Time Series EV Charging Load Forecasting using Seq2Seq and BiLSTM

Jashanpreet Singh
Netaji Subhas University of Technology
New Delhi, India
jashan.preet.ug23@nsut.ac.in

Veer Rathi
Netaji Subhas University of Technology
New Delhi, India
veer.rathi.ug23@nsut.ac.in

Yash Satsangi
Netaji Subhas University of Technology
New Delhi, India
yash.satsangi.ug23@nsut.ac.in

Abstract--- **Forecasting the charging power consumption of Electric Vehicles (EVs) is increasingly critical to keeping a grid stable and efficient. The study compares the Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM) with a new method, Residual Sequence to Sequence (Seq2Seq) Model for forecasting. Furthermore, Principal Component Analysis (PCA) is applied to decrease the computational demand of models by reducing dimensions. The Residual Seq2Seq model enhances the conventional Seq2Seq architecture by integrating residual connections, enabling it to more effectively capture dependencies over time and mitigate the vanishing gradient issue. Experimental findings demonstrate that this approach facilitates accurate predictions of time-series EV charging load data, characterized by complex temporal patterns. Secondly, PCA analysis when performed on the input data shows that this leads to some gain in model accuracy. Certainly, the Residual Seq2Seq model stands out now, especially when pca-transformed data is used - such a low rank compared to counterparts in LSTM and BiLSTM. In-depth comparative analysis was conducted to specify the necessity of advanced neural network architectures, and dimensionality reduction techniques for performing accurate EV charging intensity forecasting. The research results suggest some useful implications for intelligent grid management and smooth integration of EVs within a smart grid setup.**

Keywords—LSTM; BiLSTM; Seq2Seq; Electric vehicles; Machine learning; EV Power station load forecasting

I. Introduction

Accurately predicting the energy consumption for electric vehicles (EVs) is crucial to guide the planning of new charging stations and the associated power distribution networks. It also helps in identifying potential risks [1] and formulating effective charging and discharging strategies [2] to achieve multiple objectives. EVs can be powered by renewable energy sources or conventional power plants [3]. The use of renewable energy sources is entirely carbon-free [4], while conventional power plants, despite emitting carbon dioxide, enhance the efficiency of fossil fuel utilization compared to burning fossil fuels directly.

