Load Forecasting for Dhaka City Using RNN, LSTM, and GRU Architectures with Meteorological and Temporal Data

Abstract—**The paper introduces a machine learning model designed to forecast daily maximum load demand (MW) for the Dhaka city grid, utilizing historical time-series data and meteorological factors. The proposed model integrates advanced Recurrent Neural Network (RNN) architectures, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) layers, to capture both short-term and long-term dependencies in the data. The dataset incorporates weather features such as temperature, humidity, wind speed, and precipitation, along with temporal factors like government holidays that significantly influence electricity demand. Data pre-processing involves scaling, feature engineering, and a sliding window technique to capture temporal dependencies. The model architecture includes SimpleRNN layers for short-term pattern recognition, followed by LSTM and GRU layers for long-term dependency modeling, with batch normalization and dropout techniques applied to prevent over-fitting and enhance convergence. The results show that the model provides highly accurate load forecasts, which can assist utility companies in resource management, cost reduction, and energy planning in Dhaka city. This study presents a custom-built dataset and a specialized deep learning model, offering a valuable contribution to future advancements in electrical power management systems.**

Index Terms—**Load forecasting, RNN, LSTM, GRU, Temporal dependency, MAE, RMSE**

I. Introduction

Power generating stations must reliably meet consumption demands, requiring load forecasting to predict future electricity needs. This process analyzes historical data to estimate future load requirements and informs capital procurement and energy projections, crucial for effective planning. Accurate forecasting aids in power system planning, infrastructure development, financial management, and efficient operations by predicting electricity sales and addressing gaps between generation capacity and demand, ultimately reducing costs. [1]

Machine learning (ML) techniques, especially RNNs, are widely used for load forecasting. Historical load data is combined with factors like weather and events, cleaned, and analyzed to improve forecasting accuracy. Regional differences and equipment variations are also considered to ensure effective models. [1]

This project focuses on forecasting the Dhaka city grid using an advanced RNN model with LSTM and GRU mechanisms. The dataset integrates various weather variables and government holidays, collected from credible sources, to enhance prediction accuracy and ensure innovative results. The primary contributions of this research are highlighted as follows:

1) Development of meteorological and temporal Data for Forecasting
2) Establish correlations among the data for accurate evaluation
3) Application of RNN, LSTM, and GNN architectures for effective forecasting

II. Key Aspects of Load Forecasting

Load forecasting predicts future electricity demand using historical data and factors like weather and economic patterns. Proper forecasting is essential for effective power generation, distribution, and planning. Important considerations include data preparation, model design, and location-specific variables affecting demand.

A. Load Forecasting Categories and Procedures

Load forecasting provides utilities with efficient resource management by predicting future energy demand using various approaches. Categorizing forecasts into short-term, medium-term, and long-term groups allows utilities to use customized methods for reliable projections that meet their operating demands.

1) Categories of Load Forecasting: There are two categories of load forecasting

1) **Demand Forecasting:** Power consumption is forecast using historical data, trends, and affecting factors. Energy management relies on it to meet consumer demand through utilities and power producers. Weather, economics, calendar, and technology affect accurate predicting. These traits assist utilities in optimizing resource allocation, operational efficiency, and cost while ensuring electricity reliability. Demand projections are essential for energy planning and system stability. [2]
2) **Energy Forecasting:** Energy forecasting optimizes resource management and service reliability by predicting energy demand and supply. It uses machine learning to assess historical data, consumption trends, and weather. Demand forecasting predicts consumption trends, and supply forecasting assesses energy sources, notably renewables. Accurate energy forecasting increases operational efficiency, reliability, and strategic planning, lowering costs and ensuring energy sustainability. [3]

2) Forecasting Procedure: Based on historical data and affecting factors, load demand is projected systematically. For accurate energy management forecasts, it usually involves data collection, pre-processing, model selection, and validation. There are three load forecasting methods.

- **Short Term Load Forecasting (STLF):**Short-term load forecasting predicts energy consumption for a few minutes to a few days. This forecasting helps utilities manage real-time operations, optimize resource allocation, and maintain grid stability. Time series analysis, regression models, and machine learning algorithms use current load data and external factors like weather and exceptional events to increase accuracy. [4]
- **Medium Term Load Forecasting (MTLF):** Medium-term load forecasting (MTLF) lasts one week to two years. MTLF balances demand and generation through maintenance scheduling, load dispatch coordination, and price settlement. This forecasting method helps utilities optimize operations and make resource management decisions in the intermediate term.
- **Long Term Load Forecasting (LTLF):** Long-term load forecasting (LTLF) covers a few years to 10–20 years. System expansion planning—generation, transmission, and distribution—is its main focus. LTLF can also affect new generating unit investments, helping utilities prepare for future energy and infrastructure needs. [5]

B. Factors Affecting Load Forecasting

There are various factories that affect the load forecasting.

