

Short-Term Aggregated Residential Load Forecasting using BiLSTM and CNN-BiLSTM

Bharat Bohara, *Student Member, IEEE*
ECE Department
University of Houston
Houston, United States
bbohara@uh.edu

Raymond I. Fernandez
ECE Department
University of Houston
Houston, United States
rifernan@uh.edu

Vysali Gollapudi
ECE Department
University of Houston
Houston, United States
vgollap2@uh.edu

Xingpeng Li, *Senior Member, IEEE*
ECE Department
University of Houston
Houston, United States
xli82@uh.edu

Abstract—Higher penetration of renewable and smart home technologies at the residential level challenges grid stability as utility-customer interactions add complexity to power system operations. In response, short-term residential load forecasting has become an increasing area of focus. However, forecasting at the residential level is challenging due to the higher uncertainties involved. Recently deep neural networks have been leveraged to address this issue. This paper investigates the capabilities of a bidirectional long short-term memory (BiLSTM) and a convolutional neural network-based BiLSTM (CNN-BiLSTM) to provide a day ahead (24 hr.) forecasting at an hourly resolution while minimizing the root mean squared error (RMSE) between the actual and predicted load demand. Using a publicly available dataset consisting of 38 homes, the BiLSTM and CNN-BiLSTM models are trained to forecast the aggregated active power demand for each hour within a 24 hr. span, given the previous 24 hr. load data. The BiLSTM model achieved the lowest RMSE of 1.4842 for the overall daily forecast. In addition, standard LSTM and CNN-LSTM models are trained and compared with the BiLSTM architecture. The RMSE of BiLSTM is 5.60%, 2.85% and 2.60% lower than LSTM, CNN-LSTM and CNN-BiLSTM models respectively. The source code of this work is available at <https://github.com/Varat7v2/STLF-BiLSTM-CNNBiLSTM.git>.

Index Terms—Long short-term memory (LSTM), Bidirectional long short-term memory (BiLSTM), Deep Neural Networks (DNN), Load forecasting, Optimal load dispatch, Energy Management

I. INTRODUCTION

A load forecasting is a technique used by the power/energy utilities to estimate the future load profiles so as to balance the power demand-supply chain. A short-term load forecasting (STLF) ranges from estimating the load demand for the next half an hour up to the next two weeks. An accurate load forecasting techniques help the utilities to plan or schedule the power generations optimally. It has a greater economic impact on the utilities as it plays a critical role in optimal load scheduling, economic load dispatch, superlative load flow, and contingency analysis, and intelligent planning, operations, and the maintenance of the power systems [1].

*Note: Only the personal use of this preprint is permitted - solely for the purpose of research, and republication/redistribution requires IEEE permission. Check http://www.ieee.org/publications_standards/publications/rights/index.html for more information. Please cite this paper from this link: <https://ieeexplore.ieee.org/document/9990696>.

Indeed, the economy of operations and control of power systems are sensitive to positive and negative forecasting errors. Under-estimates have a negative impact on demand response and power system operations, whereas overload conditions challenging to manage. Over-estimates impacts power system operation and hence the system's efficiency [2]. Even 1% reduction in the average STLF error can potentially save a typical utility company millions of dollars [3]. Thus, STLF has emerged as an essential area of research for efficient and reliable power system operation.

At the residential level, higher penetration of renewables, integration of smart home technologies, and the electrification of both vehicles and heating/cooling systems present a challenge to grid stability as utility-customer interactions add complexity to power system operations. In response, short-term residential load forecasting has become an increasing area of focus [4]. However, forecasting at this level, even for aggregated loads, is challenging because of the diverse electricity consumption behaviors and patterns. Thus it adds up a rapid fluctuation and variability in the data, making the prediction job more challenging. Residential household load profiles vary significantly depending on the house size, occupancy, presence of solar panels, electric vehicle usage, and other socio-demographic factors [5]. STLF for smaller residential communities is a complex process because of the non-smooth and nonlinear temporal load profile behaviors. Thus, accurate methods for STLF at the residential level are needed and will provide valuable information for the effective generation, and distribution of the electricity [6].

Machine learning techniques are widely becoming popular in Machine learning techniques are widely popular in predicting the aggregated residential electric load. Zhang *et al.* [7] have applied a combination of state vector machine (SVM) based clustering and decision tree techniques to smart meter data to build a short-term load forecasting framework. The model can accurately predict the power usage pattern with real-time operation constraints. Stephen *et al.* [8] have applied a Gaussian-Markov chain sampling technique to smart meter data to predict residential power usage. Hence, machine learning-based models achieved equivalent accuracy but were observed to be computationally expensive compared to a traditional autoregressive integrated moving average (ARIMA)

model.