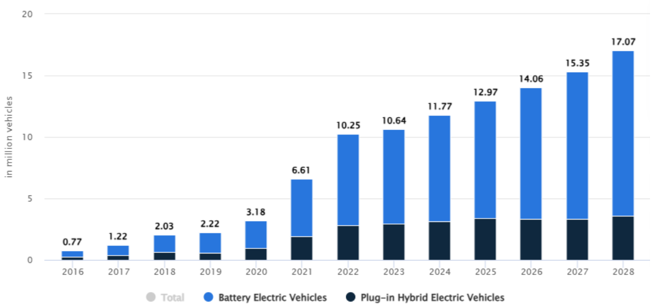

Fig. 1. Increasing Global Adoption of EVs, 2016-2028

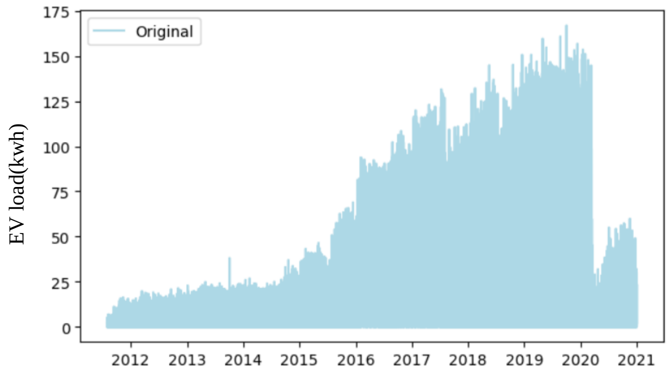

Fig. 2. EV Charging Station load Dataset Plot

Electric power consumption forecasting techniques can be broadly classified into classical quantitative approaches and modern artificial intelligence method [5]. Classical techniques feature models including ARIMA [6] and various regression methods [7]. Alternatively, artificial intelligence approaches span a variety of techniques including neural network [8], support vector machine [9], and deep learning [10]. Recently, advancements in computing hardware, like graphics processing units [11], along with notable successes such as AlphaGo [12], have significantly increased the focus on deep learning. This technique is now widely utilized in fields like image processing [13], natural language processing [14], and other scientific and technical areas [15]. Deep learning models [16] are characterized by their complex network structures, which include many hidden layers and recursive components [17]. Compared to conventional artificial neural networks, deep learning methods offer enhanced learning and adaptability, which makes them particularly valuable in the domain of load forecasting.

Errors in long-term forecasts of EV charging loads can lead to substantial issues in making decisions about the development of charging infrastructure. Similarly, short-term uncertainties in EV charging load predictions can affect the effectiveness of coordinating EVs with renewable energy sources. Generally, two primary challenges exist in EV load forecasting. Firstly, the charging processes are often inadequately accounted for in forecasting models. Secondly, the current forecasting methods still fall short of achieving optimal performance.

To manage these complexities, this paper presents a comparative study of forecasting models using encoder-decoder LSTM or Seq2Seq and BiLSTM algorithms. The research includes an optimized feature engineering process to extract core elements such as periodic patterns, historical patterns over time, seasonal variations, and residual trends.

Figure 3 illustrates these features graphically.

II. PROPOSED METHODOLOGY

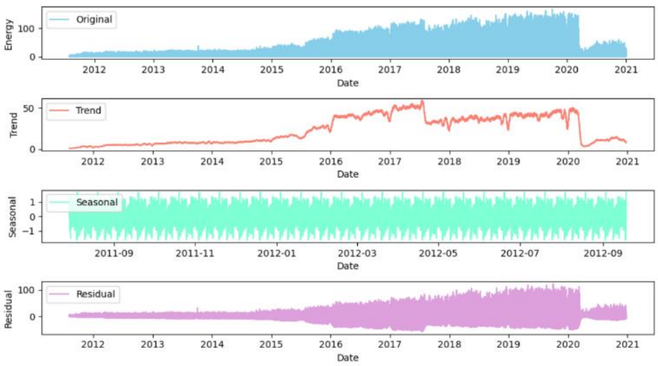

Fig. 3. Energy Time Series Decomposition (Yearly)

A. *Methodological Framework*

The approach outlined in this research employs a comprehensive comparative analysis approach [23], by comparing LSTM, BiLSTM, and Seq2Seq models. This design aims to thoroughly capture the essential attributes of EV energy consumption data for accurate prediction. Given that EV energy consumption is influenced by numerous factors, the combined use of Seq2Seq and BiLSTM models offers an improved method for identifying the time-dependent and non-linear relationships among attributes and results.

B. *Description of ML Algorithms*

1) Long Short-Term Memory (LSTM) Model

The Long Short-Term Memory (LSTM) network, an evolution of Recurrent Neural Networks (RNNs), excels in capturing extended dependencies within sequences. The blueprints of the LSTM network is depicted in Fig. 4, primarily consists of Forgetting Gate, Input Gate, and Output Gate. The corresponding mathematical equations are listed below;

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (1)$$

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (4)$$

$$\tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (5)$$

The input gate of neural network is constructed from h_{t-1} and C_{t-1}, which are the outputs from the prior unit, along with the current input X_t. The sigmoid function σ serves as the transfer function, generating an output vector that acts as the forgetting gate. This output vector is then utilized for the previous memory C_{t-1} through a matrix dot product to produce a new memory value C_t. Here, i_t denotes input gate, f_t denotes forgetting gate, c_t denotes current state storage unit, o_t denotes output gate, and h_t denotes the mid-level output state. Additionally, W denotes weight matrix, while b_i, b_f, b_c and b_o refers to the bias vectors.

