Contemporary Mathematics
https://ojs.wiserpub.com/index.php/CM/

Research Article

An Accurate Load Forecasting and Scheduling of Charging for Electric Vehicles Using Deep Learning Techniques

Rupesh Rayalu Onteru*, V Sandeep, VB Murali Krishna

Department of Electrical Engineering, National Institute of Technology Andhra Pradesh, Tadepalligudem, 534 101, India
E-mail: rupesh.onteru.ee@gmail.com

Received: 5 November 2024; **Revised:** 28 November 2024; **Accepted:** 6 December 2024

Abstract: The increasing adoption of Electric Vehicles (EVs) has increased the need for an efficient management of charging infrastructure. Load forecasting is one of the most prominent challenges in EV operation and is an important factor for achieving effective scheduling. Various techniques have been deployed for forecasting the load demand in EVs. However, in comparison to traditional loads, the charging and scheduling of EVs is different because of its dynamic fluctuations and periodic variation. These issues affect the performance of conventional load forecasting techniques. The emergence of Deep Learning (DL) models provides a potential solution for addressing the drawback of conventional forecasting methods. Because of its excellent learning ability and capacity to handle large-scale datasets, DL models are extensively used to perform forecasting tasks. This research analyzes the application of different DL models for forecasting the load demand and scheduling in EVs. In this work, various models such as Convolutional Neural Networks (CNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) and hybrid models are deployed for accurately predicting the load. In addition, a dynamic pricing algorithm is implemented for achieving effective scheduling. A Graphical User Interface (GUI) is designed for verifying charging scheduling and management. Simulation results of different charging stages validate the effectiveness of the proposed framework.

Keywords: electric vehicle, load forecasting and scheduling, deep learning techniques

MSC: 68T30, 68T20, 68T27

1. Introduction

Renewable energy source driven generation stations play a significant role in the modern power sector for adopting many advantages [1–3]. On the other hand, Electric Vehicles (EVs) have transformed the transportation sector in terms of providing a clean and pollution-free environment [4]. The EVs have reduced the dependence on conventional fuel-based vehicles which is the main source of CO_2 emissions [5]. EVs offer several advantages such as lower lifetime cost, improved safety, and zero vehicle emission. Despite the advantages of EVs, the challenges related to the charging and scheduling process restrict the adoption of EVs [6]. Through optimal scheduling EVs can be charged when the peak demand is low and thereby help the utility system to store the excess energy [7]. Accurate load forecasting plays an important role in managing the charging and scheduling process [8]. But it is difficult to forecast the load demand considering the highly stochastic nature of the charging mechanism [9]. Traditional model-based load forecasting methods incorporate

DOI: https://doi.org/10.37256/cm.6420256032

mathematical models and statistical tools [10]. These models struggle to capture complex and intricate patterns of EV charging which are highly nonlinear [11]. In addition, the charging load is characterized by various complex factors such as volatile weather conditions, and uncertain demand, which might not be adequately modeled using traditional approaches. These factors can have a significant impact on the accuracy of traditional load forecasting techniques [12]. The drawbacks of traditional mathematical and statistical techniques can be addressed using Artificial Intelligence (AI) based Machine Learning (ML) and DL models. Advanced ML and DL models can automatically learn from past historical data and predict the target. Various ML and DL models such as Artificial Neural Networks (ANN) [13], Support Vector Regression (SVR) [14], Gradient Boosting algorithms [15], CNN [16], and Recurrent Neural Networks (RNN) [17] etc. are predominantly used in the forecasting tasks. Compared to ML algorithms, DL models, especially RNN models such as LSTM and GRU have more potential in achieving accurate forecasting since they can process nonlinear time-series problems effectively. Considering the advantages of DL models, this research presents a comprehensive analysis of different DL algorithms for load forecasting and scheduling in EVs.

The highlights of the proposed work are:

1. In this work, the LSTM, GRU, Bidirectional LSTM (BiLSTM), Bidirectional GRU (BiGRU) and hybrid models are analyzed for forecasting the load and charging scheduling in EVs.

2. A dynamic pricing algorithm is developed in this study managing the charging scheduling. The algorithm helps the users and companies to adjust the prices based on the load demand.

3. A GUI is developed using the tkinter library for displaying visual information about the availability of the charging station, and time slots for charging EVs.

The organization of this work is structured as follows: various existing methods on load forecasting and scheduling using different DL algorithms are discussed in Section 2. Section 3 discusses the application of various forecasting techniques for load demand estimation. Section 4 discusses the application of a dynamic pricing algorithm for scheduling. Section 5 discusses the development of GUI for demonstrating the scheduling of EV's based on dynamic pricing and Section 6 concludes the paper with prominent research observations and future directions.

2. Related works

Accurate load forecasting and charging scheduling in EVs is one of the most extensively researched topics in recent times. Considering the popularity of EVs, the importance of load forecasting and optimal charging scheduling is significantly increasing. The work presented in [18] employed a LSTM model for predicting the charging demand in EVs. A novel Arithmetic Optimization Algorithm (AOA) is implemented for modeling to overcome the vanishing and exploding gradient issues. The model is tested for the data collected from the dataset of Georgia Tech, Atlanta, USA. Results show that the LSTM model with AOA achieved an enhanced accuracy of 97.14%. A comparative analysis of different DL models for load forecasting in EVs is presented in [19]. The review states that the accuracy of load forecasting is directly related to the optimization of the DL models. The study discusses the application of various DL models such as LSTM, GRU, hybrid CNN-LSTM, hybrid CNN-GRU, multivariate LSTM and GRU in terms of forecasting the charging load. In [20], the authors have proposed a novel ensemble learning approach for forecasting the load. Three different algorithms such as ANN, RNN, and LSTM are integrated to design the ensemble model. In addition, a linear regression model is employed for learning the weight of the network parameters. The efficacy of the ensemble learning model is tested on a real-world dataset and results show that the ensemble model enhanced the forecasting accuracy by 5.79%, 2.48%, and 0.83% compared to individual models such as ANN, RNN and linear regression. A hybrid forecasting approach is presented in [21] for predicting EV load. A Gradient Boosting Decision Trees (GBDT) is combined with the Time Convolutional Network (TCN) for assisting the EV users to make timely decisions. The GBDT algorithm classifies the rate of battery discharge in EVs with a phenomenal accuracy of 92%. Besides, the TCN model captures the temporal attributes for predicting the loads. The GBDT-TCN model outperforms other state of art models in terms of load forecasting in real time with a minimum error. A short-forecasting model for EV is presented in [22] which uses three learning algorithms such as Autoregressive (AR) models [22], SVR [23], and LSTM algorithms for predicting short term loads in EVs [24]. In comparison to SVR and AR models, the LSTM achieves an outstanding performance in terms of reducing the Mean Absolute Error (MAE) to 4 kW and RMSE of 5.9 kW. The efficacy of DL models is validated in several research works and results show that RNN models such as LSTM [25] exhibit excellent forecasting performance with reduced error in comparison to conventional ANN and CNN models [26, 27]. Despite the availability of different works that have deployed LSTM models [28, 29] for load forecasting [30], there is a great scope of research in terms of exploring different models [31] to achieve better forecasting [32] and charge scheduling performance [33]. This research intends to contribute to this research gap by investigating the performance of different DL models including advanced LSTM and GRU models along with hybrid models for EV charging load forecasting.