Deep learning has become an active area of research in short-term aggregated residential load forecasting. Specifically, long short-term memory (LSTM) architectures are becoming an increasing area of focus. One of the first attempts to use an LSTM was conducted by Marino *et al.* [9]. Standard LSTM and LSTM-based sequence-to-sequence (S2S) architectures were employed to predict the aggregated electricity consumption of residential customers. The study demonstrated that the S2S LSTM architecture could accurately forecast loads with both one-minute and one-hour temporal resolution. This result marked a notable improvement compared to previous studies with mostly shallow artificial neural networks. Similar studies have addressed residential short-term load forecasting as well. Kong *et al.* [10] have tested an LSTM neural network on residential smart meter data and compared the performance to a comprehensive set of benchmarks in the field of load forecasting. Hence, from the above studies, it is observed that the LSTM approach outperforms other state-of-the-art forecasting algorithms in forecasting short-term loads.

Other studies have addressed the more challenging task of individual residence load forecasting. Wang *et al.* [11] have studied a short-term residential load forecasting using an LSTM model that takes a two-dimensional feature of load and weather as input, and a mean absolute percentage error (MAPE) score of 48.46% is achieved. This marked a 9.8% accuracy improvement compared to a MAPE score of 53.77% achieved using the one-dimensional load input. In addition to the multiple feature input data, researchers have also experimented with more complex LSTM architectures. Alhussein *et al.* [12] has proposed a deep learning framework consisting of combined convolutional neural network (CNN) and LSTM networks. The hybrid CNN-LSTM model is tested on individual household electrical load data, and a better result with an average MAPE of 40.38% is achieved compared to other forecasting techniques. In this study, MAPE score of 8.18% and 44.6% is achieved for aggregated and individual household load predictions, respectively. Hence, because of the possibility of better model accuracy, an aggregated household load is chosen over the individual household for further analytical comprehension.

There is very limited work done on short-term load forecasting using BiLSTM and CNN-BiLSTM. Wu *et al.* [13] has proposed an attention-based CNN-LSTM-BiLSTM model to forecast one hour ahead load profiles of regional integrated energy systems with historical load, cooling load, temperature, and gas consumption for the past 5 days as the input features. The authors have stated that the proposed method outperformed all other models such as CNN-BiLSTM, CNN-LSTM, BiLSTM, LSTM, random forest regression (RFR), and support vector machine regression (SVR). Similarly, Miao *et al.* [14] has proposed a short-term load forecasting with CNN-BiLSTM for highly accurate time series prediction with Bayesian Optimization (BO) to tune the model hyperparameters and Attention Mechanism (AM) to focus on the important part of the BiLSTM layers. The proposed model takes

historical load profiles, time slots and meteorological data to predict load for the next 24 hours. The attention mechanism-based learning is not included in our work, however, it can be bookmarked for the future work.

This paper proposes a bidirectional LSTM (BiLSTM) and convolutional BiLSTM (CNN-BiLSTM) to predict the aggregated household power consumption in a small residential community. The main idea is to forecast a day ahead load profile, given the previous day's load at an hourly resolution. The contribution of this work is to push the potential of the regular and convolutional BiLSTM further to model the short-term forecasting at the lower time resolution without any significant loss in the model performance.

II. BACKGROUND

A. LSTM Architecture

Recurrent Neural Networks (RNNs) are a branch of deep neural networks that are designed to process sequential data [15]. One of the main problems faced in RNN is *gradient vanishing* issue. With more depth of neural layers, the gradients assigned to the weight matrix go on diminishing and finally have no effect on the output. The other problem observed in conventional RNNs is *gradient exploding* - again, with the increasing number of hidden layers, the gradients assigned to the weight matrices grow exponentially, thus making output incomprehensible. Both the gradient vanishing and the gradient exploding make RNNs hard to train.

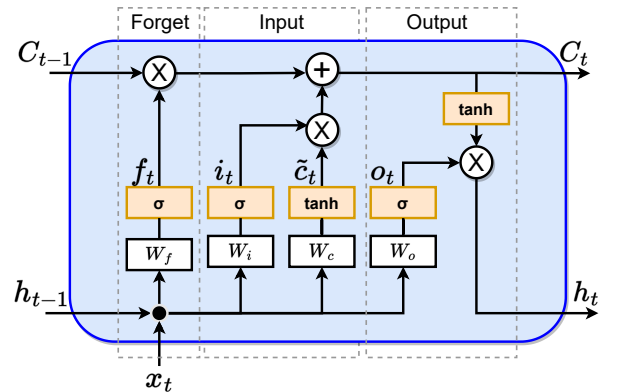


Fig. 1: LSTM cell with gating controls

LSTM - a gated RNN, overcomes the difficulty faced by traditional RNNs in learning the long-term dependencies of the sequential input data [16]. It can leverage the long-term previous states sequential information with the involvement of gates - forget gate (f_t), input gate (i_t), and output gate (o_t). The LSTM cell architecture with the gating controls is shown in fig. 1, where all the gates $f_t, i_t, o_t \in \{0, 1\}$: 0 means that the gate is closed and no information passes through it, and 1 means the gate is open to allow the information flow. The forget gate ($f_t \in \{0, 1\}$) is constituted of a sigmoid function that determines whether to retain or discard the information from the cell memory, i.e., 0 - indicates forgetting and 1 - indicates remembering the cell memory information. The

