Evaluating RNN-Based Models for Next-2-Hour Energy Load Prediction

Prepared By:

Sana Ambreen

Namrata Sood

Yitian Liu

Priyadarshini Balasubramanian

University of North Texas

Toulouse Graduate School (College of Science)

ADTA 5560

Dr. Mehmet Orhan

Fall 2025

INTRODUCTION

Project Overview

This project aims to forecast the next two hours of electricity demand in the Electric Reliability Council of Texas (ERCOT) grid by comparing different Recurrent Neural Network (RNN) architectures. Electricity load is a time series influenced by weather, seasonal behavior, population patterns, and sudden events. Because of all this, RNN-based models, especially LSTM and GRU, are widely used to identify dependencies and evolving patterns.

This study uses five years of hourly ERCOT load data from 2020 to 2025, including eight regional load features and total system demand. The goal is to determine which RNN architecture yields the lowest RMSE for short-term load prediction and whether including regional features improves forecasting accuracy compared to using only the total load. By comparing univariate and multivariate models, the project aims to assess the extent to which regional information adds predictive value to next-2-hour forecasts.

Problem Statement

There is a requirement in the energy sector to accurately forecast the short-term electricity demand to provide the end users with a stable, cost-efficient power system. There should be a continuous balance between electricity generation and consumption by the grid operators.

The energy sector is prone to forecasting errors. There is a direct impact on end users due to forecasting errors, leading to blackouts, overloads, and costly power purchases. For example, a utility company's short-term model gives an inaccurate demand forecast of 50,000 MW. Due to weather changes, demand surges to 55000 MW, so the company would buy from extremely expensive emergency sources (gas-fired peaking plants or the market). This leads to an increase in the utility's operational costs and higher rates for end users as well. The main goal is to provide reliable, affordable, and sustainable electricity for all members of society.

Our group project aims to identify the optimal RNN architecture with the lowest RMSE to accurately predict the amount of power the Texas grid (ERCOT) will need over the next 2 Hours. Additionally, we want to know whether giving the model detailed regional power-usage data helps it make better predictions than just looking at the total power use for the whole state.

Research Questions

- Which RNN-based architecture produces the lowest RMSE for next-2-hour grid demand forecasting?
- Does including regional load features improve forecasting accuracy over using total ERCOT load alone?

Objectives

1. Implement and train Vanilla RNN, LSTM, Bidirectional LSTM and GRU using the ERCOT data.

Use the RMSE to determine the best model for accurate next-2-hour prediction of load.

2. Test the impact of regional features by developing comparative models-Univariate (ERCOT load) and Multivariate (all regional loads)

Significance and Applications

Accurate short-term electricity demand forecasting is essential for maintaining a stable, reliable, and cost-efficient power system. Grid operators must continuously balance generation with consumption, and even small forecasting errors can lead to overloads, blackouts, or expensive emergency power purchases. By identifying the best RNN architecture for forecasting next 2-hour predictions, it helps utilities optimize operations, supports renewable integration, reduces operational costs, and ultimately ensures that consumers and society receive safe, dependable, and sustainable electricity.

This project's findings would provide significant benefits in the energy sector value chain and benefit the following groups:

1. **Grid Operators & Utilities:** These people are responsible for providing uninterrupted electricity to homes, businesses, hospitals, industries, and public infrastructure. They must constantly balance the electricity being generated with the electricity being consumed, hour by hour. Accurate next-2-hour forecasting helps them schedule power generation more efficiently, anticipate sudden changes in demand, and prevent overloads or blackouts. Better forecasting reduces operational costs, minimizes the risk of system failures, and improves the overall reliability and stability of the power system.
2. **Renewable Energy Planners:** Renewable energy planners rely on accurate short-term load forecasts to coordinate renewable generation, energy storage systems, and grid support mechanisms. Improved RNN-based forecasting helps them maintain balance between supply and demand while increasing the share of clean energy in the grid.
3. **Smart Grid & IoT System Designers:** Modern smart grids depend on advanced technologies such as smart meters, automated demand-response systems, home energy management platforms, and IoT-based monitoring devices. These systems require highly accurate load forecasts to make real-time adjustments, optimize energy use, and reduce wastage.
4. **Researchers and Data Scientists:** Researchers and data scientists benefit from understanding how different RNN architectures compare in terms of accuracy, efficiency, and stability for next-hour and next-2-hour predictions. It can guide in future studies, algorithm improvements, and practical applications in the energy domain.
5. **Consumers & Society:** Accurate load forecasting leads to fewer outages, more stable electricity prices, and better service quality. A well-forecasted grid benefits all members of society by ensuring reliable, affordable, and sustainable access to electricity.

Literature Review

Recent studies have explored advanced deep learning methods to improve short-term electricity load forecasting for ERCOT and similar power grids.

1. Shohan et al., 2022introduced a Hybrid LSTM–Neural Prophet model that separates linear and non-linear patterns, achieving the lowest RMSE for hour-ahead ERCOT forecasting, proving that combining models can boost accuracy.
2. Fayyazbakhsh et al., 2025 compared LSTM, SVR, and ensemble models, finding that the LSTM alone performed slightly better, especially in capturing peak load patterns that are critical for grid reliability.
3. Yang et al., 2024analyzed weather and time-based features in RNN models for ERCOT forecasting and concluded that GRU models outperformed LSTM in minimizing error, showing their efficiency for short-term predictions.
4. Lastly, Hasanat et al., 2024 proposed a CNN - GRU hybrid model that effectively extracts spatial and temporal patterns from data, achieving the lowest RMSE, MAE, and MAPE compared to other deep learning models.

Overall, while LSTM models are reliable, hybrid and GRU-based architectures often deliver higher accuracy and lower error rates, especially for complex short-term forecasting tasks such as predicting ERCOT's next-hour load.

This research differs from existing work in a few important ways. Most previous studies focused on one-hour-ahead forecasting, while this project predicts the next two hours, giving grid operators more lead time for decision-making. Many earlier models relied on large multivariate feature sets, including weather data or regional loads, but our results showed that a simpler univariate Bidirectional LSTM performed better than more complex multivariate models. This challenges the common assumption that adding more features always improves accuracy. Finally, unlike most research that stops model evaluation, this project builds a full real-time deployment pipeline using APIs, FastAPI, and Tableau, demonstrating how a forecasting model can operate continuously with live ERCOT data. This makes the work more practical and directly usable in real-world grid monitoring.

DATA AND DATA ANALYSIS

DATA GENERATION

This dataset was chosen because it is large-scale and time-based, making it suitable for testing Recurrent Neural Network models, which are designed to learn from sequential data and make time-based predictions. It directly aligns with the project goal of predicting next-2-hour electricity demand using deep learning. The data is recorded every hour, making it ideal for studying overtime. It has many related columns that show power usage across different regions, which helps the model learn both temporal and regional patterns. It also spans 5 years (2020-2025), providing ample data to train and test the models accurately.

SOURCE

The dataset is taken from the Electric Reliability Council of Texas (ERCOT) official website (ercot.com). ERCOT manages the flow of electric power for most of Texas, and it publishes real-time and historical electric load data. They publish this data to show how much electricity people use in different regions and at different times. The dataset is official, real, and regularly updated, making it trustworthy and accurate. Their data is often used by researchers, energy companies, and engineers to study power demand, predict usage, and improve grid stability.

DESCRIPTION:

The dataset contains hourly electricity demand (in megawatts) for different regions in Texas from 2020 to 2025. It contains 50399 rows and 10 columns. Each row represents the total energy usage in a specific hour in a specific region.

DATA DICTIONARY

Variable_Name	**Description**	**Type**
Hour_Ending	Timestamp representing the end of each hourly interval.	Datetime
COAST	Hourly electricity load for the Coast region.	Numeric
EAST	Hourly electricity load for the East region.	Numeric
FWEST	Hourly electricity load for the Far West region.	Numeric
NORTH	Hourly electricity load for the North region.	Numeric
NCENT	Hourly electricity load for the North Central region.	Numeric

SOUTH	Hourly electricity load for the South region.	Numeric
SCENT	Hourly electricity load for the South Central region.	Numeric
WEST	Hourly electricity load for the West region.	Numeric
ERCOT *(Target Variable)*	Total system-wide electricity load across all ERCOT regions.	Numeric

Hour Ending- Date and Time for each hour.

Regional Loads - COAST, EAST, FWEST, NORTH, NCENT, SOUTH, SCENT, WEST showing how much power was used in each ERCOT region.

Ercot -The total system load across all regions.

The target variable here is the ERCOT, which shows the overall energy usage of the entire state at that particular time.

The goal of the project is to predict the ERCOT value for the next two hours using the historical values. The model learns from regional load inputs to forecast total demand shortly thereafter. This target variable is perfect for time series forecasting, because it changes over time and depends on historical demand patterns, time of day, and region-wise usage. In addition to forecasting total ERCOT demand, the same model can be trained to predict electricity usage region-wise. This helps identify region-specific demand patterns and supports localized energy planning across Texas.

EXPLORATORY DATA ANALYSIS

We have selected a unified time-series dataset and performed Exploratory Data Analysis (EDA) on the dataset. The ERCOT data set includes 50,399 hourly records across 10 features. EDA was performed to ensure data quality, understand temporal patterns, and prepare the data for subsequent modeling. The primary objectives of the EDA were to validate the time-series integrity, handle any missing values, and analyze the seasonal and long-term trends present in the ERCOT electrical load data.

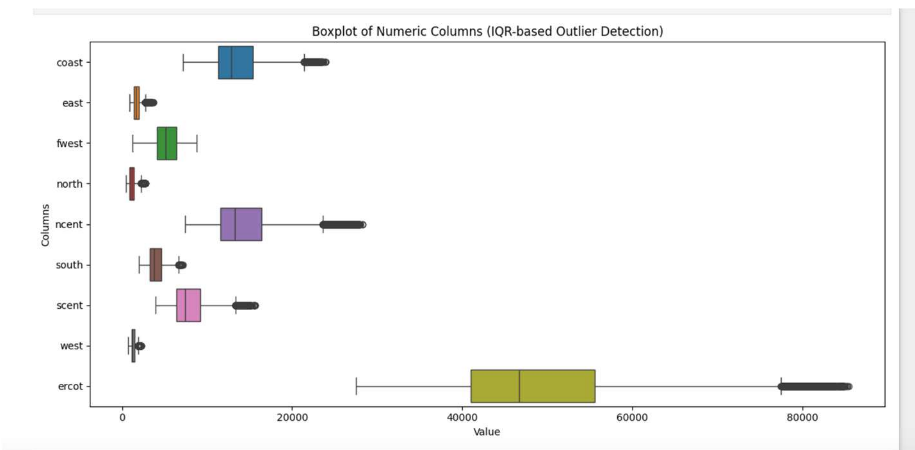
Descriptive Statistics:

```
df[numeric_cols].describe().T
```

	count	mean	std	min	25%	50%	75%	max
coast	50399.0	13588.539137	3031.431797	7128.496204	11333.033987	12830.923243	15359.422892	23963.415268
east	50399.0	1713.455953	427.070989	863.809399	1411.928467	1602.427675	1941.242421	3696.721811
fwest	50399.0	5280.179574	1348.545655	1242.334939	4090.090082	5117.274624	6372.041735	8773.080101
north	50399.0	1166.747913	405.423039	488.272293	844.005142	1099.660684	1413.635734	2732.085977
ncent	50399.0	14356.071007	3958.840715	7404.964154	11583.591723	13254.891917	16402.920821	28312.541008
south	50399.0	3945.253615	949.679577	1958.876190	3208.507217	3765.588877	4576.872843	7155.621786
scent	50399.0	7953.338252	2160.546604	3955.451332	6355.435654	7434.263629	9159.492725	15664.703570
west	50399.0	1289.408360	264.479334	735.793806	1099.808103	1220.942179	1428.124117	2230.497660
ercot	50399.0	49294.216672	11298.194328	27491.364599	40986.340961	46688.425731	55602.964596	85464.116394

Missing Values: Only 1 missing for datetime, date, year, month, hour, so we can drop this 1 row safely