3. Proposed research methodology

The preliminary objective of this research is to analyze the performance of different load forecasting techniques based on deep learning for predicting EV charging and scheduling. The steps involved in the proposed approach are briefed in the below subsections.

3.1 *Data collection*

In this work, datasets are collected from PJM, UK Household of London and Panama for the experimental analysis on EV charging. From all three datasets the energy consumption data was collected for analysis. The subsequent three-subsections brief about these three data sets.

3.1.1 *PJM hourly energy consumption data*

PJM Interconnection LLC (PJM) is a Regional Transmission Organization (RTO) in the United States. PJM mainly coordinates the distribution of electricity in all or parts of 13 states in the Eastern United States namely Delaware, Illinois, Indiana, Kentucky, Maryland, Michigan, New Jersey, North Carolina, Ohio, Pennsylvania, Tennessee, Virginia, West Virginia, and the District of Columbia. The hourly energy consumption data provided by PJM is accessed. The data fundamentally provides information about the energy demand across different regions within the PJM Interconnection. The hourly energy consumption data collected from PJM's website are measured in Megawatts (MW).

3.1.2 *Panama case study*

The dataset with the details of panama case study is used to train the forecasting models and the results can be compared with the official forecasting data obtained from the weekly pre-dispatched reports. The original data sources provide the details about the post-dispatch electricity load daily and weekly pre-dispatch electricity load forecast data on a weekly basis. Both daily and weekly data are provided in separate files and the data are collected with hourly granularity. The data collected during holidays and school periods are sparse along with websites and PDF files. On the other hand, the weather data is collected only on daily NetCDF files. The published datasets are preprocessed by combining all data sources based on the date-time index. The data records are presented in the .CSV file with all required variables in the form of a single continuous dataset. The file also contains the data from the weekly pre-dispatch reports about the load forecast. In addition, the dataset also contains two excel files which incorporate the data related to suggested regressors and 14 training/testing datasets pairs.

3.1.3 *A dataset from household EV charging data in London*

In this work, a dataset is considered, which contains one-year measurements of demand with an average of 11 kWh/day, electric vehicle charging of 3 kW, and PV generation 3.3 kWp for a household in London city. The detail of the dataset is obtained from the reference [24]. The dataset contains the data starting from 1^{st} January to 31^{st} December with a time gap of 30 minutes, i.e, 17,520 points/year. All energy measurement data are in kW wherein the first column denotes the demand (kW), second column denotes data related to PV (kW) and third column denotes EV (kW).

3.2 *Data preprocessing*

Preprocessing is the preliminary stage of the forecasting process wherein the raw data obtained from the dataset is cleaned and prepared for enhancing the accuracy of forecasting. In general, the data consists of redundancies, and uncertainties in the form of missing data, null values, presence of artifacts etc. which affect the performance of the deep learning models. Preprocessing is performed using different steps such as identifying missing data, data normalization and data splitting.

3.2.1 *Handling of missing data*

The presence of missing data affects the ability of DL models to accurately capture patterns and generate accurate predictions. Considering the impact, it is crucial to identify and eliminate them from the data. The outliers and missing data identified from the dataset are replaced with interpolated values as shown in Equation (1):

$$y_{dt} = \frac{y_{dt-1} + y_{dt+1}}{2} \tag{1}$$

In (1), the interpolation equation calculates the value of y_{dt-1} at the position y_{dt} along the straight line.

3.2.2 *Feature scaling*

The feature scaling is accomplished to normalize (or standardize) a range of independent variables (features) wherein the features of different ranges are normalized into a specific range. By scaling the features, it is ensured that all features are used to learn the process and enhance the performance efficiency of the DL algorithms. Feature scaling is performed using two main methods namely standardization and normalization. Standardization transforms the features in such a way that they have a mean of value, 0 and standard deviation of value, 1. The standardization (X'_{stand}) is given by (2).

$$X'_{stand} = \frac{x - \text{mean}(x)}{a} \tag{2}$$

In (2), X'_{stand} is the new value, 'a' is the standard deviation and x is the actual or the original value.

On the other hand, normalization also known as min-max scaling scales the features to a specific range, usually in the range of [0, 1]. Mathematically, normalization (X'_{norm}) is performed using Equation (3).

$$X'_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{3}$$

In (3), X'_{norm} is the new value and 'x' is the actual value.

The normalized data provides better results than the actual data since it has a common range.

3.2.3 *Data splitting and time interval processing*

The data is split into training and testing data wherein the models are trained using the training data for forecasting and the performance is validated using the testing data. An appropriate splitting ratio is selected such as 80 : 20 or 70 : 30 for clustering the training and testing data. Before splitting, the data is discretized to obtain a timestamp. This is achieved using the Pandas tool which belongs to the python library. The standard data models and large number of data libraries help in large scale data sets. In this research, Pandas are used for aggregating the data related to charging based on the timestamps and for splitting the data into 24-time instances for each day. In this way, a 60-minute interval is obtained for the analysis. The pseudocode for the discretization of continuous data is given below:

```
Pseudocode for discretization:
Initialization
Define the frame 'kWhDelivered'
Define start date and time
Update the data and time after each interval
Define the index for connection time [0]
Start date, time = = Connection time
Update the 'kWhDelivered' data [i]
End the Process.
```

3.3 *Deep learning architecture for load forecasting*

As observed from existing works, RNN models are the most effective and potential tools for solving time series problems and handling nonlinear data. In addition, RNN models such as LSTM and GRU can process sequential data successfully compared to other DL models. The models used for EV charging load forecasting are discussed in the below sub sections.

3.3.1 *LSTM*

The LSTM is characterized by its superior memory wherein it can remember the long-term sequences. Because of its excellent memory, LSTM is considered in this research for load forecasting. The architecture of the LSTM model consists of memory units composed of three gates such as the input gate, the output gate and the forget gate. These units will enable the LSTM to either remember or forget the data at any time by controlling the flow of information through that unit. This will enable the LSTM to track only relevant data. This mechanism will overcome the problem of data disappearing gradient and helps the model to remember the data stored on a longer run.

