Vasanthan BUVANESAN, PhD Candidate (corresponding author)
dr.vasanthaneee@rediffmail.com
Coimbatore Institute of Technology, Tamil Nadu, India

Manikandan VENUGOPAL, PhD
manikantan.cit@gmail.com
Coimbatore Institute of Technology, Tamil Nadu, India

Kabil MURUGAN
kabilm1012@gmail.com
Coimbatore Institute of Technology, Tamil Nadu, India

Venbha VELUMANI SENTHILKUMAR
venbhavetha17@gmail.com
Coimbatore Institute of Technology, Tamil Nadu, India

An Optimised Deep Learning Model for Load Forecasting in Electric Vehicle Charging Stations

Abstract. *Accurate short-term load forecasting in the Electric Vehicle Charging Stations Network enhances power grid management. Existing methods often overfit the highly fluctuating energy consumption data from charging stations, creating a gap in developing accurate models. This paper tackles this challenge by proposing a ConvLSTM-BiLSTM, based encoder-decoder network, where convolutional layers are used to capture spatial trends along with recurrent layers for temporal dependencies. Furthermore, the model's hyperparameters are tuned using Levy Flight Particle Swarm Optimisation, enhancing its performance. The proposed model is evaluated on a publicly available Electric Vehicle Charging Stations dataset from Palo Alto City. The accuracy of the ConvLSTM-BiLSTM architecture with LFPSO optimisation surpasses that of conventional LSTM, BiLSTM models, and other encoder-decoder configurations. Significant improvements in RMSE, MAPE, and MSE were achieved, with reductions of around 37.14%, 62.13%, and 61.17%, respectively. The enhanced overall forecasting accuracy aids in better resource allocation and improves grid stability.*

Keywords: *Deep learning, electric vehicle, electric vehicle charging stations, encoder-decoder network, Levy flight particle swarm optimisation, load forecasting.*

JEL Classification: C32, C33, C61.

1. Introduction

Global adoption of electric vehicles (EVs) has increased in recent years, indicating a considerable transition in the automotive industry towards cleaner and sustainable transportation (Ravi & Aziz, 2022). Organisations throughout the world are adopting EVs as a crucial component to reduce carbon emissions, prevent climate

DOI: 10.24818/18423264/59.1.25.20

change, and promote energy efficiency. Advances in battery technology, stronger pollution restrictions, and greater awareness of the need for sustainable alternatives to fossil fuel-powered vehicles are all driving up EV demand (Rasoulinezhad, 2022). As a result, EVs are becoming more accessible to consumers, and forecasts indicate that their market share will continue to expand in the future decades (Ravi & Aziz, 2022). Although the shift to EVs is desirable, there are certain challenges, especially in the field of energy management. Load forecasting plays a vital role in energy management, as it ensures the stability and reliability of the power grid (Habbak et al., 2023). Load forecasting of charging stations refers to the prediction of future energy demand based on several factors. Since the broad adoption of EVs has a major effect on trends in energy consumption, an accurate estimation of the load in charging stations is particularly necessary. EV charging consumes a large amount of energy, particularly when charging during peak hours. Poor management of this increased demand could result in excessive stress leading to power breakdown in the grid. By addressing these energy management challenges, load forecasting plays a key role in maximising the benefits of EV adoption while maintaining grid stability and efficiency. Load forecasting can be considered a time series problem, in which historical data is utilised for forecasting future energy demand patterns. Time-series data often includes components such as trend, seasonality, autocorrelation, and noise. Noise levels are especially high when it comes to EV charging due to variations in charging habits and external influences. This makes it difficult for predictions and necessitates accurate methodology. Current techniques for load forecasting are broadly classified into two types: statistical methods, which rely on known mathematical models to capture data patterns, and data-driven methods, which use machine learning and advanced deep learning algorithms to analyse large data sets and learn trends.

Traditional statistical methods, such as the autoregressive integrated moving average (ARIMA) model, are commonly employed for load forecasting. Jubieras et al. (1999) used ARIMA for hourly load forecasting, adding weather factors to increase short-term prediction accuracy. Akshay et al. (2024) demonstrated that SARIMA outperformed ARMA, ARIMA models in forecasting EV loads, obtaining greater accuracy with lower error metrics. Lo Franco et al. (2023) combined the statistical approach with the machine learning model to forecast the load at EV charging stations, considering the SOC data for better accuracy. Overall, while traditional statistical models provide fundamental insights into time-series forecasting, the complexities of modern EV charging necessitate the use of advanced methods like machine learning and deep learning. These methods improve accuracy and address uncertainties, making them more applicable to real-world EV infrastructure.

In recent years, data-driven approaches to forecast EV charging loads have gained popularity, due to the advancements in machine learning and deep learning. Park et al. (1991) were the first to use artificial neural networks (ANN) for forecasting electric loads, demonstrating the model’s ability to detect nonlinear patterns in data. Li et al. (2018) used deep learning models like Long Short-Term

Memory (LSTM) and Gated Recurrent Units (GRU) to forecast EV charging station loads. GRU outperformed other models in terms of accuracy. Dabbaghjamanesh et al. (2021) introduced Q-learning for load prediction, which significantly improved forecast accuracy in both coordinated and uncoordinated EV charging scenarios. Huang et al. (2023) introduced MetaProbformer, the first transformer based load forecasting model utilising meta-learning to improve generalisation ability on new stations with scarce historical records. Yang et al. (2023) proposed a hierarchical load forecasting problem via attention-based LSTM model by capturing more complex trends in the data. Li et al. (2023) proposed a model of deep reinforcement learning with application in probabilistic charge load forecasting from real charging data using LSTM and proximal policy optimisation (PPO). Mohammad et al. (2023) developed ConvLSTM and BiConvLSTM encoder-decoder models to capture spatio-temporal correlations for forecasting energy consumption at EV charging stations. Table 1 summarises a comprehensive review of recent literature relevant to this research.