1) Meteorological Factors: Critical parameters affecting load forecasts include temperature, humidity, precipitation, wind speed, and cloud cover. Several components are interdependent, including as temperature, cloud cover, precipitation, and humidity. Key factors impacting load predictions include:

- **Weather Factor:** This prediction uses average temperature, maximum temperature, lowest temperature, relative humidity, and rainfall to evaluate meteorological parameters in load forecasting. [6] Weather factors include:
  - Temperature: Due to its impact on electricity demand, temperature is vital to load forecasting. Higher summer temperatures boost air conditioner use, which increases power use. In winter, lower temperatures increase heating system demand and energy usage. This shows the need of appropriately adding temperature data into load forecasting algorithms to predict electricity consumption year-round. [7]
  - Relative Humidity: Impacts energy use via affecting comfort. High humidity increases cooling need in warm areas and decreases heating needs in cool climates.
  - Wind Speed: Changing comfort affects energy use. Higher wind speeds limit summer cooling and boost winter heating. [7]
  - Precipitation; Rain and snow increase interior energy use for heating, lighting, and appliances, which helps load forecasting. Disrupting renewable energy generation's supply-demand balance. Understanding precipitation's influence helps forecasts, especially in weather-sensitive areas.
  - Wind Pressure: Wind pressure influences load forecasting by increasing energy consumption in colder, windy situations due to higher heating demand. It affects wind energy generation efficiency and energy supply. Wind pressure helps predict electricity demand based on wind conditions and energy use.
  - Dew Point: At dew point, air becomes saturated, changing load calculations. High dew points indicate humidity, which increases cooling appliance and electricity use in warm weather. Cold-weather low-dew point air may reduce heating needs. Utility load forecasting systems examine humidity and temperature-based electricity demand fluctuations using dew point.

2) Temporal and Calender Factors: Calendar influences impact load forecasting by affecting daily consumption patterns between seasons and years. Energy demand varies over time, including summer and winter, weekdays and weekends, and festivities like Diwali and New Year, resulting in increased load consumption. Generally, calendar elements fall under these categories:

- **Government Holidays:** Holidays change power demand patterns because home consumption rises for cooking and entertaining, while commercial load falls when enterprises close. Holidays are unpredictable, thus load forecasting models must account for them to match electricity supply with demand.
- **Working Days:** Electricity usage differs between working days and vacations, resulting in different load patterns. Tuesdays to Thursdays have similar electricity use, however Mondays and Fridays often vary owing to weekend breaks. [7]
- **Time Factor:** Due to human and economic activity, electric load peaks during the day and decreases at night. Seasonal, weekend, and hourly factors also affect this fluctuation. Holiday load forecasting is difficult due of its irregularity. Daily load patterns match human activities including work, pleasure, and rest, with lower electricity demand on weekends and holidays than weekdays. [7]

3) Economic Factor: Electricity costs, load control, and industrialization affect average load and peak demand. Forecasting also depends on consumer behavior, tariff rate fluctuations, appliance types, population density, equipment age, and employment levels. These economic factors affect public behavior, load generation, and demand, making them essential for long-term load forecasting. This impacts data collection and model selection. [8]

III. Methodology

This section outlines the approach used to develop a machine learning model for forecasting daily *Max Demand (MW)* using time-series load data and weather information.

A. Data Collection

The dataset is essential for predicting Dhaka's load demand. It contains the 2020 government holidays, maximum demand, and weather information. The required meteorological data was obtained from *Dhaka Weather Data for 2020* [9]. *Government Holidays in Bangladesh 2020* [10] provided information about government holidays. Furthermore, the *Daily Generation Archive* [11] was a source of demand data.Consequently, the following make up the data set: *1. Date, 2. Dew Point, 3. Humidity, 4. Wind Velocity, 5. Pressure, 6. Temperature, 7. Precipitation, 8. Government Holidays, and 9. Max Demand* to forecast Dhaka city's load.

