Design and Implementation of Short-Term Load Forecasting using STM

Balaguruperumal A.[1], Hemavarshini P.[2], Lakshan Damodharasami[3], Tejaswini A A.[4], Manikandan V.[5]

Department of Electrical and Electronics Engineering, Coimbatore Institute of Technology, Coimbatore, India

E-mail: [1]balaguruperumal2003@gmail.com, [2]hemavarshini2209@gmail.com, [3]lakshandamu2904@gmail.com, [4]tejaswinialagappan16@gmail.com. [5]manikandan@cit.edu.in

Abstract

Residential energy consumption constitutes a significant portion of the total electricity demand, emphasizing the urgent need for accurate short-term load forecasting to facilitate efficient energy management. Initially, a Bidirectional Long Short-Term Memory (BiLSTM) model was employed to analyze household energy consumption patterns using publicly available datasets. To further enhance forecasting accuracy and improve adaptability to real-world scenarios, a real-time data collection system was developed utilizing an ESP32. This system was designed to capture key parameters, including voltage, current, power, and energy consumption, thereby generating a custom dataset. Subsequently, this custom dataset was used to train the BiLSTM model, which was then deployed on an STM32 microcontroller for edge-based forecasting. For fine-tuning the model's hyperparameters, the Particle Swarm Optimization (PSO) technique was implemented. Pre-processing techniques were applied to filter and reduce noise within the datasets. Comparative studies conducted between the BiLSTM models trained on publicly available data and the customized data demonstrated the superiority of the customized data in terms of forecasting accuracy and sensitivity in edge-based performance. The proposed methodology outperforms traditional forecasting techniques by enabling a scalable, adaptive, and effective solution for residential energy management.

Keywords: Load Forecasting, BiLSTM, STM32, Particle Swarm Optimization.

DOI: https://doi.org/10.36548/jscp.2025.1.002
Received: 08.02.2025, received in revised form: 22.03.2025, accepted: 05.04.2025, published: 16.04.2025

1. Introduction

Optimization of energy consumption in homes, reduction in electricity usage and consumption cost, and the adoption of sustainable energy practices are facilitated by accurate short-term load forecasting. Household energy consumption exhibits non-linear behavior influenced by various factors, such as the number of occupants and the frequency of appliance usage. Furthermore, environmental factors, including temperature and daylight hours, also impact consumption patterns. This inherent non-linearity and unpredictability of electricity use are not adequately captured by traditional forecasting methods, leading to energy waste and increased costs. Both homeowners and utility providers face challenges due to inaccurate predictions, resulting in higher bills and difficulties in maintaining grid stability. Consequently, advanced forecasting methods are essential for improving energy efficiency and establishing a sustainable energy system at the household level. Short-term load forecasting enables the accurate prediction of electricity demand, thereby enhancing the optimization of energy consumption. Temporal Convolutional Networks (TCN) and LightGBM have been employed to improve load prediction accuracy, optimizing energy use for smarter residential energy management [1]. Support Vector Machine (SVM) decomposition combined with XGBoost can effectively address non-linear energy patterns, reducing errors and enhancing prediction reliability [2]. To incorporate variables such as weather conditions, daily routines, and appliance usage, probabilistic forecasting with machine learning techniques is utilized [3]. Optimization techniques, like Levy Flight-Particle Swarm Optimization (LF-PSO), fine-tune the model parameters, to ensure faster convergence, and minimize prediction errors. BiLSTM models, when integrated with such approaches, effectively capture complex energy consumption patterns [4]. The capability to predict energy demand across diverse datasets has been extensively analyzed using deep learning-based models, including variations of Long Short-Term Memory (LSTM) networks. For further enhancing the prediction accuracy, hybrid models combining BiLSTM with attention mechanisms, which focus on important temporal features, have been implemented [5]. Moreover, edge-based forecasting enhances the practicality of these models by deploying BiLSTM models on edge devices such as STM32 microcontrollers, enabling real-time, low-latency load forecasting for home energy management [6]. The integration of optimization techniques, such as Modified Particle Swarm Optimization with attention-based LSTM, allows for more accurate and adaptive predictions, making it ideal for real-time residential energy management applications [7]. By utilizing both temporal dependencies and feature extraction capabilities, a BiLSTM model combined with a wide neural network architecture has been proposed to enhance short-term load forecasting. This hybrid method proves effective for energy-efficient home automation systems by improving the accuracy and precision of load forecasting methods [8]. This research emphasizes the feasibility of implementing deep learning models on microcontrollers, demonstrating that efficient computation and real-time prediction can be achieved even with resource-constrained hardware [9]. For more accurate short-term electricity load predictions, a modified Sparrow Search Algorithm is used to optimize BiLSTM training [10]. These advancements contribute to the scalability and adaptability of energy forecasting solutions, making them suitable for dynamic residential environments. The current forecasting techniques have proven to be effective and eco-friendly in managing the energy consumption of a typical household through the application of deep learning, optimization techniques, and edge computing.