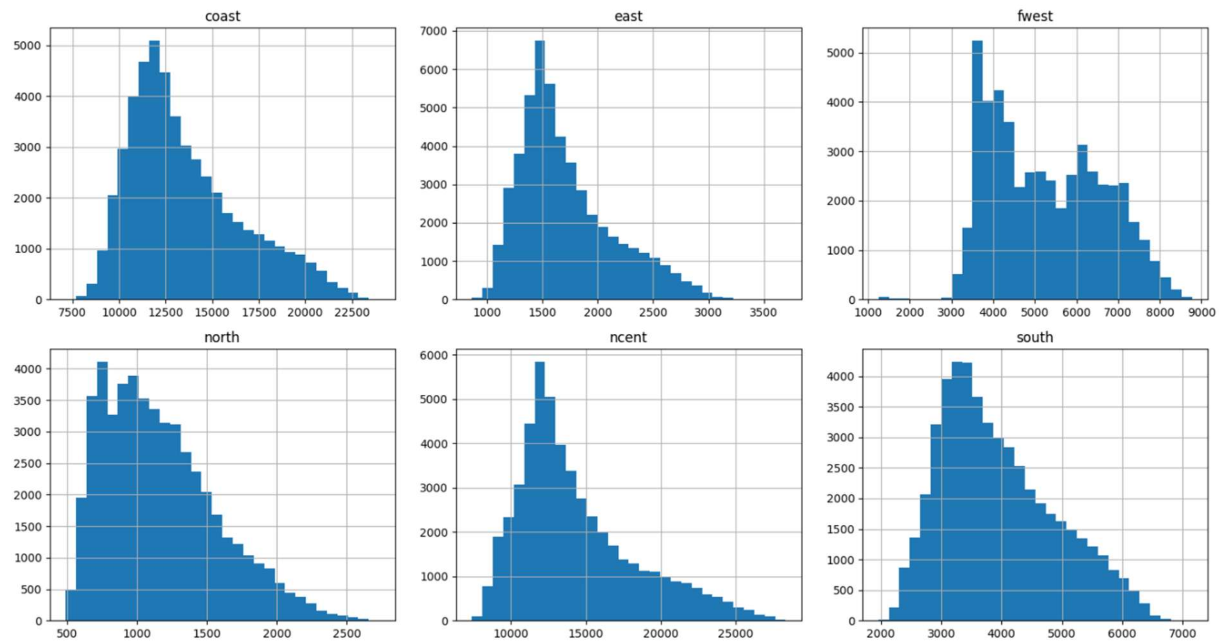
Outliers:

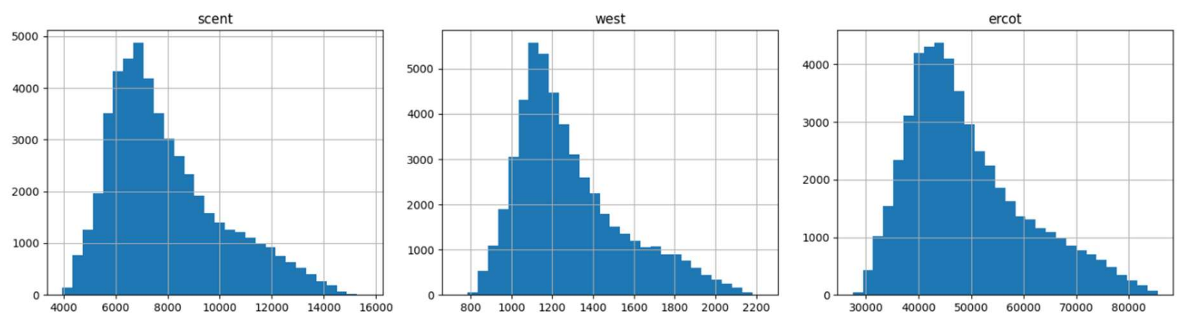


IQR-based outlier detection identifies some extreme values in ERCOT and regional load series. Because these represent real operational conditions and not data errors, they are not removed during cleaning. Instead, they are retained to preserve the integrity of the time-series structure. Outliers are only tagged for reference and potential downstream modeling choices.

Understanding Seasonality and Understanding Trends: Analyze load data by year, month, day, or hour for time-series forecasting and modeling.

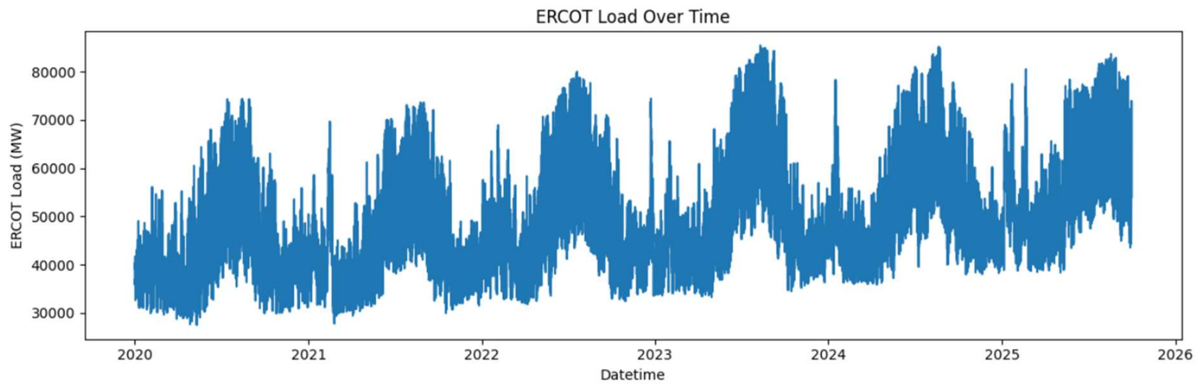
Distribution of Regional Loads





The load distributions across all regions are right-skewed, meaning most hours have moderate demand while extreme high-load events occur less frequently but still significantly impact the grid. Regions like NCENT and SCENT show the highest variability and widest range, reflecting large population centers and high industrial activity. In contrast, FWEST has a more irregular shape, likely influenced by renewable generation patterns and lower baseline demand.

ERCOT Load Over Time



1.Strong Seasonal Pattern (Yearly Cycle):

Lowest demand in January–March, Sharp peak in June–September

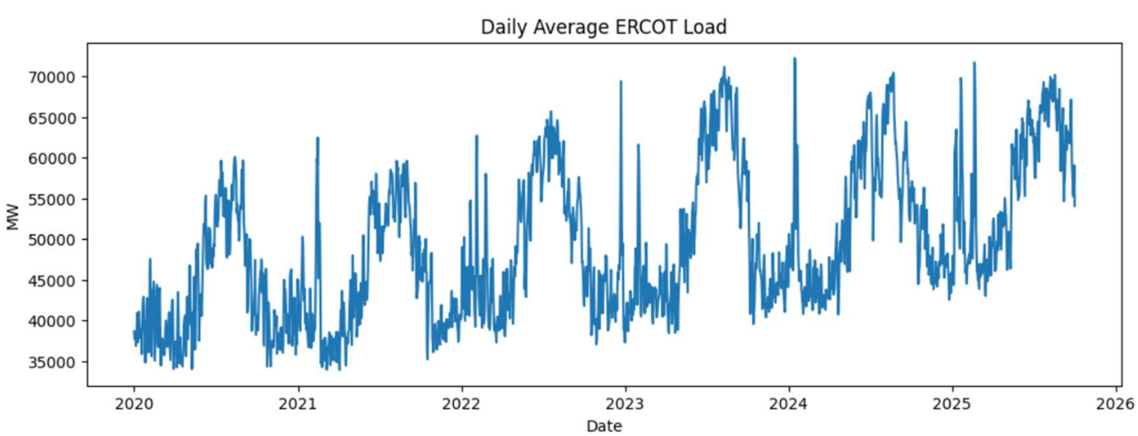
2. There are noticeable spikes and anomalies in certain years:

1) One or more abrupt peaks or drops in early 2021. During the freeze, ERCOT experienced grid stress and extreme demand.

2)Texas has had multiple record heatwaves causing (in 2022, 2023, 2024) very high summer loads.

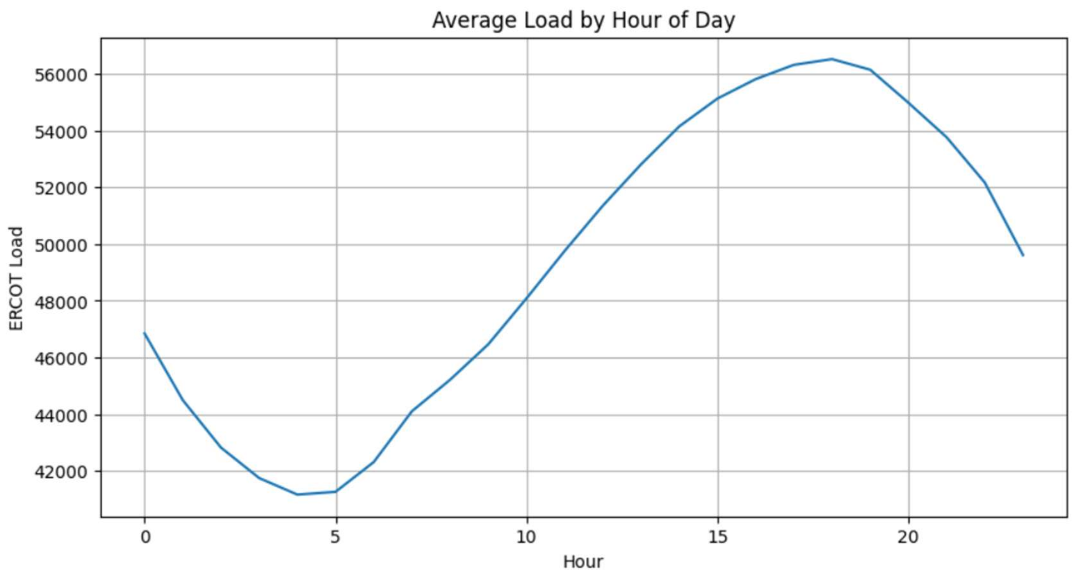
3. COVID-19 period (2020): Reduced commercial and industrial use, more residential use (remote work)

Daily_Average_ERCOT_Load



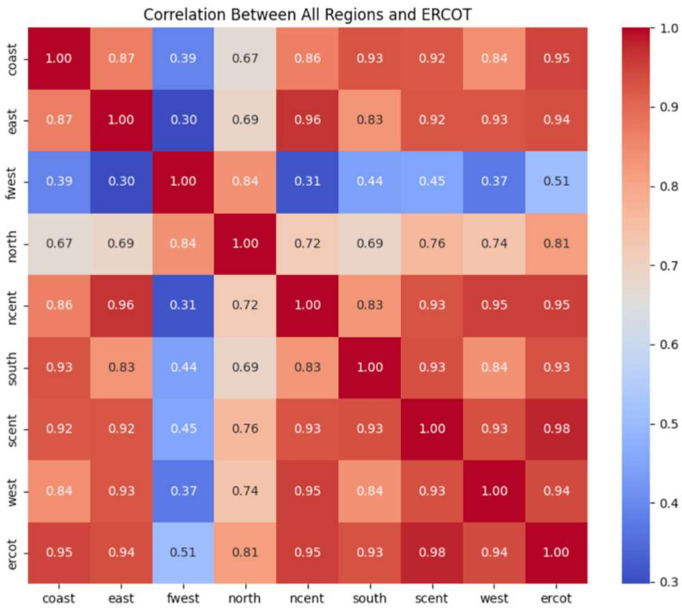
The daily average ERCOT load shows a clear seasonal pattern, with electricity demand rising sharply during summer months and falling during winter. Across the five-year period, summer peaks grow progressively higher, reflecting population growth and increasingly extreme heat events in Texas. Occasional sharp spikes represent major weather or grid-stress events and highlight the system's sensitivity to extreme conditions.

