined and tuned values. Randomly selecting hyperparameter values and running the model repeatedly is not scientific and may not be feasible for a large range of hyperparameters. The R^2 score improved from 0.9198 for default hyperparameter to 0.9838 for the optimized value of hyperparameters. This indicates that HPO is crucial and can enhance the model performance significantly. However, using the HPO algorithm can lead to longer execution time and may fail to identify the optimal solution if the defined range for hyperparameters is excessively broad.

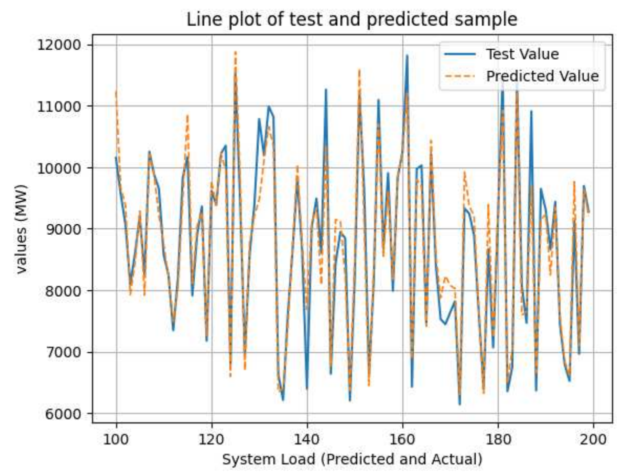

Fig. 7: Line plot of 100 evaluated and 100 actual samples before HPO.

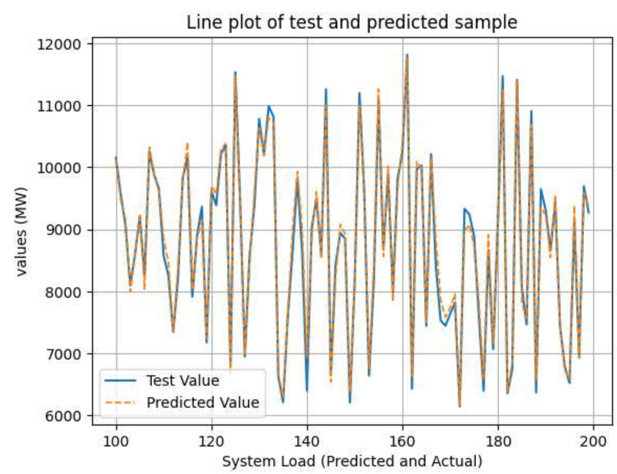

Fig. 8: Line plot of 100 evaluated and 100 actual samples after HPO.

TABLE III: Performance metrics values before and after HPO.

Performance metrics	before HPO	after HPO
MSE	161161.0005	32546.6836
MAE	298.0643	132.9852
R^2 Score	0.9198	0.9838

The improvement of prediction accuracy of the LF can be observed from Fig. 7 and Fig. 8. The enhancement in system load prediction accuracy in Fig. 8 is due to the implementation of HPO. By incorporating all significant factors such as dry

TABLE IV: Hyperparameter values.

Hyperparameters	Range defined	Tuned value
Hidden layer1 size	(8, 256)	165
Hidden layer2 size	(8, 128)	99
Hidden layer3 size	(8, 128)	55
Number of epochs	(10, 110)	97
Learning rate	(1e-5, 1e-1)	0.0088

bulb, dew point, wet bulb, humidity, DoM, DoW, month, year, and ToD during model training, we have enhanced the practicality and reliability of the load forecast model. The major factors that can affect electricity demand are considered, enabling the proposed DNN method to correctly learn the relationship of input features with the system load. This enhancement contributes to the overall reliability of the LF. Additionally, the implementation of HPO has ensured that the results are optimized across the defined range of hyperparameters.