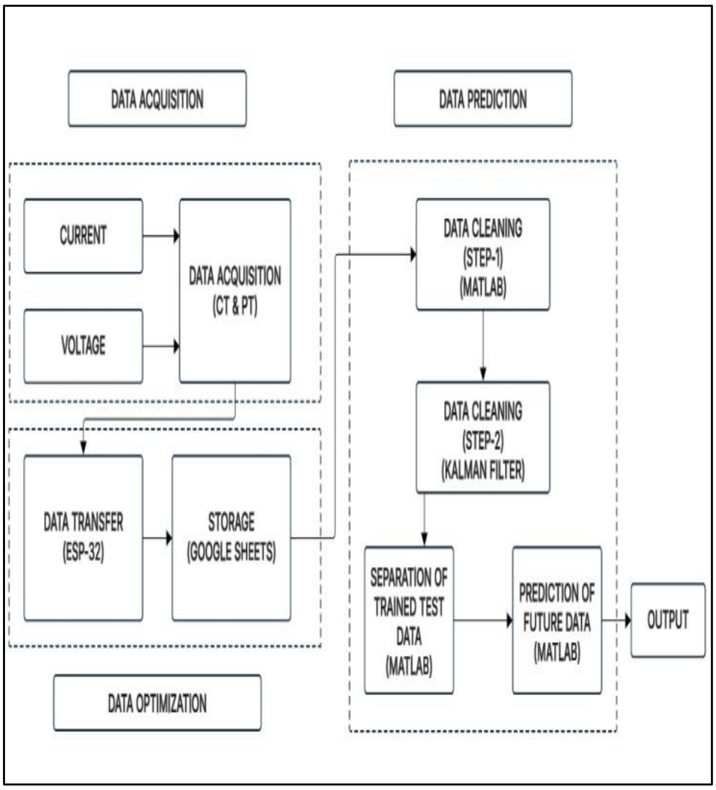
DATA ACQUISITION
DATA PREDICTION
CURRENT
VOLTAGE
DATA ACQUISITION (CT & PT)
DATA CLEANING (STEP-1) (MATLAB)
DATA CLEANING (STEP-2) (KALMAN FILTER)
DATA TRANSFER (ESP-32)
STORAGE (GOOGLE SHEETS)
SEPARATION OF TRAINED TEST DATA (MATLAB)
PREDICTION OF FUTURE DATA (MATLAB)
OUTPUT
DATA OPTIMIZATION

Figure 1. Proposed Block Diagram for Energy Optimization

2. Proposed Methodology

The proposed methodology for short-term load forecasting for residential energy focuses on precise, flexible, and real-time predictions to meet the complexities of fluctuating consumption patterns. This approach combines state-of-the-art machine learning approaches, optimization techniques, and data pre-processing (Figure. 1) to develop an accurate forecasting model that is suitable for home energy management.

This research is a comparative evaluation of short-term load prediction based on two different data sources: publicly provided online datasets and real-time measurements from a hardware system. The forecasting model takes a Bidirectional Long Short-Term Memory (BiLSTM) model, trained independently with both types of data, to evaluate their efficiencies in predicting the consumption of energy by residential households. Historical consumption trends are available from the online dataset, whereas hardware-based data, as received from an ESP32 microcontroller, indicates [illegible] ntial activities and climate. A performance analysis is extensively done to understand the accuracy, adaptability, and response of every method in order to reflect on their relevance to real-time home energy management.

