Menemui Matematik
(Discovering Mathematics)

journal homepage: https://myjms.mohe.gov.my/index.php/dismath/

LSTM Model Optimization Based on the Epoch Numbers for Forecasting Battery Electric Vehicle Charging Patterns

Syahrizal Salleh[1], Roslinazairimah Zakaria[2*], and Siti Roslindar Yaziz[3]
[1,2,3]*Centre for Mathematical Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah, Kuantan, Pahang, Malaysia*

[1]pjz23006@student.umpsa.edu.my, [2]roslinazairimah@umpsa.edu.my, [3]roslindar@umpsa.edu.my
*Corresponding author

Received: 12 June 2024
Accepted: 5 November 2024

ABSTRACT

To address the scarcity of high-resolution BEV charging data, a novel feature engineering technique was applied, transforming start-stop electricity charging data sourced from the My Electric Avenue project into the count of concurrent charging events. Recognizing the nonlinear, dynamic, and noisy nature of BEV charging patterns, a Long Short-Term Memory (LSTM) network was chosen to model the electricity demand arising from multiple concurrent BEV charging events. The selected LSTM network comprises a single layer with 125 units of LSTM cells employing a tanh activation function and a single dense output layer. This study investigates the error distribution of LSTM networks across a range of epochs beyond the point of initial convergence, focusing on the Mean Absolute Percentage Error (MAPE) as the primary error metric. Unlike previous analyses that often concentrate on specific epochs or report loss values without considering the variability of performance metrics across epochs, this study examines MAPE values at intervals between 10 and 50 epochs, with increments of 10 epochs. The LSTM network converges earlier than epoch 10, and the lowest MAPE was achieved at epoch 20. The lowest recorded MAPE was 1.19%, with a corresponding Root Mean Squared Error (RMSE) of 0.51 . The findings contribute to optimizing LSTM training and improving the generalization of the model to unseen data.

Keywords: Battery Electric Vehicles, Long Short-Term Memory, Charging Patterns

INTRODUCTION

The widespread embrace of Battery Electric Vehicles (BEVs) on a global scale has resulted in numerous nations making exclusive commitments to the sale of BEVs for new passenger cars, spanning the cutoff period between 2035 and 2040. This dedicated approach signifies a strategic initiative aimed at mitigating the release of greenhouse gases (GHGs) into the atmosphere, given that BEVs depend entirely on onboard batteries for propulsion. Notably, since 2016, the carbon dioxide (CO_2) emissions stemming from Internal Combustion Engine Vehicles (ICEVs) have contributed to roughly one-quarter of global emissions (Yi et al., 2022).

The simultaneous charging of multiple BEVs imposes strain on the electric grid, giving rise to challenges associated with grid capacity. Electric utility companies are proactively projecting the demand capacity originating from BEVs to avert losses stemming from overgeneration. Furthermore, these companies are compelled to enhance electricity generation and grid infrastructure to accommodate escalating demands that surpass existing capacity levels. The haphazard deployment of BEV charging stations and the surge in charging power demand may

exert an adverse impact on the reliability of power grids (Chang et al., 2021). This uncoordinated demand for EV charging power leads to deteriorating voltage profiles, overloading, harmonic distortions, and accelerates the aging of power equipment (Chang et al., 2021).

Electricity possesses a distinctive characteristic wherein, in principle, demand and supply equilibrium is imperative and must be maintained continuously (Nguyen & Hansen, 2017). Fast charging stations, owing to their elevated power requirements, accentuate the significance of accurately predicting charging demand. This predictive capability contributes significantly to ensuring the reliability and resilience of the power grid (Yi et al., 2022). The uncoordinated charging patterns observed in the substantial adoption of plug-in electric vehicles introduces a distinct periodicity and fluctuation, making conventional load forecasting techniques ineffective (Zhu et al., 2019).

Many studies have been conducted to model BEV charging data collected at public charging stations (Buzna et al., 2019; Chang et al., 2021; Dokur et al., 2022; Kim et al., 2019; Koohfar et al., 2023; Mohsenimanesh et al., 2022; Yi et al., 2022; Zhang et al., 2022; Zhu et al., 2019) It is acknowledged that acquiring charging data from private charging locations poses challenges. In contrast, this study aims to model BEV charging data collected specifically at private charging locations.