V. Conclusion

This paper implemented a DL approach for STLF using weather, time, and cost factors along with the application of HPO algorithm. HPO technique is implemented to acquire the optimized hyperparameters to improve the effectiveness of the LF model. Consideration of all the major factors affecting the system load makes this method more reliable. The work has proved that using weather factors, electricity price, and time factors along with implementing HPO makes forecasting more realistic and pragmatic for real-world electrical LF applications. The result indicates that the proposed DNN-based STLF model has a robust generalizing capability and it has an excellent forecasting performance. This method helps the system operator for efficient planning as well as execution of the power grid because of the reliable and accurate LF. Additionally, the operation and maintenance cost of the system can also be reduced significantly considering the advantages of the proposed method.

Acknowledgment

This material is based upon work supported by the National Science Foundation under an NSF CAREER Award Number 2339456.

References

[1] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, Oct. 2015.

[2] S. Khatoon, A. K. Singh, *et al.*, "Effects of various factors on electric load forecasting: An overview," in *2014 6th IEEE Power India International Conference (PIICON)*, IEEE, Dec. 2014, pp. 1–5.

[3] M. T. Haque and A. M. Kashtiban, "Application of neural networks in power systems; a review," *International Journal of Energy and Power Engineering*, vol. 1, no. 6, pp. 897–901, Jun. 2007.

[4] C. A. Moturi and F. K. Kioko, "Use of artificial neural networks for short-term electricity load forecasting of kenya national grid power system," *International Journal of Computer Applications*, vol. 63, no. 2, Jan. 2013.

[5] S. Singh, S. Hussain, and M. A. Bazaz, "Short term load forecasting using artificial neural network," in *2017 Fourth International Conference on Image Information Processing (ICIIP)*, IEEE, Dec. 2017, pp. 1–5.

[6] S. Stock, D. Babazadeh, and C. Becker, "Applications of artificial intelligence in distribution power system operation," *IEEE access*, vol. 9, pp. 150098–150119, Nov. 2021.

[7] S. A. Kalogirou, "Applications of artificial neural-networks for energy systems," *Applied energy*, vol. 67, no. 1-2, pp. 17–35, Sep. 2000.

[8] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, "A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids," *Renewable and Sustainable Energy Reviews*, vol. 144, pp. 110992, Jul. 2021.

[9] M. U. Fahad and N. Arbab, "Factor affecting short term load forecasting," *Journal of Clean Energy Technologies*, vol. 2, no. 4, pp. 305–309, Oct. 2014.

[10] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using deep neural networks," in *IECON 2016-42nd annual conference of the IEEE industrial electronics society*, IEEE, Oct. 2016, pp. 7046–7051.

[11] C.-L. Hor, S. J. Watson, and S. Majithia, "Analyzing the impact of weather variables on monthly electricity demand," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 2078–2085, Oct. 2005.

[12] P.-H. Kuo and C.-J. Huang, "A high precision artificial neural networks model for short-term energy load forecasting," *Energies*, vol. 11, no. 1, pp. 213, Jan. 2018.

[13] A. S. Khwaja, A. Anpalagan, M. Naeem, and B. Venkatesh, "Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting," *Electric Power Systems Research*, vol. 179, pp. 106080, Feb. 2020.

[14] G. Hafeez, K. S. Alimgeer, and I. Khan, "Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid," *Applied Energy*, vol. 269, pp. 114915, Jul. 2020.

[15] D. Zhang, S. Wang, Y. Liang, and Z. Du, "A novel combined model for probabilistic load forecasting based on deep learning and improved optimizer," *Energy*, vol. 264, pp. 126172, Feb. 2023.

[16] https : / / s3 - eu - west - 1 . amazonaws . com / valohai - examples/load-forecasting/ercot-dataset.csv, Accessed: 2024-06-09, 2024.

[17] M. Khashei and M. Bijari, "A novel hybridization of artificial neural networks and arima models for time series forecasting," *Applied soft computing*, vol. 11, no. 2, pp. 2664–2675, Mar. 2011.