input or update gate is composed of two layers: 1) a sigmoid function ($i_t \in \{0, 1\}$) - that determines whether to update the cell input or not, and 2) a tanh function ($\tilde{c} \in \{-1, 1\}$) - that creates a vector of new candidate values to be added to the memory cell. At the end, the combination of both the layers and the forget layer updates the LSTM memory cell as shown in equation 4. The output gate is also composed of a sigmoid, and a tanh function: $\sigma \in \{0, 1\}$ determines if LSTM-cell information has contributed to the overall cell output, and hyperbolic tangent function maps the final cell output between -1 and 1 . In other words, the output gate controls the output of the information and determines if the current hidden state will be passed to the next sequence network. The mathematical framework of a LSTM-cell is depicted in equations 1, 2, 3, and 4 [16]–[18].

$$\begin{aligned}
i_t &= \sigma(w_{ix}x_t + w_{ih}h_{t-1} + b_i) \\
f_t &= \sigma(w_{fx}x_t + w_{fh}h_{t-1} + b_f) \\
\tilde{c}_t &= \tanh(w_{cx}x_t + w_{ch}h_{t-1} + b_c) \\
o_t &= \sigma(w_{ox}x_t + w_{oh}h_{t-1} + b_o)
\end{aligned} \tag{1}$$

where, i_t is a input gate, f_t is a forget gate, \tilde{c}_t is a candidate hidden state, and o_t is an output gate of a LSTM cell. Eqn. 1 can be rearranged in a vectorized form as shown in eqn. 2.

$$\begin{aligned}
\begin{bmatrix} i_t \\ f_t \\ o_t \end{bmatrix} &= \sigma \left(\begin{bmatrix} w_{ix} & w_{ih} \\ w_{fx} & w_{fh} \\ w_{ox} & w_{oh} \end{bmatrix} \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + \begin{bmatrix} b_i \\ b_f \\ b_o \end{bmatrix} \right) \\
\tilde{c}_t &= \tanh \left(\begin{bmatrix} w_{cx} & w_{ch} \end{bmatrix} \cdot \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b_c \right)
\end{aligned} \tag{2}$$

The vectorized eqn. 2 can be condensed to the eqn. 3.

$$\begin{aligned}
\mathbf{y}_1 &= \sigma(\mathbf{w}_1 \mathbf{x}_1 + \mathbf{b}_1) \\
\mathbf{y}_2 &= \tanh(\mathbf{w}_2 \mathbf{x}_2 + \mathbf{b}_2)
\end{aligned} \tag{3}$$

where,

$$\begin{aligned}
\mathbf{y}_1 &= [i_t \quad f_t \quad o_t]^T, \quad \mathbf{w}_1 = \begin{bmatrix} w_{ix} & w_{fx} & w_{ox} \\ w_{ih} & w_{fh} & w_{oh} \end{bmatrix}^T, \\
\mathbf{x}_1 &= [x_t \quad h_{t-1}]^T, \quad \mathbf{b}_1 = [b_i \quad b_f \quad b_o]^T, \\
\mathbf{y}_2 &= [\tilde{c}_t], \quad \mathbf{w}_2 = [w_{cx} \quad w_{ch}]^T, \\
\mathbf{x}_2 &= [x_t \quad h_{t-1}]^T, \quad \mathbf{b}_2 = [b_c]
\end{aligned}$$

The final cell output at each time step will be as shown in eqn. 4.

$$\begin{aligned}
c_t &= f_t * c_{t-1} + i_t * \tilde{c} \\
h_t &= o_t * \tanh(c_t)
\end{aligned} \tag{4}$$

where, c_t is cell states and h_t is a vector of cell output values.

B. BiLSTM Architecture

The BiLSTM neural network architecture used in this study is illustrated in Fig. 2. To overcome the shortcoming of a single LSTM cell that can capture only the previous state information but not the future state information, a Bi-LSTM model is used in this paper. It is built upon two independent

LSTM networks: a forward LSTM cell that passes information from the back to the front direction and a reverse LSTM cell that passes information from the front to back direction. This allows the network to leverage the learning from both the forward and reverse direction of the input sequence at every time step. Through this flexibility, the network can look at both the previous and future context information to predict the next step sequence.

