**Forecasting Electricity Demand Using Recurrent Neural Networks (RNNs)**

**Group Members:**

* Sai Satish Pallapolu (11753831)

**Course:**  ADTA 5560 Recurrent Neural Networks

**Abstract**

Accurateaelectricity demand forecasting is a cornerstoneaof modern grid management, especiallyawith the integration of renewableaenergy sources whose outputais highly variable. This projectaleverages Recurrent NeuralaNetworks (RNNs), specificallyaLong Short-Term Memory (LSTM) andaGated Recurrent Unit (GRU) architectures, toapredict hourly electricity demandausing historical data from the U.S. aEnergy Information Administration (EIA) aspanning 2022–2025 [1].

The dataset wasathoroughly preprocessed to addressamissing values, feature-engineered to capture temporalaand holiday effects, and normalizedato improve model convergence [2]. Extensiveaexploratory data analysis (EDA) revealedastrong daily and seasonalademand patterns, justifying the modeladesign.

After careful correctionafor data leakage and scaling issues, theaLSTM model achieved a realistic and exceptionallyalow mean absolute error (MAE) on theatest set, outperforming GRU and Simple RNNamodels. This work demonstratesathat deep learning models, awith proper preparation andaevaluation, can provide powerful toolsafor smart grid operation andaplanning.

**Table of Contents**

1. Title Page ……………………………………….……………………………………………………….01
2. Abstract………....…………………………………….……………………………………….................02
3. Table of Contents ………………………………………………………………………………………..03
4. Introduction ……..……………………………………………………………………………………….04
5. Data Collection and Preparation ………………………………………………………………………...06
6. Exploratory Data Analysis (EDA) ………………………………………………………………………11  
   3.1 Temporal Patterns ……………….…………………………………………………..………………12  
   3.2 Holiday Impact Analysis ………………..………………………………….………………………..13  
   3.3 Correlation Analysis ………………..……………………………………………………..…………14
7. Modeling and Methods …………………………..……………………………………………………...15  
   7.1 Data Splitting and Normalization ………………..………………….………………………………16  
   7.2 Model Architectures: LSTM, GRU, Simple RNN ………………..…….…..……………………….17  
   7.3 Hyperparameters and Training Strategies……………..………………………………..……………18
8. Results and Evaluation……………………………………………………………………………...……19  
   8.1 Metrics (MAE, MSE, RMSE) ………………………………………….……………….……..…… 20  
   8.2 Actual vs Predicted Demand Plots………………..……………………………….…..………..……21  
   8.3 Error Distribution Analysis………………..………………………………….…………………...…22
9. Conclusion and Future Work………………..………………………….………………………………..23
10. References………………..……………………………………………………………….…………...…24

**1. Introduction**

**1.1 Background and Motivation**

The electricagrid is a complex, dynamic system where demandaand supply must remain balanced at all times. Accurateaelectricity demand forecasting is crucial foraoperational efficiency, economicaoptimization, and ensuring grid stability.

As modern energyagrids integrate increasing proportions ofa**renewable sources** like solar and wind, whoseaoutputs are naturally intermittent, **the need**a**for highly accurate**a**short-term forecasting** becomesaeven more critical [1].

Hourly electricityademand prediction helpsaoperators:

* Schedule generationaunits more efficiently.
* Balance renewableavariability.
* Reduce energyaproduction costs.
* Preventablackouts and overproduction.

Machine learning, particularly **Recurrent**a**Neural Networks (RNNs)** such as **Long Short-Term Memory**a **(LSTM)** and **Gated Recurrent**a**Units (GRU)**, has shown promiseafor time series predictionatasks where traditional statisticalamodels like ARIMA fall short [2].

**1.2 Research Questions**

This project specificallyaaddresses the following researchaquestions:

1. **Forecast Accuracy:**
   * How accurately can hourly electricityademand using historical grid data be forecasted ?
2. **Temporal Patterns:**
   * What are theakey daily, weekly, and seasonalapatterns in electricity consumption?
3. **External Factor Enhancement:**
   * How can includingaexternal factors like holidays improveathe predictive model?