Fig. 1.System architecture for load forecasting

2.1 Data Acquisition

The online dataset used in this work is based on Australian residential electricity consumption data which is available in https://www.kaggle.com/datasets/julianlee/australian-energy-household-dataset. It contains historical consumption data of power from smart meters installed in several houses, thus enabling a rich and complex dataset of the residential energy consumption patterns. Some of the key features of the dataset include consumption data in the time series format which helps in identifying changes in power consumption over time, as well as weather data such as temperature and humidity that are key drivers of energy consumption. The model is therefore able to use this dataset to predict the trends, seasons, and other factors that affect electricity demand in order to improve the accuracy of short-term load forecasting. The hardware-based data acquisition system is designed to sample real time electrical parameters using an ESP32 microcontroller to accurately measure the energy consumption of the house. The system stores time data, voltage, current, power and energy data, so that the data to be used for load forecasting is detailed. The system records time data, voltage, current, power, and energy, ensuring a comprehensive dataset for load forecasting. The dataset is summarized in Table 1. The ZMPT101B sensor records voltage, whereas an SCT-013 current sensor records current measurements. The ESP32 computes these readings and determines power and energy consumption. The gathered data is forwarded to Google Sheets through an HTTP POST request where a Google App Script extracts and logs the readings in structured format, timestamps the entry, for further analysis. This data acquisition system guarantees the sequential flow of household load pattern supporting the demand side management strategies. this system provides a representative dataset, facilitating enhanced forecasting accuracy. This real-time monitoring method boosts forecasting accuracy by recording actual household consumption pattern, providing a more valid substitute for online datasets. This data acquisition system improves the potential and accuracy of the forecasting model, as it gives real time reading with respect to physical parameter using sensors.

Table 1. Real Time Data Collection

Date	Voltage	Current	Power	Energy	Predicted
03/02/2025	220.44	1.2309	270.97	5511.27	6023
03/02/2025	220.03	1.467	323.66	5516.66	6001

2.2 Pre-processing Data

The online data obtained from publicly available residential electricity consumption records proves to be unreliable as it contains missing values, outliers, and inconsistencies due to logging errors or variations in sampling rates. To improve data quality, the unwanted fluctuations has been removed by filtering, while maintaining of power consumption values within a consistent range is ensured by normalization. For identification of prominent consumption patterns, caused by external factors such as time-based trends and seasonal variations, feature extraction has been performed. The long-term energy consumption behaviours have been extracted to train BiLSTM model, with the help of these pre-processed inputs. To pre-process the dataset, MATLAB’s built-in filter feature has been used. Additionally, to reduce the impact of sudden variations and sensor inaccuracies, the Kalman filter is applied to dynamically estimate true consumption values.

The real-time data collected using ESP32 microcontroller, records voltage, current, power, energy, and timestamps. Compared to online data, this data is more vulnerable to transient disturbances due to sensor noise, inference of environmental factors, and fluctuations in household loads. To clean the recorded values, MATLAB’s built-in filter feature is applied, also preserving the essential variations in load consumption. This step is essential, as these real-time variations directly impact forecasting accuracy. For real-time prediction; unlike online data, immediate preprocessing is required before being fed into the BiLSTM model (Figure. 2). For maintaining the consistency in recorded values, normalization is performed. To improve model performance, additional factors such as time of day and appliance-specific power usage trends are incorporated. To stabilize the real-time sensor readings, the Kalman filter is applied, enhancing the accuracy and reliability of short-term load forecasting in residential settings.

2.3 Model Development

The BiLSTM model for online data is trained on historical residential electricity consumption records to capture long-term usage patterns and seasonal variations. It uses multiple BiLSTM layers to effectively pattern power consumption dependencies, outperforming traditional LSTM networks. Dense layers enhance the learned representations, while an output layer predicts future consumption values. Key hyperparameters such as the number of BiLSTM units, batch size and learning rate are optimized for their performance. Dropout layers prevent overfitting and the Adam Optimizer increases convergence speed.