As per the findings presented by Dokur et al. (2022), and Chang et al. (2021), publicly accessible chargers constitute only 25% and 8%, respectively. Despite private chargers drawing power at a rate ranging from 1% to nearly 13% compared to public chargers, the sheer quantity of private chargers can be up to 11 times higher than that of public chargers. Nevertheless, it is noteworthy that despite this prevalence, most time series studies on BEV charging patterns tend to concentrate on public chargers, given the limited availability of data on private chargers.

Artificial intelligence exhibits the capability to effectively handle nonlinear, dynamic, and noisy time series data. Various techniques have been employed for time series analysis, including Artificial Neural Network (ANN), Hidden Markov Model (HMM), Genetic Algorithm (GA), Decision Tree (DT), Rough Set Theory, Bayesian Analysis (BA), K-Nearest Neighbors (KNN), Particle Swarm Optimization (PSO), Multilayer Perceptron (MLP), Recurrent Neural Networks (RNN), LSTM, and Convolutional Neural Network (CNN) (Lin et al., 2021).

Recurrent Neural Network (RNN) is a specific type of neural network designed for processing sequence data. However, RNN has limitations related to gradient vanishing issues during error backpropagation, making it challenging to learn from long historical data. Addressing this limitation, LSTM networks have been introduced. The LSTMs overcome the drawbacks of traditional RNNs by integrating short-term memory with long-term memory through gate control mechanisms (Wu et al., 2020). The LSTM has demonstrated superior performance in long time horizon forecasting compared to other artificial intelligence methods. This efficacy is particularly pronounced when utilizing past load data to discern the effects and relationships among time series (Zheng et al., 2017). In contrast to traditional power load, BEV load exhibits stronger random fluctuations (Shen et al.,2022)

The effectiveness of LSTM is typically assessed by analyzing their performance across different epochs during training. However, it has been observed that many studies limit their analysis to specific epochs, potentially overlooking significant variations in the error function as training progresses. This selective focus may lead to an incomplete understanding of the model's true performance over the entire training process. The selective focus on specific epochs, often at

the later stages of training, raises concerns about the generalizability and robustness of the conclusions drawn from these studies. LSTM networks, like other deep learning models, may exhibit fluctuations in the error function even after convergence, indicating that the model's performance can vary significantly across epochs.

Chang et al. (2021) set the epoch hyperparameter at 50, 100, and 150 epochs while the analysis focused exclusively on the results at 150 epochs, disregarding variations in the error function at subsequent epochs. Dokur et al. (2022) only considered LSTM performance at 3 epochs, which may not capture the full convergence patterns of the model, potentially leading to incomplete conclusions about the model's generalization ability. Kim et al. (2019) set the epoch hyperparameter at 10, 30, 50, 70, 80, 100, 120, and 140 epochs. Despite this range, the analysis centered solely on the results at 100 epochs, missing out on potential insights from other epochs. Analysis by Koohfar et al. (2023) was limited to 100 epochs. This selective reporting does not reflect possible trends in the error function before or after 100 epochs, potentially skewing the understanding of the model's performance. Yi et al. (2022) focused on LSTM performance at 200 epochs. While this might capture the model's state at convergence, it overlooks the variations in the error function across other epochs, which could provide valuable insights into the training dynamics. Zhu et al. (2019) analyzed the training and loss function at 1, 10, 20, and 30 epochs. However, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) metrics were only reported at 30 epochs, neglecting the patterns of these error metrics at earlier epochs.

In this study, we aim to investigate the error distribution of an LSTM model across a range of epochs beyond its initial point of convergence. Specifically, we will analyze the MAPE as our primary error metric. Unlike traditional analyses that often focus on the loss function or specific epoch numbers, this study will provide a detailed examination of how MAPE values behave over a series of epochs after the model has converged. To explore the variability of error metrics post-convergence, we will evaluate the MAPE across epochs ranging from 10 to 50, with increments of 10 epochs. This range is selected to capture the performance of the model not only at early stages of training but also as it progresses through additional iterations. By analyzing the error distribution across these epochs, this study aims to uncover any underlying trends or fluctuations that might be overlooked when focusing on a single epoch, as seen in some prior studies.

Siami-Namini et al. (2019) provided an extensive report of loss values across multiple epochs, offering a useful reference for understanding the loss landscape during training. However, their study did not delve deeply into specific error metrics like MAPE, nor did it examine the distribution of errors across different epochs post-convergence.