Table 1. Overview of Existing Research on Load Forecasting for Electric Vehicle Charging Stations (EVCS)

Reference	Method	EVCS Dataset	Metrics Used	Drawbacks	Advantages
(Kim, 2021)	ARMA,TBATS,ARI -MA,ANN,LSTM	Korea	MAPE	Best results are shown only with past data	Consideration of external variables improves microscale forecast accuracy.
(Huttel et al., 2023)	The tobil Model,Censored quantile regression,T-GCNs	Denmark, Copenhagen	ICP,MIL	Lacks factors like departure time and route, making it difficult to validate	Employs real-world censoring instead of artificial censoring schemes
(Zhang et al., 2020)	Probabilistic Distribution Models	U.S. National House hold travel survey(NHTS)	R^2	User preferences are not considered, which could affect future predictions	Refined probabilistic models to provide more accurate daily charging profiles
(Li et al., 2023)	CEEMDAN,LSO-VMD,BiLSTM	Shenzhen,China	RMSE,MAE, MAPE,R^2, WMAPE	The short-term forecasts and lack of precise EV charging behavior data	Accurate prediction of EV load in different periods helps for resource allocation in powergrid
(Shanmuganathan et al., 2022)	EMD–AOA–DLSTM	Georgia tech,Atlanta,USA	MAE,MSE, RMSE, A_{pre}	Limited external factor consideration, require further validation and model comparison	Improved EV charging demand forecasting accuracy and minimized errors
(Marzbani et al., 2023)	ARMA,MLR,PCA, ANN,SVR,KNN	UK,Shandong, Shenzhen,Georgia Tech,Korea	MAE, MAPE, RMSE, R^2 , MED	Don't consider both training and testing errors, potentially leading to inaccurate models	Hybrid models predict EV energy consumption better than linear and non-linear models
(Mohammad et al., 2023)	ConvLSTM-BiLSTM, BiConvLSTM-LSTM	Palo Alto,Boulder, Dundee,Perth	MSE,MAPE, RMSE	Hyperparameter tuning techniques are not considered and the prediction is offline	Accurate load forecasting using spatiotemporal analysis
(Huang et al., 2023)	Probformer, Metaprobformer	Palo,Boulder, Perth,NL	MAE,MSE,RMSE	Challenging to deal with brand new charging station without historical data	Improved short-term and long-term probabilistic forecasting
(Yang et al., 2023)	Enhanced Attention-Based LSTM	Shenzhen,China	R^2,MAE, RMSE,NRMSE	Needs more adaptive algorithms to handle varied energy situations accurately	High-resolution EV charging demand forecasting using real-world data
(Nespoli et al., 2023)	LSTM	Georgia Tech,Atlanta,USA	MAE,MSE and RMSE	Many deep layers may increase the model's temporal complexity	LSTM demonstrated improved accuracy of 97.14% with minimal MSE
(Dabbaghjamanesh et al., 2021)	Q-learning	Custom Dataset	MSE	Hyperparameter tuning techniques are not considered	Q-learning tracks PHEV load more accurately than ANN and RNN
(Rasheed et al., 2023)	Supervised learning RNN,LSTM,GRU	Boulder	NMAE,NRMSE	Must integrate optimization algorithms for real-world constraints	The ISL method improves FC and CNN forecasting accuracy
(Huttel et al., 2021)	T-GCN	Palo Alto	RMSE	Need more investigation since models assume a static charging station network	GCN models capture spatio-temporal correlations better than other methods

Source: Authors' analysis.

Although several methods, such as ARIMA, SARIMA, LSTM, and CNN-based architectures, have been developed for EV load forecasting, most approaches fail to fully capture the intricate spatio-temporal dependencies in charging station networks. Additionally, many models overfit or struggle with generalisation, particularly in high-variability scenarios like EV load demand. Moreover, while some studies focus on long-term or probabilistic forecasting, there is a critical gap in state-of-the-art methods regarding the availability of novel approaches for accurate short-term load forecasting that are able to capture spatial interdependencies and temporal dynamics within electric vehicle charging station networks. Addressing these gaps will be critical to improving the accuracy of load forecasting and enabling effective grid management.

To address the aforementioned gaps, a data-driven deep learning model is proposed for short-term load forecasting offering the following main contributions.

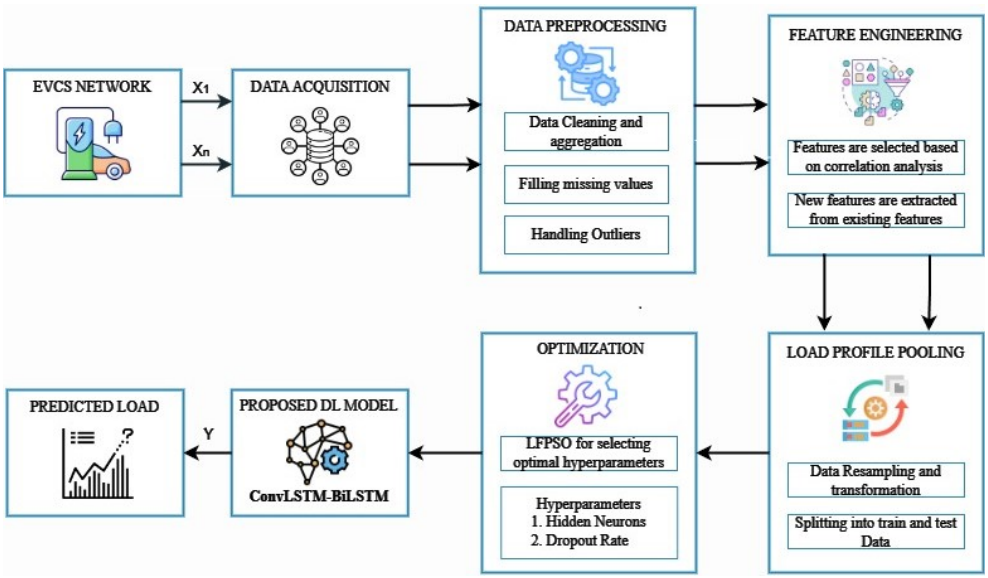

Figure 1. Block diagram of the proposed framework
Source: Authors' own creation.