The BiLSTM model for hardware data is designed for real-time prediction and processing using data sensed from the ESP32 microcontroller, including voltage, current, power, energy and timestamps. It shares similar architecture with the online model but is optimized with real-time inference with lightweight structure and in smaller batches to ensure faster computation. For edge-based implementation, STM32 microcontroller deployment is considered, enabling real time forecasting without depending on cloud processing. MATLAB’s inbuilt filter smooths sensor data before input, while the Kalman filter reduces noise and enhances stability of prediction. The model is trained on pre-processed data using MATLAB’s deep learning functions, with RMSE as evaluation parameter is calculated as shown in (1).

Error calculated from

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

Where

y_i = Actual load values

$\hat{y}_i$ = Predicted load values

N = Number of observations

The model adjusted its weights using backpropagation through time (BPTT), minimizing the loss function, as defined in (2). At each training step, the weight is updated.

$$W^{t+1} = W^{t} - \eta \frac{\partial L}{\partial W} \tag{2}$$

Where

W = Model parameters

η = Learning Rate

$\frac{\partial L}{\partial W}$ = Gradient of Loss Function

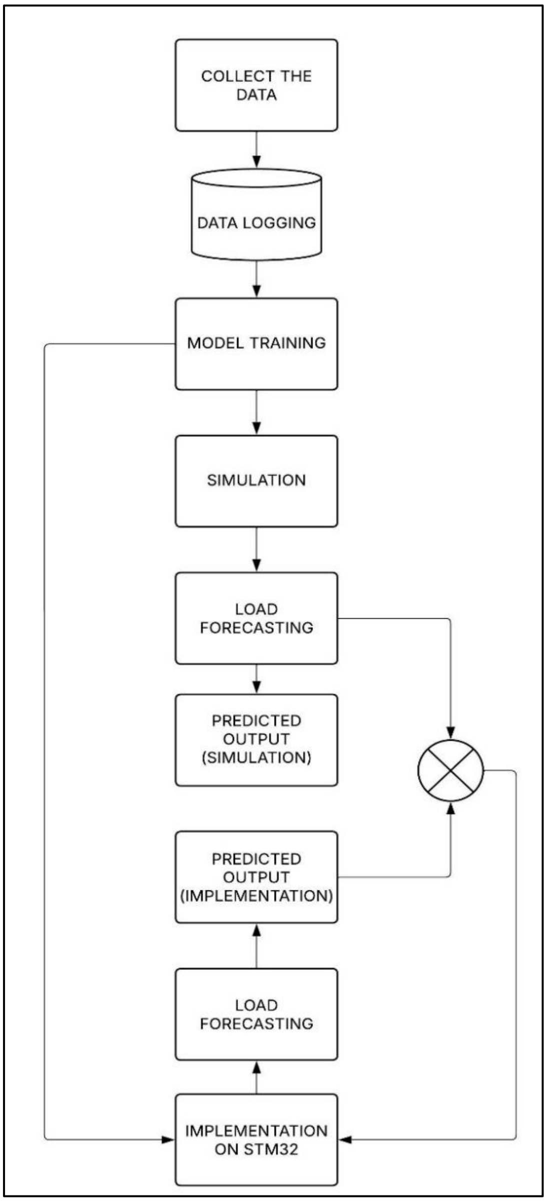

Figure 2. Flow Diagram

The BiLSTM model had an initial load prediction value of 6057, while the actual load was 5751, resulting in a squared error of 93636. The subsequent readings yielded comparable values such as (6080, 6105) and (6102, 6090), the total squared errors of which summed to 237023. With N set to 5, the early RMSE had a value of 217.7, denoting incorrect predictions. As training continued, weight updates through backpropagation and gradient descent changed those predictions into values such as (5751, 5750.98) and (6080, 6080.02), this rebuilding the value of squared errors to 0.0043. Rescaling the RMSE gave the last resulting value of 0.06, which proved the model better at load forecasting.