The detailed analysis of MAPE across a range of epochs is not only a methodological refinement but also has significant implications for model generalization. This study will examine how the error distribution evolves with continued training, the point at which additional epochs cease to provide meaningful improvements or may even lead to overfitting. This insight is crucial for optimizing LSTM training processes and ensuring robust model performance on unseen data.

METHODOLOGY

The study of BEV charging patterns has predominantly relied on the analysis of authentic data obtained from public charging facilities. Specifically, Buzna et al. (2019) employed actual BEV load data derived from a dataset collected in 1700 charging stations in the Netherlands. Meanwhile, Chang et al. (2021) utilized a dataset encompassing BEV charging power demand from fast

charging stations across the entire Jeju Island area. Dokur et al. (2022) conducted their study using 7891 charging events collected from various public stations in Elaad Netherlands in 2019.

As per the findings presented by Dokur et al. (2022), and Chang et al. (2021), publicly accessible
chargers constitute only 25% and 8%, respectively. Despite private chargers drawing power at a rate ranging from 1% to nearly 13% compared to public chargers, the sheer quantity of private chargers can be up to 11 times higher than that of public chargers. Nevertheless, it is noteworthy that despite this prevalence, most time series studies on BEV charging patterns tend to concentrate on public chargers, given the limited availability of data on private chargers.

EA Technology initiated a BEV loaner program (My Electric Avenue). This program involved providing a select group of participants with a few hundred Nissan Leaf first generation BEVs, spanning between 2013 and 2015. The objective of the program was to comprehensively study user charging patterns, with a particular focus on understanding BEV charging patterns. Participants were equipped with Level 1 home chargers as part of the initiative. At the conclusion of the program, EA Technology gathered comprehensive data from various aspects of the initiative. The collected data encompassed information such as the locations of the chargers, the start and stop timestamps of charging events, the battery charge levels at the onset and conclusion of charging events, and the recorded distances traveled by the participating vehicles.

Forecasting methods can be categorized according to various techniques, including classical, machine learning, and deep learning approaches (El-Azab et al., 2023). The majority of statistical or traditional methods, such as Autoregressive integrated moving average (ARIMA), Seasonal Autoregressive Integrated Moving Average (SARIMA), Generalized Autoregressive Conditional Heteroskedasticity (GARCH), Linear Regression, Linear Discriminate Analysis (LDA), and Support Vector Machine (SVM), rely on assumptions of linearity, stability, and normally distributed data (Lin et al., 2021). However, the stochastic nature of BEV charging demand, influenced by individual BEV owners' driving and travel patterns, introduces variability and unpredictability (Buzna et al., 2019).

Figure 1 visualizes a single LSTM cell, its input layer, output layer, and hidden internal layers. The LSTM cell takes input from the previous cell state memory (c_{t-1}), previous cell hidden state (h_{t-1}), and current input data (x_t). Internally, the intermediate state of forget gate (f_t), input gate (i_t), and prior cell state ($\acute{c}_t$) is calculated. Outputs from the LSTM cell are current cell state (c_t), current hidden cell state (h_t), and output gate state (o_t).

The LSTM cell is defined by Eq. (1-6).

$$f_t = \sigma_g\left(W_f * x_t + U_f * h_{t-1} + b_f\right) \quad (1)$$
$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + b_i) \quad (2)$$
$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + b_o) \quad (3)$$
$$\acute{c}_t = \sigma_g(W_c * x_t + U_c * h_{t-1} + b_c) \quad (4)$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \acute{c}_t \quad (5)$$
$$h_t = o_t \cdot \sigma_c(c_t) \quad (6)$$

where, the constants W_f, W_i, W_o, W_c, U_f, U_i, U_o, and U_c are weight matrices. The constants b_f, b_i, b_o, and b_c are biases. Both weight matrices and biases are not time-dependant. The σ_g is a sigmoid function defined by Eq. (7), and the σ_c is tanh hyperbolic tangent function defined by Eq. (8).