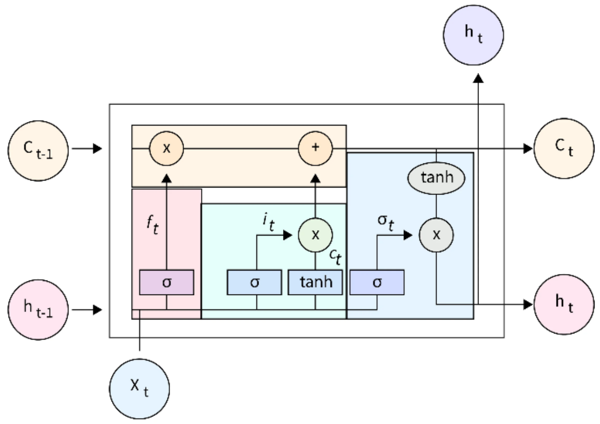

Fig. 4. Flow Diagram representing LSTM Algorithm

2) Bi-Directional LSTM (BiLSTM) Model

BiLSTM is an advanced variant of the traditional LSTM architecture. It involves two parallel LSTM networks: one analyzes the data in a forward direction, while the other analyze it backward. This dual approach is especially helpful for tasks in natural language processing (NLP). The blueprints of BiLSTM is represented in Fig. 5.

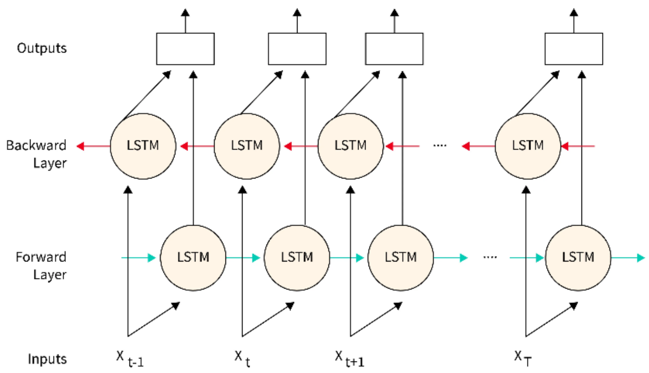

Fig. 5. Flow Diagram representing BiLSTM Algorithm

3) Sequence to Sequence LSTMs or RNN Encoder-Decoders

A Seq2Seq model includes two primary components: Encoder and Decoder. The Encoder generates a Context Vector, which is then provided to the Decoder. In the context of language translation, the input is a sentence. This sentence is processed by the Encoder, which learns and represents the context of the entire sentence in a Context Vector. Once the Context Vector is learned, it is passed to the Decoder, which translates it into the target language and outputs the translated sentence.

An Encoder is essentially an LSTM network designed to learn representations. The key difference is that, rather than focusing on the output, we utilize the hidden state of the last cell, which encapsulates the context of all inputs. This hidden state acts as the context vector and is then passed to the Decoder.

The Decoder is also an LSTM network, but rather than starting with random values for the hidden state, it uses the context vector. The first input to the Decoder is set to <bos> (beginning of sentence). The output from the first LSTM cell then act as the input for the next cell.

C. Detailed Procedures

Data Preprocessing

The raw data includes several preprocessing steps, including imputation of missing values, normalization, and dataset partitioning. Key features such as weekday, weekend, and seasonal information were extracted from the date. Missing values were addressed by using previous day's data. Then we categorize the dataset as training and testing sets with a 7:3 ratio.

Transform data using PCA

Principal Component Analysis (PCA) is widely utilized for data preprocessing in machine learning applications. It extracts the most informative features from large datasets, preserving the key information from the original data. To standardize the data, PCA is applied to the scaled data, reducing its dimensionality to five principal components (PC). The transformed data is then converted into a NumPy array for further analysis. This dimensionality reduction retains the most significant features, enhancing the model's efficiency and performance. The Pareto and Scree plots of the principal components with their variance ratios are depicted in Fig.6 and Fig. 7 respectively;