B. Data Preprocessing

The dataset includes historical*Max Demand (MW)* data for the Dhaka district and corresponding weather data (temperature, humidity, etc.). The load data consists of daily maximum demand values, while weather features are obtained from public weather databases. The following preprocessing steps were applied:

- **Scaling:** All features were normalized using MinMax scaling to ensure uniformity across the input features, particularly for the weather data.
- **Feature Engineering:** Temporal features such as the day of the week, month, and whether the day was a holiday were added. Weather features like temperature, humidity, and wind speed were also included.
- **Sliding Window:** A sliding window technique was used to capture temporal dependencies. A sequence of past load values over a fixed window size was used as input, with the next day's*Max Demand (MW)* as the target.

C. Model Architecture

In this work, we designed a deep recurrent neural network architecture combining SimpleRNN, LSTM, and GRU layers to capture both short-term and long-term dependencies in the load demand time series data. The model is built in a sequential manner, where each layer serves a specific purpose. Below is a detailed explanation of each layer and its role in the architecture.

TABLE I: Model Architecture Details

Layer (Type)	Output Shape	Param #
SimpleRNN (SimpleRNN)	(21, 250)	63,000
BatchNormalization	(21, 250)	1,000
Dropout (Dropout)	(21, 250)	0
SimpleRNN_1 (SimpleRNN)	(21, 200)	90,200
BatchNormalization_1	(21, 200)	800
Dropout_1 (Dropout)	(21, 200)	0
LSTM (LSTM)	(21, 100)	120,400
BatchNormalization_2	(21, 100)	400
Dropout_2 (Dropout)	(21, 100)	0
GRU (GRU)	(50)	22,800
BatchNormalization_3	(50)	200
Dropout_3 (Dropout)	(50)	0
Dense (Dense)	(1)	51
Total Parameters	298,851	

1) SimpleRNN Layers: The model begins with two *SimpleRNN* layers, which are used to capture short-term dependencies in the sequential data. These layers allow the model to process sequences step by step by maintaining a hidden state that evolves over time. The equations governing *SimpleRNN* are given by:

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b) \tag{1}$$

where h_t is the hidden state at time step t, and W_x and W_h are the weight matrices [12]. In our architecture, the first SimpleRNN layer has 250 units, followed by a second *SimpleRNN* layer with 200 units. These layers are responsible for learning temporal patterns over short time intervals. The total number of parameters for these layers are 63,000 and 90,200, respectively, which include the recurrent weights and biases.

2) Batch Normalization: After each SimpleRNN layer, we apply *BatchNormalization* to stabilize and accelerate training. Batch normalization normalizes the output of the preceding layer by scaling and shifting the activations to ensure that they have zero mean and unit variance. This is particularly useful in recurrent architectures, as it helps mitigate internal covariate shifts and allows the model to converge faster. The parameter count for the batch normalization layers are 1,000 and 800 for the first and second SimpleRNN outputs, respectively.

3) Dropout Layers: To prevent over-fitting, *Dropout* is applied after each batch normalization layer. Dropout is a regularization technique that randomly drops a fraction of neurons during training, forcing the network to generalize better. In our architecture, dropout with a rate of 20% is applied to all recurrent layers. Dropout does not have any learnable parameters, and its function is solely to improve model robustness.

4) LSTM Layer: The third recurrent layer in the model is an *LSTM* layer with 100 units. LSTM (Long Short-Term Memory) networks are designed to capture long-term dependencies by maintaining a cell state along with hidden states. The cell state allows the network to retain information over long sequences. The update equations for the LSTM layer are:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{3}$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{4}$$

$$h_t = o_t \cdot \tanh(C_t) \tag{5}$$

where f_t, i_t, and o_t are the forget, input, and output gates, respectively. The LSTM layer captures long-term temporal dependencies, making it essential for accurate forecasting [13]. The number of parameters in this LSTM layer is 120,400.

5) GRU Layer: Following the LSTM layer, we include a *GRU* layer with 50 units. GRU (Gated Recurrent Unit) is a simplified version of LSTM that combines the forget and input gates into a single update gate, reducing the number of parameters while still addressing the vanishing gradient problem [14] [15]. The GRU update equations are:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{6}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{7}$$

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tanh(W_h \cdot [r_t \cdot h_{t-1}, x_t] + b_h) \tag{8}$$

where z_t is the update gate and r_t is the reset gate. GRU is used to further refine the temporal patterns and improve model efficiency. This layer has 22,800 parameters.