$$\sigma_g = (1 + e^{-x})^{-1} \tag{7}$$

$$\sigma_c = \frac{e^{2x}-1}{e^{2x}+1} \tag{8}$$

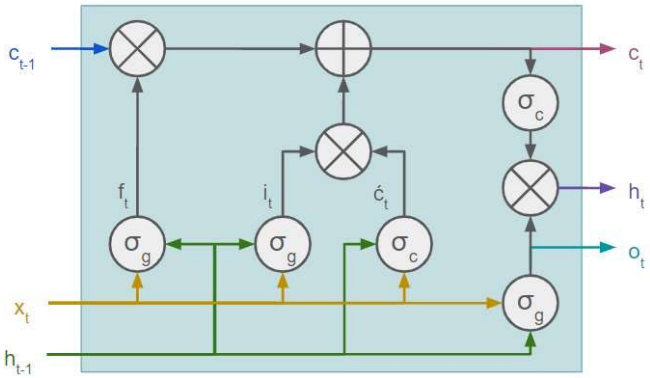


Figure 1: Visualization of single LSTM cell

The LSTM network architecture utilized for this study is depicted in Figure 2. The hidden LSTM
layer is structured as a sequential single layer, encompassing 125 units of LSTM cells. In the training process, the data was divided into sequences of 60 data points, each associated with a single expected output. The expected output corresponds to the subsequent data point in the training dataset. The structure of the data aligns with the LSTM network for effective training and prediction.

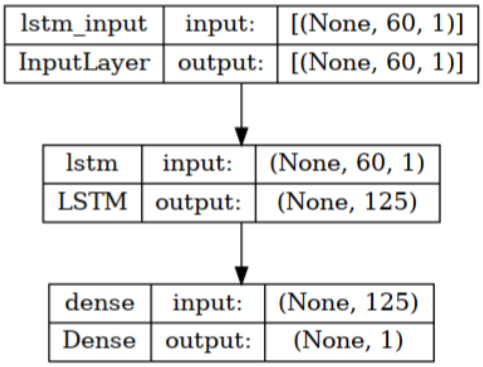


Figure 2: LSTM network architecture for this study

The LSTM model employed in this study is configured for supervised learning. To impart the patterns of the data to the LSTM network, the feature-engineered data, as depicted in Figure 3, is partitioned into training and testing sets. Specifically, the training data comprises 80% of the dataset, starting from the initial entry, while the testing data consists of the remaining subsequent 20% of the dataset (Koohfar et al., 2023). From the total 1-minute interval feature engineered BEV concurrent active charging data, the training dataset contains 452,304 data points, while the testing dataset consists of the remaining 50,256 data points.

The purpose of the training data is to enable the LSTM network to fine-tune its weight matrices and biases. Through exposure to the training dataset, the network refines its parameters to learn and capture the underlying patterns and relationships within the data. This process

enhances the model's ability to make accurate predictions when presented with new, unseen data. The testing data is the unseen data in this study.

The performance of the proposed forecasting method is rigorously evaluated using two widely recognized error metrics: MAPE, and RMSE. These metrics are chosen to provide a comprehensive assessment of the model's accuracy, robustness, and overall predictive capability. Using both MAPE and RMSE together provides a robust evaluation of the forecasting model. MAPE gives an understanding of the relative error and is beneficial for interpreting the model's performance across different magnitudes of data. RMSE, on the other hand, focuses on absolute errors, highlighting the significance of large discrepancies between predicted and actual values.

MAPE is calculated for each forecasted data point and provides a measure of the average relative error between the predicted and actual values, expressed as a percentage of the actual value. MAPE is defined mathematically as:

$$\text{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t-\hat{y}_t}{y_t}\right| \times 100 \quad (9)$$

where y_t represents the observed value, $\hat{y}_t$ is the forecasted value at time *t*, and *n* is the number of observations. MAPE expresses the prediction error as a percentage, offering an intuitive measure of the model accuracy across different scales of data. It is particularly useful in providing insight into the model generalization capability, as it averages the errors over all data points.

RMSE is another critical metric that measures the absolute magnitude of prediction errors, penalizing larger errors more heavily. It is computed as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (10)$$

RMSE is valuable for understanding the overall accuracy of the model in the same units as the data, making it easier to grasp the practical impact of the forecast errors. By squaring the errors before averaging them, RMSE emphasizes larger deviations, which is particularly important in applications where minimizing significant errors is crucial.

RESULTS AND DISCUSSIONS

The dataset utilized in this study is secondary data sourced from the EA Technology program (My Electric Avenue). The relevant data consists of discrete records capturing the start and stop times of BEV charging sessions, with timestamps recorded at minute intervals. To facilitate time-series analysis, the discrete start and stop timestamps were transformed into a continuous time series. This transformation was achieved through a custom feature engineering process, which involved aggregating the data to represent the number of concurrent active charging events at each minute. The result is a continuous time series that tracks the dynamic patterns of EV charging activities over time.