Average load by hour of day



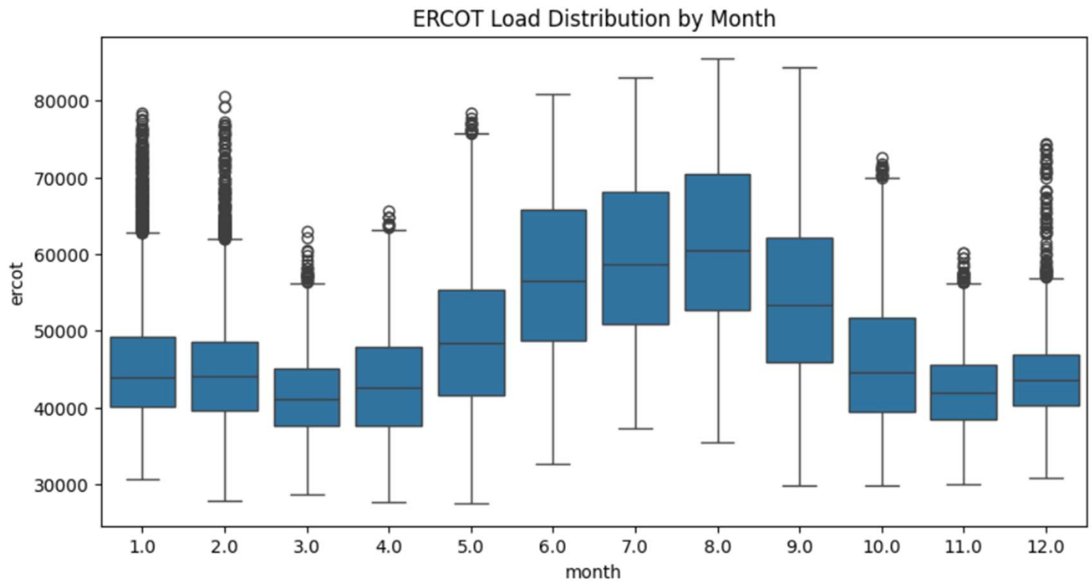
The average ERCOT load by hour of day shows a clear daily cycle, with demand lowest between 3–6 AM and rising steadily throughout the morning. Electricity usage peaks around 4–6 PM, reflecting afternoon heat and residential cooling demand.

Correlation between Regions



The correlation heatmap shows that most ERCOT regions are highly correlated with each other and with the total system load, especially NCENT, COAST, SOUTH, and SCENT, which exhibit correlations above 0.90 with ERCOT. FWEST is the only region with noticeably lower correlations (0.30–0.50), reflecting its unique and steadily increasing trend driven by energy production activity rather than traditional demand patterns.

ERCOT Load by Month



The monthly distribution plot highlights pronounced seasonality, with ERCOT loads peaking in the summer months (June–August) and reaching their lowest levels in the winter and early spring. Summer months not only have higher medians but also much wider spreads and more extreme outliers, reflecting intense and variable cooling demand during heat events. In contrast, winter show lower and more stable load distributions, indicating more consistent electricity usage outside of peak cooling season.

METHODOLOGY

Overview of the Modeling Approach

For this project, we chose Recurrent Neural Network (RNN)-based deep learning models because ERCOT load data is sequential, highly seasonal, and depends on past temporal patterns. We implemented and compared four architectures:

- Vanilla RNN
- LSTM (Long Short-Term Memory)
- BiLSTM (Bidirectional LSTM)
- GRU (Gated Recurrent Unit)

Each model was trained to predict the next 2 hours of ERCOT electricity demand, both in:

1. Univariate mode -only the ERCOT total load is used
2. Multivariate mode**-** all 8 regional loads + datetime features (hour, day, month)

This setup let us analyze not just which model performs best, but also whether regional information improves forecasting accuracy.

RNNs work well for ERCOT load because the electricity demand changes in predictable patterns over time. It rises and falls each day, follows weekly routines, shifts with the seasons, and depends a lot on what the demand was a few hours before. Weather and human habits also affect it, but usually with a delay. Since RNNs remember what happened in previous time steps, they can naturally learn these patterns instead of treating each moment on its own.

When we say the model predicts

$$\mathbf{y_{t+1:t+2} = f(y_t, y_{t-1}, \ldots, y_{t-24})},$$

we just mean it looks at the past 24 hours of data to figure out what the next few hours will look like. A 24-hour window makes sense because electricity usage follows a daily rhythm, so giving the model a full day of history helps it make better predictions.

MLPs and normal regression models don't work well here because they look at each data point on its own. They don't understand what happened an hour ago affects what happens next.

ARIMA and SARIMA can handle time series, but they fall apart from big datasets or multiple regions. They also don't deal well with complex, nonlinear patterns, and you have to spend a lot of time tuning them.

Transformers are super powerful, but they're too heavy for this job. They need a lot of compute, a lot of data, and they're meant for much bigger forecasting tasks. For a simple 2-hour forecast, they're more than you need.

RNNs sit right in the middle - they understand time, they run efficiently, and they're easier to work with. That makes them a great fit for this project.

Methodology Details

The data is prepared using a sliding-window setup where each training example looks at the past 24 hours to predict the next 2 hours. For every position in the time series, we take the past 24 time steps as the input window (using only ERCOT load for the univariate models, and all available features for the multivariate models), and the values y_{t+1} and y_{t+2} of the ERCOT load as the targets. In code, this is done by appending arr[i:i+window] to X and arr[i+window:i+window+horizon] to y as we slide through the dataset. After collecting all samples, the input is reshaped into the format RNNs expect: (samples, 24, num_features), meaning every example is a sequence of 24 time-steps. Before training, the data is scaled with MinMaxScaler fit only on the training split which helps keep the values in a stable range so the RNN can learn smoothly without exploding or vanishing gradients.

Model Architecture

Vanilla RNN: This model is trained using a single feature – ERCOT load over a 24-hour sliding window to predict the next 2-hour ERCOT load. The architecture is as follows:

- Simple RNN layer (64 units, return_sequences=True)
- Dropout layer (0.3)
- Simple RNN layer (32 units)
- Dense layer with 32 ReLU units
- Output dense layer with 2 neurons (predicting next-2-hour load)

Vanilla RNN (Multivariate): The multivariate Vanilla RNN is trained using 12 input features (8 regional loads, total ERCOT load, and 3 datetime features: hour, day of week, month) over a 24-hour sliding window. The model predicts the next 2 hours of ERCOT demand.

- Simple RNN layer (64 units, return_sequences=True, input_shape=(window, 12))
- Dropout layer (0.3)
- Simple RNN layer (32 units)
- Dense layer with 32 ReLU units
- Output dense layer with 2 neurons (predicting next-2-hour load)

Long Short-term Memory (Univariate): This model is trained using a single feature – ERCOT load over a 24-hour sliding window to predict the next 2-hour ERCOT load. The architecture is as follows:

- LSTM layer (64 units, return_sequences=True)
- Dropout layer (0.3)
- LSTM layer (32 units)
- Dense layer with 32 ReLU units
- Output dense layer with 2 neurons

Long Short term Memory (Multivariate) : The multivariate LSTM is trained using 12 input features (8 regional loads, total ERCOT load, and 3 datetime features: hour, day of week, month) over a 24-hour sliding window. The model predicts the next 2 hours of ERCOT demand. The architecture is as follows:

- LSTM layer (64 units, return_sequences=True, input_shape=(window, 12))
- Dropout layer (0.3)
- LSTM layer (32 units)
- Dense layer with 32 ReLU units
- Output dense layer with 2 neurons

Bidirectional LSTM (Univariate): The univariate Bidirectional LSTM model uses only the historical ERCOT total load. Each training sample consists of a 24-hour sliding window (scaled with MinMaxScaler), reshaped into a single-feature sequence. The model predicts the next 2 hours of ERCOT demand.

- Bidirectional LSTM layer (64 units, return_sequences=True)
- Dropout layer (0.3)
- Bidirectional LSTM layer (32 units)
- Dense layer with 32 ReLU units
- Output Dense layer with 2 neurons (predicting next-2-hour load)

Bidirectional LSTM (Multivariate): In the multivariate setup, the model receives 24-time steps × 12 features, including all regional loads and datetime features (hour, day of week, month). The goal is to predict the next 2 hours of ERCOT demand.

Architecture:

- Bidirectional LSTM layer, 64 units
- Dropout layer, 0.3
- Bidirectional LSTM layer, 32 units
- Dense layer with 32 ReLU units
- Output Dense layer with 2 neurons (predicting next 2-hour load)

GRU Univariate: The univariate GRU model uses only the historical ERCOT total load to forecast the next 2 hours. Each sample consists of a 24-hour sliding window, scaled between 0 and 1, and reshaped into the RNN format.

Architecture:

- GRU layer, 64 units, return_sequences=True
- Dropout layer, 0.3
- GRU layer, 32 units
- Dense layer with 32 ReLU units
- Output Dense layer with 2 neurons

GRU Multivariate: The multivariate GRU model uses 12 input features (8 regional loads, total ERCOT load, and 3 datetime features: hour, day of week, month) over a 24-hour sliding window. The goal is to predict the next 2 hours of ERCOT load.

Architecture:

- GRU layer, 64 units, return_sequences=True
- Dropout layer, 0.3
- GRU layer, 32 units

- Dense layer with 32 ReLU units
- Output Dense layer with 2 neurons

Performance Optimization

For the multivariate BiLSTM and GRU models, we used Optuna to automatically search for the best model settings instead of choosing them by hand. In each trial, Optuna builds a new model using different combinations of hyperparameters. These include the number of units in the first and second LSTM/GRU layers, the dropout rate, the size of the dense layer, the learning rate, and the batch size. Optuna also decides whether the model should use one or two recurrent layers.

Every trial trains for up to 40 epochs with early stopping, and the validation MSE is used to judge how good each model is. After running all trials, Optuna picks the hyperparameters that gave the lowest validation loss. Using those settings, we rebuild the final BiLSTM and GRU models and train them again using the full training setup (up to 100 epochs with early stopping).

This tuning process lets the models automatically choose the right amount of complexity like how many units to use or how much dropout is needed based on the data, rather than relying on fixed or manually guessed values.

Parameters selected after hyperparameter tuning

Multivariate BiLSTM

Hyperparameter	Best Value	Description
Lstm1_units	96	Units in the first Bidirectional LSTM layer
Lstm2_units	64	Units in the second Bidirectional LSTM layer
Dropout_rate	0.2	Dropout applied after the recurrent layer
Dense_units	64	Units in the dense layer before the Output
Learning_rate	0.0023255786	Learning rate for the Adam optimizer
Use_two_layers	True	Indicates two BiLSTM layers were used
Batch_size	128	Batch Size used during training

Multivariate GRU

Hyperparameter	Best Value	Description

gru1_units	96	Units in the first GRU layer
gru2_units	64	Units in the second GRU layer (used because use_two_layers=True)
Dropout_rate	0.1	Dropout applied after the recurrent layer
Dense_units	32	Units in the dense layer before the Output
Learning_rate	0.0019249653	Learning rate for the Adam optimizer
Use_two_layers	True	Indicates two GRU layers were used
Batch_size	32	Batch Size used during training

Evaluation Metrics

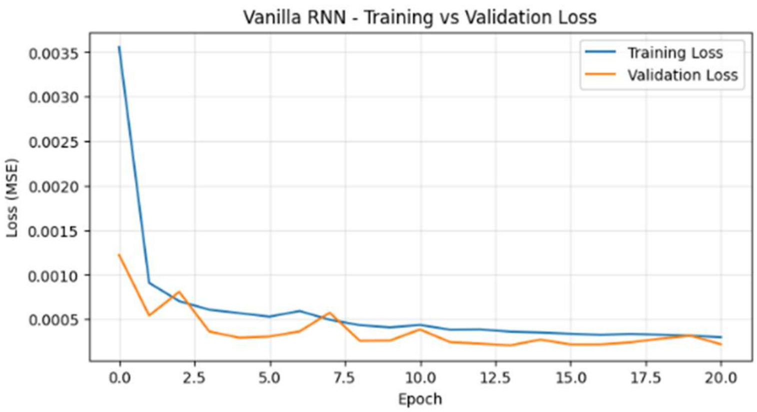
To measure how well each model performed, several evaluation metrics were calculated using the predicted values and the actual ERCOT load values. The main metric was Root Mean Squared Error **(RMSE)**, which tells us how large the prediction errors are on average. A lower RMSE means the model is making predictions that are closer to the real values. Another metric used was Mean Absolute Percentage Error (MAPE), which expresses the prediction error as a percentage of the actual load value. MAPE helps show how large the errors are relative to the size of the actual demand, which is useful when comparing different models or datasets.