6) Final Dense Layer: The final layer of the architecture is a *Dense* layer with a single output unit. This layer serves as the output layer of the network, responsible for producing the final prediction. In our case, the Dense layer outputs a single scalar value representing the predicted maximum demand for a future time step. The number of parameters in this layer is 51.

7) Summary: The model architecture consists of multiple recurrent layers, each designed to capture different aspects of temporal dependencies in the data. The use of SimpleRNN layers helps with short-term patterns, while LSTM and GRU layers capture long-term dependencies. Batch normalization and dropout are applied throughout the model to improve generalization and stability. The final Dense layer outputs the prediction for load demand, completing the sequence-to-point forecasting task.

D. Training and Hyper-parameter Tuning

The dataset was first scaled using standard scaling techniques to ensure all features had comparable ranges. For testing purposes, the dataset was split into training and test sets using an 80-20 train-test split.

The training data was further used for K-fold cross-validation, while the test data was held out and used exclusively for final performance evaluation. This approach ensured the model's performance was assessed on unseen data.

The model was implemented using Python 3 with TensorFlow. We divided the dataset into training and validation sets for cross-validation to evaluate the model's performance during training. To further assess generalization ability and prevent over-fitting, we employed K-fold cross-validation on the training set.

Hyper-parameter tuning was conducted using techniques such as grid search or random search, focusing on optimizing key parameters, including the learning rate, batch size, and the number of neurons in each layer. The learning rate was set to $1e-3$, and the model was trained for a maximum of 100 epochs with a batch size of 32. Early stopping was applied with a patience of 5 epochs to mitigate the risk of over-fitting.

E. Evaluation Metrics

The model's performance was primarily evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics were selected due to their direct relevance in minimizing forecast error. Special emphasis was placed on the validation RMSE ('val_rmse') during model tuning to ensure minimal over-fitting and robust model performance.

In addition to validation performance, the final evaluation of the model was conducted on the test set. The test MAE and test RMSE were calculated as follows:

$$\text{Test MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_{\text{true},i} - y_{\text{pred},i}| \tag{9}$$

$$\text{Test RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_{\text{true},i} - y_{\text{pred},i})^2} \tag{10}$$

These metrics provided a comprehensive assessment of model performance on unseen data, highlighting its ability to generalize beyond the training set. Special attention was given to minimizing the test RMSE to ensure a high level of prediction accuracy on future load demand scenarios.

IV. Results and Analysis

A. Correlation Analysis

To better understand the relationship between the dependent variable, *Max Demand (MW)*, and other features, we performed a correlation analysis. The correlation coefficient measures the strength and direction of the linear relationship between two variables, with values ranging from -1 to 1. A positive value indicates a direct relationship, while a negative value indicates an inverse relationship.

TABLE II: Correlation of *Max Demand (MW)* with other features

Feature	**Correlation with *Max Demand (MW)***
Dew Point (°C)	0.550
Humidity	0.522
Month	0.412
Time	0.408
Temperature (°C)	0.254
Precipitation (mm)	0.111
Wind Speed (km/h)	0.075
Government Holidays	-0.313
Pressure (inch Hg)	-0.458

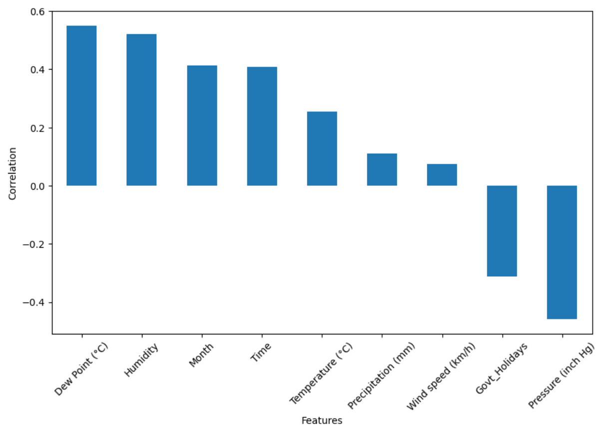

Fig. 1: Correlation with Columns

As seen in Table II and Figure 1, the highest positive correlations are observed with *Dew Point (°C)* (0.550), *Humidity* (0.522), and *Month* (0.412), indicating a significant positive relationship with *Max Demand (MW)*. These variables are likely to increase as demand increases.