The transformed dataset, hereafter referred to as the observed data, is illustrated in Figure 3. This dataset forms the foundation for the subsequent analysis, facilitating the investigation of the BEV charging patterns and the assessment of the forecasting model's performance. Upon visual

inspection, the observed data reveals a clear daily seasonality in BEV charging patterns. Specifically, charging activity begins to increase around 6:00 PM, reaching a peak at approximately 8:00 PM. This patterns is highlighted in the snippet window of Figure 3, which provides a closer view of the daily fluctuations. Additionally, the main plot in Figure 3 indicates the presence of another concurrent seasonal patterns, suggesting that multiple layers of seasonality may be influencing the charging patterns. These preliminary observations underline the complexity of the dataset and underscore the importance of capturing these seasonal effects in the forecasting model.

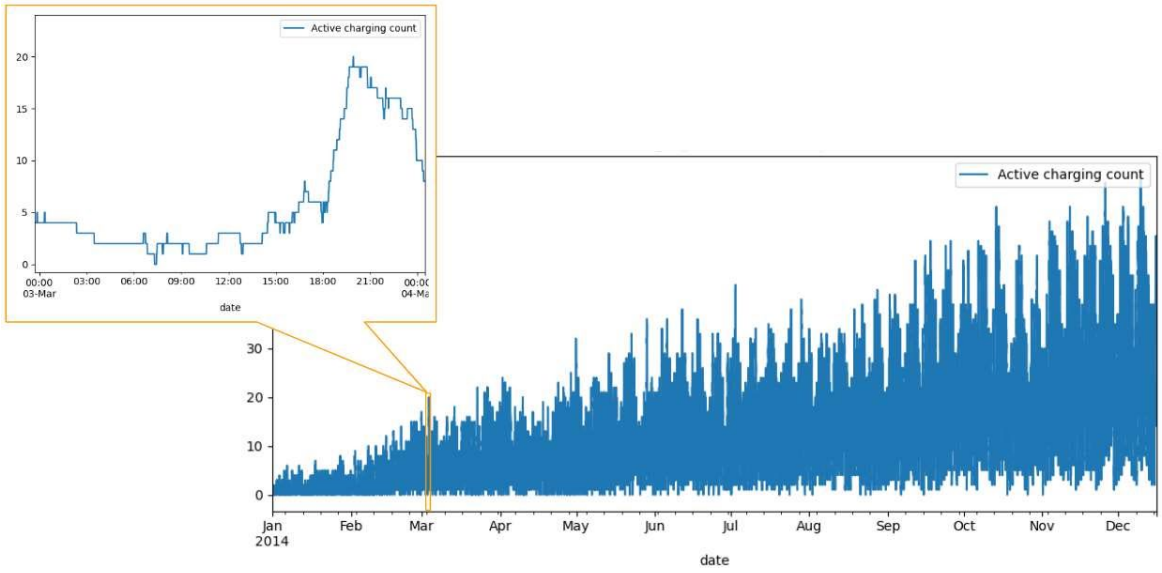


Figure 3: Concurrent active charging count by minutes

To facilitate the development and evaluation of the forecasting model, the observed data was systematically partitioned into two distinct subsets: a training set and a testing set. The training set, comprising 80% of the total dataset, was used to train the model, enabling it to learn the underlying patterns and relationships within the data. The remaining 20% of the data was reserved as the testing set, which was used to evaluate the model's predictive performance on unseen data. This methodical splitting of the dataset is critical for validating the model's accuracy in forecasting the BEV charging patterns.

In Figure 4, the training set is represented by the first 80% of the data, capturing the early and mid-range periods of the observed dataset. This portion of the data includes various patterns of daily and potential longer-term seasonal patterns, which the model utilizes to learn the dynamics of EV charging activities. The testing set, comprising the final 20% of the data, is shown as the latter segment of the observed data. This partition is crucial for evaluating the model's predictive performance, as it contains data that the model has not encountered during the training process. The visual distinction between the training and testing sets in Figure 4 highlights the continuity of the charging events and ensures that the structure of the data is preserved. This continuity is essential for maintaining the integrity of the time-series data, allowing for a more accurate assessment of the model's ability to forecast future charging patterns.