To better understand the behavior of each model during training and evaluation, several plots were generated. Training curves (Training loss vs Validation loss over epochs) were used to check whether the model was learning properly and whether it was overfitting. Actual vs. predicted graphs showed how closely the model's predictions matched real ERCOT load values on the test set. Finally, next-2-hour forecast visualizations illustrated the model's ability to look ahead and predict future demand based on the most recent input window. These visual tools helped verify both the accuracy and stability of the forecasting models.

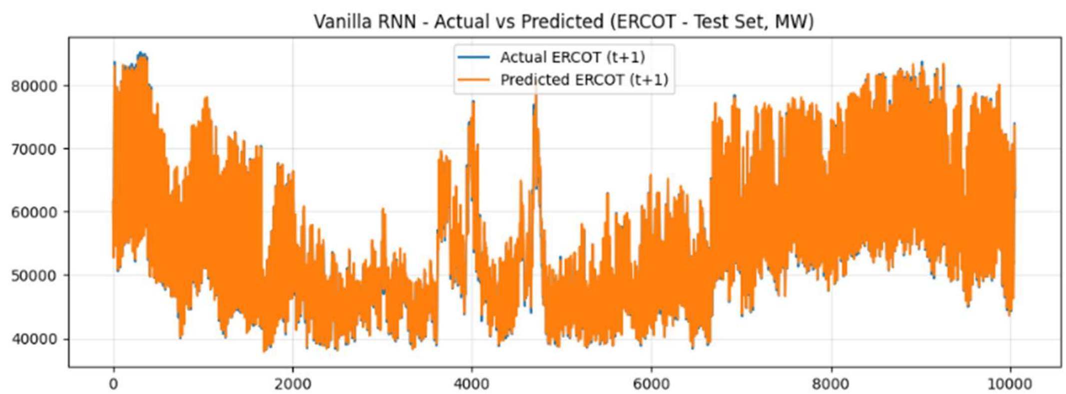
CHARTS

Univariate Models

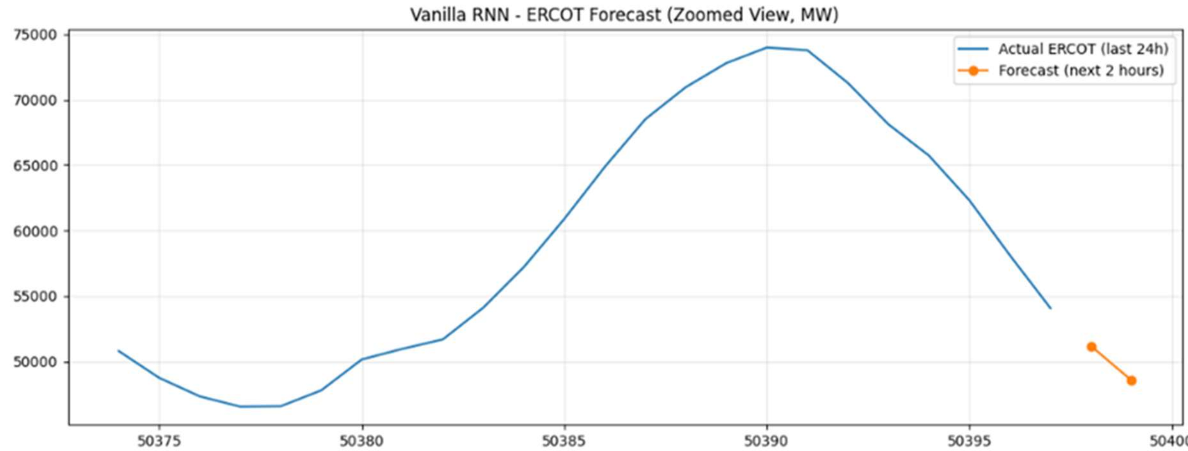
Vanilla RNN



Both Training and validation loss decrease smoothly, indicating stable learning and no signs of overfitting.

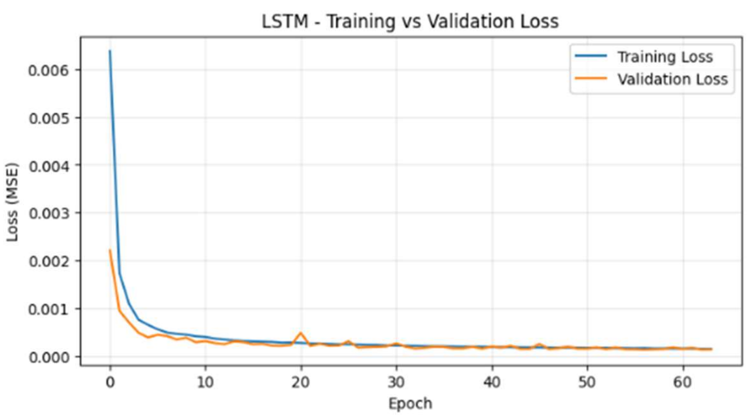


The model captures the overall structure, seasonal cycles, and daily variations, but there is a slight underfitting of extreme peak.

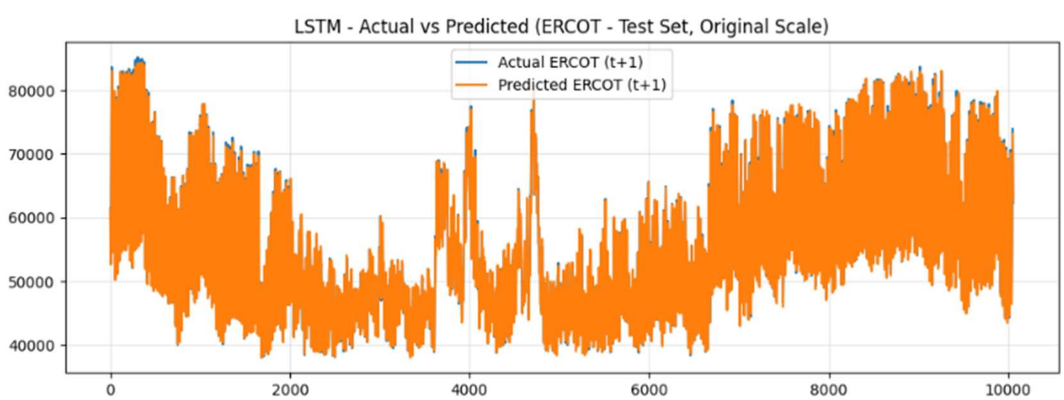


The model predicts the next 2 hours following the same direction as the ongoing trend and there is no unrealistic jumps observed.

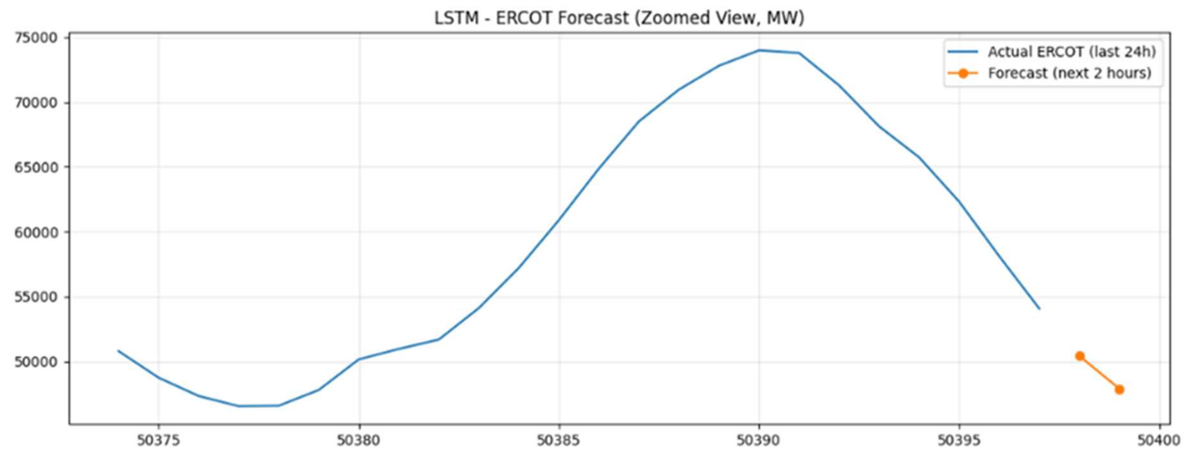
LSTM



The loss drops to zero within very few epochs, which shows that the model learns very quickly. There are no signs of overfitting, which indicates that the model generalizes extremely well.

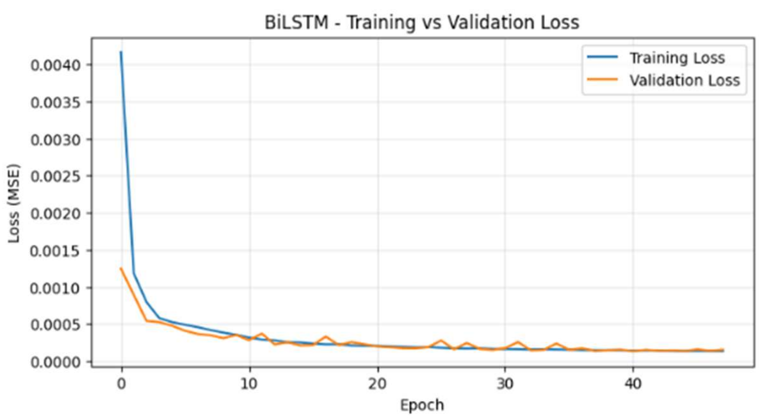


Predicted values (orange) overlap extremely closely with actual ERCOT load (blue). There are slight signs that the model isn't able to predict extreme spikes.

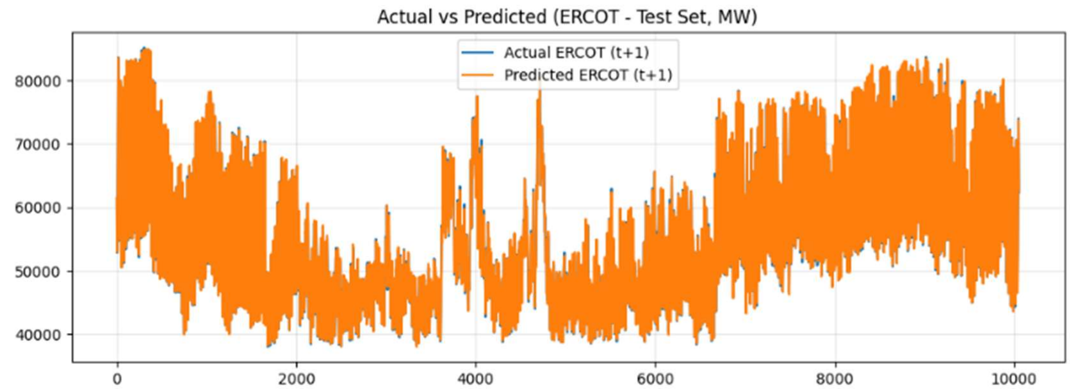


The model predicts the next 2 hours following the same direction as the ongoing trend and there is no unrealistic jumps observed.

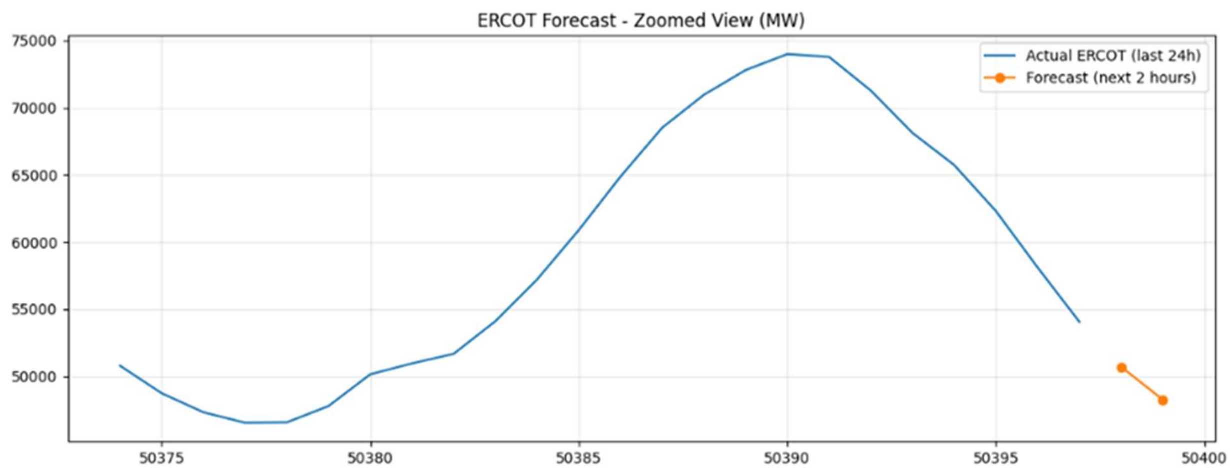
Bidirectional LSTM



The loss drops to zero within very few epochs, which shows that the model learns very quickly. There are no signs of overfitting, which indicates that the model generalizes extremely well.