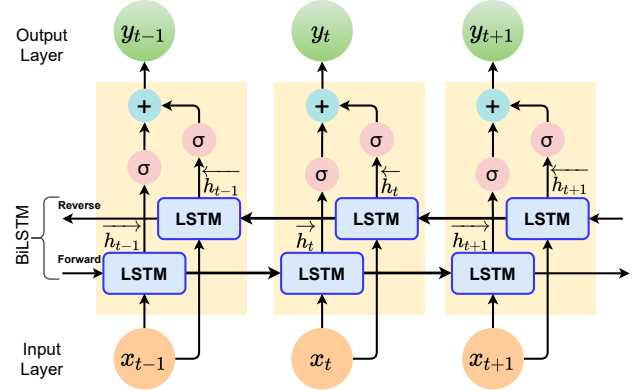


Fig. 2: Architecture of the BiLSTM network

A BiLSTM cell is fed with a input sequence of $x = (x_1, x_2, \dots, x_n)$, where n represent the length of the sequence. In reference to a BiLSTM cell architecture shown in fig. 2, $\vec{h}_t = (\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n)$ is forward hidden sequence, $\overleftarrow{h}_t = (\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n)$ is reverse hidden sequence, and $y_t = (y_1, y_2, \dots, y_n)$ is the output sequence. The final encoded output vector is the combined effect of both forward and reverse information flow i.e., $y_t = f(\vec{h}_t, \overleftarrow{h}_t)$ [17]. The mathematical framework of the BiLSTM neural networks architecture is shown in eqn. 5.

$$\begin{aligned}
\vec{h}_t &= \sigma(w_{\vec{h}x}x_t + w_{\vec{h}h}h_t + b_{\vec{h}}) \\
\overleftarrow{h}_t &= \sigma(w_{\overleftarrow{h}x}x_t + w_{\overleftarrow{h}h}h_t + b_{\overleftarrow{h}}) \\
y_t &= w_{y\vec{h}}\vec{h}_t + w_{y\overleftarrow{h}}\overleftarrow{h}_t + b_y
\end{aligned} \tag{5}$$

C. CNN-BiLSTM Architecture

A convolution neural network (CNN) is cascaded on top of the BiLSTM networks to extract the prior feature knowledge of the inputs before fetching them into the BiLSTM layers. Although CNN layers add additional computational cost, the CNN-BiLSTM model is not observed to perform equivalent to the BiLSTM. Its poor performance might be because of losing some information due to the max-pooling layers.

As shown in fig. 3, first, the entire dataset is sliced into per day (24 hr.) load profiles, and each slice is considered a single input to the 1D CNN layer. 64 filters with a kernel size of 3x1 and a default stride of 1 are used to extract the feature vectors from the input load profile data. Max-pooling is applied on the convoluted feature maps. It reduces the dimension but also might have lost some vital information. It is the reason behind its lower accuracy compared to the BiLSTM networks. In

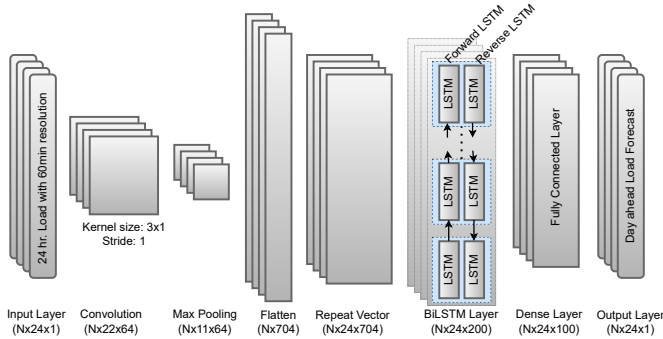


Fig. 3: Architecture of the CNN-BiLSTM network

order to perform sequential load demand forecasting, BiLSTM networks are used on the flattened feature vectors. Finally, the load profile for the next day is outputted by the fully-connected layers with the *ReLU* activation function. The only difference in CNN-LSTM is that it uses regular forward-directed LSTM layers instead of the bidirectional layers for sequential time-series predictions.

D. Naive Forecasting Technique

The main idea of the naive forecasting technique is that it assumes the previous day's load profile as the forecast for the next day's load demand. This technique applies only to data with a periodic repeating and recurring sequence pattern. Since the hourly load profile data has a strong repetitive seasonal component, as observed in fig. 5, this technique is used in this study for comparison purposes. On the other hand, the naive forecasting technique is usually used in any recurring sequential time-series data because of its simplicity and inexpensive computational cost.

$$D(t+1) = D(t) \quad \forall t = \{1, 2, 3, \dots, n\} \quad (6)$$

where, D is an hourly load profile for a particular day, t is a previous day, $t+1$ is the next day and n is the total length of the sequence in days.