The preliminary step in the LSTM architecture is to control the flow of information and decide the amount of information to eliminate from the cell state. This is achieved using the forget gate which consists of a sigmoid activation function and the same is defined in the given Equation (4).

$$f_t = \sigma(x_t U^f + h_{t-1} W^f) \tag{4}$$

Where f_t is the forget gate, h_t is the hidden state, h_{t-1} is the previous hidden state, W^f is the input word sequence, x_t represents the bottleneck features and σ is the sigmoid function. In the next state, the LSTM cell decides the amount of new information to store in the cell state. This is achieved using two parts wherein the first sigmoid layer called as input gate decides the candidate values of the new information to be stored and in the second part, a tanh layer creates a vector of new candidate values which can be added to the cell state, as shown in Equation (5).

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \tag{5}$$

In Equation (5), the i_t is the input gate.

In the next stage, the new candidate values and old values are combined to update the state wherein the old cell stage C_{t-1} is updated into the new cell state C_t as shown in below equations.

$$\hat{C}_t = \tanh(x_t U^g + h_{t-1} W^g) \tag{6}$$

$$C_t = \sigma(f_t C_{t-1} + i_t \hat{C}_t) \tag{7}$$

As shown in Equation (7), the old cell state is multiplied with f_t to control the information that has to be forgotten. Further, the new candidate values denoted by i_t are multiplied with the new cell state to determine the update state value for each cell. Finally, the output will be generated based on the cell state. The sigmoid layer defines the part of information that is generated as output. For this, the cell state is run through the tanh function (to push the values to be between -1 and 1) and multiply it by the output of the sigmoid gate. In this way, the relevant output is generated, which is shown in Equations (8) and (9).

$$o_t = \sigma(x_t U^o + h_{t-1} W^o) \tag{8}$$

$$h_t = \tanh(C_t) * o_t \tag{9}$$

In (9), o_t is the output gate and h_t is the hidden state.

3.3.2 *GRU*

GRU is a similar version of the LSTM model except for the number of gates. The GRU consists of two gates including a reset gate that controls the addition of new input and calibrates the previous memory state with the newly added input, which is shown in Equations (10) and (11).

$$z = \sigma(W_z * x_t + U_z * h_{t-1} + b_z) \tag{10}$$

$$r = \sigma(W_r * x_t + U_r * h_{t-1} + b_r) \tag{11}$$

An update gate is incorporated to control the storing of previous memory state and the hidden state is represented in Equation (12) and (13).

$$\hat{h} = \tanh(W_h * x_t + r * U_h * h_{t-1} + b_z) \tag{12}$$

$$h = z * h_{t-1} + (1 - z) * h \tag{13}$$

3.3.3 *Bidirectional LSTM*

The Bidirectional LSTM (Bi-LSTM) is a cutting-edge version compared to the conventional LSTMs. It helps to improve the performance of a system architecture in view of issues of sequence classifications, where the input of all steps is accessible. Further, the Bi-LSTM can train two-LSTMs on the sequence of input instead of one-LSTM.

Unlike the conventional LSTM model, the Bi-LSTM model consists of an additional layer known as backward LSTM, which reverses the flow of information. The hidden layer is responsible for synthesizing the information processed through the forward and backward layer. The BiLSTM has a training phase and the prediction phase. Based on the data set samples, the input data mapped with the charging load during training stage. In the next stage, during the prediction, feature extractor employs to attain feature vectors from the patterns and the models to generate prediction results. The Bi-LSTM model has the capability to learn complex contextual and temporal features related to load forecasting. The forward, backward information flow and the final output of hidden layers of the BiLSTM network are given by Equations (14), (15), and (16), respectively.

$$h_f = f(W_{f1}x_t + W_{f2}h_{t-1}) \tag{14}$$

$$h_b = f(W_{b1}x_t + W_{b2}h_{t+1}) \tag{15}$$

$$y_i = g(W_{o1} * h_f + W_{o2} * h_b) \tag{16}$$

3.3.4 *Bidirectional GRU*

The BiGRU model incorporates bidirectional processing and gated mechanisms to capture and model sequential patterns in data. The architecture of Bi-GRU is similar to Bi-LSTM except for the inclusion of a cyclic unit. The two bidirectional networks of GRU can simultaneously use forward and backward information. The structure of the GRU unit is simpler than the LSTM unit hence the Bi-GRU model is simpler than the Bi-LSTM model.

3.3.5 *1-D convolutional networks*

The 1-D convolution network is considered for the analysis. The 1-D convolution block is composed by a specific number of filters, where the filter has a fixed size. A convolution operation is performed between the vector and the filter, to generate the new vector as an output. The output consists of multiple channels as the number of filters. The layers in the convolutional block are activated using an activation function.

3.3.6 *Hybrid model*

The hybrid model is designed by combining both LSTM layers and 1-D CNN. In the hybrid architecture, the first layer includes the 1D CNN which extracts relevant features from the training data to forecast EV charging load. The details of the architecture for the hybrid model is defined in Table 1.

Table 1. Architecture details of the hybrid model

Architecture	Network parameters
CNN1	Conv1d (1, 64, kernel size = 3), Stride = 1
LSTM 1	LSTM (64, 128, batch_first = true)
LSTM 2	LSTM (128, 32, batch_first = true, bidirectional = true)
Fully connected layer	Linear (input features = 64, output features = 1, bias = true)

3.4 *Training of the model*

The model is trained using the training data samples wherein the input and output of the LSTM and GRU models are mapped for solving the forecasting problem. In this research, a Stochastic Gradient Descent (SGD) algorithm is employed for solving the optimization problem. SGD is an iterative algorithm wherein the parameters of the models are updated after each iteration using the backpropagation algorithm. The SGD algorithm selects the number of components and hyper parameters such as loss function, batch size, learning rate, initialized weights and number of epochs. The performance of the deep learning models in terms of load forecasting is discussed in Section 5.

3.5 *Dynamic pricing of EV*

This research employs a dynamic pricing algorithm wherein the algorithm provides the real-time pricing in order to assist the companies or the service providers for adjusting the prices according to the market demands. Dynamic pricing defines the structure of the utility rate wherein the charges for per kWh varies every hour based on the real-time production costs of the utility. The dynamic pricing estimation using the proposed algorithm is determined using the below given equations.

$$LP = K \tag{17}$$

$$(L^T L)P = L^T K \tag{18}$$

$$P = (L^T L)^{-1} L^T K \tag{19}$$

Where, L represents the forecasted load in slots-wise, P defines the price in slots-wise, and K represents the total forecasted load price. The slot wise load array and the load price is determined for 24 hours.