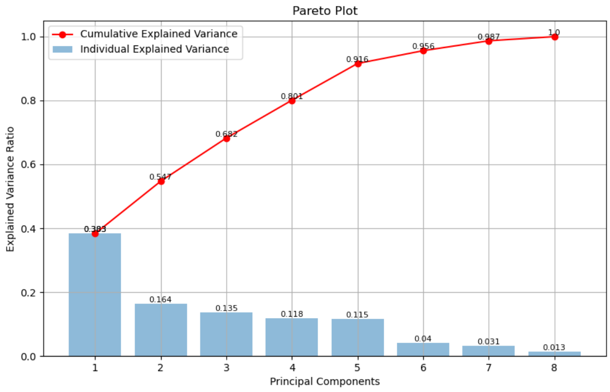

Fig. 6. Pareto Plot of Principal Components

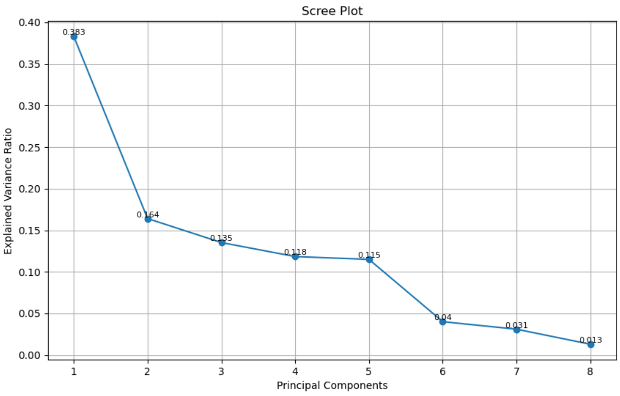

Fig. 7. Scree Plot of Principal Components

Model evaluation index

In this research, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are chosen as metrics for evaluating the performance of the various forecast models. The formulas for these evaluation metrics are provided below;

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_p - y_i| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_p - y_i)^2} \tag{7}$$

D. Key Contribution

In this research, we present a comprehensive comparative analysis of LSTM, BiLSTM, and Seq2Seq models. Utilizing a robust dataset of charging sessions, our study evaluates the predictive accuracy, computational efficiency, and robustness of each model. The results reveal significant insights into the pros and constraints of these deep learning techniques. Notably, the Seq2Seq model demonstrated superior performance in capturing complex temporal patterns, while the BiLSTM provided enhanced accuracy with bidirectional context.

III. Case Study

A. Description of Dataset

The EV Charging Station data provides detailed insights into the consumption trends of EV power stations in Palo Alto, California. This publicly available dataset delivers valuable information on the supply, usage, and requirement for EV charging infrastructure. It includes details such as charging station locations, types of available connectors, charging session start and end times, session duration, power consumption, and other essential indicators. The dataset spans from 29-07-2011 to 31-12-2020, and contains a total of 2,59,353 recorded values.

B. Study Results

The forecasted values using various algorithms, along with the comparison for a sample dataset, are displayed in Figures 8, 9, 10 and 11. All algorithms forecasting curves closely align with the trend of the actual energy consumption curve. It is evident that the energy comsumption is nearly zero in morning (from 0:00 to 8:00), peaks around noon (from 10:00 to 14:00), fluctuates afterwards, and then decreases at night. Figure 10 shows that the Seq2Seq LSTM-based model produces lower prediction errors and fits the true charging load curve more accurately, demonstrating its efficiency and superiority over various algorithms.

The forecasting efficiency on the test set is summarized in Table-I. As shown, Seq2Seq algorithms achieve lower RMSE and MAE compared to LSTM and BiLSTM, highlighting the benefits of deep learning techniques. Additionally, the Seq2Seq model yields the least predictive errors, with an RMSE of 0.111 and an MAE of 0.073. Thus, in conclusion we can say that in this research the Seq2Seq model enhances forecast accuracy compared to LSTM and BiLSTM, based on RMSE and MAE as error metrics.