To ensure the consistency and robustness of the results, the modeling and forecasting processes were executed on two distinct CPU configurations. The first computational environment utilized an AMD Ryzen 7 5800HS processor with 16GB of memory, referred to as CPU_A. In parallel, a second environment employed an Intel i7-8700T processor with 8GB of memory, designated as CPU_L. This dual-CPU approach was implemented to evaluate whether hardware

differences could influence the model's performance and the reproducibility of the results. The LSTM model was iteratively trained and evaluated at epoch intervals of 10, 20, 30, 40, and 50. For each configuration (CPU_A and CPU_L), the models were trained independently at these epoch intervals to assess the impact of training duration on the model's accuracy. After each iteration, the forecasted values generated by the model were validated against the testing dataset. The performance of each model was quantitatively assessed using two key metrics: RMSE and MAPE.

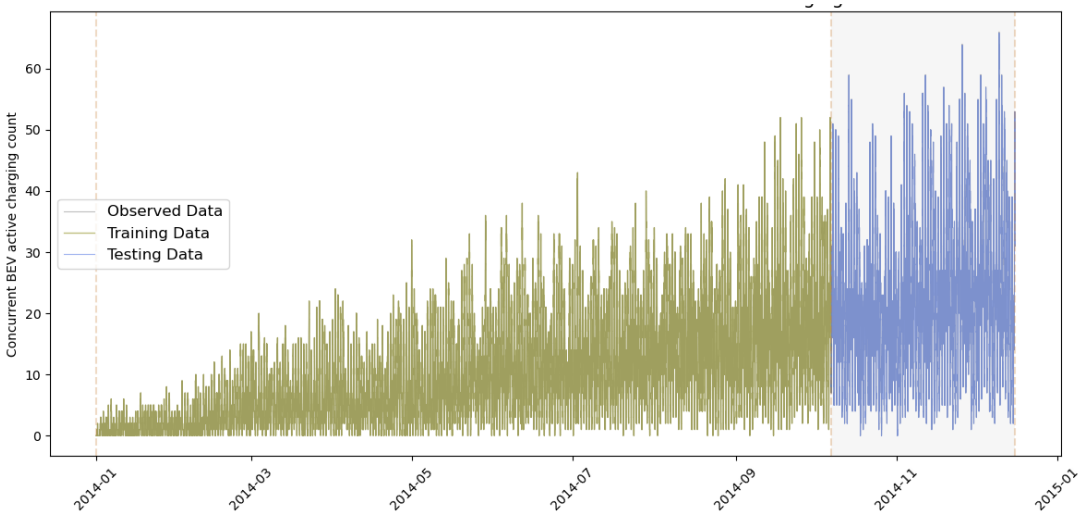
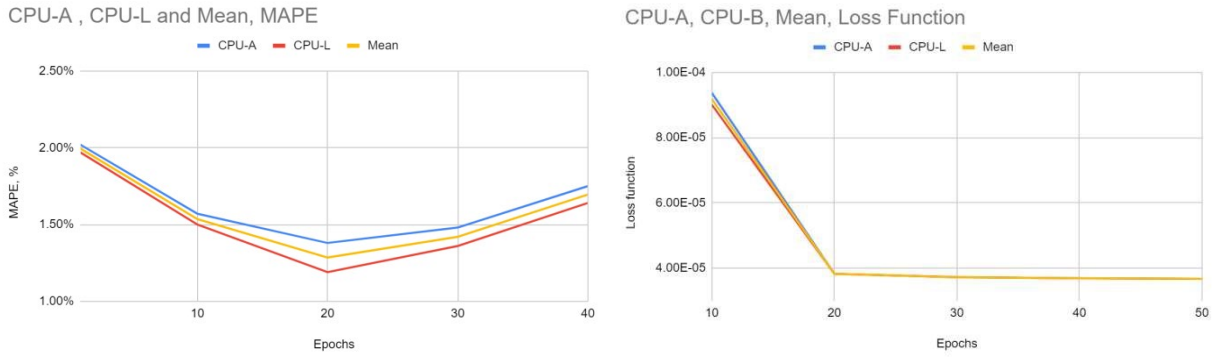


Figure 4: Training and testing set partitioned into 80:20 ratio

Figure 5(a) illustrates the relationship between MAPE and the number of epochs for models developed using both CPU_A and CPU_L. The figure provides a comparative visualization of how the model's accuracy evolved as the number of training epochs increased. This comparison reveals the model's sensitivity to different hardware configurations and training durations, offering insights into the optimal conditions for achieving accurate forecasts. A key observation from Figure 5(a) is that the LSTM model consistently reached its minimum MAPE at epoch 20, regardless of the CPU used. This finding suggests that, for this particular dataset and model configuration, 20 epochs represent the optimal balance between underfitting and overfitting, leading to the best generalization performance on the testing data. The convergence of results across both CPUs further reinforces the reliability of this epoch selection, indicating that the model's performance is robust to variations in computational resources.