III. METHODOLOGY

A. Dataset Preparation

An aggregated residential electrical load profiles measured in 38 single-family houses in Northern Germany [19] is used in this study. The dataset consists of active and reactive power per household from 10 seconds to 60 minute temporal resolution dated from May 2018 to December of 2020. The data is open-sourced and consists of seven hierarchical data format version 5 (HDF5) files for each year - that consists of different hierarchies as shown in fig. 4. The dataset used in this study is a subset of the larger HDF5 dataset published in [19]; the subset is generated with the red-dashed arrowed direction as shown in fig. 4 - with filename *data_spatial*, top-level node *NO_PV*, middle-level node *60min*, and low-level node *HOUSEHOLD*. The generated subset dataset consists of a timestamp and active and reactive power. For load forecasting, only active power in *kW* is used. The dataset is divided

into training, testing, and validation sets based on the date, as shown in the table I. A Min-max normalization eqn. 7 is applied to the training, validation, and test dataset before feeding into the models.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (7)$$

where X is a regular and X_{norm} is the normalized data.

TABLE I: Data split into training, validation, and test set

Category	Load Profile From	Load Profile To
Training set	5/31/2018	12/31/2019
Validation set	1/1/2020	6/30/2020
Test set	7/1/2020	12/31/2020

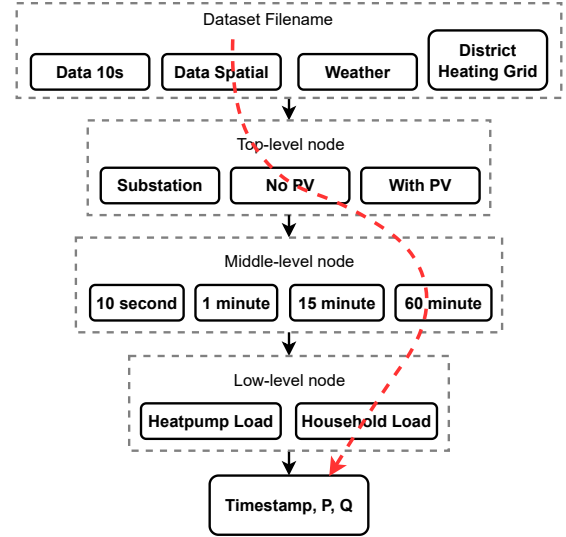


Fig. 4: Hierarchy of dataset published in [19], where red-dashed arrow indicates the data-subset used in this study

An additive time series decomposition is applied to the dataset in order to split the time series data into several components such as seasonality, trend, and noise. Seasonality gives the periodic signal in the time series data; trend gives the hidden pattern if present in the data, and noise gives the remaining random signals. Each of these components will be helpful in data preparation, model selection, and model finetuning. A representative example of an arbitrary 30 day period of data is additively decomposed, as depicted in fig. 5, at an hourly resolution. The data has a strong seasonality and significant noise, but no distinct trend exists. A complex trend and a noise component make it difficult to forecast the power demand with typical statistical techniques. However, the strong daily seasonality pattern suggests that a naive forecast made by using a day before load to predict the day ahead load profile can provide a good forecast baseline measurement.

B. Objective and Metrics

The main objective of the load forecasting technique is to minimize the error between the actual and predicted daily load forecasts at an hourly resolution. Root mean squared error

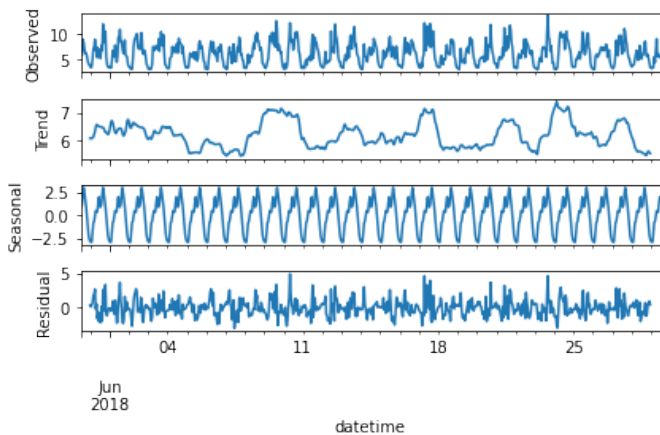


Fig. 5: Time-series decomposition of the data

(RMSE) metric is used to evaluate the performance of the trained model.

$$J(\theta) = \min_{w,b} \frac{1}{n} \sqrt{\sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (8)$$

where $n \in \{1, 2, 3, \dots, 24\}$ is the total number of data points for 24 hrs. with an hourly resolution, y_i is the actual value at i^{th} hour, \hat{y}_i is the predicted value at i^{th} hour, the cost function may be terminated via an early stopping technique to prevent the model from overfitting, and parameters $\theta \in \{w, b\}$ are weights and biases of the neural networks that are learned during the model training with the observation data as shown in table I.