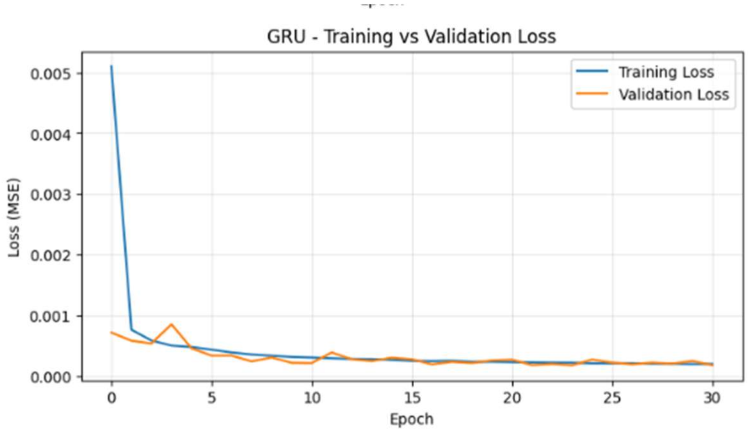


Predicted values (orange) overlap extremely closely with actual ERCOT load (blue). Compared to LSTM, bidirectional LSTM performs slightly better in predicting the extreme peaks.

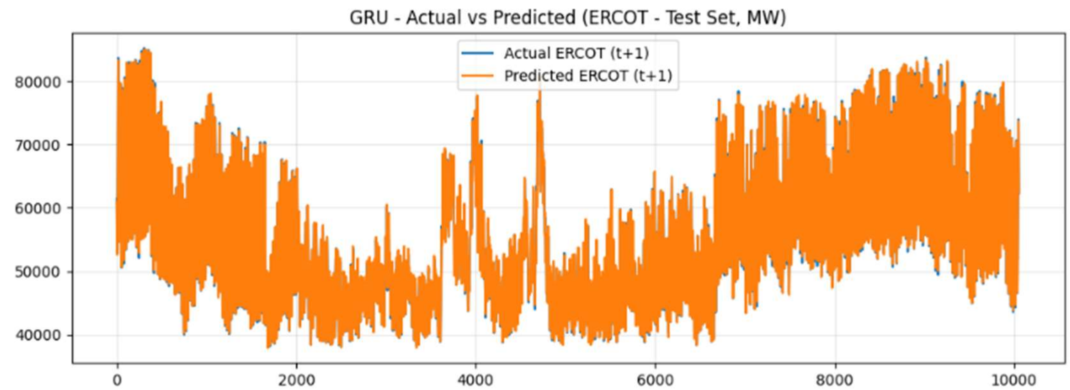


The model predicts the next 2 hours following the same direction as the ongoing trend and there is no unrealistic jumps observed.

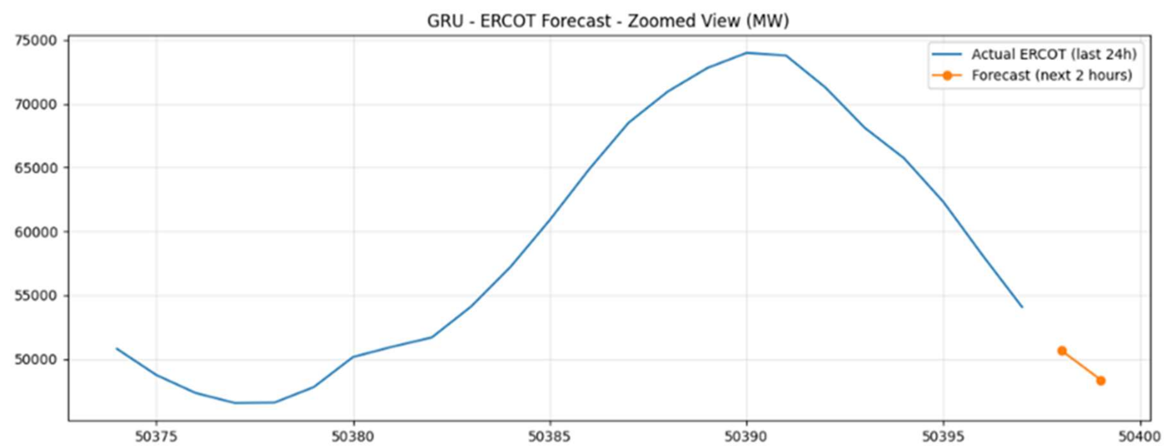
GRU



The loss drops to zero within very few epochs, which shows that the model learns very quickly. There are no signs of overfitting, which indicates that the model generalizes extremely well.



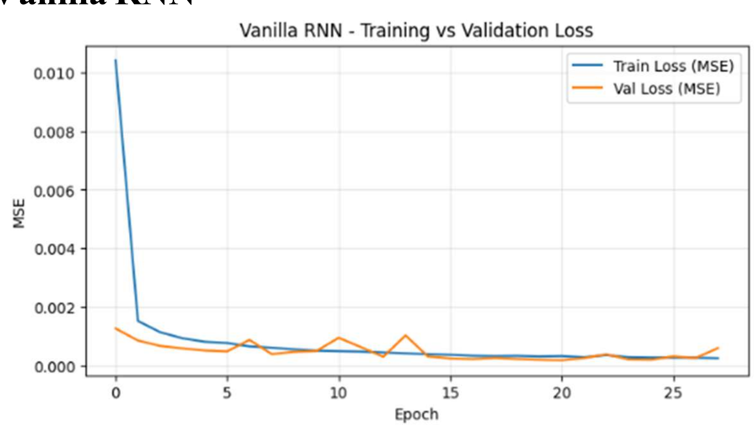
Predicted values (orange) overlap closely with actual ERCOT load (blue). Some extreme values are slightly smoothed, overall the model's forecasting accuracy is pretty good.



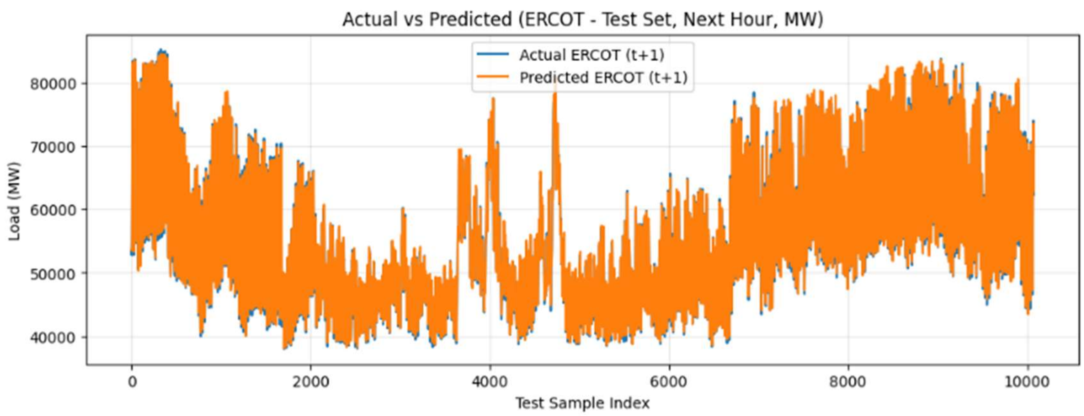
The model predicts the next 2 hours following the same direction as the ongoing trend and there is no unrealistic jumps observed.

Multivariate Models

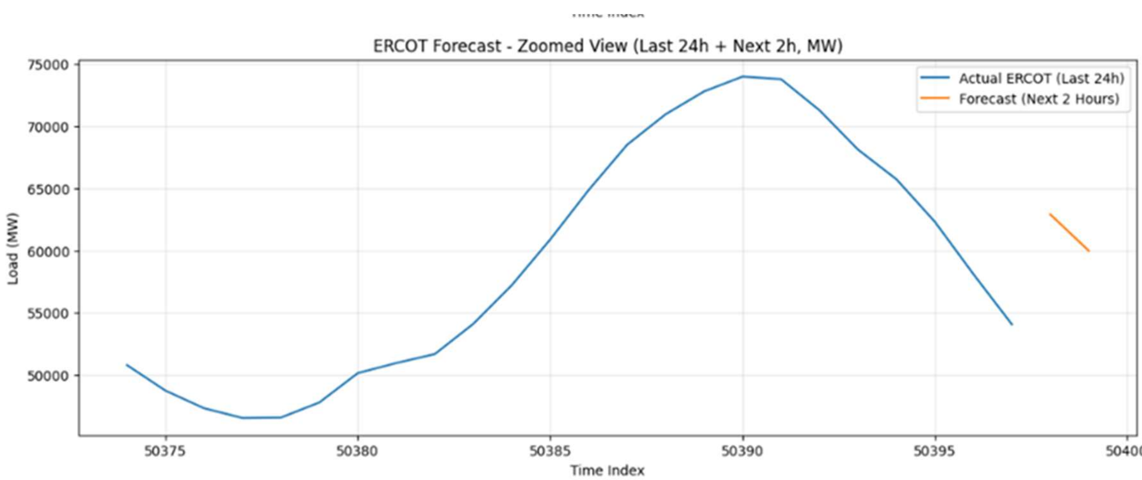
Vanilla RNN



Both Training and validation loss decrease smoothly, indicating stable learning and no signs of overfitting. Slight fluctuations appear in the beginning, but they settle quickly

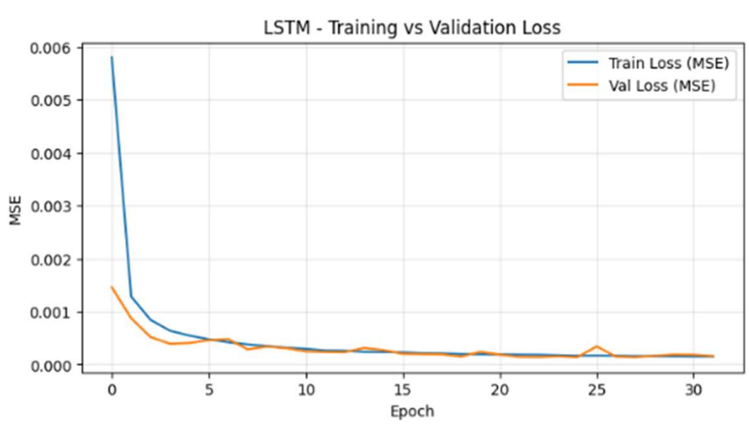


The model captures the overall structure, seasonal cycles, and daily variations, but there is visible underfitting of the extreme peak.

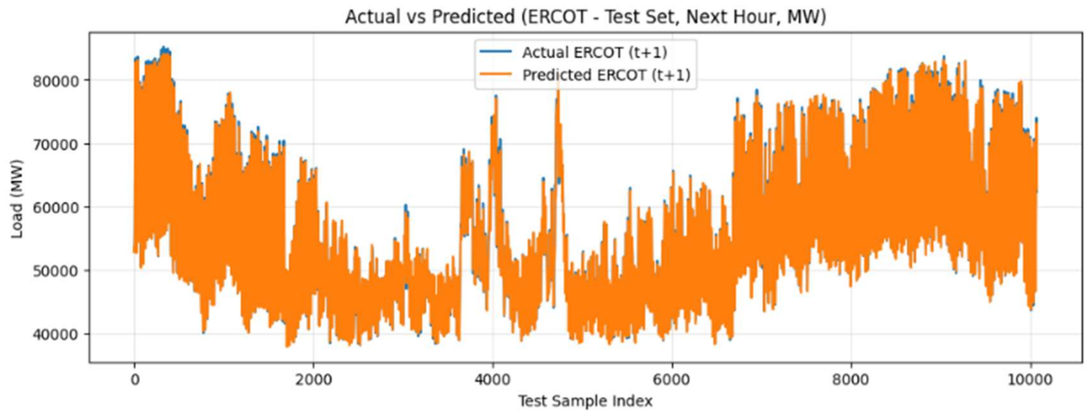


The model predicts the next 2 hours following the same direction as the ongoing trend, and there are no unrealistic jumps observed.