Throughout each epoch run, the value of the loss function was meticulously recorded to monitor the model's performance during training. Achieving the lowest possible value of the loss function is crucial, as it indicates that the model is making predictions on training data that closely align with the actual data. In the context of neural networks, the goal is to reach the global minimum of the loss function. This global minimum represents the point where the model's predictions are most accurate across the training dataset. However, in practice, the model may only approach this minimum, depending on various factors such as the complexity of the model, the learning rate, and the number of epochs.

Figure 5(b) illustrates the recorded loss function values across the epochs during the training process. As observed, the LSTM model's loss function consistently decreased as the number of epochs increased, indicating that the model was progressively improving its fit to the training data. Notably, the model had significant learning occurred within the first few epochs as shown by drastic drop in the loss function for both CPU_A and CPU_L. Subsequent to epoch 10, the learning rate which indicated by reduction in the loss function reduces slightly at every epoch.



(a) (b)

Figure 5: (a) MAPE values over number of epochs run, (b) Loss function values over number of epochs run

Table 1: Forecast evaluation and loss function using LSTM

	CPU_A			CPU_L		
Epoch	RMSE	MAPE (%)	Loss Function	RMSE	MAPE (%)	Loss Function
1	0.58	2.02	9.3675×10^{-5}	0.58	1.97	9.0025×10^{-5}
10	0.52	1.57	3.8275×10^{-5}	0.51	1.50	3.8266×10^{-5}
20	0.51	1.38	3.7209×10^{-5}	0.51	1.19	3.7259×10^{-5}
30	0.52	1.48	3.6928×10^{-5}	0.51	1.36	3.6870×10^{-5}
40	0.53	1.75	3.6713×10^{-5}	0.52	1.64	3.6734×10^{-5}
50	0.53	1.79	3.6650×10^{-5}	0.55	2.09	3.6594×10^{-5}

Table 1 provides a comprehensive summary of the relationship between the number of epochs and the corresponding values of RMSE, MAPE, and the loss function for both CPU_A and CPU_L. This table serves as a key reference for understanding how the model's performance metrics evolved with additional training. It reveals that the LSTM model's error metrics, such as RMSE and MAPE, were closely aligned with the patterns of the loss function, with the most significant improvements occurring between epochs 10 and 20. For both CPU_A and CPU_L, the results consistently show that the model achieved its lowest RMSE and MAPE values at epoch 20, which corresponds to the point where the loss function was minimized. This correlation between the loss function and error metrics underscores the importance of selecting an appropriate number of training epochs. Training beyond epoch 20 showed diminishing returns, with minimal improvements in the loss function and error metrics, suggesting that this is an optimal stopping point for this specific dataset and model configuration.

CONCLUSIONS

The LSTM model developed in this study has achieved successful one-step-ahead forecasts with the lowest MAPE recorded as 1.19% on CPU_L at epoch 20 and 1.38% on CPU_A at a comparable epoch (epoch 20). The corresponding loss function values for the lowest MAPE on CPU_A and CPU_L are 3.7209×10^{-5} and 3.7259×10^{-5}, respectively.

The use of two different CPUs highlights considerations for computational efficiency. While CPU_A, with its higher memory capacity, may offer faster processing times and potentially allow for larger batch sizes or more complex model architectures, the comparable performance achieved with CPU_L indicates that similar results can be obtained even with less powerful hardware. This insight is valuable for practitioners who may need to balance model performance with available computational resources, particularly in real-world applications where hardware constraints are a common consideration. By systematically analyzing the model's performance across different epochs and CPU configurations, this study not only identifies the optimal training parameters for the LSTM model but also demonstrates the importance of considering hardware environments in the evaluation process. The consistent results across CPUs validate the robustness of the forecasting approach, ensuring that the findings are both reliable and reproducible across different computational setups.