(I) We propose a ConvLSTM-BiLSTM based encoder-decoder model that effectively captures the spatio-temporal dependencies within the charging station network.

outperforming the performance of traditional methods.

(II) The proposed methodology considers the interconnected nature of charging station networks rather than isolated stations. The forecasting accuracy has improved due to this deeper understanding of the network's dynamics.

III) Key features, such as peak and off-peak hours, are extracted from the existing features through feature engineering, as part of the preprocessing of the Palo Alto dataset.

(IV) Further, we employ Levy Flight Particle Swarm Optimisation (LFPSO) for optimising the hyperparameters of the DL model, which enhances the model's performance.

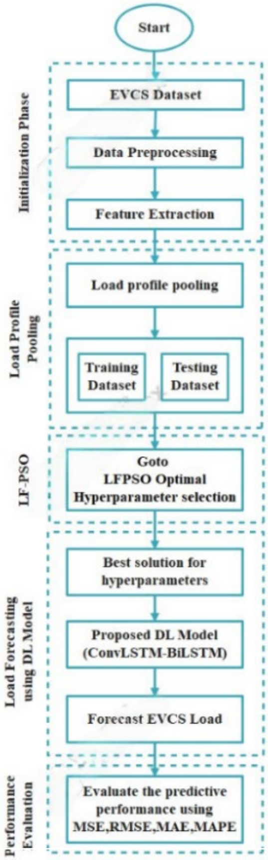

Figure 2. Work flow of proposed deep learning approach

Source: Authors' own creation.

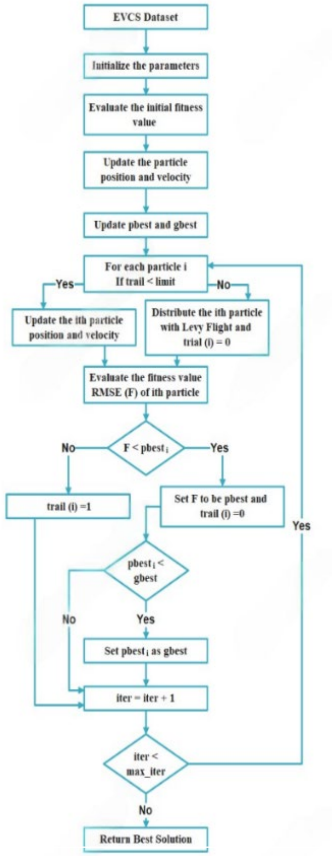

Figure 3. Flow chart describing the workflow of LFPSO

Source: Authors' own creation.

The subsequent sections of this paper are structured as follows. Section II provides a detailed explanation of the methodology that is proposed. Section III discusses the proposed model's performance within the EVCS dataset. Finally, Section IV concludes by discussing key findings and future research directions.

2. Proposed Methodology

Dynamic characteristics of the demand within Electric Vehicle Charging Stations (EVCS) Networks and their integration with the power grid constitute a significant challenge to forecast electric vehicle charging load. Existing methods often fail to capture both spatial and temporal correlations among charging stations, leading to inaccurate predictions. To tackle this challenge, our research introduces an innovative short-term load forecasting approach for EVCS that employs a deep learning (DL) based prediction model. Specifically, within an encoder-decoder architecture, we propose a combination of Convolutional Long Short Term Memory (ConvLSTM) and Bidirectional Long Short Term Memory (BiLSTM) networks. This architecture captures complex relationships within the EVCS and predicts future charging demands accurately. For hyperparameter tuning, this study employs Levy Flight Particle Swarm Optimisation (LFPSO) as a novel methodology. The LFPSO's exploration and exploitation capabilities are utilised to efficiently search the hyperparameter space, thereby balancing the exploration of new regions with the exploitation of promising areas. For identifying the optimal hyperparameter configuration, this ability to maintain a balance is crucial, which greatly reduces overall performance metrics, ultimately leading to a more accurate and robust deep learning model for EVCS load forecasting. The proposed framework of the EVCS load forecasting model, which employs both encoder-decoder architecture and LFPSO for hyperparameter tuning, is depicted as a block diagram in Figure 1. A flowchart explaining the complete workflow of the proposed approach is given in Figure 2.

2.1 *Development of LFPSO for optimising the hyperparameters*

Levy Flight Particle Swarm Optimisation (LFPSO) is used for hyperparameter optimisation. PSO struggles with global search and often becomes stuck in local optima. LFPSO addresses this issue by implementing the Levy flight mechanism. This approach enables particles to explore search space more efficiently, perhaps avoiding local minima and identifying superior solutions than standard PSO (Haklı & Uğuz, 2014). This section explains the methodology used to optimise the hyperparameters. The aim is to determine the best forecasting configuration to minimise RMSE. The flowchart describing the workflow is provided in Figure 3. Tables 2 and 3 provide the pseudocode for Levy Flight (LF) and LFPSO, respectively. The following are LFPSO procedures:

Table 2. Pseudo Code for Levy Flight

Algorithm 1: Levy Flight Algorithm
1: Input: Current Position, Target Position, Bounds)
2: Output: New Position
3: Start
4: Calculate the step length s based on Mantegna's algorithm
5: Calculate the different factor DF = $0.01 \times s \times$[Current Position - Target Position]
6: Compute the actual random walk with
7: New Position = Current Position+$s \times \text{rand}() \times$[size(CurrentPosition)]
8: Check the bounds of New Position to ensure that it is within the limits of the search space
9: Return New Position
10: End

Source: Authors' own creation.