C. Proposed Method

A dataset consisting of the aggregated household load is first divided into training, validation, and test sets. Training and validation sets are used to train the BiLSTM and CNN-BiLSTM networks. The test set is held back to evaluate the model's performance. Training and validation sets are normalized and then fed into the model. The model initially takes in the first 24 hrs. of data and attempts to predict the following 24 hrs. load profile. In the second training iteration, the window is shifted ahead by 1 hour. The window shifting continues until the dataset is completely exhausted. Simultaneously, the validation set evaluates the model performance and monitors the issues such as overfitting or underfitting. The model accuracy can be further improved by increasing the training set and varying the batch size and the total number of BiLSTM cells and layers. Adam optimizer adaptively optimizes the learning rate based on the error value and the rate of change of the error. Once the model is trained, a walk-forward test strategy is employed to evaluate the model's performance. During the model evaluation, the model initially takes in the first 24 hrs. of test data and uses the trained model to predict the following 24 hrs. load profile. During the second test iteration, again, the window is shifted ahead by 24 hrs. time frame throughout the available test data points.

D. Hyperparameter Tuning

The hyperparameters such as batch size, number of BiLSTM cells, and the network layer size are tuned by using a grid search method. The batch size and the number of total cells in the BiLSTM layer are varied, and the corresponding RMSE values are reported in table II. The parameters - batch size of 384 and LSTM cells of 200, resulted in the lowest average RSME value of 1.4842 with minimum and maximum values ranging between [1.4800, 1.5020]. However, with multiple folds grid search method, these tuned parameters resulted in the lowest RMSE value multiple times. Hence these parameters are finally considered for model training.

TABLE II: RMSE Values for Grid Search Parameters

Batch size	150 cells	200 cells	400 cells
96	1.5053	1.5023	1.5240
192	1.5518	1.5279	1.5039
384	1.4981	1.4842	1.5171

IV. RESULTS AND DISCUSSION

A. Naive Forecasting Technique

Since the naive forecasting technique is generally adopted in real-life forecasting problems, it is considered the baseline in this study. It is, by default, human nature to assume that similar events will get repeated in the subsequent days. The hourly RMSE error between the actual and predicted load demand is calculated as shown in fig. 6(a), and the resulting prediction output is shown in fig. 6(b). Lower error values indicate better prediction is observed in the early morning and late-night hours compared to the mid-day. An average daily RSME value for the naive forecasting technique is observed to be $1.7282kW$.

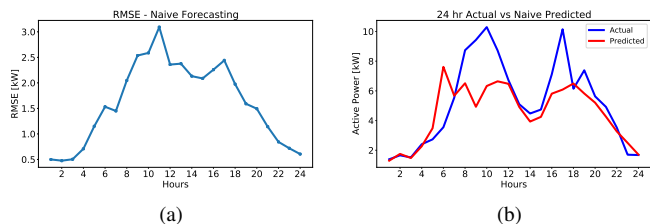


Fig. 6: (a) RMSE and (b) predicted load profile using naive forecasting technique

B. BiLSTM Forecast

Fig. 7(a) shows the hourly RSME plot between the actual and predicted load demand. Like the naive forecasting technique, the lower RMSE values indicate better predictions are observed in the early morning and late-night hours. An average daily RMSE score for the BiLSTM forecast method is observed to be $1.4842kW$, which shows a 13.53% decrease in the average error compared to the naive forecast.

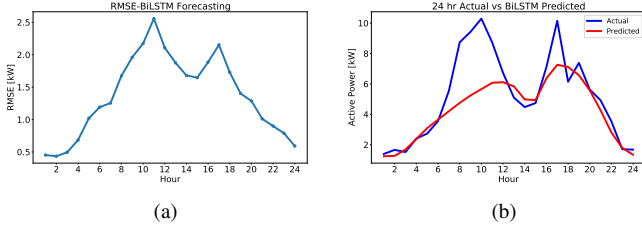


Fig. 7: (a) RMSE between actual and predicted load, (b) hourly forecasted load profile using BiLSTM networks

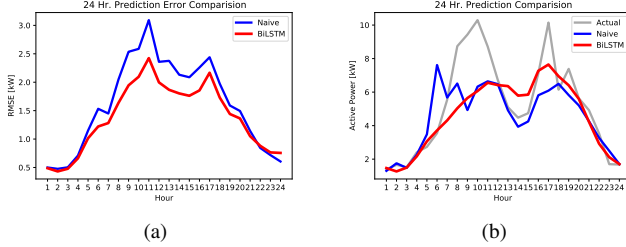


Fig. 8: (a) Hourly RMSE, (b) Hourly load forecast comparison of naive forecasting vs BiLSTM

C. CNN-BiLSTM Forecast

As shown in table IV, an average RMSE of the BiLSTM model is 2.6% lower than that of CNN-BiLSTM. However, if additional and more complex data are involved, such as weather, renewable sources (PV, wind turbine, etc.), and individual household demand response, CNN would perform better.