2.4 Model Optimization and Hyperparameter Tuning

To enhance the performance of both BiLSTM models “Particle Swarm Optimization” is used to fine-tune hyperparameters such as the number of LSTM units, batch size, learning rate and layer depth. For the online data model, PSO iteration searches for the best hyperparameter set by balancing exploitation and exploration, minimizing error metrics like RMSE,MAPE and MAE .The dataset is split into training and validation sets ,and multiple model configurations are tested to ensure optimal performance .In addition to this,L2 regularization and dropout techniques are incorporated to prevent overfitting.

For the hardware data model, PSO is applied with constraints to account for real-world sensor inaccuracies and computational limitations on embedded systems like STM32.The optimization ensures real time feasilbility by improving the model for efficient execution under memory constraints. Quantization and model pruning are used to reduce memory usage and inference time ensuring smooth deployment on edge devices. Further improvements include evaluating fixed point vs floating point computation and adjusting the model structure to fit within STM32’s RAM and flash memory. These optimizations ensure that both models achieve high accuracy while maintaining efficiency for real world application.

2.5 Particle Swarm Optimization

PSO efficiently searches for optimal parameters by balancing exploration and exploitation. In load forecasting it fine-tunes BiLSTM hyperparameters to minimize prediction errors like RMSE and MAPE.

PSO Update Equations

1. Velocity Update

$$Vi(t+1) \; = \; w.vi(t) + c1.r1.(\,Pbesti \; - \; xi(t)) + c2.r2.(\,Gbest \; - \; x\,i\,(t)) \qquad (3)$$

Where

$Vi(t+1)$ = Current velocity of particle I at iteration t.

$xi(t)$ = Current position of particle i.

w = Inertia weight.

$c1, c2$ = Acceleration coefficients.

$r1, r2$ = Random variables between 0 and 1

$Pbest$ $_i$= Best position found by particle i.

$Gbest$ = Best position found by the entire swarm

2. Position Update

$$Xi(t+1) \; = \; xi(t) \; + \; v(t+1)i \qquad (4)$$

During the initial iterations of the Particle Swarm Optimization (PSO) process, the RMSE fluctuated between 0.08 and 0.06, as the algorithm explored different parameter values. For example, in iteration 1, the model parameters were set at (W1 = 0.5, W2 = 1.2, Bias = 0.3), and for an actual load of 5751 kW, the model predicted 6057 kW, leading to a large squared error. Using velocity update (3), the particles adjusted their velocity based on their personal best (Pbest) and global best (Gbest) solutions. Similarly, their updated position was calculated (4), by this RMSE gradually improved. By iteration 5, the updated parameters (W1 = 0.7, W2 = 1.0, Bias = 0.25) reduced the RMSE to 0.06, showing improved forecasting accuracy.

As more iterations progressed, the particles improved their positions further, leading to better parameter optimization. By iteration 10, with (W1 = 0.85, W2 = 0.95, Bias = 0.22), the RMSE improved to 0.05, demonstrating the model's ability to capture load variations more effectively. Eventually, by iteration 15, the parameters stabilized at (W1 = 0.9, W2 = 0.92, Bias = 0.20), with an RMSE of 0.04, indicating that the swarm had converged to an optimal solution. At this stage, the BiLSTM model provided highly accurate predictions, such as (5751, 5750.98) kW, significantly reducing forecasting errors. The reduction in RMSE highlights the effectiveness of PSO in fine-tuning model parameters, ensuring precise short-term load forecasting, which is essential for power system management, grid stability, and efficient energy planning.

3. Performance Evaluation

The online model, trained on extensive historical datasheets, achieved a lower RMSE of 0.04, as shown in Table 2 , highlighting its superior accuracy . This performance

stems from access to a large volume of diverse training data, effective feature engineering and the ability to utilize advanced tuning techniques such as Particle Swarm Optimization. Cloud based processing enables the use of complex architectures like BiLSTM which further refines predictions. However, the model depends on internet connectivity which can introduce the delays and reduce real time responses. The online model accuracy could be further improved by integrating adaptive learning algorithms where model continuously updates itself with new data and applying outlier detection techniques to eliminate erroneous values from online source.