Step 1 Initialisation of Hyperparameter Search Space (i) Number of hidden neurons in BiLSTM layer1 (φ_1) (ii) Dropout rate in layer1 (φ_2) (iii) Number of hidden units in BiLSTM layer2 (φ_3) (iv) Dropout rate in layer2 (φ_4) (v) Number of dense neurons (φ_5) (vi) L2 Regularizer λ in Encoder Layer (φ_6) (vii) L2 Regularizer λ in Decoder Layer 1 (φ_7) (viii) L2 Regularizer λ in Decoder Layer 2 (φ_8) are initialised as follows.

$$\varphi_{max} \leq \varphi \leq \varphi_{min} \tag{1}$$

The search space ranges are $1 \leq \varphi_1 \leq 500$, $0 \leq \varphi_2 \leq 1$, $1 \leq \varphi_3 \leq 500$, $0 \leq \varphi_4 \leq 1$, $1 \leq \varphi_5 \leq 200$, $0.1 \leq \varphi_6 \leq 0.01$, $0.1 \leq \varphi_7 \leq 0.01$, $0.1 \leq \varphi_8 \leq 0.01$

Table 3. Pseudo Code for LFPSO

Algorithm 2: LFPSO Algorithm

```
Initialize the parameters (Number of Particles, Bounds, Max_iter, c1,
c2, V_min, V_max)
trail = 0 (Set the limit for each particle)
Randomly initialize the positions of all particles X_i =
{X_i1, X_i2, X_i3, ...., X_iD}
Evaluate the initial fitness value
Set the initial fitness value as the pbest
Set the particle with the lowest fitness as the gbest
While iter < Max _Iter do
   for i=1:No of Particles
      if trail(i) < limit
         Update the velocity V_i of particle
         Update the position X_i of particle
      else
         trail(i) = 0
         Update the particle's position using Levy_Flight mechanism
         Handle new position if it is outside the bounds
      end if
      Evaluate the fitness value
      if X_i is better than pbest_i
         trail(i) = 0
         Set pbest_i = X_i
      Else
         trail(i) = trail(i) +1
      end if
      If X_i is better than gbest
         Set gbest = X_i
      end if
   end for
   iter = iter + 1
end while
```

Source: Authors' own creation.

Step 2 The parameters of LFPSO are configured as follows. (i) Swarm size comprising 20 particles was chosen. (ii) The cognitive and social coefficients c_1 and c_2: $c_1 + c_2 \geq 2$ (iii) The bounds of the search space are defined according to the ranges specified in Step 1. (iv) The maximum iterations for the algorithm was set to 20. (v) A threshold value, termed the trail limit, was set to 10.

Step 3 The initial position of each particle are randomly selected inside the search space for the five hyperparameters (φ_1 to φ_8).

Step 4 The fitness of each paricle is measured using the Root Mean Squared Error (RMSE).

Step 5 The beginning position of each particle is assigned as its personal best (pbest). The particle with the lowest fitness value is chosen as the global best (gbest).

Table 4. Optimised Hyperparameters are highlighted in BOLD

Hyperparameter	Searching Bound
Epoch	[1-500] **297**
Number of encoder layers	[1-5] **1**
Number of decoder layers	[1-5] **2**
Activation Function	[ReLU,LeakyReLU] **ReLU**
Batch Size	[16,32,64,128,256] **32**
Filters in ConvLSTM Layer	[32,64,128] **64**
Kernel Size	[1x3,1x5]**1x3**
Hidden neurons in Decoder Layer 1	[1-500] **298**
Hidden neurons in Decoder Layer 2	[1-500] **401**
Hidden neurons in Dense Layer	[1-200] **141**
L2 Regularizer λ in Encoder Layer	[0.1-0.01] **0.007**
L2 Regularizer λ in Decoder Layer 1	[0.1-0.01] **0.002**
L2 Regularizer λ in Decoder Layer 2	[0.1-0.01] **0.009**
Dropout Rate in Encoder Layer	[0-1] **0.30**
Dropout Rate in Decoder Layer 1	[0-1] **0.033**
Dropout Rate in Decoder Layer 2	[0-1] **0.092**

Source: Authors' processing.

Step 6 A trail counter is implemented for all particles. If the particle's fitness does not improve relative to its pbest throughout iteration, this counter increases by one.

Step 7 The trail limit of each particle is checked. The standard PSO will update a particle's velocity and position using (2-4) if its trail counter has not exceeded the limit.

$$V_{i,d}^{t+1} = \omega^t V_{i,d}^{t+1} + c_1 r_1(\text{pbest}_{i,d}^t - X_{i,d}^t) + c_2 r_2(gbest_d^t - X_{i,d}^t) \tag{2}$$

$$\omega = (Max\_iter - iter)/ Max\_iter \tag{3}$$

$$X_{i,d}^{t+1} = X_{i,d}^t + V_{i,d}^{t+1} \tag{4}$$

Step 8 If a particle fails to improve its $pbest_i$ over a certain number of iterations and exceeds the trailing limit, the Levy flight distribution function updates its position using (5-6).

$$X_i(t+1) = T\,P + \alpha_{1\text{Neighbour}} + \text{rand}() \times \alpha_2 \times \frac{(T\,P + \propto_3 X_{\text{Leader}})}{2} - X_i(t) \tag{5}$$

$$X_{New}^i(t+1) = LevyFlight(X_i(t+1), T\,P, LB, UB) \tag{6}$$

Step 9 The proposed deep learning model is trained, and the resulting RMSE is used to evaluate particle fitness.

Step 10 Each particle's fitness will be compared to pbest. The fitness value will be $pbest_i$ if it is better than the particle's best position.

Step 11 If a particle's better position is better than the global best, *gbest* becomes $pbest_i$.

Step 12 Repeat steps 7-11 until reaching Max iter.

Step 13 The DL model uses optimal hyperparameters φ_1 -φ_8 obtained from LFPSO.