3.6 *EV scheduling*

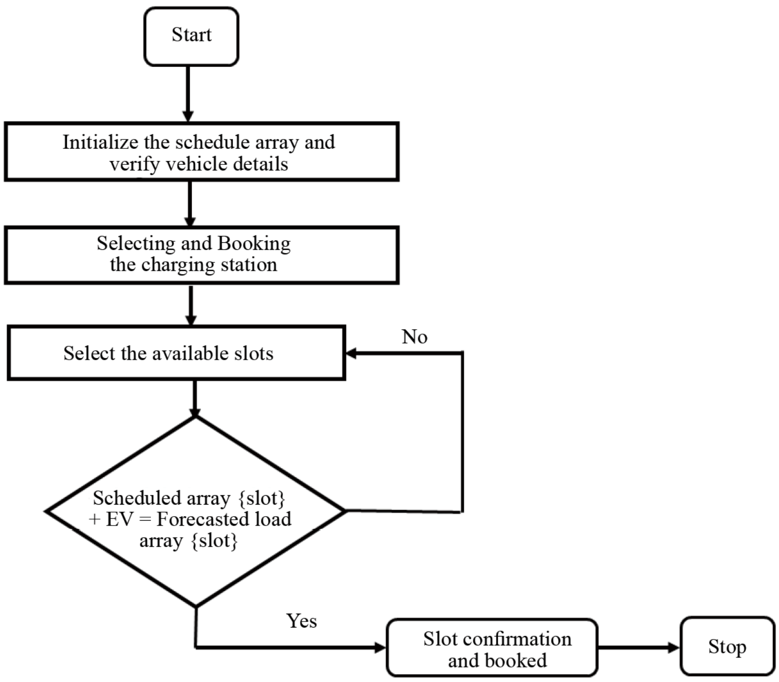

Figure 1. Flowchart of the proposed scheduling process

As discussed in the previous section, in this work, two arrays are forecasted. The first load array is forecasted in a slot-wise manner and the second array, dynamic array is forecasted in a slot-wise prices. In this section, charging scheduling in EVs is analyzed while maintaining a temporary array for recording the scheduled loads. The sum of the battery KWH booked in a slot are considered as scheduled loads here. The process followed by the proposed algorithm for EV scheduling for a particular slot and time are as follows:

1. If battery KWH + scheduled load < slot value in the forecasted array; then the EV user is allowed to charge.
2. If battery KWH + scheduled load > slot value in the forecasted array; then the EV user is suggested to charge with another best slot.

The flowchart of the proposed dynamic pricing algorithm is illustrated in Figure 1.

4. Experimental results

This section discusses the results of the experimental analysis with respect to load forecasting, dynamic pricing and scheduling. The parameters of the neural network are tuned using the optimization algorithm for enhancing the accuracy of forecasting. The performance of the models are determined with respect to the training and testing losses for all three datasets i.e, PJM dataset, Panama dataset, and UK dataset. The hyper parameters selected for the analysis are shown in Table 2.

Table 2. Hyper parameters

Hyper parameter	PJM dataset	Panama dataset	UK dataset
Loss Function	RMSE	RMSE	RMSE
Batch Size	32	32	32
Learning Rate	0.004	0.0035	0.004
Weight Initialization	Xavier Normal	Xavier Normal	Xavier Normal
Epochs	5	7	6

The loss function is used to estimate the performance of the proposed DL models using a specific set of weights for the data samples obtained from the training dataset. The optimization algorithm begins the search process using the initial model parameters or weights. Since the random values are selected as initial model weights, there is a need to determine the weights using efficient techniques and this process is known as weight initialization. In this research, the Xavier Normal technique is used for initializing the weights. It is worthy to note that, at the beginning of the training process, the initial small random values are assigned to model weights. While updating the model, a few samples from the training dataset are used for calculating the model error and this process is defined as the loss and the selection of number of training samples for model update is termed as batch size. In other words, batch size is also defined as the number of samples used for estimating the error gradient before updating the parameters of model for a particular instant. After estimating the error gradients, the derivative of the error is calculated and the same is used for updating the parameters of the DL models. The amount that each model parameter is updated per cycle of the learning algorithm is defined as learning rate. It is noted that the models are trained for multiple times and the total number of iterations of the process is defined by the number of iterations through the training dataset after which the training process is terminated; to achieve a better forecasting. This number is defined as an epoch. Although there are several parameters responsible for controlling the performance of the DL models, this research identifies five hyper parameters as shown in Table 2, for enhancing the forecasting accuracy of the DL models. The training and testing loss of the DL models is illustrated in Figure 2 for (a) Panama, (b) PJM and (c) UK.

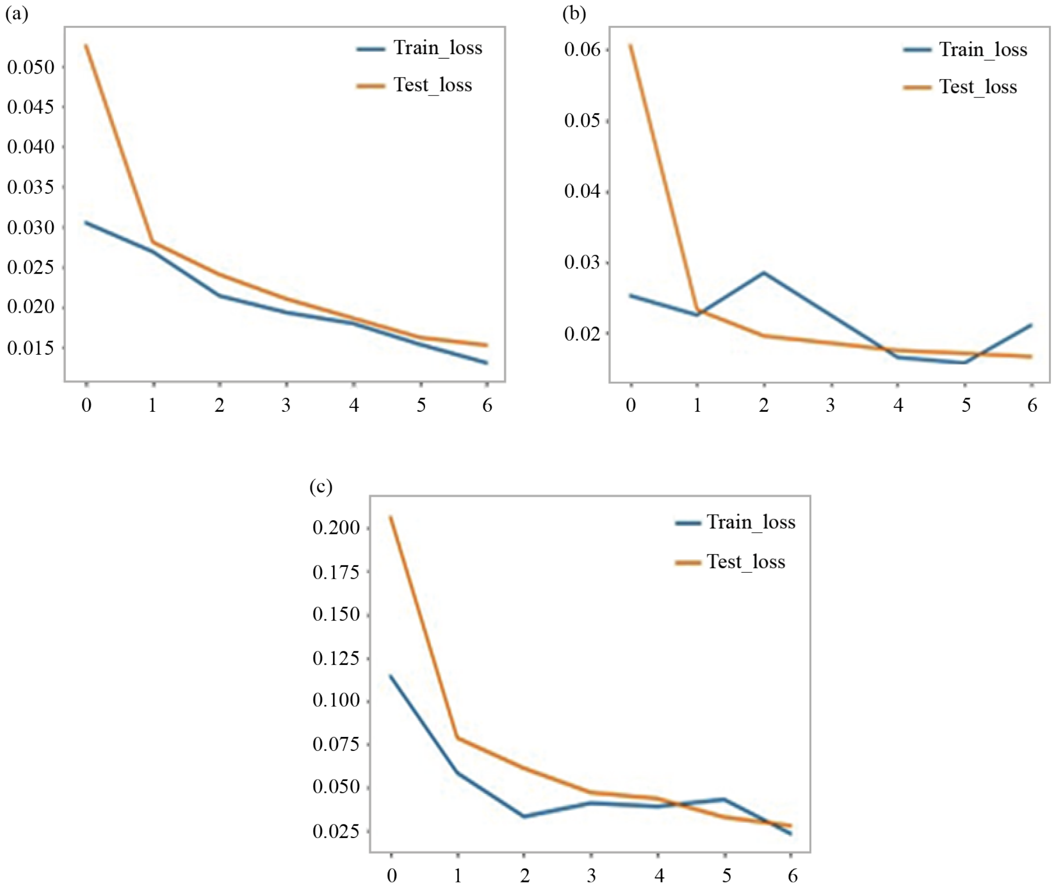

Figure 2. Testing and training loss for different datasets: (a) Panama, (b) PJM and (c) UK