D. Models Comparison and Discussion

To observe the effectiveness of the BiLSTM model, table III compares the method to a similar study [20] that uses machine learning algorithms, i.e., support vector machine (SVM), LSTM, multi-layered perceptron (MLP) to forecast daily electricity consumption at an hourly resolution for a micro-grid consisting of 38 homes. Compared to those models, the proposed method is able to achieve an RMSE reduction of between 50.31 – 76.03%.

TABLE III: Comparison of the proposed model with the similar study [20]

Algorithms	RMSE (Average)	BiLSTM Improvement (% Δ)
SVM	6.191	75.76
MLP	5.654	73.45
LSTM	2.987	49.75
BiLSTM (this paper)	1.501	-

In addition, the BiLSTM and CNN-based BiLSTM models are compared with the naive forecasting technique, improved LSTM, and CNN-LSTM models that are trained during the study. BiLSTM's RMSE is lower than any of the models, as shown in table IV, and fig. 10, and observed to be 16%, 5.60%,

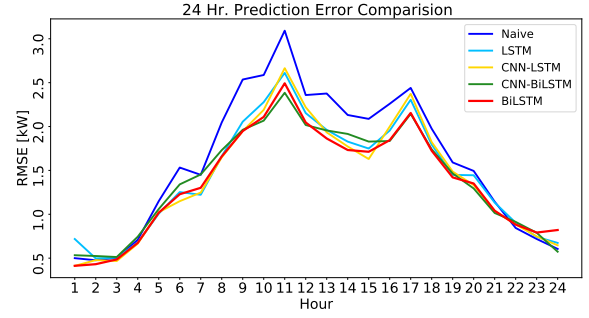


Fig. 9: Comparison of hourly RMSE of different models

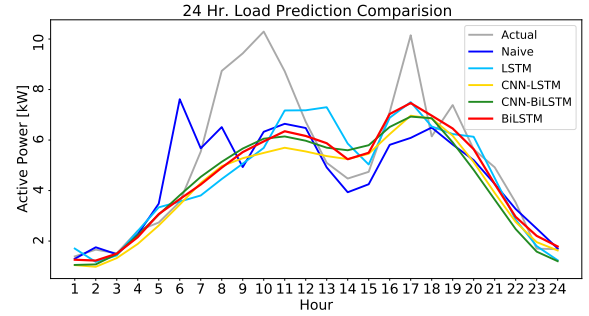


Fig. 10: Comparison of hourly prediction of different models

2.85% and 2.60% better than the naive forecasting, improved LSTM, CNN-LSTM and CNN-BiLSTM models respectively.

The data has a strong seasonal component but lacks a trend and a substantial noise as the dataset comprises 38 residential homes. It is possible to consider the utility of the BiLSTM network for making more accurate short-term forecasting to support the economic load dispatch of the community microgrid, similar to the paper [20]. The load forecasting provided by machine learning can be used to model the load variation over a period in the future and thus, allow for matching supply and demand in community microgrid optimal load dispatch models for grid-connected community microgrids.

V. CONCLUSION

Different deep learning models, i.e., standard LSTM, CNN-LSTM, bidirectional LSTM (BiLSTM), and CNN-BiLSTM, are trained to perform a day ahead load forecast of the aggregated residential load of 38 homes given the previous day load profile. The RMSE of BiLSTM architecture is observed to be 16%, 5.60%, 2.85%, and 2.60% better than the naive forecast, LSTM, CNN-LSTM, and CNN-BiLSTM respectively. Because of the information loss during max-pooling, CNN-BiLSTM cannot perform better than the regular BiLSTM model. However, it is believed that with the increased diversity and variability in the data - more than just the seasonal pattern, the CNN-BiLSTM would perform better than BiLSTM. Hence, in the future, the performance of BiLSTM and CNN-BiLSTM models will be evaluated on more dynamic

TABLE IV: Comparative study of the proposed model with other frequently adopted models for dynamic load forecasting

Algorithms	No. of trainable parameters	RMSE (Average)	BiLSTM Improvement (% Δ)
Naive	0	1.787	16
LSTM	51,001	1.590	5.60
CNN-LSTM	332,457	1.545	2.85
BiLSTM	101,801	1.501	-
CNN-BiLSTM	664,457	1.541	2.60

and uncertain data such as weather data and the more extensive load profiles involved.