2.2 Development of the Proposed ConvLSTM-BiLSTM Architecture

A fully connected vanilla LSTM is capable of capturing the temporal characteristics of time-series data, but it lacks the capability to interpret the spatial characteristics of the data. In order to address this problem, we propose a ConvLSTMBiLSTM based encoder-decoder architecture that can effectively capture the spatio-temporal characteristics (Mohammad et al. 2023).

The ConvLSTM model overcomes vanilla LSTM limitations in understanding the spatio-temporal dynamics within the input sequences by utilising its convolutional principles. To learn the spatio-temporal dynamics, the proposed model comprises two modules, an encoder module and a decoder or prediction module. This encoder-decoder architecture allows us to capture complex relationships within the EVCS and predict future charging demands accurately. The spatio-temporal encoder model captures charging station spatial correlations, whereas the temporal decoder model forecasts energy demand using historical data. The encoder's role is to encode the input sequence's temporal and spatial information into spatio-temporal feature vectors. The equations (7-12) describe the operations of ConvLSTM,

$$i_t = \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \tag{7}$$

$$f_t = \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \tag{8}$$

$$\tilde{C}_t = tanh(W_{xc} * X_t + W_{hc} * H_{t-1} + b_c) \tag{9}$$

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{10}$$

$$o_t = \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \tag{11}$$

$$H_t = o_t \circ tanh(C_t) \tag{12}$$

where i_t, f_t, and o_t represent the input, forget, and output gates, respectively. C_t is the cell state, updated using the candidate cell state $\tilde{C}_t$, while Ht is the hidden state. The functions σ and tanh control the flow of information, capturing both spatial and temporal dependencies. Following the preprocessing procedures, the time series data is structured as a 3D array [samples, sequences, features]. As required by the ConvLSTM layer, this 3D array is reshaped into a 5D array [samples, timesteps, rows, columns, channels]. The encoder section uses a single ConvLSTM layer with 1 × 3 kernels and 64 filters. The ConvLSTM produces feature vectors, which are subsequently passed to the flatten layer which reorganises them into a one-dimensional array. After flattening, the resultant 1D array is passed via the repeat vector layer, which duplicates the feature vectors seven times to reproduce into a 2D array. This replicated sequence is then fed into the decoder module having two BiLSTM layers with 298 and 401 neurons. The BiLSTM architecture utilises a backward-propagating LSTM to analyse sequence input, capturing information from both earlier and later time steps, enhancing the model's understanding of temporal dynamics (Schuster & Paliwal, 1997). Table 4 shows the best hyperparameters set based on several experiments and LFPSO optimisation. The rectified linear unit (ReLU) is used as an activation function for all layers to introduce non-linearity to the architecture. The decoder generates individual predictions for each day in the output sequence. A single final output covering the week's prediction is then produced using a time distributed fully connected dense layer. The proposed architecture has two dense layers. The first dense layer has 141 hidden neurons with ReLU activation function and operates on each day of the decoder's sequence independently. The second dense layer has a single neuron and acts on each day. The final output from the second layer represents the predicted energy demand for a single day. These individual day predictions are then aggregated with the subsequent six-day predictions from the decoder sequence to produce the final forecast for the entire week. Two regularisation methods improved model performance and training. The first strategy, Dropout, randomly deactivates % of hidden neurons throughout training iterations.

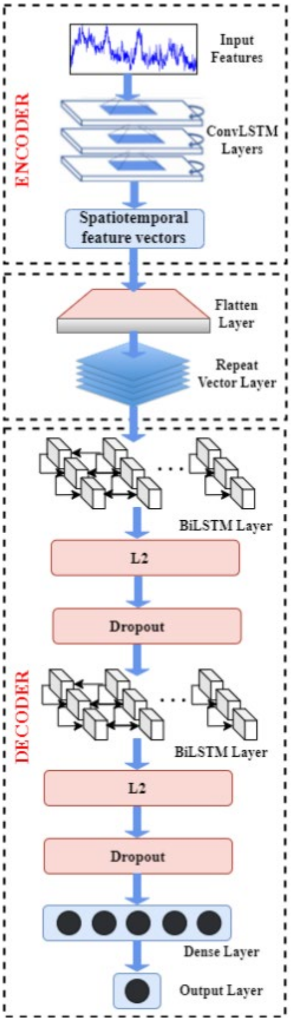

Figure 4. Proposed ConvLSTM-BiLSTM based Encoder-Decoder Architecture

Source: Authors' own creation.

L2 regularisation, also known as weight decay was the second method. This technique penalises the high value weights and thus reduces the variance. Both these regularisation techniques are used to prevent overfitting of model. The proposed encoder-decoder architecture used for load forecasting is given in Figure 4. The training process was improved by using the Adam optimiser with learning rates ranging from $1e^{-4}$ to $1e^{-2.}$ Also, the model was tested with 16, 32, 64, 128, and 256 batches. Extensive trials showed that a batch size of 32 with a learning rate of $1e^{-4}$ performed better in terms of time complexity and loss function optimisation.

3. Data Analytics

This section describes the load forecasting dataset preprocessing methods. The energy consumption trends in Palo Alto, California, over 10 years are used in this research. The Palo Alto dataset describes electric vehicle charging transactions in detail (City of Palo Alto, 2020). The dataset includes 2,59,415 charge transactions from 2011 to 2020 with 33 data fields each. The target variable for our prediction model is the energy demand in kWh.

3.1 Data Preprocessing

Data acquired through the communication infrastructure may be in many sources and unstructured formats, hindering analysis. Data preprocessing removes incorrect or corrupted entries to improve model training. Outliers and missing values in unprocessed data could negatively impact model performance. For best results, data-driven models need structured, high-quality data.

1) Handling Missing Values and Outliers: Real-world datasets with missing values and outliers require robust analytical methods for reliable results. The Palo Alto dataset includes nine missing entries in the “Port type” field. Mode imputation fills missing data with the most repeated data. Outliers are the data points that differ from the majority, and can negatively impact data driven methods. This study detects outliers using Interquartile Range (IQR). Outliers are 1.5 IQR below the first quartile (Q1) and above the third quartile (Q3). After detection, the outliers are linearly interpolated.