The Figure 3 presents the test predictions and true labels for three distinct test samples: Panama, PJM, and the UK, shown in subfigures as Figure 3a, b, and c, respectively. For Panama (a), the predictions closely align with the true labels, indicating good model performance, though minor deviations suggest areas for improvement. In contrast, the PJM dataset (b) exhibits more significant discrepancies between predicted and true values, highlighting potential challenges in the model's ability to generalize to this sample, which may require further investigation into feature representation or overfitting. The UK dataset (c) shows reasonably accurate predictions, but there are small errors, suggesting the model is generally effective but could benefit from fine-tuning or incorporating region-specific features. Overall, the model demonstrates strong performance on Panama and the UK datasets, while the PJM dataset may need additional optimization to improve prediction accuracy. The corresponding validation losses for different DL models are computed and the results are tabulated in Table 3.

Table 3. Metric comparison for various DL models

Metric Model	MAE	MSE	RMSE	MAPE	MASE
RNN	0.2719	0.09756	0.3123	204.89	0.8318
CNN	0.2696	0.09584	0.3095	203.23	0.8248
CNN-LSTM	0.2693	0.09505	0.3094	201.24	0.8241

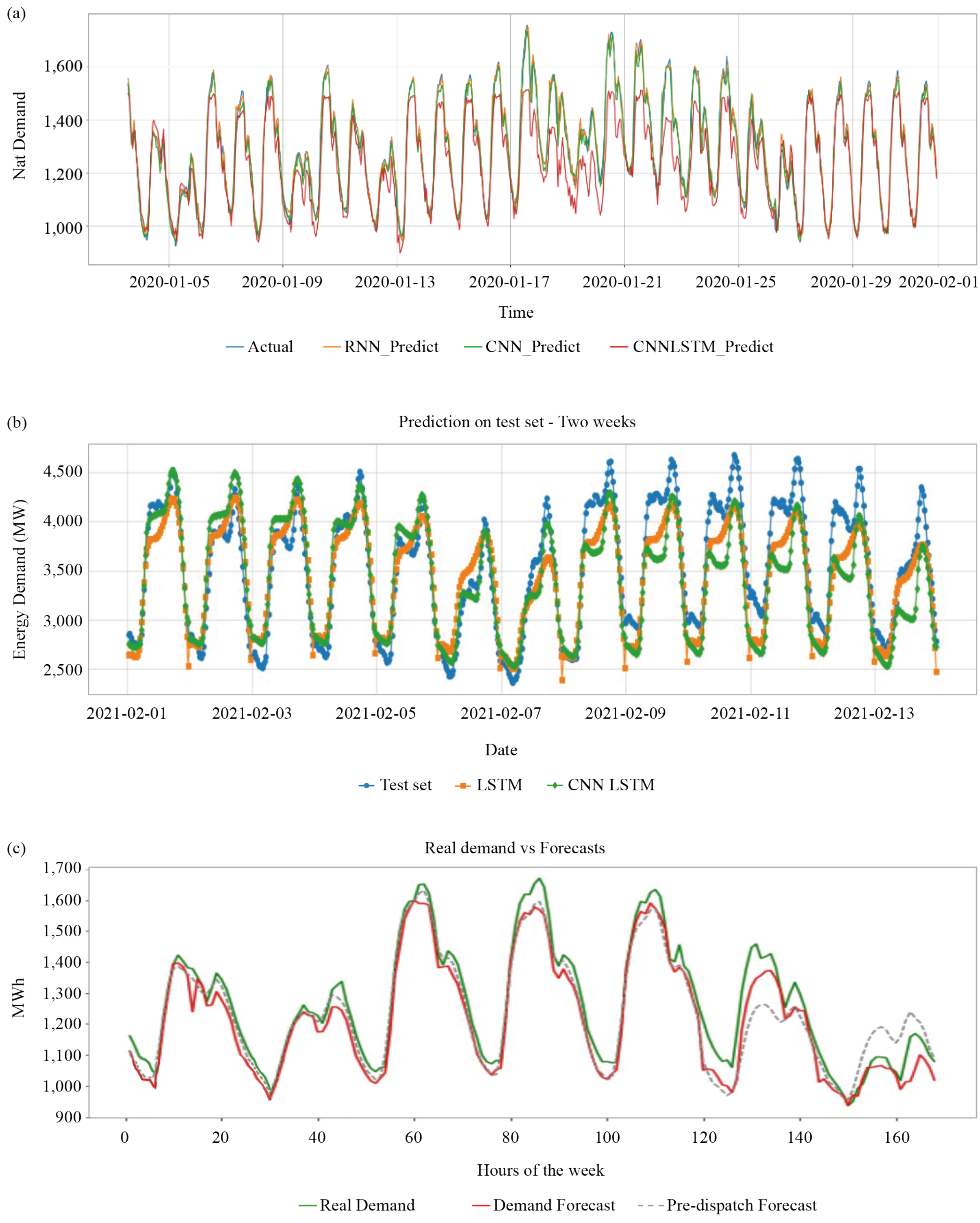

Figure 3. Test prediction and true labels of test sample

Table 3 compares the performance of various Deep Learning models, specifically RNN, CNN, and a hybrid CNN-LSTM model, across several key evaluation metrics. These metrics include MAE, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Mean Absolute Scaled Error (MASE). Across all models, the CNN-LSTM combination demonstrates marginally better performance compared to the other models, achieving the lowest MAE (0.2693) and MSE (0.09505), which suggests it has the most accurate predictions in terms of the average magnitude of error.

The RNN model, while still performing relatively well, shows slightly higher error rates across all metrics compared to CNN and CNN-LSTM. Specifically, it has the highest MAE (0.2719) and MSE (0.09756), indicating a higher deviation in predictions. Additionally, the MAPE for the RNN (204.89) is noticeably higher than that of CNN (203.23) and CNN-LSTM (201.24), highlighting that the RNN model's performance diminishes as the percentage error increases. Overall, while the differences in performance between the CNN and CNN-LSTM models are minimal, the latter's superior results in the MAE, MSE, and MAPE suggest that the integration of LSTM with CNN enhances the model's capacity to capture sequential dependencies, thus improving prediction accuracy.