Table 2. Performance Evaluation of Online Model

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch RMSE	Mini-batch Loss	Base Learning Rate
1	250	00:00:08	0.05	1.2e-03	0.0050
1	300	00:00:10	0.05	1.2e-03	0.0050
1	350	00:00:11	0.05	1.2e-03	0.0050
1	400	00:00:13	0.04	8.5e-04	0.0050
1	450	00:00:14	0.04	7.7e-04	0.0050

The STM32 model, is trained using real time data collected through ESP32 which, recorded an RMSE of 0.06, as shown in Table 3, which is slightly higher due to factors like sensor noise, limitless training data, and computational constraints . Since embedded systems lack the processing power of cloud based models, optimization techniques such as model quantization, lightweight BiLSTM architectures, and the noise filtering algorithms can be applied to reduce RMSE. Additionally, incremental learning methods can help the STM32 model adapt to changing energy patterns over time, the STM32 model offers the advantage of real-time, on-device forecasting without dependency on external servers. The online model provides higher accuracy due to its access to extensive datasheets and cloud-based optimization, making it ideal for long-term forecasting and large-scale applications. However the STM32 model, despite its slightly higher RMSE (Figure. 3),excels in real-time responsiveness and autonomy. RMSE for the STM32 model can be further reduced by implying advanced filtering techniques, real-time data augmentation, and lightweight neural architecture. A hybrid approach that combines clous-based training for accuracy with edge-based employment for real-time processing clous provide an optimal balance between precision and efficiency in smart energy forecasting systems.

Table 3. Performance Evaluation of STM Model

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch RMSE	Mini-batch Loss	Base Learning Rate
1	1	00:00:01	0.86	0.4	0.0050
1	50	00:00:03	0.10	5.2e-03	0.0050
1	100	00:00:04	0.07	2.4e-03	0.0050
1	150	00:00:06	0.06	1.8e-03	0.0050

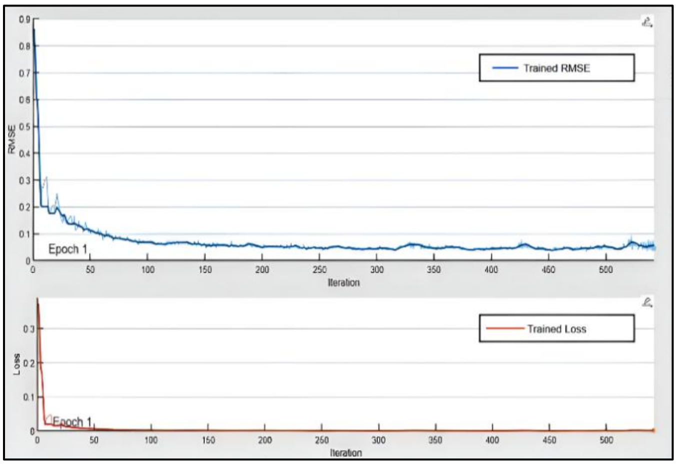

Figure 3. RMSE Graph

Table 4. Memory Usage and Model Complexity

Name	Used RAM	Used Flash	Complexity
Network	3.57 KB	376.48 KB	91601 MACC
Library	2.19 KB	0.00 B	-
Total (1)	5.76 KB	376.48 KB	91601 MACC

A BiLSTM model was constructed on real-time data and then executed on STM32 microcontroller (Figure. 4) as an embedded system for load forecasting in real-time. The model depicts efficiency in resource usage with 3.57 KB of RAM allotted to do network operations while the total RAM consumption stands at 5.76 KB in consideration of the other runtime processes. Flash memory consumption is 376.48 KB, taken up with the model architecture and configuration that prove conducive to microcontroller uses. The computational complexity is rated 91,601 MACC, which is a point that shows how light the model is. Additionally, the MATLAB-built-in library used in runtime takes up only 2.19 KB of memory space without necessarily prompting for flash, thus making the whole system more efficient. All these aggregated measures are accounted in Table4 verifies that the BiLSTM model is always at a fixed place within the frequency range of the edge devices with fast prediction.