2) Feature Engineering: Feature engineering is the process of selecting and extracting useful features. Feature engineering directly enhances the model’s performance. We extracted the hour component from the ’Start Date’ feature and categorised it into ’Off-Peak’ and ’Peak’ intervals. The categorisation is based on the definitions provided by the U.S. Energy Information Administration (EIA), where peak hours are from 7:00 a.m. to 11:00 p.m. on weekdays, and off-peak hours are from 11:00 p.m. to 7:00 a.m. on weekdays and all day on weekends (Blink Charging, n.d.). The resulting data includes the number of transactions during both peak and off-peak hours each day. This helps to assess how peak and off-peak hours affect charging behaviour. Correlation analysis is also used to assess the linear relationships between features to identify the most informative features of the dataset.

3) Data resampling and Standardisation: This study downsamples hourly energy consumption data to daily totals. This reduces data points but provides a suitable format for daily prediction. Downsampling is done using Pandas resample() method. Using resample('D') along with sum(), date-time indexed data is grouped by day. The energy consumption and other feature values are added up to create a daily aggregated data. Figure 5 illustrates the visualisation of resampled daily data points of features over the years 2016 to 2019. Data normalisation after resampling ensure consistent numerical scales across features. The mathematical formulation for standard scaler is presented in (13).

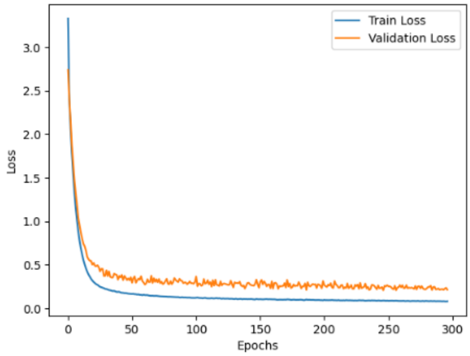

Figure 5. Convergence of losses over epochs

Source: Authors' processing.

$$z = \frac{(x - \mu)}{\sigma} \tag{13}$$

where x, μ, σ, and z represents the original data value, the mean value, standard deviation of the feature, and the normalised value respectively.

3.2 Experimental Results and Discussion

The Palo Alto dataset was used for evaluating the performance of proposed DL model. The pre-processed dataset is split into train and test arrays. The train array covers the years 2016-2018, while the test data is from the year 2019. The train and test arrays are split into standard weeks, starting from Sunday to Saturday, which is required for weekly predictions. The details of train-test split were given in Table 5. The model was trained using the train data to learn the complex patterns. The performance of the model was monitored during training, as shown in Figure 8, where the blue curve indicates the training loss and the orange curve indicates the validation loss (MSE). The epochs were set to 500, concluded at 297 epochs using an early stopping technique. Beyond epoch 297, the losses remained constant, signifying the model's successful convergence. Lower values of both curves indicate better model performance. Upon completion of the training phase, the model's effectiveness was assessed using a separate set of unseen test data. Several metrics were used to evaluate the proposed model's performance with the test data, including Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Squared Error (MSE). To justify the superiority of the proposed model over other conventional approaches, this study conducted experiments using several baseline models, such as LSTM, BiLSTM, CNNLSTM, CNN-BiLSTM, LSTM-LSTM, LSTM-BiLSTM, and BiLSTM-BiLSTM, on the EVCSN dataset. Table 6 compares the proposed model's performance against different baseline models.

Table 5. Details of the Train-Test Period

Data	From	To	Number of Weeks	Number of Days	Percentage of Data%
Train	Jan 03 2016	Dec 29 2018	156	1092	75
Test	Dec 29 2018	Dec 28 2019	52	364	25

Source: Authors' processing.

Table 6. Comparison Results of Proposed Model Vs Other Baseline Models

Metric	LSTM	BiLSTM	CNN +LSTM	CNN +BiLSTM	LSTM +LSTM	BiLSTM +BiLSTM	LSTM +BiLSTM	Proposed Method	Impro-vement%
RMSE	0.7078	0.7316	0.4991	0.5585	0.6350	0.6169	0.6581	**0.4463**	**10.57**
MAPE	135.104	130.794	90.190	76.122	85.105	76.109	79.741	**69.35**	**8.87**
MSE	0.5010	0.5352	0.2491	0.3119	0.4332	0.3805	0.4332	**0.1991**	**20.07**
MAE	0.6190	0.6380	0.3785	0.4410	0.5571	0.5173	0.5571	**0.3378**	**10.72**

Source: Authors' processing.

Table 7. Comparison Results of Proposed Model Vs other Paper Models

Metric	(Mohammad et al., 2023) ConvLSTM-BiLSTM	(Mohammad et al., 2023) BiConvLSTM-LSTM	Proposed Method	Improvement %
RMSE	0.71	0.73	**0.4463**	**37.14**
MAPE	183.16	245.81	**69.35**	**62.13**
MSE	0.878	0.87	**0.3378**	**61.17**

Source: Authors' processing.

Furthermore, a comparative analysis was carried out between the proposed model in this study and one previously published paper (Mohammad et al. 2023) which employed identical architectures, including ConvLSTM-BiLSTM as model I and BiConvLSTM-LSTM as model II to forecast the forthcoming 7-day load of EVCS. The comparison showed considerable improvements in critical metrics including RMSE, MAPE, and MSE, exhibiting respective reductions of around 37.14%, 62.13%, and 61.17%. The comparative results were illustrated in Table 7, indicating clearly that our proposed model outperformed the models presented in (Mohammad et al., 2023). Figure 7 presents actual vs. predicted graphs of the proposed model across each step as well as an aggregated perspective by combining all seven steps. Significantly, the figure illustrates a more precise convergence of the actual and predicted values for the suggested model, implying an enhanced level of accuracy compared to the baseline models. A comparison of the proposed model's evaluation metrics at each step and when aggregated is displayed in Figure 9. The effectiveness of our model in capturing the spatio-temporal dynamics of EVCS for precise short-term load forecasting is confirmed by these meticulous comparisons. The deep learning models were developed utilising the Keras frontend library together with the TensorFlow back-end framework. Experimental results indicate that the proposed DL model outperforms every benchmark model, including well-established, state-of-the-art methods and alternative encoder-decoder architectures. The improved performance of the proposed approach is mainly due to the use of the Levy Flight Particle Swarm Optimisation (LFPSO) algorithm to optimise hyperparameters in addition to extensive preprocessing.