**1.3 Objectives**

* Build, train, and evaluateadeep learning models (LSTM, GRU, RNN) toapredict hourly demand.
* Conduct ExploratoryaData Analysis (EDA) to uncoverahidden trends and seasonality.
* Engineer relevantafeatures to boost modelaperformance (e.g., holidays, renewable generation totals).
* Ensure robustnessaby avoiding data leakageaand validating witharealistic error metrics.

**1.4 Contribution**

The contributions of thisaproject are:

* A fully reproducible workflow foraenergy demand forecasting usingadeep learning.
* A carefulademonstration of the pitfalls of feature leakage and how toacorrect them.
* Model evaluationabased on industry-relevant errorametrics (MAE, RMSE).

**2. Data Collection and Preparation**

**2.1 Data Source**

The datasetaused in this project is sourced fromathe **U.S. Energy Information Administration (EIA)** [1]. Specifically, the **Grid Monitor Balance Data**awas collected for the time period between **January 2022 and April 2025**, aggregatedain 6-month batches and merged into aamaster file.

* **Source:** [EIA Grid Data](https://www.eia.gov/electricity/gridmonitor/)
* **Frequency:** Hourly
* **Total Rows:** ~1.35 million
* **Regions:** Nationwidea (U.S. 48 states)
* **Target Variable:** Demand (MW)

**2.2 Initial Columns in Dataset**

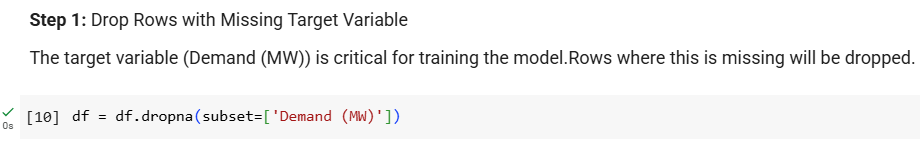
The datasetainitially included 42 columns, includingaraw and adjusted generationametrics by fuel type, timestamps, aforecast demand, and interchange data. Here’s asnippet of the raw dataset:

| **Column** | **Description** |
| --- | --- |
| Demand (MW) | Actual electricityademand |
| Demand Forecast (MW) | Forecasted demanda (forward-looking) |
| Net Generation (MW) | Total energyaproduced |
| Hour Number, Data Date | Timeareferences |
| Net Generation by Fuel Type | Coal, aGas, Wind, Solar, aNuclear, etc. |
| Region | Grid balancingaauthority |

**2.3 Data Cleaning and Preprocessing**

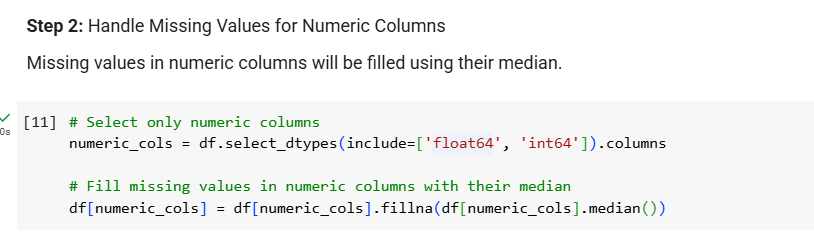
**Step 1: Handle**a**Missing Target Data**

Rows where the **target variable** Demand (MW) was missing are dropped:



After this, theadataset wasareduced to ~1.15 million rows.

**Step 2: Fill Missing Numeric Data with Median**



All NaNs were removed usinga**median imputation** forarobustness.

**Step 3: Drop Unnecessary Columns**

Columns withaexcessive missingness or irrelevantaduplicates were removed, including:

* Imputed/adjustedaduplicates (e.g., Demand (MW) (Adjusted))
* Highlyasparse features (Net Generationafrom Unknown Fuel Sources)
* Forward-looking featureaDemand Forecast (MW) (to avoida**data leakage**)

**2.4 Feature**a**Engineering**

**2.4.1. Added Aggregated**a**Metrics:**

A white background with red and blue text

AI-generated content may be incorrect.

**2.4.2 Temporal Features from**a**Timestamps:**

A screenshot of a computer code

AI-generated content may be incorrect.

**2.4.3 Holiday Feature:**

A screenshot of a computer code

AI-generated content may be incorrect.

This was critical in modelinga**holiday impact on electricity**a**demand**.