VI. ACKNOWLEDGEMENT

The authors would like to thank the deHumlaTek AI Research and Mr. Saurav Adhikari for their helpful suggestions to improve the overall paper quality.

REFERENCES

- [1] M. Jacob, C. Neves, and D. Vukadinović Greetham, "Short Term Load Forecasting," pp. 15–37, 2020. **1**
- [2] Y. Chen, P. B. Luh, C. Guan, Y. Zhao, L. D. Michel, M. A. Coolbeth, P. B. Friedland, and S. J. Rourke, "Short-term load forecasting: Similar day-based wavelet neural networks," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 322–330, feb 2010. **1**
- [3] H. K. Alfares and M. Nazeeruddin, "Electric load forecasting: Literature survey and classification of methods," vol. 33, no. 1, pp. 23–34, jan 2010. **1**
- [4] A. Ahmad, N. Javaid, A. Mateen, M. Awais, and Z. A. Khan, "Short-Term Load Forecasting in Smart Grids: An Intelligent Modular Approach," *Energies 2019, Vol. 12, Page 164*, vol. 12, no. 1, p. 164, jan 2019. **1**
- [5] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, "Short term load forecasting and the effect of temperature at the low voltage level," *International Journal of Forecasting*, vol. 35, no. 4, pp. 1469–1484, oct 2019. **1**
- [6] N. Amjady, F. Keynia, and H. Zareipour, "Short-term load forecast of microgrids by a new bilevel prediction strategy," *IEEE Transactions on Smart Grid*, vol. 1, no. 3, pp. 286–294, dec 2010. **1**
- [7] P. Zhang, X. Wu, X. Wang, and S. Bi, "Short-term load forecasting based on big data technologies," *CSEE Journal of Power and Energy Systems*, vol. 1, no. 3, pp. 59–67, oct 2015. **1**
- [8] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1591–1598, jul 2017. **1**
- [9] D. L. Marino, K. Amarasinghe, and M. Manic, "Building energy load forecasting using Deep Neural Networks," *IECON Proceedings (Industrial Electronics Conference)*, pp. 7046–7051, dec 2016. **2**
- [10] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 841–851, jan 2019. **2**
- [11] Y. Wang, N. Zhang, and X. Chen, "A Short-Term Residential Load Forecasting Model Based on LSTM Recurrent Neural Network Considering Weather Features," *Energies 2021, Vol. 14, Page 2737*, vol. 14, no. 10, p. 2737, may 2021. **2**
- [12] M. Alhussein, K. Aurangzeb, and S. I. Haider, "Hybrid cnn-lstm model for short-term individual household load forecasting," *IEEE Access*, vol. 8, pp. 180 544–180 557, 2020. **2**
- [13] K. Wu, J. Wu, L. Feng, B. Yang, R. Liang, S. Yang, and R. Zhao, "An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system," *International Transactions on Electrical Energy Systems*, vol. 31, no. 1, p. e12637, jan 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/2050-7038.12637> **2**
- [14] K. Miao, Q. Hua, and H. Shi, "Short-Term Load Forecasting Based on CNN-BiLSTM with Bayesian Optimization and Attention Mechanism," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12606 LNCS, pp. 116–128, 2021. [Online]. Available: <https://link.springer.com/chapter/10.1007/978-3-030-69244-5-10> **2**
- [15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>. **2**
- [16] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series," in *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*. Institute of Electrical and Electronics Engineers Inc., dec 2019, pp. 3285–3292. **2, 3**
- [17] L. Cai, S. Zhou, X. Yan, and R. Yuan, "A Stacked BiLSTM Neural Network Based on Coattention Mechanism for Question Answering," *Computational Intelligence and Neuroscience*, vol. 2019, 2019. **3**
- [18] Y. Yu, C. Hu, X. Si, J. Zheng, and J. Zhang, "Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset," *Neurocomputing*, vol. 402, pp. 134–147, aug 2020. **3**
- [19] M. Schlemminger, T. Ohrdes, E. Schneider, and M. Knoop, "Dataset on electrical single-family house and heat pump load profiles in Germany," *Scientific Data 2022 9:1*, vol. 9, no. 1, pp. 1–11, feb 2022. [Online]. Available: <https://www.nature.com/articles/s41597-022-01156-1> **4**
- [20] L. Wen, K. Zhou, S. Yang, and X. Lu, "Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting," *Energy*, vol. 171, pp. 1053–1065, mar 2019. **6**