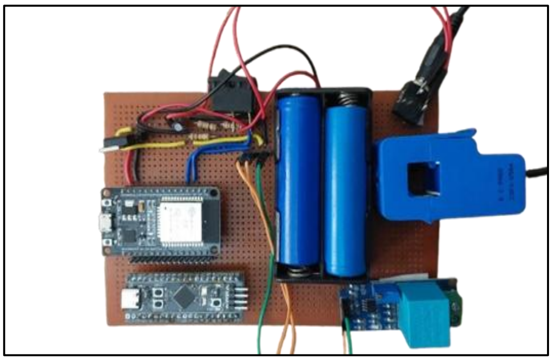

Figure 4. Hardware Prototype

4. Conclusion

This paper presents a compact comparison of short-term load forecasting, using online datasets and real-time hardware-collected data for residential energy management.

Bidirectional Long Short-Term Memory (BiLSTM) model is optimized with particle swarm optimization, and applied to both data sources for evaluation of forecasting accuracy. While the online dataset(numbers of years of energy consumption on the residential level) gave trends and patterning of the long run, the real-time data (live measurement of energy parameters: voltage, current, power, and energy) was collected using the ESP32-based implementation. Deployment of BiLSTM onto an STM32 microcontroller for hardware data showed the efficiency in performing real-time forecasting with small characteristics of deferral, whereas a centralized processing strategy for online datasets showed up. A comparative analysis noted that while real-time data respond better to sudden changes in energy demand, patterns of seasonality and trends with respect to time are more discernible with online datasets. These combined practices provide the feasibility of combining historical and real-time data to improve the accuracy of forecasting, optimization the utilization in households reduce costs and aid in sustainability energy management policies.

References

[1] Y. Wang et al.,” Short-Term Load Forecasting for Industrial Customers Based on TCN-LightGBM,” in IEEE Transactions on Power Systems, vol. 36, no. 3, May2021,https://doi.org/10.1109/TPWRS.2020.3028133. 1984-1997.

[2] Yuanyuan Wang, Shanfeng Sun, Xiaoqiao Chen, Xiangjun Zeng, Yang Kong, Jun Chen, Yongsheng Guo, Tingyuan Wang, “Shortterm load forecasting of industrial customers based on SVMD and XGBoost”, International Journal of Electrical Power & Energy Systems, Volume 129,2021,106830, ISSN 0142-0615, https://doi.org/10.1016/j.ijepes.2021.106830.

[3] Hong, Tao, and Shu Fan. "Probabilistic electric load forecasting: A tutorial review." International Journal of Forecasting 32, no. 3 (2016): 914-938.

[4] Kiruthiga, D., and V. Manikandan. "Levy flight-particle swarm optimization-assisted BiLSTM+ dropout deep learning model for short-term load forecasting." Neural Computing and Applications 35, no. 3 (2023): 2679-2700.

[5] Dai, LuPing. "Performance analysis of deep learning-based electric load forecasting model with particle swarm optimization." Heliyon 10, no. 16 (2024).

[6] Han, Lijia, Xiaohong Wang, Yin Yu, and Duan Wang. "Power Load Forecast Based on CS-LSTM Neural Network." Mathematics (2227-7390) 12, no. 9 (2024).

[7] Sun, Yiyang, Xiangwen Wang, and Junjie Yang. "Modified particle swarm optimization with attention-based LSTM for wind power prediction." Energies 15, no. 12 (2022): 4334.

[8] Cai, Changchun, Yuan Tao, Tianqi Zhu, and Zhixiang Deng. "Short-term load forecasting based on deep learning bidirectional lstm neural network." Applied Sciences 11, no. 17 (2021): 8129.

[9] Elsts, Atis, and Ryan McConville. "Are microcontrollers ready for deep learning-based human activity recognition?." Electronics 10, no. 21 (2021): 2640.

[10] Zhang, Chenjun, Fuqian Zhang, Fuyang Gou, and Wensi Cao. "Study on short-term electricity load forecasting based on the modified simplex approach sparrow search algorithm mixed with a bidirectional long-and short-term memory network." Processes 12, no. 9 (2024): 1796.