Enhancing Energy Demand Forecasting: A Comparative Study of LSTM, BiLSTM, and Residual Seq2seq Models for EV Charging Consumption Patterns

Belal Mahmud Fahim (40290617)

Github Link: github.com

Abstract—**The forecasting of charging power load for Electric Vehicles (EVs) has become crucial for ensuring grid stability and efficiency. In this study, the forecasting capabilities of Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), and a novel Residual Seq2Seq model are evaluated. Additionally, Principal Component Analysis (PCA) is employed to enhance model performance through dimensionality reduction. The traditional Seq2Seq architecture is extended by the proposed Residual Seq2Seq model, which incorporates residual connections to better capture long-term dependencies and mitigate the vanishing gradient problem. Experimental results demonstrate the efficacy of this novel approach in accurately predicting the intricate temporal patterns inherent in EV charging load data. Furthermore, significant improvements in model accuracy are observed through PCA analysis when applied to the input data. The Residual Seq2Seq model, particularly when coupled with PCA-transformed data, emerges as the top performer, surpassing LSTM and BiLSTM counterparts. Through comprehensive comparative analysis, the importance of advanced neural network architectures and dimensionality reduction techniques for precise EV charging load forecasting is emphasized. These findings highlight implications for optimizing grid management and facilitating the seamless integration of electric vehicles into the smart grid infrastructure.**

Index Terms—**Principal Component Analysis, Residual Sequence to Sequence, Long Short-Term Memory, Bidirectional LSTM**

I. Introduction

Electric Vehicles (EVs) are currently seen as an opportunity to reduce greenhouse gasses in the transport sector, help address local air pollution, and overcome the uncertainties posed by the dependence on fossil fuel importas spotlighted by recent geopolitical events. Many governments are consequently incentivizing EV use, and some regions of the world have even planned to ban the sale or use of combustion-engine vehicles in the mid-term (by 2025 in Norway, by 2030 in Germany, the UK, and the Netherlands, and by 2040 in France, for instance). More recently, COP26 signatories have agreed to commit to 100% zero emissions transport by 2040 [3].

Furthermore, battery costs are expected to decrease in the coming decades. Therefore, the share of EVs in the transport sector is forecasted to surge in the next few years. EV sales in Europe continued to increase by more than 65% in 2021. The International Energy Agency's 2030 scenario forecasts that half of all vehicle sales in Europe could be EVs by 2030 and stresses the critical importance of careful planning and fostering smart charging [4]. In the case of France, the country has set 2040 as the target year for ending sales of new fossil fuel-powered passenger cars and light commercial vehicles. In 2021, France experienced a significant increase in electric Light Duty Vehicles (LDV) sales, likely due to its ecological bonus program which provides subsidies for Zero Emission Vehicles (ZEV) purchases under its Covid-19 economic package, France Relance, which was extended to mid-2022. The French president has also set a target of 1 million Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) to be produced in France by 2025 [5].

The latest power load prediction models can be broadly classified into traditional statistical models and artificial intelligence (AI) models. Traditional methods encompass time series analysis, autoregressive integrated moving average (ARIMA), regression analysis, Kalman filtering, and others. [7], [8]. Conversely, AI approaches include artificial neural networks (ANNs), support vector machines (SVMs), and deep learning techniques. Prior to the 21st century, ANNs gained prominence in load forecasting due to their robust adaptive, self-learning, and generalization capabilities. A review by Hippert et al. [9] highlighted the effectiveness of ANNs in load forecasting, emphasizing their accuracy and efficiency.

In [10], the authors introduced a hybrid deep neural network comprising a CNN module, an LSTM module, and a feature-fusion module, which they evaluated using 2-year electric load datasets from the Italy-North region. Their model was compared with several other machine learning models, such as Decision Tree, Random Forest, DeepEnergy, LSTM, and CNN, yielding superior performance. Despite achieving promising results with their proposed model, the authors did not subject it to more challenging datasets or test its predictive capabilities across different time horizons.