In addition to the validation loss, the load forecasting performance is determined in Figure 4 (a) and (b), respectively.

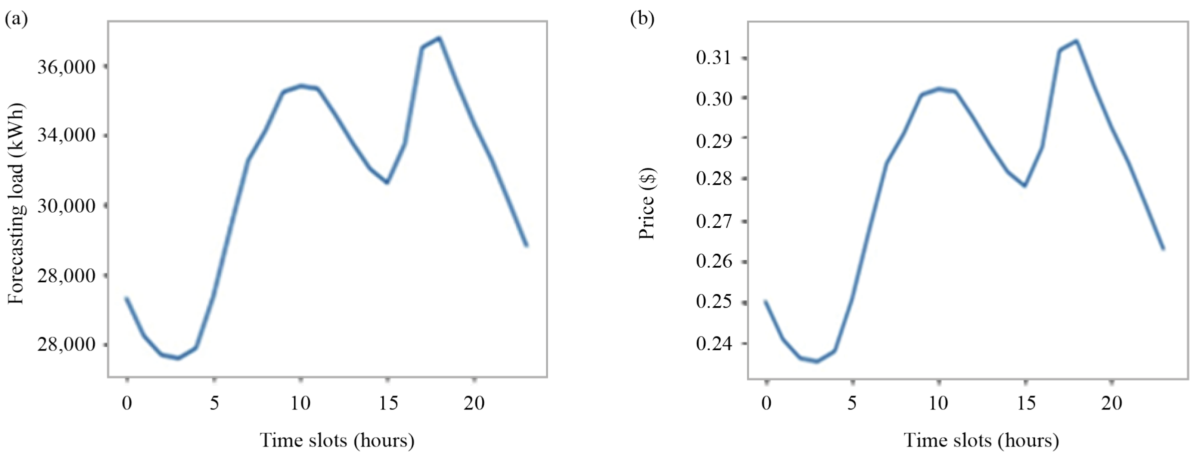

Figure 4. Load forecasting performance

The outcome of the forecasting analysis shows that the load curve is aligning with the price curve which validates the fact that the proposed dynamic pricing algorithm achieved precise real time pricing based on the load.

4.1 *Development of GUI for charging process management*

In this study, a Graphical User Interface (GUI) has been developed to visualize the forecasting performance of the proposed approach. The GUI is implemented using the Tkinter platform, which is a widely used standard library for creating graphical interfaces in Python. The Tk GUI toolkit enables the development of intuitive and efficient user interfaces, facilitating the creation of user-friendly applications. Through this interface, users are able to access and interact with information related to charging stations, time slot availability, and scheduling, thereby enhancing the overall user experience. Figure 5 demonstrates the process flow for the deployment of the GUI application, providing a clear and structured representation of the steps involved from the initial application launch to the final user interface. This flowchart is essential for understanding the underlying architecture of the system and how the various components interact to facilitate the application's functionality.

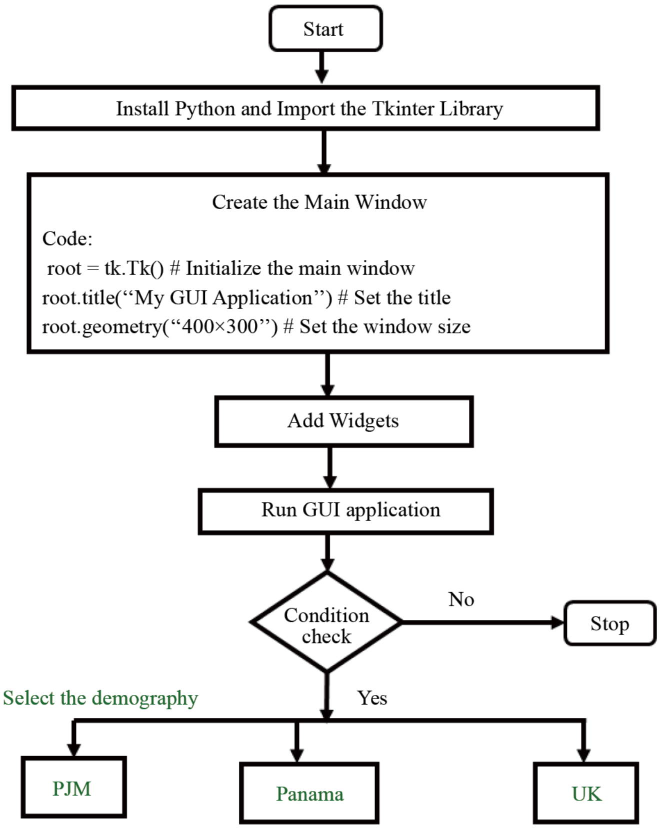

Figure 5. Flowchart of GUI application deployment

Figure 6 displays a screenshot of the user interface that allows users to select a charging station. The visual representation highlights the simplicity and intuitiveness of the design, showcasing how users can interact with the application to make selections. This feature is crucial in providing a user-friendly experience for selecting charging stations from a list of available options.

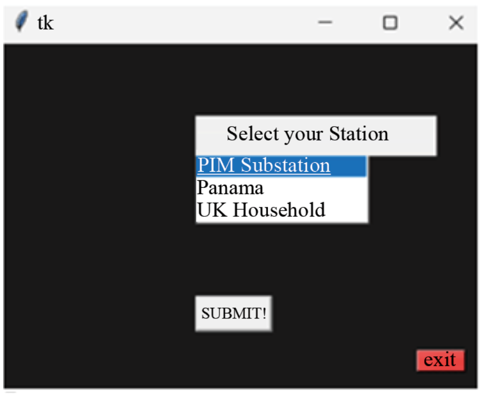

Figure 6. A window screenshot of GUI application for selecting of charging station

Figure 7 further extends the user interaction by illustrating the interface for selecting a range of time slots. This is particularly significant when users are choosing between multiple charging stations, such as the Panama and UK household charging stations, as shown in the figure. The ability to select time slots offers a level of customization and flexibility for users, enabling them to plan their charging sessions more effectively and according to their availability.

Together, the Figures 6, 7 demonstrate the user-centric design of the GUI application, focusing on the deployment process, ease of navigation, and functionality aimed at enhancing the user experience in selecting and managing charging stations. The flowchart and screenshots emphasize the application's design and usability, which are key to ensuring efficient and accessible interactions for users.