On the other hand, *Pressure (inch Hg)* (-0.458) and *Government Holidays* (-0.313) exhibit negative correlations, implying that higher atmospheric pressure or government holidays tend to reduce the demand for electricity.

Interestingly, variables like *Temperature (°C)* (0.254) and *Precipitation (mm)* (0.111) show weak correlations, indicating they may have a limited impact on the load demand.

This correlation analysis provides valuable insight into which features should be emphasized in the load forecasting model.

B. Temporal Dependency Analysis

In time series forecasting, temporal dependency on previous values plays a critical role in predicting future values. To investigate the temporal dependency of the 'Max Demand(MW)' on its own previous values, we calculated the correlation of the target variable with its lagged versions. Specifically, we explored the correlations for lags ranging from 1 to 60 time steps.

The results of this analysis, shown in Fig. 2, indicate that the highest correlation occurs at a lag of 1 time step, with a value of 0.838. This suggests that the most recent demand value is a strong predictor of the next demand value, which aligns with the intuition that electricity demand on consecutive days tends to be closely related.

As the lag increases, the correlation gradually decreases, confirming that more distant past values of the target variable have a diminishing impact on predicting the next day's demand. This temporal dependency analysis is essential in defining the lookback period used in our recurrent neural network models to capture the sequence-based patterns effectively.

C. Experimental Setup

The experiments were conducted using a local machine with a GPU. The RNN model was trained on the preprocessed

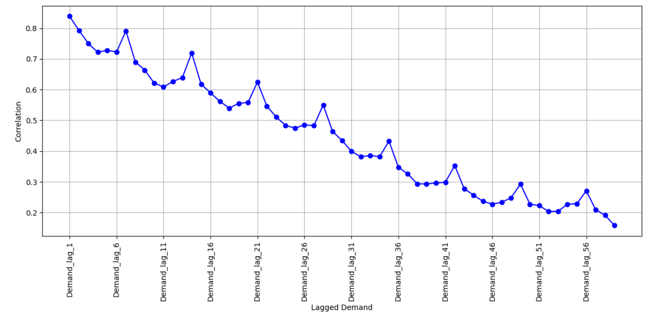

Fig. 2: Correlation of Max Demand with Lagged Demand

dataset using a batch size of 1 and a learning rate of 0.1 with momentum (SGD optimizer). The model's performance was evaluated using k-fold cross-validation with $k = 5$. Each fold was trained for 100 epochs, utilizing early stopping to prevent overfitting.

D. Results

The trajectory of the model's performance is visualized in Figure 3, where the training process across different epochs is shown. The performance stability of the model across the validation sets is further emphasized. Additionally, the final bar comparison of key metrics is illustrated in Figure 4, providing a clear comparison between training and validation performance across different folds.

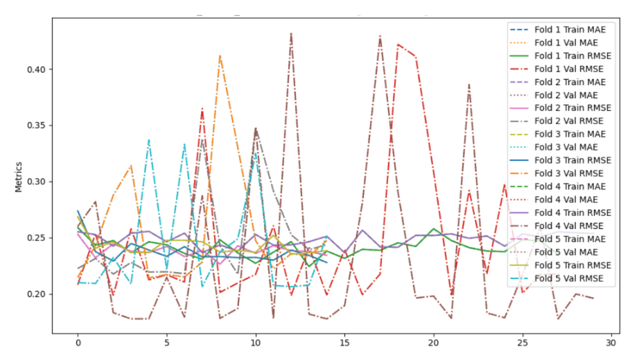

Fig. 3: Trajectory of the model's performance.

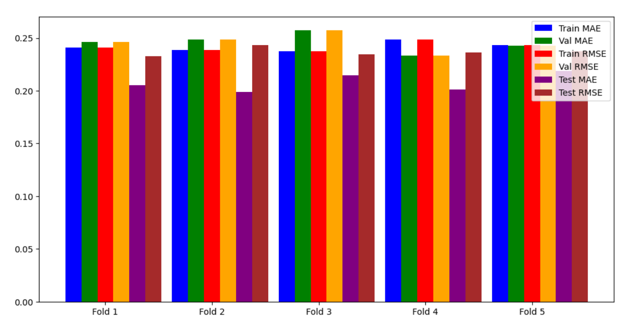

Fig. 4: Final bar comparison of metrics.