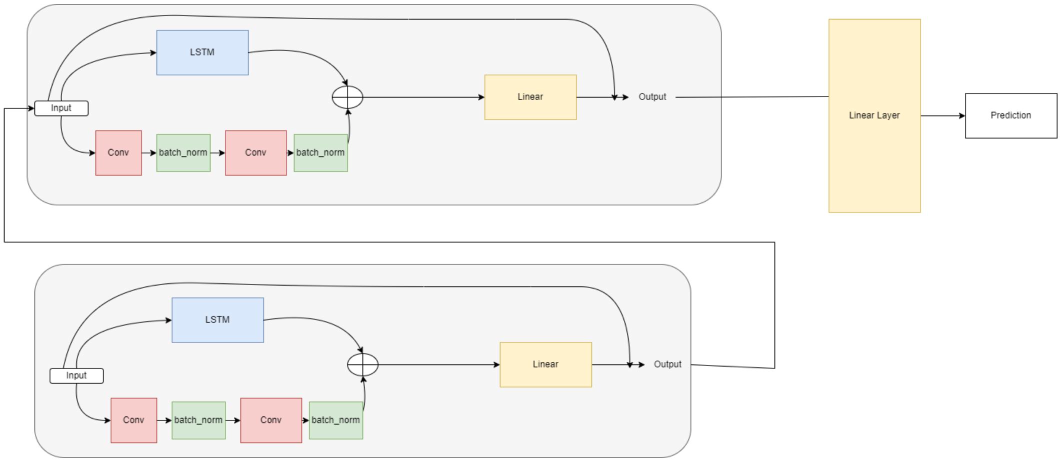

Fig. 1. Residual Seq2seq

The main contributions of this article are listed as follows:

- We propose a new model called Residual Seq2seq combined with Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Residual network for short term load forecasting.
- We examine the influence of Principle Component Analysis (PCA) on forecasting models.

The rest of the report is organized as follows: Section II describes the PCA methodology, Section III gives an overview of the three forecasting algorithms, Section IV provides the EV charging dataset description, Section V discusses about PCA results, Section VI provides experimental results. Finally, in section VII, the conclusion is drawn.

II. Principle Component Analysis

Principal Component Analysis (PCA) is a dimensionality reduction technique used to transform high-dimensional data into a lower-dimensional representation while preserving the most important information [6]. It achieves this by identifying the principal components, which are orthogonal vectors that capture the directions of maximum variance in the data. PCA is widely utilized in various fields such as machine learning, data analysis, and signal processing to simplify complex datasets, remove redundant information, and facilitate visualization and analysis. PCA can be applied to any data matrix using the following steps [1]:

A. Standardization

Standardize the features by subtracting the mean $\bar{X}$ and dividing by the standard deviation σ. This step ensures that all features have a similar scale, preventing variables with larger magnitudes from dominating the analysis.

$$X_{\text{std}} = \frac{X - \bar{X}}{\sigma} \tag{1}$$

- X: Original data matrix.
- $\bar{X}$: Mean vector of the features.
- σ: Standard deviation vector of the features.

B. Compute Covariance Matrix

Calculate the covariance matrix of the standardized data. The covariance matrix captures the relationships between pairs of features and provides insights into how features vary together

$$\text{Cov}(X) = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X}) \tag{2}$$

- X: Original data matrix.
- $\bar{X}$: Mean vector of the features.
- n: Number of samples.

C. Compute Eigenvectors and Eigenvalues

Compute the eigenvectors $\mathbf{v}_i$ and corresponding eigenvalues λ_i of the covariance matrix. Eigenvectors represent the directions of maximum variance in the data, while eigenvalues indicate the magnitude of variance along each eigenvector.

$$\text{Cov}(X)\mathbf{v}_i = \lambda_i \mathbf{v}_i \tag{3}$$

- $\mathbf{v}_i$: Eigenvectors.
- λ_i: Eigenvalues.

D. Select Principal Components

Sort the eigenvectors based on their corresponding eigenvalues in descending order. The top k eigenvectors, known as principal components, capture the most variance in the data and form the basis for dimensionality reduction.

E. Projection

Project the original data onto the subspace spanned by the selected principal components. This projection transforms the data into a lower-dimensional space while retaining the most important information captured by the principal components.

$$X_{\text{proj}} = X_{\text{std}}\mathbf{V}_k \tag{4}$$

- $\mathbf{V}_k$: Matrix of the top k eigenvectors.