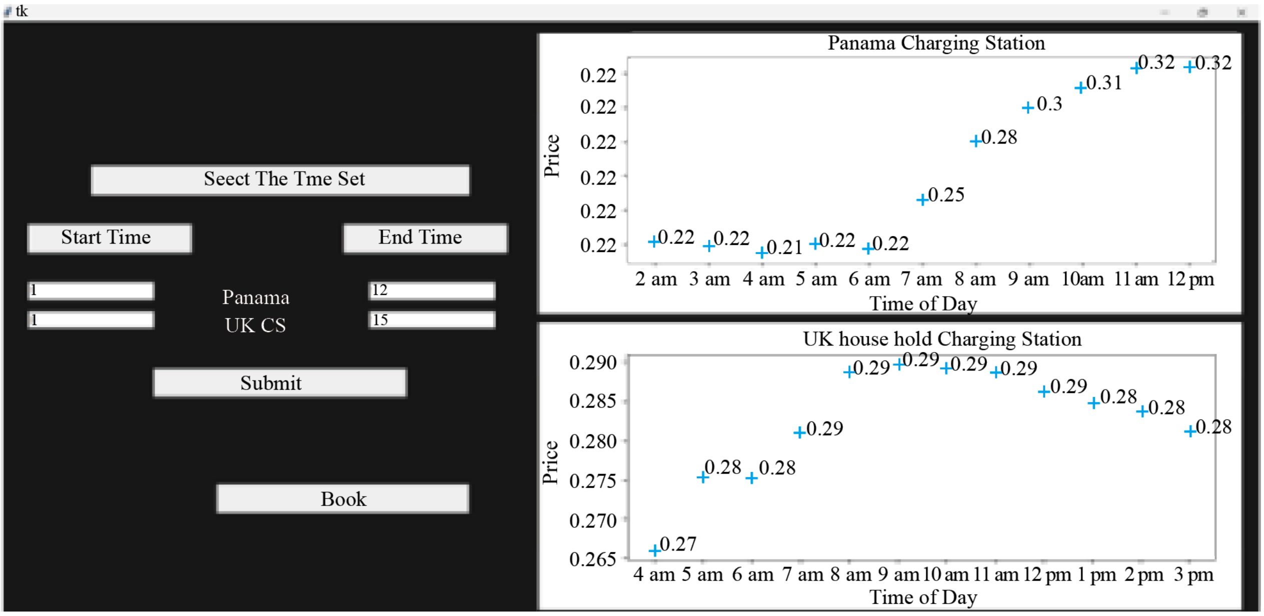

Figure 7. A window screenshot of GUI application for selecting range of time slots when two charging station, such as Panama and UK household charging stations are chosen

5. Conclusion

This research presented a comprehensive analysis of different deep learning models for achieving efficient load forecasting and charging scheduling in EVs. Different DL models such as LSTM, GRU, BiLSTM, BiGRU, and hybrid models are tested and evaluated for determining the forecasting performance. The study analyzes the optimization of DL models for identifying the hyper parameters which control the performance of the forecasting models. Experimental analysis is conducted to determine the effectiveness of the proposed approach. Results show that the dynamic pricing algorithm implemented in this research provides optimal real-time pricing for the EV companies to adjust the price based on the load demand. In addition, the efficacy of the scheduling process is validated using the GUI developed. Experimental analysis also shows that the hybrid model achieves better performance in terms of achieving minimum validation loss. For future scope, more charging station data can be included in the training for better results and the implementation of RES can be deployed to meet the demand of the EV users.

6. Limitations

The analysis is limited to specific models (e.g, CNN, GRU, LSTM, and hybrids). While these models are effective, there are other advanced architectures like Transformers that could offer enhanced performance. While the dynamic pricing algorithm is implemented, its real-world validation or impact on customer behavior and satisfaction is not discussed.

7. Future scope

In addition, this research can also be extended to design a EV charging planner to minimize monthly expenses on EV charging and which acts like a personal guide and recommendation system can be included with GPS so that all nearby CS are prioritized to reduce traveling time. Further, the study explores the integration of renewables [26] with maximum capabilities [29].

Conflict of interest

The authors declare no conflict of interests.

References

[1] Krishna VB, Sandeep V, Murthy SS, Yadlapati K. Experimental investigations on performance comparison of self-excited induction generator and permanent magnet synchronous generator for small-scale renewable applications. *Renewable Energy*. 2022; 195: 431-441. Available from: https://doi.org/10.1016/j.renene.2022.06.051.

[2] Krishna VB, Duvvuri SSSR, Sobhan PVS, Yadlapati K, Sandeep V, Narendra BK. Experimental study on excitation phenomena of renewable energy source-driven induction generator for isolated rural community loads. *Results in Engineering*. 2024; 21: 101761. Available from: https://doi.org/10.1016/j.rineng.2024.101761.

[3] Pidikiti T, Sheedevi G, Mopidevi S, Krishna VB. Design and control of Takagi-Sugeno-Kang fuzzy controller-based inverter for power quality improvement in grid-tied PV systems. *Measurement: Sensors*. 2023; 25: 100638. Available from: https://doi.org/10.1016/j.measen.2022.100638.

[4] Kumar RR, Alok K. Adoption of electric vehicles: A literature review and prospects for sustainability. *Journal of Cleaner Production*. 2020; 253: 119911. Available from: https://doi.org/10.1016/j.jclepro.2019.119911.

[5] Subbarao M, Dasari K, Duvvuri SSSR, Prasad KRKV, Narendra BK, Krishna VB. Design, control and performance comparison of PI and ANFIS controllers for BLDC motor-driven electric vehicles. *Measurement: Sensors*. 2024; 31: 101001. Available from: https://doi.org/10.1016/j.measen.2023.101001.

[6] Zhu J, Yang Z, Guo Y, Zhang J, Yang H. Short-term load forecasting for electric vehicle charging stations based on deep learning approaches. *Applied Sciences*. 2019; 9(9): 1723. Available from: https://doi.org/10.3390/app9091723.

[7] El-Azab HAI, Swief RA, El-Amary NH, Temraz HK. Seasonal electric vehicle forecasting model based on machine learning and deep learning techniques. *Energy and AI*. 2023; 14: 100285. Available from: https://doi.org/10.1016/j.egyai.2023.100285.

[8] Gharibi MA, Nafisi H, Askarian-Abyaneh H, Hajizadeh A. Deep learning framework for day-ahead optimal charging scheduling of electric vehicles in parking lot. *Applied Energy*. 2023; 349: 121614. Available from: https://doi.org/10.1016/j.apenergy.2023.121614.

[9] Al-Ogaili AS, Hashim TJT, Rahmat NA, Ramasamy AK, Marsadek MB, Faisal M, et al. Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations. *IEEE Access*. 2019; 7: 128353-128371. Available from: https://doi.org/10.1109/ACCESS.2019.2939595.

[10] Yang Z, Hu T, Zhu J, Shang W, Guo Y, Foley A. Hierarchical high-resolution load forecasting for electric vehicle charging: A deep learning approach. *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*. 2022; 4(1): 118-127. Available from: https://doi.org/10.1109/JESTIE.2022.3218257.