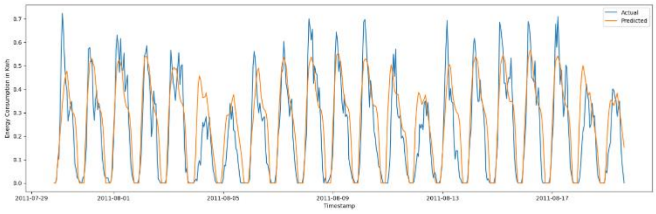

Fig. 8. LSTM Prediction on Dataset

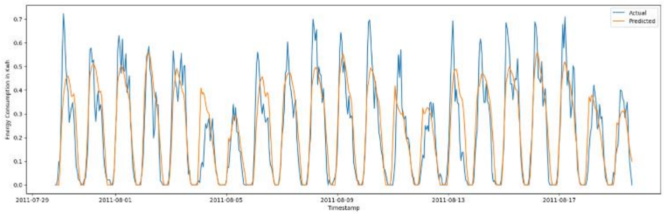

Fig. 9. BiLSTM Prediction on Dataset

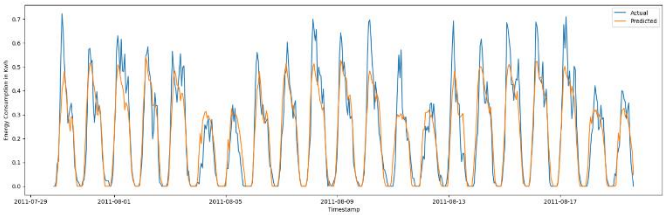

Fig. 10. Seq2Seq Prediction on Dataset

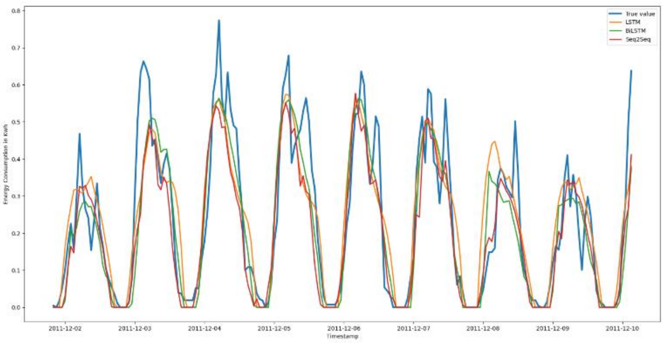

Fig. 11. Prediction by various Models

TABLE 1.

PERFORMANCE COMPARISON OF VARIOUS MODELS

Model	LSTM	BiLSTM	Seq2Seq
RMSE	0.143	0.118	0.111
MAE	0.109	0.080	0.073

IV. DISCUSSION

The outstanding performance of deep learning-based approaches in prediction tasks can be attributed to the "iterative" optimization algorithms they employ, which aim to achieve the best possible results. Iterative in this context means repeatedly obtaining results and selecting the ideal ones, i.e., the cycle that minimizes errors. This process helps transform an under-fitted model into one that is ideally aligned with the data. In deep learning these iterative optimization algorithms typically involve refine model parameters using Gradient Descent here gradient denotes rate of slope inclination, and descent refers to the process of descending. The gradient descent algorithm determines the speed at which the Neural Network components are trained. Various factors might necessitate more iterations when training a network compared to others. A primary reason is when the dataset is too large to be processed by the model all at once. An effective solution is distributing the data among smaller batches and train the model iteratively on these smaller chunks (i.e., batch size and iteration). The weights of the neural network parameters are revised at the conclusion of each iteration to reflect the fluctuation introduced by the newly introduced training set. Additionally, training a network with the same dataset multiple times (i.e., epochs) can further optimize the model parameters.

A. Iteration and Batch Size

If a dataset is very large to be processed by a neural network all together, it can be divided into smaller batches, allowing the network to be trained in multiple stages. The batch size indicates the number of training examples in each batch. Essentially, when a large dataset is split into smaller chunks, each chunk is termed as batch. Iteration, however, denotes the number of batches required to complete one full cycle of training using the entire dataset. In other words, the number of batches equals the number of iterations for one complete round of training. For example, if there are 1,000 training examples, and they are divided into batches of 250, it would take four iterations to train the model on the entire dataset in one round.

B. Epochs

An epoch depicts the overall frequency a dataset is utilized in the training process. Particularly, an epoch means the complete dataset is propagated in both forward and backward directions across the network in a single pass. Due to the face that deep learning algorithms use gradient descent to enhance their models, it is beneficial to pass the entire dataset across the network repeatedly to update the weights and improve the model's accuracy. However, determining the exact number of epochs required to achieve optimal weights is not straightforward. Different datasets have varying characteristics, and therefore, the number of epochs needed for optimal training can vary.