III. Deep-learning based algorithms

A. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem in traditional RNNs, which can struggle to capture long-term dependencies in sequential data. LSTM introduces memory cells and various gating mechanisms to selectively retain or discard information over time, allowing it to better handle long-range dependencies. In load forecasting, many studies used LSTM and improved their approaches by finding the dependency within load series data. The following equations show the computation of LSTM in details:

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

Where:

- x_t is the input at time step t,
- h_t is the output at time step t,
- c_t is the cell state at time step t,
- i_t, f_t, and o_t are the input, forget, and output gate vectors at time step t respectively,
- σ denotes the sigmoid function,
- $\odot$ denotes element-wise multiplication,
- W and b are the weight matrices and bias vectors respectively.

B. Bidirectional Long Short-Term Memory (BiLSTM)

Bidirectional Long Short-Term Memory (BiLSTM) is an extension of the LSTM architecture that incorporates information from both past and future contexts. It consists of two LSTM layers: one processes the input sequence in a forward manner, while the other processes it in a backward manner. The outputs of both layers are concatenated to capture information from both directions. The equations governing the BiLSTM architecture are similar to LSTM, but with separate computations for the forward and backward passes.

C. Residual Seq2seq

This section introduces a novel methodology that integrates LSTM, CNN, and residual network architectures. Unlike existing approaches discussed in the introduction, the methodology presented here offers a distinct approach. For instance, in [2], a CNN-LSTM model was proposed where the CNN is utilized to extract input data features, subsequently fed into the LSTM. However, this sequential approach can lead to feature interference affecting LSTM training. In contrast, our model independently extracts information from the input data using CNN and LSTM modules. Each model operates on the input data separately, and their outputs are combined and fed into a linear layer. Additionally, a residual network is employed to address the issue of vanishing gradients. This entire process constitutes a block, and the model integrates multiple such blocks. The output of one block serves as the input for the next. Fig 1 illustrates the framework of the proposed methodology.

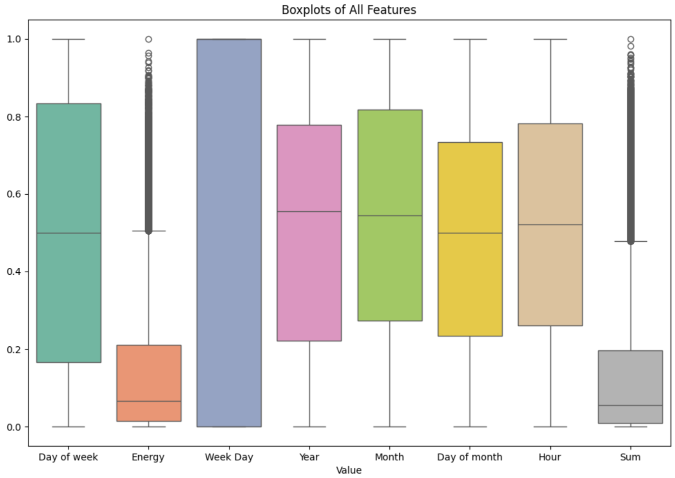

Fig. 2. Box Plot

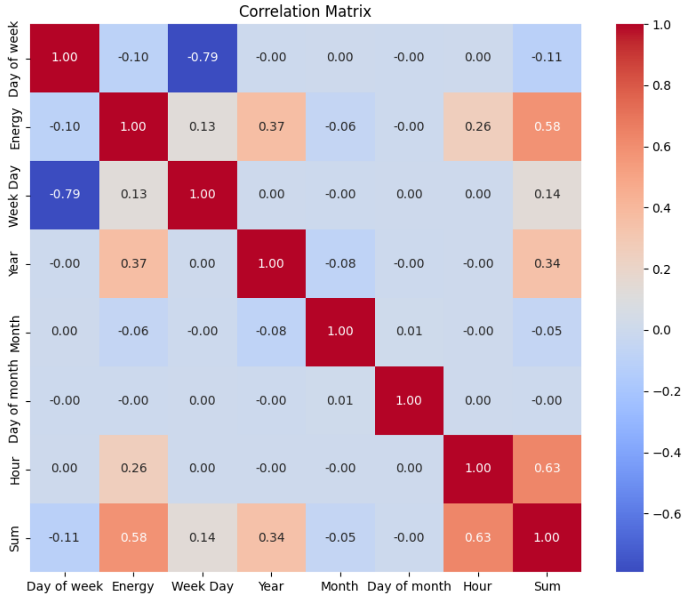

Fig. 3. Correlation Matrix

IV. Evaluation Metrics

A. Root Mean Square Error (RMSE)

The Root Mean Square Error (RMSE) is a widely used metric for evaluating the accuracy of forecasting models. It measures the average magnitude of the errors between predicted and observed values. RMSE is particularly useful because it penalizes large errors more heavily than smaller ones, providing a comprehensive assessment of the model's performance.The RMSE is calculated using the following equation:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

where n is the number of observations, y_i represents the observed values, and $\hat{y}_i$ represents the predicted values. In forecasting models, lower RMSE values indicate better predictive accuracy, as they signify smaller deviations between predicted and actual values.

B. Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is another commonly used metric for assessing the performance of forecasting models. Unlike RMSE, which calculates the square of the errors, MAE computes the absolute differences between predicted and observed values, making it less sensitive to outliers. The MAE is calculated using the following equation

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

where n is the number of observations, y_i represents the observed values, and $\hat{y}_i$ represents the predicted values.

V. Dataset Description

The dataset utilized in this study comprises EV charging session data spanning from July 29, 2011, to December 31, 2020, within the Palo Alto area. Collected from an open-source database, a comprehensive view of EV charging behaviors over nearly a decade is provided by this dataset. It includes essential details such as charging start and end times, energy consumption, Day of week, Week Day, Year, Month, Day of month, and daily cumulative sum.

The existing models were implemented and optimized based on 60-minute intervals of daily time-series patterns, which were normalized using the MinMaxScalar Normalization function. Therefore, the described datasets were also rearranged into day-to-day matrices, downsampled to a 60-minute frequency, and normalized using the same MinMaxScalar function.

Utilizing the boxplot and whiskerplot and their five-number summaries on the dataset, the distributions, central values, and variability of the features were measured. The box plot of the features of the dataset is illustrated in Fig 2. It can be observed that only the energy-related features have points outside the interquartile range; others are more related to datetime. Correlation matrix is a good way to visualize the correlations of the features. The correlations between the features of the dataset are illustrated in Fig 3.

As mentioned before, load consumption data are time series. Thus, to forecast future load consumption, some time-series analyses are needed. Time series have important attributes such as trend, seasonality, etc. Most load series data have all trends, seasonal, noise attributes simultaneously. For instance, decomposition of a seasonal load series data is shown in Fig 4. The original data is depicted in red, the trend is shown in blue, the seasonality of data is displayed in green, and the noise is represented in purple. A library from Python called seasonal - decompose() had been used to decompose all the seasonal data in this article.

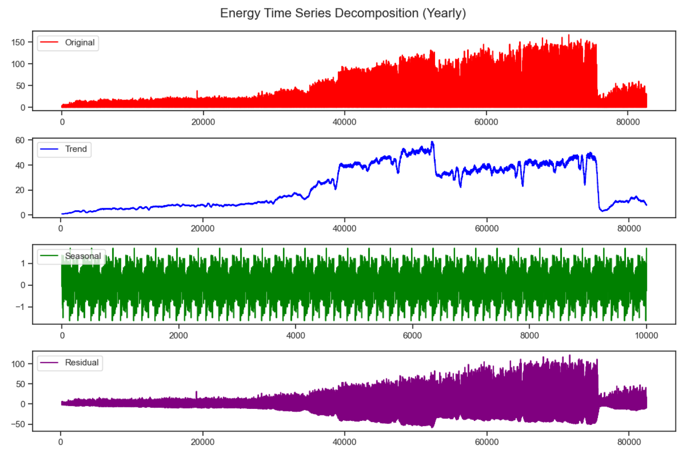

Fig. 4. Decomposition of Seasonal Load Series Data

VI. PCA Results

In this article, PCA is applied to the EV charging dataset using the Python library. By applying PCA steps, the feature set of 8 can be reduced to r numbers of features where r < 8. The original n × p dataset is reduced using the eigenvector matrix A, with each column of A represented by a principal component (PC). Each PC captures an amount of data that determines the dimension (r). The corresponding eigenvalues are:

$$\lambda = \begin{bmatrix} 0.28689099 \\ 0.12285223 \\ 0.10125487 \\ 0.08863008 \\ 0.08610584 \end{bmatrix}$$