[11] Zhang X, Chan KW, Li H, Wang H, Qiu J, Wang G. Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model. *IEEE Transactions on Cybernetics*. 2020; 51(6): 3157-3170. Available from: https://doi.org/10.1109/TCYB.2020.2975134.

[12] Hammad MA, Jereb B, Rosi B, Dragan D. Methods and models for electric load forecasting: A comprehensive review. *Logistics and Sustainable Transport*. 2020; 11(1): 51-76. Available from: https://doi.org/10.2478/jlst-2020-0004.

[13] Morsalin S, Mahmud K, Town G. Electric vehicle charge scheduling using an artificial neural network. In: *2016 IEEE Innovative Smart Grid Technologies-Asia (ISGT-Asia)*. Melbourne, VIC, Australia: IEEE; 2016. p.276-280. Available from: https://doi.org/10.1109/ISGT-Asia.2016.7796398.

[14] Li Y, Che J, Yang Y. Subsampled support vector regression ensemble for short-term electric load forecasting. *Energy*. 2018; 164: 160-170. Available from: https://doi.org/10.1016/j.energy.2018.08.169.

[15] Xue M, Wu L, Zhang QP, Lu JX, Mao X, Pan Y. Research on load forecasting of charging station based on XGBoost and LSTM model. *Journal of Physics: Conference Series*. 2021; 1757(1): 012145. Available from: https://doi.org/10.1088/1742-6596/1757/1/012145.

[16] Rasheed T, Bhatti AR, Farhan M, Rasool A, El-Fouly TH. Improving the efficiency of deep learning models using supervised approach for load forecasting of electric vehicles. *IEEE Access*. 2023; 11: 91604-91619. Available from: https://doi.org/10.1109/ACCESS.2023.3307022.

[17] Chang M, Bae S, Cha G, Yoo J. Aggregated electric vehicle fast-charging power demand analysis and forecast based on LSTM neural network. *Sustainability*. 2021; 13(24): 13783. Available from: https://doi.org/10.3390/su132413783.

[18] Shanmuganathan J, Victoire AA, Balraj G, Victoire A. Deep learning LSTM recurrent neural network model for prediction of electric vehicle charging demand. *Sustainability*. 2022; 14(16): 10207. Available from: https://doi.org/10.3390/su141610207.

[19] Sasidharan MP, Kinattingal S, Simon SP. Comparative analysis of deep learning models for electric vehicle charging load forecasting. *Journal of The Institution of Engineers (India): Series B*. 2023; 104(1): 105-113. Available from: https://doi.org/10.1007/s40031-022-00798-4.

[20] Huang X, Wu D, Boulet B. Ensemble learning for charging load forecasting of electric vehicle charging stations. In: *2020 IEEE Electric Power and Energy Conference (EPEC)*. Edmonton, AB, Canada: IEEE; 2020. p.1-5. Available from: https://doi.org/10.1109/EPEC48502.2020.9319916.

[21] Zhang T, Huang Y, Liao H, Liang Y. A hybrid electric vehicle load classification and forecasting approach based on GBDT algorithm and temporal convolutional network. *Applied Energy*. 2023; 351: 121768. Available from: https://doi.org/10.1016/j.apenergy.2023.121768.

[22] Vishnu G, Kaliyaperumal D, Pati PB, Karthick A, Subbanna N, Ghosh A. Short-term forecasting of electric vehicle load using time series, machine learning, and deep learning techniques. *World Electric Vehicle Journal*. 2023; 14(9): 266. Available from: https://doi.org/10.3390/wevj14090266.

[23] Zhu J, Yang Z, Mourshed M, Guo Y, Zhou Y, Chang Y, et al. Electric vehicle charging load forecasting: A comparative study of deep learning approaches. *Energies*. 2019; 12(14): 2692. Available from: https://doi.org/10.3390/en12142692.

[24] Mohamed AAR, Best RJ, Liu X, Morrow DJ. A comprehensive robust techno-economic analysis and sizing tool for the small-scale PV and BESS. *IEEE Transactions on Energy Conversion*. 2021; 37(1): 560-572. Available from: https://doi.org/10.1109/TEC.2021.3107103.

[25] Zhang T, Xu R. Performance comparisons of Bi-LSTM and Bi-GRU networks in Chinese word segmentation. In: *Proceedings of the 2021 5th International Conference on Deep Learning Technologies*. 2021. p.73-80. Available from: https://doi.org/10.1145/3480001.34800.

[26] Mrudul GV, Rohit G, Harshavardhan G, Dhanush K, Anudeep B. Efficient energy management: Practical tips for household electricity conservation. *Journal of Modern Technology*. 2024; 1(1): 1-8. Available from: https://doi.org/10.71426/jmt.v1.i1.pp1-8.

[27] Rao PN, Lavanya V, Manasa D, Boggavarapu S, Soni BP. Battery models and estimation techniques for energy storage systems in residential buildings. *Journal of Modern Technology*. 2024; 1(1): 47-58. Available from: https://doi.org/10.71426/jmt.v1.i1.pp47-58.

[28] Varanasi LN, Karri SPK. Enhancing non-intrusive load monitoring with channel attention guided bi-directional temporal convolutional network for sequence-to-point learning. *Electric Power Systems Research*. 2024; 228: 110088. Available from: https://doi.org/10.1016/j.epsr.2023.110088.

[29] Solanke AV, Verma SK, Kumar S, Oyinna B, Okedu KE. MPPT for hybrid energy systems using machine learning techniques. *Journal of Modern Technology*. 2024; 1(1): 19-37. Available from: https://doi.org/10.71426/jmt.v1.i1.pp19-37.

[30] Varanasi LN, Karri SPK. STNILM: Switch transformer-based non-intrusive load monitoring for short and long duration appliances. *Sustainable Energy, Grids and Networks*. 2024; 37: 101246. Available from: https://doi.org/10.1016/j.segan.2023.101246.

[31] Al-Amri RM, Hadi AA, Kadhim MS, Mousa AH, Matloob AZK, Hasan HF. Enhancement of the performance of machine learning algorithms to rival deep learning algorithms in predicting stock prices. *Babylon Journal of Artificial Intelligence*. 2024; 2024: 118-127. Available from: https://doi.org/10.58496/BJAI/2024/012.

[32] Noori WE, Albahri AS. Towards trustworthy myopia detection: integration methodology of deep learning approach, XAI visualization, and user interface system. *Applied Data Science and Analysis*. 2023; 2023: 1-15. Available from: https://doi.org/10.58496/ADSA/2023/001.

[33] Nafea AA, Alameri SA, Majeed RR, Khalaf MA, AL-Ani MM. A short review on supervised machine learning and deep learning techniques in computer vision. *Babylon Journal of Machine Learning*. 2024; 2024: 48-55. Available from: https://doi.org/10.58496/BJAI/2024/012.