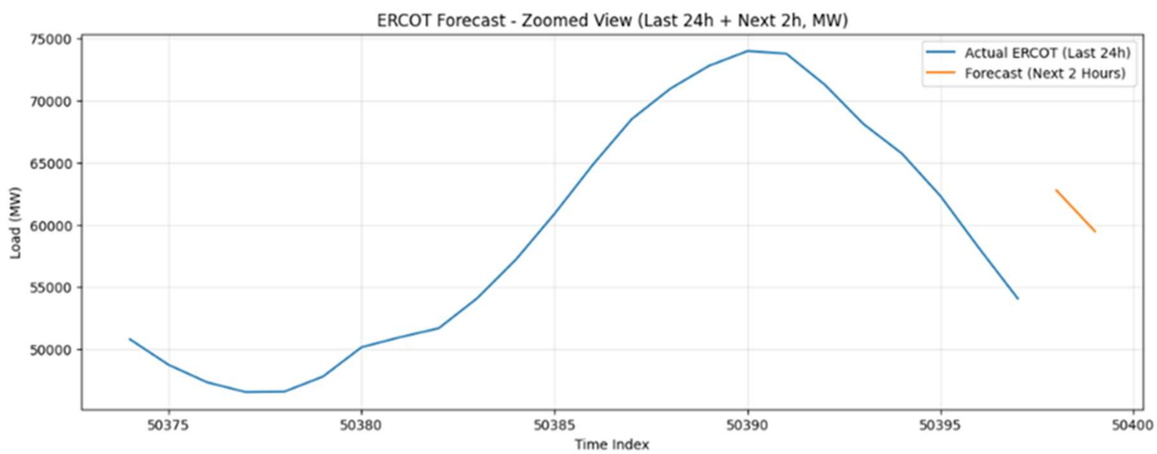
LSTM



The loss drops to zero within very few epochs, which shows that the model learns very rapidly. There are no signs of overfitting, which indicates that the model generalizes well.



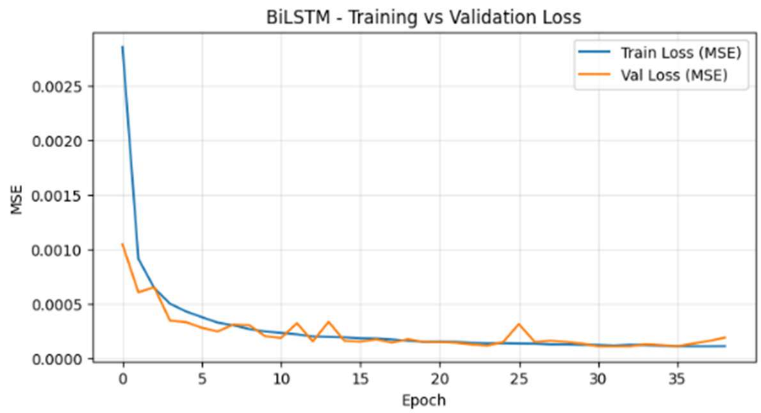
The predicted ERCOT load overlaps with the actual ERCOT load across the entire test set. There are slight signs that the model isn't able to predict extreme spikes, but it performs way better than Vanilla RNN.



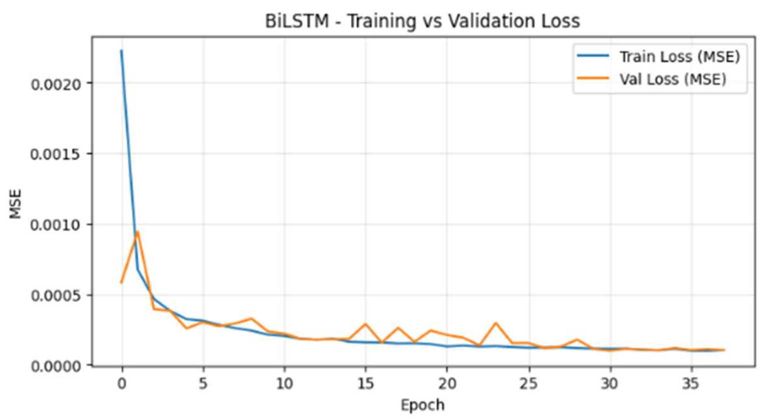
The model was able to predict the next 2-hour forecast in the same direction as the ongoing trend, and the load range remains within a realistic range, which shows that there are no abrupt or unrealistic deviations, indicating the model is performing well.

Bidirectional LSTM (Before and After Hyperparameter tuning)

Before



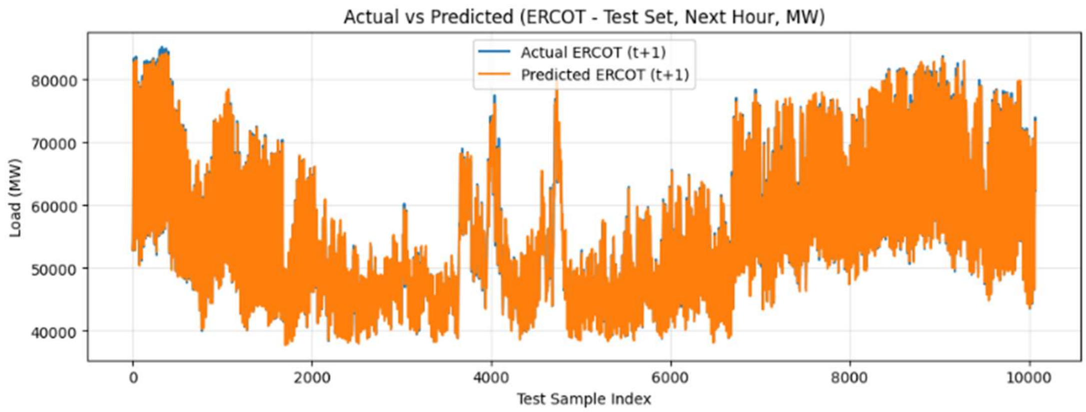
After



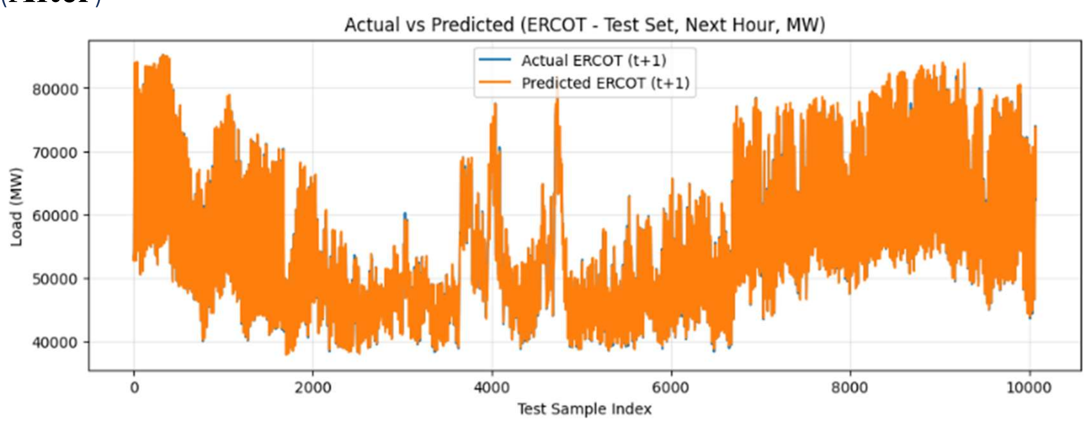
The loss drops to zero within very few epochs, which shows that the model learns very rapidly. There are no signs of overfitting, which indicates that the model generalizes well.

The final validation loss was noticeably higher in the pre-tuned model compared to the tuned model, confirming that hyperparameter tuning significantly improved the model's accuracy.

(**Before**)

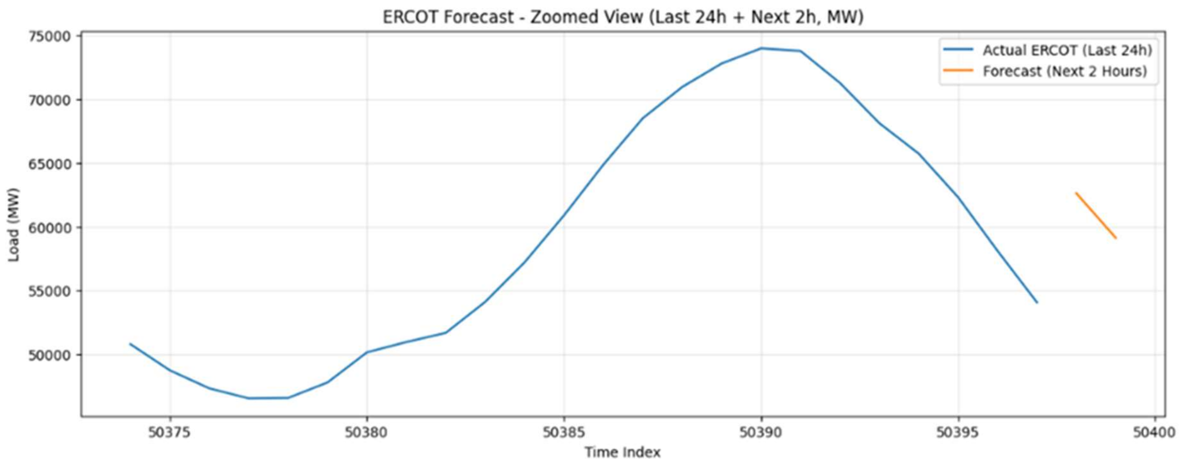


(**After**)

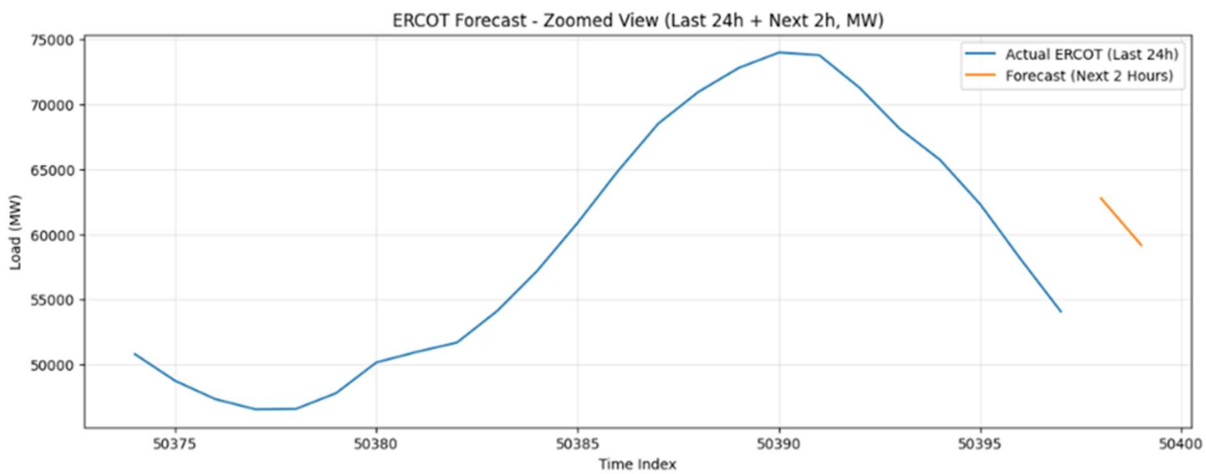


The graph shows that the predictions are much better after the model was tuned with the best parameters, as the orange lines overlap much more closely with the blue lines.