Due to its size, the complete eigenvector matrix (A) cannot be included in this document. However, the transposed version of the eigenvector matrix is presented below:

$$A^T = \begin{pmatrix} 0.562 & 0.041 & -0.006 & -0.003 & 0.000 \\ -0.050 & 0.287 & -0.011 & -0.103 & -0.011 \\ -0.823 & -0.043 & 0.007 & 0.004 & -0.000 \\ -0.012 & 0.524 & 0.314 & -0.714 & -0.068 \\ 0.004 & -0.259 & -0.796 & -0.536 & -0.114 \\ -0.001 & -0.014 & -0.051 & -0.129 & 0.990 \\ -0.015 & 0.614 & -0.496 & 0.419 & 0.039 \\ -0.062 & 0.442 & -0.142 & 0.015 & -0.001 \end{pmatrix}$$

Fig 5 and Fig 6 demonstrate the scree plot and pareto plot of the PCs. The scree plot and pareto plot display the amount of variance explained by each principal component. The percentage of variance experienced by the j-th PC can be evaluated using the following equation:

$$j = \frac{\lambda_j}{\sum \lambda_j} \times 100, \quad j = 1, 2, \ldots, p,$$

where λ_j represents the eigenvalue and the amount of variance of the j-th PC.

It is observed from both figures that the variance of the first five PCs contributes to 91.6% of the amount of variance of the original dataset. i.e., the first PC holds 38.4% of variance, the second PC holds 16.4% of variance, the third PC holds 13.5% of variance, the fourth PC holds 11.8% of variance, and the fifth PC holds 11.5% of variance.

The scree plot presents that the elbow is also located on the fifth PC. These observations imply that the dimension of the feature set can be reduced to five ($r = 5$). Based on the provided matrix:

$$\begin{aligned} Z_1 = {} & (0.562)X_1 + (-0.050)X_2 + (-0.823)X_3 \\ & + (-0.012)X_4 + (0.004)X_5 + (-0.001)X_6 \\ & + (-0.015)X_7 + (-0.062)X_8 \end{aligned}$$

Z_1 captures the most variation in the dataset, primarily influenced by features X_1, X_3, and X_8.

$$\begin{aligned} Z_2 = {} & (0.041)X_1 + (0.287)X_2 + (-0.043)X_3 \\ & + (0.524)X_4 + (-0.259)X_5 + (-0.014)X_6 \\ & + (0.614)X_7 + (0.442)X_8 \end{aligned}$$

Z_2 emphasizes features X_2, X_4, X_7, and X_8, while downplaying the influence of features X_3 and X_5.

$$\begin{aligned} Z_3 = {} & (-0.006)X_1 + (-0.011)X_2 + (0.007)X_3 \\ & + (0.314)X_4 + (-0.796)X_5 + (-0.051)X_6 \\ & + (-0.4956)X_7 + (-0.1421)X_8 \end{aligned}$$

Z_3 highlights the impact of features X_4, X_5, X_7, and X_8, with maximum contributions from other features.

$$\begin{aligned} Z_4 = {} & (-0.003)X_1 + (-0.103)X_2 + (0.004)X_3 \\ & + (-0.7143)X_4 + (-0.536)X_5 + (-0.129)X_6 \\ & + (0.419)X_7 + (0.015)X_8 \end{aligned}$$

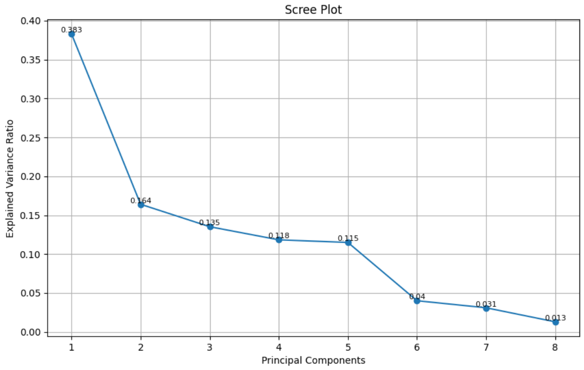

Fig. 5. Scree Plot

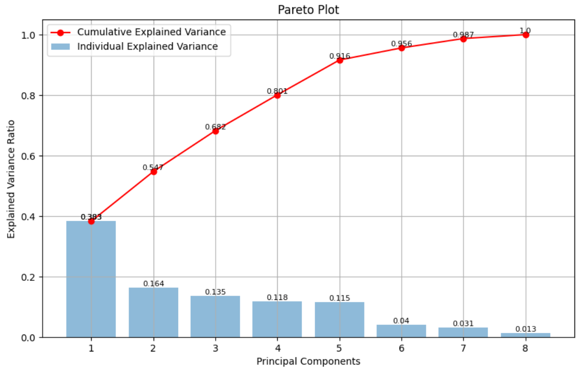

Fig. 6. Pareto Plot

Z_4 is characterized by the strong influence of features X_4, X_5, and X_7.