C. The Impact of Epochs

This section focuses on examining how the count of training cycles (epochs) affects the model when using the same dataset. We conducted a set of experiments and sensitivity studies, adjusting the epoch values and recording the error rates. For each dataset, the epoch value was set to 300. This upper limit was chosen for practical reasons and the capability of the experiments. Calculating error percentages across epoch values up to 300 required approximately 5 minutes of CPU time on a moderately performing computing cluster running the Windows operating system.

V. CONCLUSION

In conclusion, this paper presents a comprehensive comparative analysis of LSTM, BiLSTM, and Seq2Seq models for forecasting EV charging load. Our results demonstrate that the BiLSTM model surpasses both the LSTM and Seq2Seq models in prediction accuracy, displaying the lowest prediction error among the three. The superior performance of BiLSTM is attributed to its ability to integrate bidirectional context, thus capturing complex temporal dependencies more effectively. While the Seq2Seq model is adept at handling intricate sequence data, its prediction accuracy was marginally lower than that of the BiLSTM. These findings highlight the potential of BiLSTM and Seq2Seq models in achieving more accurate and reliable EV load forecasts, which is crucial for efficient energy management and maintaining grid stability amidst the growing adoption of EVs.

REFERENCES

[1] J. Brownlee, "How to Create an ARIMA Model for Time Series Forecasting with Python," https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/, 2017.

[2] J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," https://machinelearningmastery.com/time-series-prediction-lstm-recurrent-neural-networks-python-keras/, 2016.

[3] J. Brownlee, "How to develop Bidirectional LSTM for sequence classification in python in Keras" https://machinelearningmastery.com/develop-bidirectional-lstm-sequence-classification-python -keras/, 2021.

[4] J. Brownlee, "How to develop LSTM models for time series-forecasting" https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/, 2020.

[5] J. Brownlee, "How to develop a Seq2Seq Model for Neural Machine translation in Keras" https://machinelearningmastery.com/define-encoder-decoder-sequance-sequence-model-neural-machine-translation-keras /, 2019.

[6] J. Brownlee, "How to develop an Encoder-Decoder model for Sequence to Sequence prediction in Keras" https://machinelearningmastery.com/develop-encoder-decoder-model-sequence-sequence-prediction-keras/, 2020.

[7] Z. Cui, R. Ke, Y. Wang, "Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction," arXiv:1801.02143, 2018.

[8] J. Kim, N. Moon" BiLSTM model based on multivariate time series data in multiple fields for forecasting trading area." Journal of Ambient Intelligence and Humanized Computing, pp. 1-10.

[9] S. S. Namini, N. Tavakoli, and A. S. Namin. "A Comparison of ARIMA and LSTM in Forecasting Time Series." 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394-1401. IEEE, 2018.

[10] Belal Mahmud Fahim, "Deep-learning for ev charging load forecasting" https://github.com/fffahim/Deep-learning-for-electric-vehicle-charging-load-forecasting.

[11] S. S. Namini, N. Tavakoli, and A. S. Namin. "The Performance of LSTM and BiLSTM in Forecasting Time series" 2019 IEEE International Conference on Big Data.

[12] C. Li, Y. Liao, L. Zou, R. Diao, R. Sun and H. Xie. "Short-Term Forecasting of EV Charging Load Using Prophet-BiLSTM" 2022 IEEE Transportation Electrification Conference and Expo, Asia-Pacific (ITEC Asia-Pacific).

[13] X. Huang, Di Wu, B. Boulet. "Ensemble Learning for Charging Load Forecasting of Electric Vehicle Charging Stations" 2020 IEEE Electric Power and Energy Conference (EPEC).

[14] Y. Wang, S. Zhu, C. Li "Research on Multistep Time Series Prediction Based on LSTM" 20220 IEEE.

[15] https://www.ibm.com/topics/principal-component-analysis.

[16] https://www.scaler.com/topics/deep-learning/lstm/.