The analysis of the loss function across epochs provides valuable insights into the optimization process of the LSTM model. By understanding how the loss function evolves during training, practitioners can make informed decisions about the appropriate number of epochs needed to achieve optimal model performance without unnecessary computational expense. The findings suggest that early stopping could have been employed around epoch 20. This detailed examination of the loss function, coupled with the analysis of RMSE and MAPE, highlights the critical role of epoch selection in neural network training. It emphasizes the importance of monitoring the loss function as a key indicator of the model's progress and provides a practical framework for determining when to stop training to achieve the best possible outcomes.

ACKNOWLEDGEMENT

This research was funded by Universiti Malaysia Pahang Al-Sultan Abdullah (grant number RDU233008) under a special publication grant from the Research and Innovation Department, and Doctorate Research Scheme (Salleh, S.)

REFERENCES

Buzna, L., De Falco, P., Khormali, S., Proto, D., & Straka, M. (2019, May). Electric vehicle load forecasting: A comparison between time series and machine learning approaches. In 2019 1st International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED) (pp. 1-5). IEEE.

Chang, M., Bae, S., Cha, G., & Yoo, J. (2021). Aggregated electric vehicle fast-charging power demand analysis and forecast based on LSTM neural network. *Sustainability*, **13(24)**: 13783.

Dokur, E., Erdogan, N., & Kucuksari, S. (2022). EV fleet charging load forecasting based on multiple decomposition with CEEMDAN and swarm decomposition. IEEE access, **10**: 62330-62340.

El-Azab, H. A. I., Swief, R. A., El-Amary, N. H., & Temraz, H. K. (2023). Seasonal electric vehicle forecasting model based on machine learning and deep learning techniques. *Energy and AI*, **14**: 100285.

Koohfar, S., Woldemariam, W., & Kumar, A. (2023). Prediction of Electric Vehicles Charging Demand: A Transformer-Based Deep Learning Approach. Sustainability, **15(3)**: 2105.

Kim, M., Choi, W., Jeon, Y., & Liu, L. (2019). A hybrid neural network model for power demand forecasting. *Energies*, **12(5)**: 931.

Lin, Y., Yan, Y., Xu, J., Liao, Y., & Ma, F. (2021). Forecasting stock index price using the CEEMDAN-LSTM model. *The North American Journal of Economics and Finance*, **57**: 101421.

Mohsenimanesh, A., Entchev, E., & Bosnjak, F. (2022). Hybrid Model Based on an SD Selection, CEEMDAN, and Deep Learning for Short-Term Load Forecasting of an Electric Vehicle Fleet. *Applied Sciences*, **12(18)**: 9288.

My Electric Avenue. EA Technology. [Online]. Available: https://eatechnology.com/media/iwjjxzeg/1-myelectricavenue-i2ev-projectsummaryreport.pdf.

Nguyen, H., & Hansen, C. K. (2017, June). Short-term electricity load forecasting with Time Series Analysis. In 2017 IEEE International Conference on Prognostics and Health Management (ICPHM) (pp. 214-221). IEEE.

Shen, X., Zhao, H., Xiang, Y., Lan, P., & Liu, J. (2022). Short-term electric vehicles charging load forecasting based on deep learning in low-quality data environments. *Electric Power Systems Research*, **212**: 108247.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International conference on big data (Big Data) (pp. 3285-3292). IEEE.

Wu, L., Kong, C., Hao, X., & Chen, W. (2020). A short-term load forecasting method based on GRU CNN hybrid neural network model. *Mathematical problems in engineering*, 2020, 1-10.

Yi, Z., Liu, X. C., Wei, R., Chen, X., & Dai, J. (2022). Electric vehicle charging demand forecasting using deep learning model. *Journal of Intelligent Transportation Systems*, **26(6)**: 690-703.

Zhang, J., Liu, C., & Ge, L. (2022). Short-term load forecasting model of electric vehicle charging load based on MCCNN-TCN. *Energies*, 15(7), 2633.

Zheng, H., Yuan, J., & Chen, L. (2017). Short-term load forecasting using EMD-LSTM neural networks with a Xgboost algorithm for feature importance evaluation. *Energies*, **10(8)**: 1168.

Zhu, J., Yang, Z., Mourshed, M., Guo, Y., Zhou, Y., Chang, Y., Wei & Feng, S. (2019). Electric vehiclecharging load forecasting: A comparative study of deep learning approaches. *Energies*, **12(14)**: 2692.