$$\begin{aligned} Z_5 = {} & (0.0042)X_1 + (-0.011)X_2 + (-0.0004)X_3 \\ & + (-0.068)X_4 + (-0.1144)X_5 + (0.9903)X_6 \\ & + (0.0387)X_7 + (-0.001)X_8 \end{aligned}$$

Z_5 predominantly reflects the variations in features X_5, and X_6, with maximum contributions from other features.

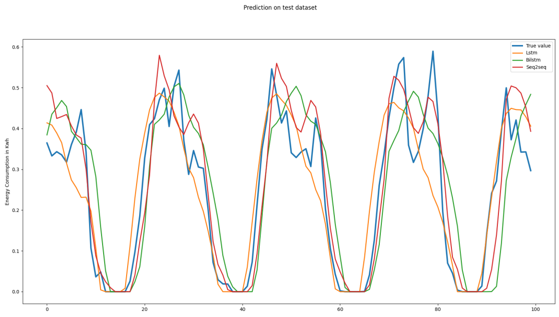

Fig. 7. Predictions on real data

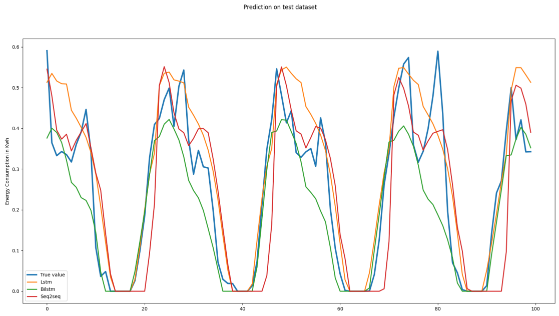

Fig. 8. Predictions on PCA transformed data

TABLE I
RMSE AND MAE VALUES FOR POWER CONSUMPTION FORECASTING

Future step	RMSE					
	Without PCA			With PCA		
	LSTM	Bi-LSTM	Seq2seq	LSTM	Bi-LSTM	Seq2seq
1 hour	0.2980	0.1806	0.1045	0.1513	0.1058	0.1185
5 hours	0.3024	0.2401	0.1731	0.1617	0.1157	0.1310
10 hours	0.3155	0.2772	0.1988	0.1829	0.1376	0.2144

Future step	MAE					
	Without PCA			With PCA		
	LSTM	Bi-LSTM	Seq2seq	LSTM	Bi-LSTM	Seq2seq
1 hour	0.2246	0.1347	0.0556	0.1093	0.0984	0.0856
5 hours	0.2516	0.1901	0.0964	0.1109	0.1078	0.0957
10 hours	0.2540	0.2226	0.1167	0.1192	0.1940	0.1728

TABLE II
MODEL TRAINING CONFIGURATION

Model	Epoch	Lag size (Previous points)	Learning rate	Weight decay
LSTM	400	5, 10	0.0001	0.005
BiLSTM	400	5, 10	0.0001	0.005
Seq2seq	300	5, 10	0.0001	0.005

VII. EXPERIMENTAL RESULTS

In this section, the performance of the three forecasting models, namely LSTM, BiLSTM, and residual Seq2seq, is examined. Optimization of the models is conducted to predict the next 1, 5, and 10 hours of power consumption. To assess the impact of PCA, the forecasting algorithms are implemented on both the original dataset and the PCA-applied dataset.

The original dataset is partitioned into training and testing sets with a ratio of 70% to 30%. Detailed parameters are provided in Table II. The numerical outcomes of the forecasting models are elaborated in Table I. The tables provide a comprehensive overview of the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics for each forecasting model across different future steps, both with and without Principal Component Analysis (PCA) applied to the dataset.