Before



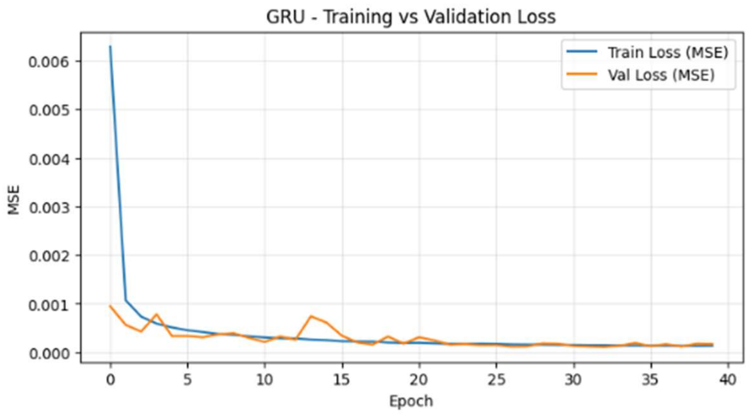
(**After**)



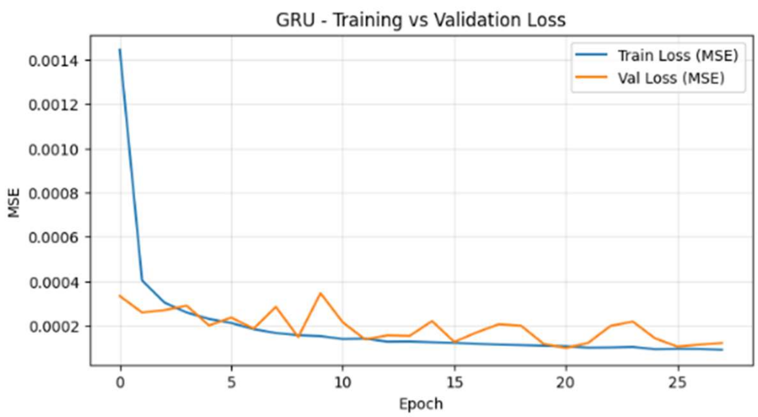
Both models were able to capture the direction of the forecast accurately. In addition, the magnitude of the load prediction remained within a realistic range.

GRU (Before and After Hyperparameter tuning)

(Before)



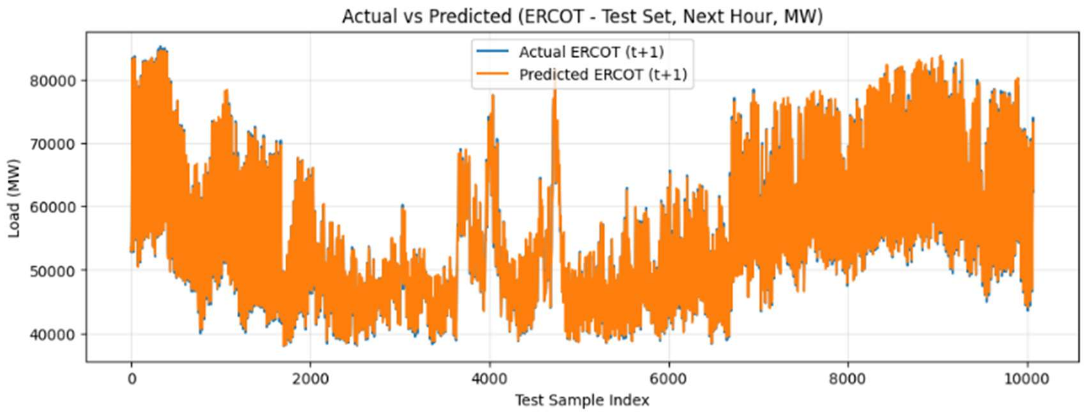
(After)



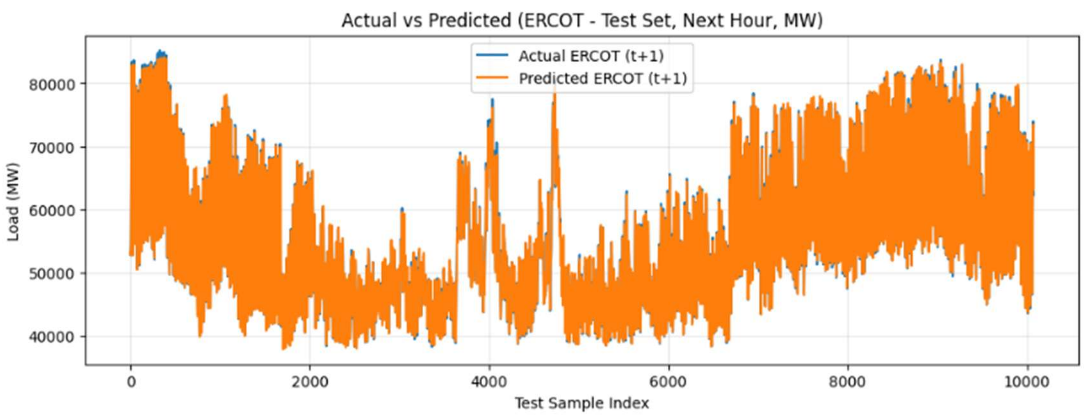
The loss drops to zero within very few epochs, which shows that the model learns very rapidly. There are no signs of overfitting, which indicates that the model generalizes well.

There are small fluctuations in the validation loss, but this is expected in time-series forecasting and is considered normal. Overall, the model is learning well.

(**Before**)

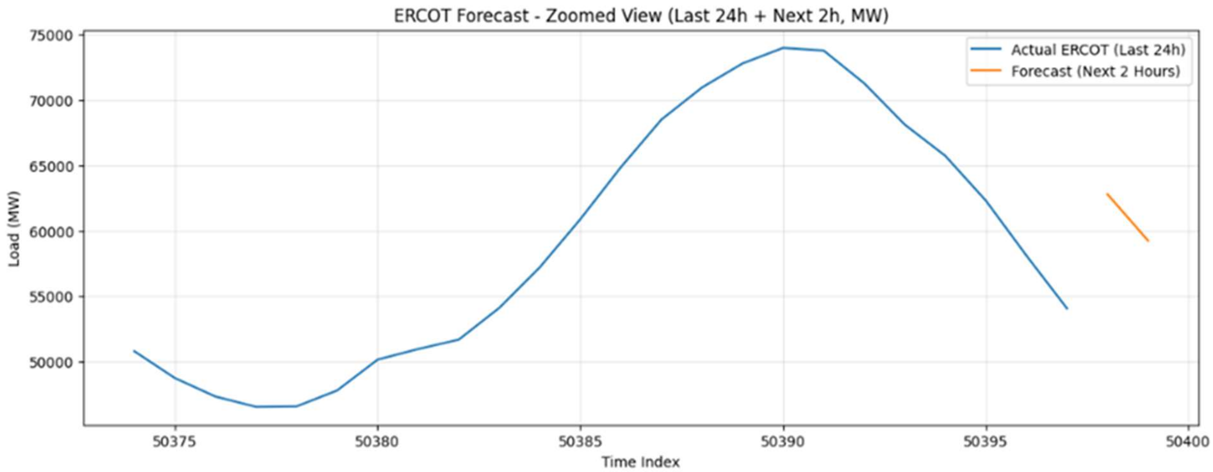


(**After**)

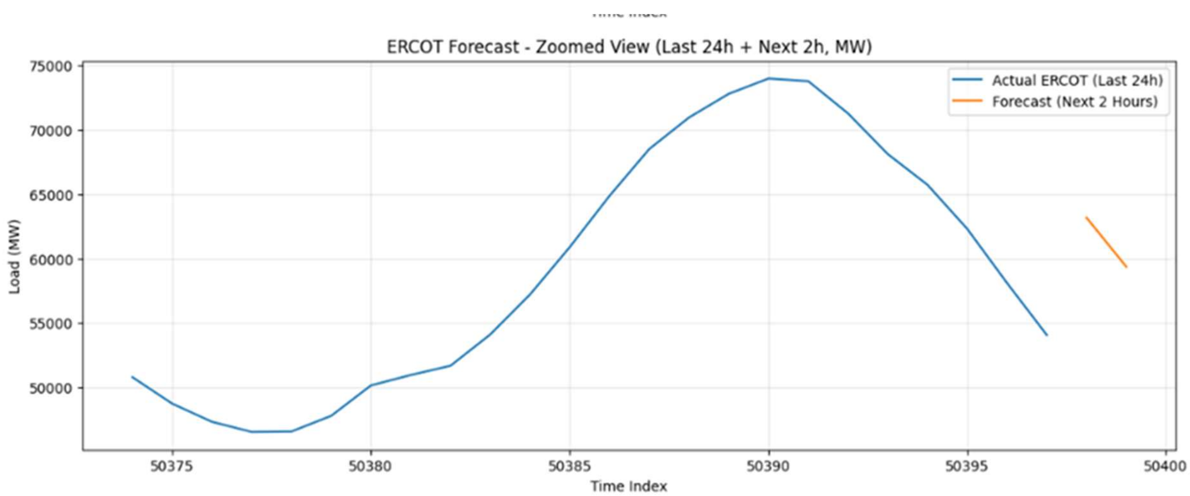


The graphs shows the orange lines overlap closely with blue lines, especially after hyperparameter tuning, the models predictions almost match with the actual data.

(**Before**)



(After)



Both models were able to capture the overall direction and trend of the forecast accurately. In addition, the magnitude of the load prediction remained within a realistic range.

RESULTS and DISCUSSIONS

Model	**MSE**	**RMSE**	**MAE**	**MAPE**
GRU (Multivariate)	377544.41	614.45	456.62	0.82
Bidirectional LSTM (Multivariate)	381273.95	617.47	448.66	0.80
Bidirectional LSTM (Univariate)	450374.84	671.10	477.74	0.86

LSTM (Univariate)	452321.16	672.55	491.69	0.89
LSTM (Multivariate)	459738.04	678.04	494.47	0.87
GRU (Univariate)	581131.32	762.32	552.37	1.00
Vanilla RNN (Multivariate)	671341.21	819.35	606.86	1.10
Vanilla RNN (Univariate)	698899.19	836.00	611.58	1.1

After Hyperparameter Tuning

Model	MSE	RMSE	MAE	MAPE
Bidirectional LSTM (Multivariate)	326789.71	571.66	410.52	0.73
GRU (Multivariate)	328965.24	573.55	413.22	0.73

Key Findings:

- Bidirectional LSTM (Multivariate) and GRU (Multivariate) are the best overall models especially after hyperparameter tuning, the model's performance improved substantially. This shows that this model captures the ERCOT's complex temporal and spatial dependencies.

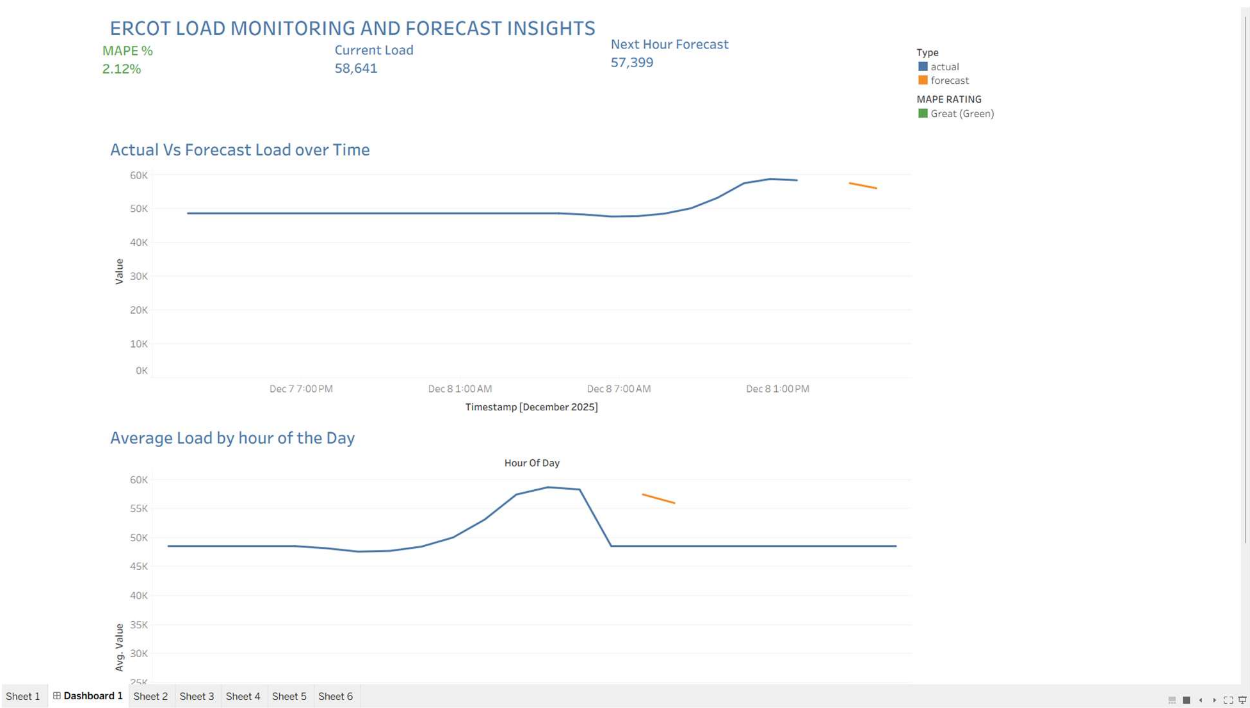
- Multivariate models outperform the univariate model except for LSTM, where the RMSE of Multivariate is slightly higher than the Univariate model. Overall, we can say that adding more features like zonal loads and time can improve the model performance.
- There was a drastic change in the RMSE value in Bidirectional LSTM and GRU when multiple features were used as input
- In univariate model, Bidirectional LSTM and LSTM performed better than other model.
- Vanilla RNN has the worst performance in both Univariate and Multivariate models which shows that this model is not suitable for long-sequence forecasting like electricity load.
- The MAPE values of the models were between 0.73-1.1%, indicating that the models were able to accurately forecast the values.