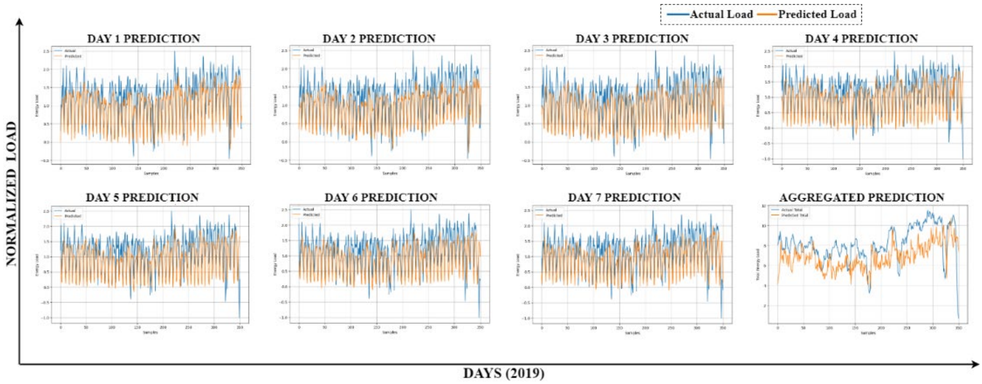

Figure 6. Comparison graphs of the proposed model across each step and aggregate
Source: Authors' processing.

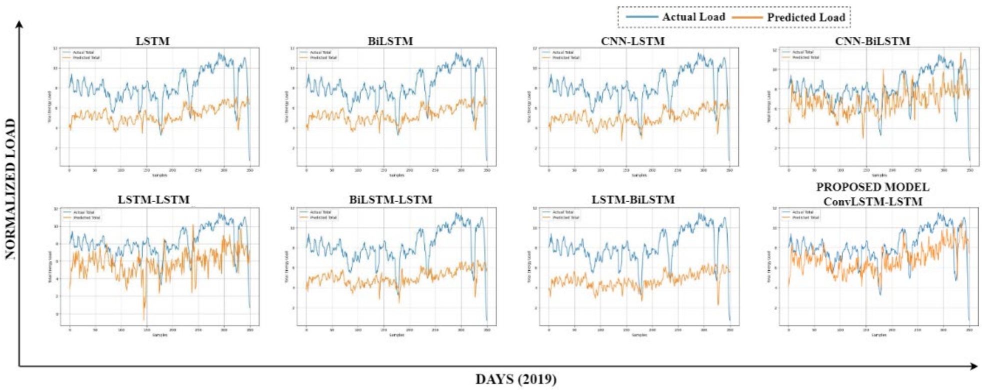

Figure 7. Comparison graphs of proposed model vs .other baseline models
Source: Authors' processing.

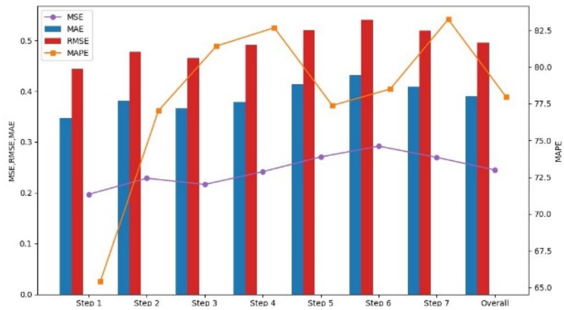

Figure 8. Comparison of metrics across each step and aggregated
Source: Authors' processing.

4. Conclusions

The important problem associated with energy demand forecast in the Electric Vehicle Charging Stations (EVCS) Network was addressed in this paper. We developed an innovative encoder-decoder network to efficiently capture the spatio-temporal dynamics of these networks. Its performance consistently outperformed that of established benchmark models. Superior performance in short-term load forecasting for the upcoming seven days is achieved by our model, which was trained using a real-world Palo Alto dataset. Meticulous preprocessing involving feature extraction and selection and the application of Levy Flight Particle Swarm Optimisation (LFPSO) for hyperparameter optimisation greatly contribute to the significant performance of the forecasting model proposed. Accurate forecasting of energy loads in EVCS is useful for various stakeholders. This forecast can be utilised by utility providers, grid administrators, and EVCS managers in order to optimise energy consumption, maintain the stability of the grid, and control peak demands. The accuracy of the proposed model ensures a sustainable and efficient electric vehicle charging infrastructure. This study provides opportunities for further investigation. Incorporating additional features can improve forecast accuracy. This study encourages researchers to experiment with the suggested methods using various time horizons. Moreover, a promising field is the study of the effectiveness of various deep learning networks, particularly transformers with attention mechanisms. In conclusion, using this knowledge to explore reinforcement learning approaches for real-time load forecasting may be productive.

References

[1] Akshay, K.C., Grace, G.H., Gunasekaran, K., Samikannu, R. (2024), *Power consumption prediction for electric vehicle charging stations and forecasting income. Scientific Reports,* 14(6497), https://doi.org/10.1038/s41598-024-56507-2.

[2] Blink Charging. (n.d.), *Day or night: When is the best time to charge your EV?* Blink Charging, https://blinkcharging.com/blog/day-or-night-when-is-the-best-time-to-charge-your-ev.