Table III presents the performance metrics (*MAE* and *RMSE*) across different folds for train, validation, and test sets. Among

TABLE III: Performance metrics across different folds

Fold	Train		Validation		Test	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Fold 1	0.241049	0.241049	0.246077	0.246077	0.205082	0.232746
Fold 2	0.238391	0.238391	0.248630	0.248630	0.198855	0.243009
Fold 3	0.237379	0.237379	0.257489	0.257489	0.214602	0.234763
Fold 4	0.248621	0.248621	0.233491	0.233491	0.201212	0.236118
Fold 5	0.243011	0.243011	0.242786	0.242786	0.218585	0.237589
Average	0.241690	0.241690	0.245695	0.245695	0.207667	0.236845
Standard Deviation	0.004463	0.004463	0.008736	0.008736	0.008563	0.003880

the folds, Fold 2 demonstrates the best performance on the test set with the lowest *MAE* (0.198855) and *RMSE* (0.243009), indicating it generalizes better compared to other folds. The average performance across all folds shows consistent training, validation, and test results, with relatively low standard deviation, signifying stable performance across the cross-validation process.

V. Conclusion

This paper presents a machine learning approach for forecasting electricity demand in Dhaka city, utilizing temporal dependencies and external factors such as weather conditions and government holidays. By integrating *LSTM* and *GRU* mechanisms in an advanced *RNN* model, we effectively predict *Max Demand (MW)*.

Our correlation analysis reveals significant positive relationships with *Dew Point* and *Humidity*, while showing negative correlations with *Pressure* and *Government Holidays*.

The experimental results indicate strong performance and minimal variation in training and validation metrics, reflecting excellent generalization and stability. This model provides an accurate and scalable solution for load forecasting in the power sector. Future work will aim to incorporate more external factors to enhance its applicability in dynamic environments.

References

[1] A. A. Mamun, M. Sohel, N. Mohammad, M. S. Haque Sunny, D. R. Dipta, and E. Hossain, "A comprehensive review of the load forecasting techniques using single and hybrid predictive models," *IEEE Access*, vol. 8, pp. 134 911–134 939, 2020.

[2] A. Acar, B. Kocaoğlu, and B. Yılmaz, "Demand forecast, up-to-date models, and suggestions for improvement an example of a business," *Journal of Global Strategic Management*, vol. 8, pp. 26–37, 06 2014.

[3] A. Mystakidis, P. Koukaras, N. Tsalikidis, D. Ioannidis, and C. Tjortjis, "Energy forecasting: A comprehensive review of techniques and technologies," *Energies*, vol. 17, no. 7, 2024. [Online]. Available: https://www.mdpi.com/1996-1073/17/7/1662

[4] S. Akhtar, S. Shahzad, A. Zaheer, H. S. Ullah, H. Kilic, R. Gono, M. Jasiński, and Z. Leonowicz, "Short-term load forecasting models: A review of challenges, progress, and the road ahead," *Energies*, vol. 16, no. 10, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/10/4060

[5] M. Kamran, "Chapter 8 - energy statistics and forecasting for smart grids," in *Fundamentals of Smart Grid Systems*, M. Kamran, Ed. Academic Press, 2023, pp. 365–392. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780323995603000077

[6] S. Di, "Power system short term load forecasting based on weather factors," in *2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM)*, 2020, pp. 694–698.

[7] S. Khatoon, Ibraheem, A. K. Singh, and Priti, "Effects of various factors on electric load forecasting: An overview," in *2014 6th IEEE Power India International Conference (PIICON)*, 2014, pp. 1–5.

[8] M. Rostum, A. Zamel, H. Moustafa, and I. Ziedan, "Electrical load forecasting: A methodological overview," *International Journal of Engineering Technology*, vol. 9, p. 842, 10 2020.

[9] Weather and Climate, "Dhaka weather data for september 2020," https://weatherandclimate.com/dhaka/september-2020, accessed: 2024-09-22.

[10] O. Holidays, "Government holidays in bangladesh 2020," https://www.officeholidays.com/countries/bangladesh/2020, accessed: 2024-09-22.

[11] B. P. D. Board, "Daily generation archive," https://misc.bpdb.gov.bd/daily-generation-archive, accessed: 2024-09-22.

[12] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111.

[15] A. Baul, G. C. Sarker, P. Sikder, U. Mozumder, and A. Abdelgawad, "Data-driven short-term load forecasting for multiple locations: An integrated approach," *Big Data Cogn. Comput.*, vol. 8, no. 2, p. 12, 2024. [Online]. Available: https://www.mdpi.com/2504-2289/8/2/12