After evaluating multiple RNN-based models, the multivariate BiLSTM model with hyperparameter tuning achieved the best overall performance, producing the lowest RMSE and

MAPE for the next-2-hour forecast. Based on these results, this model was selected as the forecasting engine for the real-time system. To enable real-time predictions, we collected up-to-date ERCOT load information through the GridStatus.io API, which provides reliable access to operational grid data and then developed a FastAPI backend that automatically retrieves the latest ERCOT values, scales and formats them, applies the trained BiLSTM model, and generates a live 2-hour-ahead forecast. This backend feeds into a custom Web Data Connector (WDC), which allows Tableau to request fresh data and predictions on every dashboard refresh.

Finally, we designed an interactive Tableau dashboard that visualizes both the actual and forecasted load, displays error metrics such as MAPE, and updates dynamically as new grid data arrives. This end-to-end system shows how machine learning models, real-time APIs, and visualization tools can work together to monitor and evaluate electricity demand forecasting in a practical, automated, and continuously updating workflow.

Tableau Dashboard:



This dashboard shows the real-time ERCOT electricity load and compares it with the forecast produced by my BiLSTM model. At the top, it displays the current load, the next-hour forecast, and the MAPE percentage, so users can quickly see how accurate the model is. The main line chart shows actual load over time alongside the forecast line, making it easy to spot whether the prediction follows the real trend. The second chart shows the average load by hour of the day, helping users understand how the current and forecasted values compare to typical daily patterns. Overall, the dashboard gives a clear and easy-to-understand view of grid demand, model accuracy, and expected near-future load, updating automatically as new data comes in.

CONCLUSION

The main contribution of this project is the development of a complete, end-to-end electricity load forecasting system that works in real time. I evaluated several RNN-based models and found that the univariate Bidirectional LSTM gave the most accurate next-2-hour predictions for ERCOT. Although I also tested multivariate models that included all regional load features, these models did not improve performance. This is likely because regional loads are already highly correlated with the total ERCOT load, so adding them created more complexity without providing new predictive information. Based on these results, I used the univariate BiLSTM as the final model and built a real-time pipeline that pulls live grid data, generates updated forecasts, and displays them through an automated Tableau dashboard. The system not only predicts future load but also monitors accuracy with metrics like MAPE and helps visualize trends in a simple, easy-to-understand format.

Below are the beneficiaries who would benefit from the project:

- Grid operators- Utilities can make better scheduling decisions, avoid blackouts, reduce costs, and improve grid reliability
- Energy planners- Energy planners can coordinate renewables more effectively
- Smart-grid and IoT system designers benefit from real-time adjustments and reduced energy wastage
- Researchers- Researchers gain benchmarks for future studies
- Consumers - consumers benefit from more stable electricity prices and fewer outages

This project also provides a practical and reusable forecasting framework that future researchers or utilities can build upon. Moving forward, future work could include:

- adding weather data,
- testing hybrid or attention-based models,
- extending the prediction window to improve long-term grid management and planning.

The findings of this project have useful impacts across several domains. For grid operators and utilities, a reliable next-2-hour forecast helps improve operational planning, reduce unexpected strain on the system, and avoid costly emergency power purchases. For energy planners and policy teams, accurate short-term forecasting supports better integration of renewable energy and battery storage, since these resources depend heavily on knowing near-future demand. In the technology and analytics domain, this project demonstrates how machine learning models especially multivariate BiLSTM can deliver strong performance when multiple regional load features were added. Finally, for researchers and data scientists, the results highlight that feeding the model with correlated multivariate inputs did improve forecasting accuracy. Overall, the work shows how well-tuned models combined with real-time data pipelines can improve decision-making in energy systems, technology design, and analytical workflows.

The predicted economic value is generated by focusing on two main areas. Firstly, the avoidance of high-cost emergency power can be achieved by identifying the optimal RNN model and feature set. This project will minimize the RBSE (forecasting error) so reduce the need to purchase power from expensive sources like gas Peaker plants.

The predicted economic value of this project stems directly from the cost savings achieved by reducing forecast errors for the critical next-2-hour operational window. The value is generated through improved efficiency in two main areas: optimized generation scheduling and avoidance of high-cost emergency power. Consumers would be provided with stabler electricity prices.

Secondly, accurate next-2-hour forecasts will lead to Optimized Generation Scheduling. Studies on large power systems often estimate that even a **1% reduction in forecast error** can result in annual savings of **millions of dollars** for a utility by preventing costly inefficiencies in energy production and trade.

Overall, the work shows how simple, well-tuned models combined with real-time data pipelines can improve decision-making in energy systems, technology design, and analytical workflows.

GROUP MEMBERS

Sana Ambreen

Namrata Sood

Yitian Liu

Priyadarshini Balasubramanian

REFERENCES:

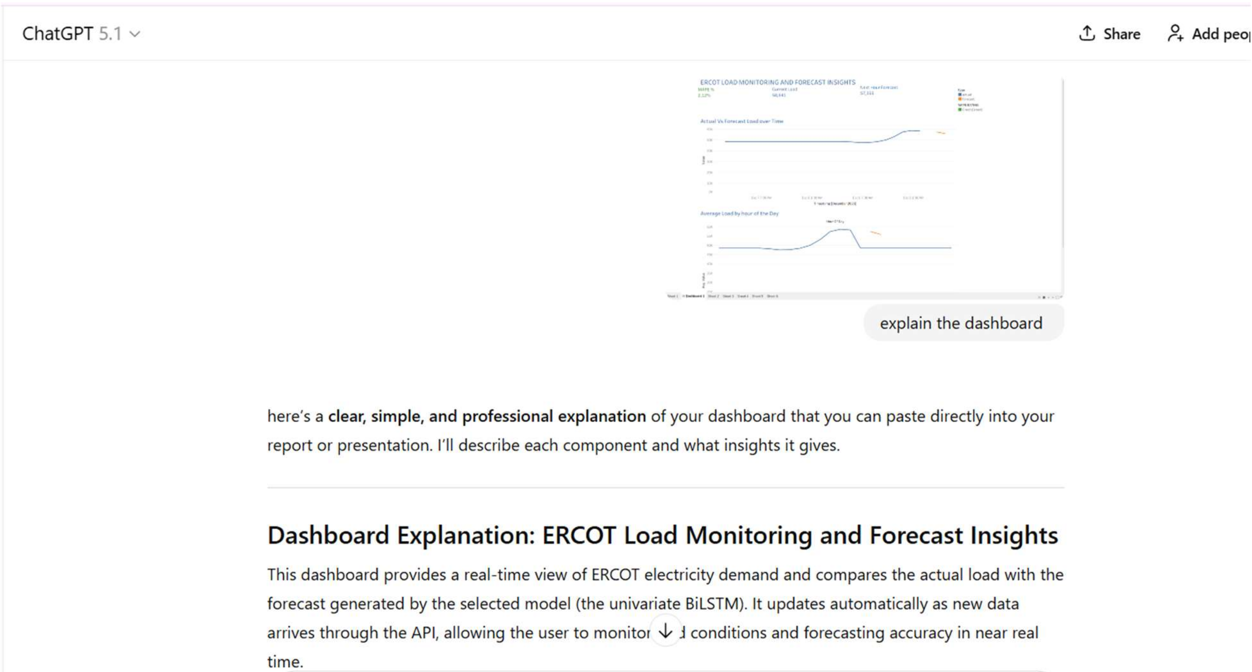
Fayyazbakhsh, A., Kienberger, T., & Vopava-Wrienz, J. (2025). Comparative Analysis of Load Profile Forecasting: LSTM, SVR, and Ensemble Approaches for Singular and Cumulative Load Categories. Smart Cities, 8(2), 65. https://doi.org/10.3390/smartcities8020065

Hasanat, S. M., Ullah, K., Yousaf, H., Munir, K., Abid, S., Bokhari, S. A. S., Aziz, M. M., Naqvi, S. F. M., & Ullah, Z. (2024). Enhancing Short-Term Load Forecasting With a CNN-GRU Hybrid Model: A Comparative Analysis. IEEE Access, 12, 184132–184141. https://doi.org/10.1109/ACCESS.2024.3511653

Shohan, M. J. A., Faruque, M. O., & Foo, S. Y. (2022). Forecasting of Electric Load Using a Hybrid LSTM-Neural Prophet Model. Energies, 15(6), 2158. https://doi.org/10.3390/en15062158

Yang, J., Tuo, M., Lu, J., & Li, X. (2024). Analysis of Weather and Time Features in Machine Learning-aided ERCOT Load Forecasting. 2024 IEEE Texas Power and Energy Conference (TPEC), 1–6. https://doi.org/10.1109/TPEC60005.2024.10472183

Chat Gpt :



This is my reasearch question • Which RNN-based architecture produces the lowest RMSE for next-2-hour grid demand forecasting? • Does including regional load features improve forecasting accuracy over using total ERCOT load alone? and I am planning to build vanilla RNN and bidirectional RNN architecture

Thought for 35s ›

Nice, this dataset is *perfect* for those questions. Let's connect your **research questions** to a concrete **experiment plan** using your ERCOT data and RNNs.

---

1. Map your research questions to models

RQ1: ***Which RNN architecture gives the lowest RMSE for next-2-hour forecasting?***

You'll compare at least:

1. **Vanilla RNN** (unidirectional)
2. **Bidirectional RNN**

Both will predict **the next 2 hours of ERCOT total load** (`ercot`) from a window of past hours.

> Target at time t:
> Input = load history from t−L+1 to t
> Output = [ercot(t+1), ercot(t+2)]

RQ2: ***Do regional loads improve accuracy bey*** ↓ ***total ERCOT alone?***

You can test this by training **the same architectures** in two settings:

+ Ask anything

3. Vanilla RNN architecture

You can implement in PyTorch or Keras. Conceptually:

- **Input**: sequence of length L, feature_dim = 1 (univariate) or 9 (multivariate)
- **RNN hidden size**: e.g., 64
- **Layers**: 1 or 2
- **Output**: Dense layer → 2 units (for t+1 and t+2)

Pseudocode (PyTorch-style idea):

python Copy code

```python
class VanillaRNN(nn.Module):
    def __init__(self, input_dim, hidden_dim, num_layers, output_dim=2):
        super().__init__()
        self.rnn = nn.RNN(input_dim, hidden_dim, num_layers, batch_first=True)
        self.fc = nn.Linear(hidden_dim, output_dim)
    def forward(self, x):
        out, h = self.rnn(x)      # out: [batch, L, hidden_dim]
        last_hidden = out[:, -1]  # last time step
        y_hat = self.fc(last_hidden)  # [batch, 2]
        return y_hat
```

For **univariate**: `input_dim = 1`
For **multivariate**: `input_dim = 9`