When analyzing the RMSE values, which measure the average magnitude of the forecasting errors, it is evident that incorporating PCA leads to a reduction in error across all models and future steps. This reduction signifies that PCA preprocessing aids in improving the forecasting accuracy of the models. For instance, at a future step of 1 hour, the LSTM model exhibits an RMSE of 0.2980 without PCA, which decreases to 0.1513 with PCA. Similar trends are observed for the Bi-LSTM and Seq2seq models across all future steps. Fig 7 and Fig 8 depicts the prediction outcomes of the forecasting models on both the original dataset and the PCA-transformed dataset.

The MAE values, which represent the average magnitude of the errors without considering their direction, mirror the patterns observed in the RMSE results. The incorporation of PCA consistently results in lower MAE values for all models and future steps, indicating improved accuracy. For example, at a future step of 1 hour, the LSTM model demonstrates an MAE of 0.2246 without PCA, which decreases to 0.1093 with PCA. Again, similar trends are observed for the Bi-LSTM and Seq2seq models across all future steps.

It appears that the residual Seq2seq model consistently demonstrates the lowest RMSE and MAE values across various future steps and PCA conditions. However, the residual Seq2seq model exhibits relatively smaller performance improvements when PCA is applied compared to the other models. This suggests that the residual Seq2seq model inherently possesses robustness and adaptability to the dataset characteristics, requiring less reliance on PCA preprocessing for performance enhancement

VIII. CONCLUSION

In conclusion, valuable insights into the realm of time series forecasting for EV charging consumption patterns have been provided by our study. The efficacy of three distinct models—LSTM, BiLSTM, and residual Seq2seq—in predicting future energy demand was explored. Through rigorous evaluation, it was discovered that model performance was significantly enhanced by the application of Principal Component Analysis (PCA), which distills essential patterns from the dataset. Notably, the top performer that emerged was the residual Seq2seq model, showcasing its ability to capture complex temporal dependencies and generate accurate forecasts.

The importance of preprocessing techniques in optimizing forecasting outcomes was underscored by our findings, particularly in dynamic energy consumption environments. Moreover, challenges such as diminishing predictive accuracy with increasing forecast horizons were identified, highlighting the need for continuous refinement and adaptation of forecasting methodologies.

Moving forward, our study sets the stage for future research endeavors aimed at integrating advanced machine learning algorithms and data augmentation strategies to further enhance forecasting accuracy and robustness. Ultimately, our insights contribute to the ongoing advancement of predictive analytics in the field of energy demand forecasting, with potential implications for sustainability and resource management.

REFERENCES

[1] A. B. Hamza, Advanced Statistical Approaches to Quality. Unpublished.

[2] A. Z. Hinchi and M. Tkiouat, “Rolling element bearing remaining useful life estimation based on a convolutional long-short-term memory network,” Procedia Comput. Sci., vol. 127, 2018, Art. no. 123132.

[3] UK, GOV. ”COP26 declaration on accelerating the transition to 100% zero emission cars and vans.” December 6 (2021). https://www.gov.uk/government/publications/cop26-declaration-zero-emission-cars-and-vans/cop26-declaration-on-accelerating-the-transition-to-100-zero-emission-cars-and-vans

[4] IEA.” By 2030 EVs represent more than 60% of vehicles sold globally, and require an adequate surge in chargers installed in buildings”.(2022). https://www.iea.org/reports/by-2030-evs-represent-more-than-60-of-vehicles-sold-globally-and-require-an-adequate-surge-in-chargers-installed-in-buildings

[5] Ministry of Economic Affairs and Finance. “Infrastructures de recharge pour véhicule électrique”. (2020). https://www.entreprises.gouv.fr/fr/infrastructures-de-recharge-pour-vehicule-electrique

[6] H. Abdi and L. J. Williams, “Principal component analysis,” Wiley interdisciplinary reviews: computational statistics, vol. 2, no. 4, pp. 433–459, 2010.

[7] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[8] L. Wei and Z. Zhen-gang, “Based on time sequence of arima model in the application of short-term electricity load forecasting,” in Research Challenges in Computer Science, 2009. ICRCCS’09. International Conference on. IEEE, 2009, pp. 11–14.

[9] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural networks for short-term load forecasting: A review and evaluation,” IEEE Transactions on power systems, vol. 16, no. 1, pp. 44–55, 2001.

[10] C. Tian, J. Ma, C. Zhang, and P. Zhan, “A deep neural network model for short-term load forecast based on long short-term memory network and convolutional neural network,” Energies, vol. 11, no. 12, p. 3493, Dec. 2018.