[3] City of Palo Alto. (2020), *Electric vehicle charging station usage (July 2011–December 2020)* [Data set]. City of Palo Alto, https://data.cityofpaloalto.org/dataviews/257812/electric-vehicle-charging-station-usage-july-2011-dec-2020/.

[4] Dabbaghjamanesh, M., Moeini, A., Kavousi-Fard, A. (2021), *Reinforcement learning-based load forecasting of electric vehicle charging station using Q-learning technique*. *IEEE Transactions on Industrial Informatics,* 17(6), 4229-4237, https://doi.org/10.1109/TII.2020.2990397.

[5] Habbak, H., Mahmoud, M., Metwally, K., Fouda, M.M., Ibrahem, M.I. (2023), *Load forecasting techniques and their applications in smart grids*. *Energies,* 16(3), Available: https://www.mdpi.com/1996-1073/16/3/1480.

[6] Franco, F.L, Ricco, M., Cirimele, V., Apicella, V., Carambia, B., Grandi, G. (2023), *Electric vehicle charging hub power forecasting: A statistical and machine learning based approach. Energies,* 16(4), 2076, https://doi.org/10.3390/en16042076.

[7] Haklı, H., Uğuz, H. (2014), *A novel particle swarm optimization algorithm with Lévy flight*. *Applied Soft Computing,* 23, 333-345, https://doi.org/10.1016/j.asoc.2014.06.034.

[8] Huang, X., Wu, D., Boulet, B. (2023), *Metaprobformer for charging load probabilistic forecasting of electric vehicle charging stations. IEEE Transactions on Intelligent Transportation Systems,* 24(10), 10445-10455, https://doi.org/10.1109/TITS.2023.3265824.

[9] Huttel, F., Peled, I., Rodrigues, F., Pereira, F. (2021), *Deep spatio-temporal forecasting of electric vehicle charging demand*. *axXiv*, https://arxiv.org/abs/2106.10940.

[10] Huttel, F.B., Rodrigues, F., Pereira, F C. (2023*), Mind the gap: Modeling difference between censored and uncensored electric vehicle charging demand*. *Transportation Research Part C: Emerging Technologies,* 153, 104189, https://doi.org/10.1016/j.trc.2023.104189.

[11] Jubieras, G., Yunta, R., Garcia Moreno, J., Mendivil, C. (1999), *A new ARIMA model for hourly load forecasting. In 1999 IEEE Transmission and Distribution Conference*, 1, 314–319, https://ieeexplore.ieee.org/document/755371.

[12] Kim, Y., Kim, S. (2021), *Forecasting charging demand of electric vehicles using time-series models*. *Energies,* 14(5), 1487, https://doi.org/10.3390/en14051487.

[13] Li, C., Liao, Y., Sun, R., Diao, R., Sun, K., Liu, J., Zhu, L., Jiang, Y. (2023), *Prediction of EV charging load using two-stage time series decomposition and deepBiLSTM model*. *IEEE Access,* 72925-72941, https://doi.org/10.1109/ACCESS.2023.3265971.

[14] Li, Y., Huang, Y., Zhang, M. (2018), *Short-term load forecasting for electric vehicle charging station based on niche immunity lion algorithm and convolutional neural network. Energies,* 11(5), 1253, https://doi.org/10.3390/en11051253.

[15] Marzbani, F., Osman, A.H., Hassan, M.S. (2023), *Electric vehicle energy demand prediction techniques: An in-depth and critical systematic review*. *IEEE Access,* 11, 96242–96255, https://doi.org/10.1109/ACCESS.2023.3322593.

[16] Mohammad, F., Kang, D.-K., Ahmed, M.A., Kim, Y.-C. (2023), *Energy demand load forecasting for electric vehicle charging stations network based on ConvLSTM and BiConvLSTM architectures*. *IEEE Access,* 11, 67350-67369, https://doi.org/10.1109/ACCESS.2023.3264823.

[17] Nespoli, A., Ogliari, E., Leva, S. (2023), *User behavior clustering based method for EV charging forecast*. *IEEE Access,* 11, 6273-6283, https://doi.org/10.1109/ACCESS.2023.3321978.

[18] Park, D.C., El-Sharkawi, M.A., Marks, R.J., Atlas, L.E., Damborg, M.J. (1991), *Electric load forecasting using an artificial neural network. IEEE Transactions on Power Systems,* 6(2), 442-449, https://doi.org/10.1109/59.76685.

[19] Rasheed, T., Bhatti, A.R., Farhan, M., Rasool, A., El-Fouly, T.H.M. (2023), *Improving the efficiency of deep learning models using a supervised approach for load forecasting of electric vehicles*. *IEEE Access,* 11, 91604-91619, https://doi.org/10.1109/ACCESS.2023.3323187.

[20] Rasoulinezhad, E. (2022), *Role of green finance in improving energy efficiency and renewable energy development. Energy Efficiency,* 15(14), https://doi.org/10.1007/s12053-022-10021-4.

[21] Ravi, S.S., Aziz, M. (2022), *Utilization of electric vehicles for vehicle-to-grid services: Progress and perspectives. Energies,* 15(2), https://www.mdpi.com/1996-1073/15/2/589.

[22] Schuster, M., Paliwal, K.K. (1997), *Bidirectional recurrent neural networks*. *IEEE Transactions on Signal Processing,* 45(11), 2673-2681, https://doi.org/10.1109/78.650093.

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

[24] Yang, Z., Hu, T., Zhu, J., Shang, W., Guo, Y., Foley, A. (2023), *Hierarchical high-resolution load forecasting for electric vehicle charging: A deep learning approach. IEEE Journal of Emerging and Selected Topics in Industrial Electronics,* 4(1), 118-127, https://doi.org/10.1109/JESTIE.2023.3267890.

[25] Zhang, J., Yan, J., Liu, Y., Zhang, H., Lv, G. (2020), *Daily electric vehicle charging load profiles considering demographics of vehicle users*. *Applied Energy,* 274, 115063, https://doi.org/10.1016/j.apenergy.2020.115063.