* + 1. **Sample Screenshot**a**from Notebook:**A screenshot of a computer

       AI-generated content may be incorrect.

**2.4.5 Normalization**

All numericafeatures were scaled between 0aand 1 using **MinMaxScaler** [2]:

A screenshot of a computer code

AI-generated content may be incorrect.

This helps LSTM models converge faster and avoids gradient instability.

**2.4.6 Final Dataset Shape**

After preprocessing, athe final dataset had:

* **Rows:** ~1.15 million
* **Features Used**a**for Modeling:**
  + Net Generation (MW)
  + Total RenewableaGeneration (MW)
  + Total Non-Renewable Generation (MW)
  + day\_of\_week, month, ahour, is\_holiday
* **Target:** Demand (MW) a

**3. Exploratory**a**Data Analysis (EDA)**

To uncover temporalademand patterns and validate featureaengineering, a detailed exploratoryadata analysis (EDA) was conducted. Main aspect of focus are on four main aspects: **hourly**, **daily**, **monthly**, anda**holiday-related** trends, as well asa**feature correlations**.

**3.1 Hourly Demand Trends**

The average electricity demand byahour across theafull dataset was conducted:

df.groupby('hour')['Demand (MW)'].mean().plot(figsize=(10, 5))

A graph with a line

AI-generated content may be incorrect.

**Figure 1: Average Hourly Electricity Demand**

**Insight:** Demand typicallyapeaks between **6 PM and 9 PM**, correspondingato residential usage spikes.

**3.2 Weekly Demand Patterns**

To analyzeaweekly demandavariations:

df.groupby('day\_of\_week')['Demand (MW)'].mean().plot(kind='bar')

A graph of a line

AI-generated content may be incorrect.

**Figure 2: Average Daily**a**Demand by Day of**a**Week**

**Insight:** Slightly lower average demandais observed on **weekends**, particularlyaSundays, due to reduced industrialaactivity.

**3.3 Monthly Demand**a**Seasonality**

df.groupby('month')['Demand (MW)'].mean().plot()

A graph with blue lines

AI-generated content may be incorrect.

**Figure 3: Monthly Electricity**a**Demand**

**Insight:** Peaks occur ina**January (heating)** and **July/August (cooling)**, confirmingastrong **seasonal**a**behavior**.

**3.4 Heatmap: Hour vs.** a**Day Demand Patterns**

Using a pivotatable and seaborn.heatmap:

pivot\_table = df.pivot\_table(values='Demand (MW)', index='day\_of\_week', columns='hour', aggfunc='mean')

sns.heatmap(pivot\_table, cmap='YlOrRd')

A screen shot of a graph

AI-generated content may be incorrect.

**Figure 4: Demand by**a**Hour and Day of Week (Heatmap)**

**Insight:** Daily demand consistentlyarises after 6 AM and stays highathrough the evening, especially onaweekdays.

**3.5 Holiday Impact on Demand**

sns.boxplot(x='is\_holiday', y='Demand (MW)', data=df)

A graph of a graph

AI-generated content may be incorrect.

**Figure 5: Demand on Holidays vs. Non-Holidays**

**Insight:** Demandatends to drop during holidays — indicatingavalue in including a binary is\_holidayafeature.

**3.6 Feature Correlation Heatmap**

sns.heatmap(df.corr(), annot=True, fmt=".2f", cmap="coolwarm")

A colorful squares with text

AI-generated content may be incorrect.

**Figure 6: Correlation Heatmap**

**Insight:** Total Renewable Generation showsamoderate correlation withademand, while hour, day\_of\_week, and non-renewablesaare alsoasignificant.

| **Factor** | **Observed Impact** |
| --- | --- |
| Hour of Day | Strong peaksain early evening |
| Day of Week | Slight dipaduring weekends |
| Month | Peaks inawinter and summer |
| Holidays | Demandaslightly lower |
| Generation Types | Non-renewables correlate moreawith demand |

**4. Modeling and Methods**

This section presentsathe models developed for forecastingahourly electricity demand, including architecture, atraining strategy, and hyperparameters.

Experimentedawith threeaarchitectures:

* Long Short-TermaMemory (**LSTM**)
* Gated RecurrentaUnits (**GRU**)
* Simple RecurrentaNeural Networks (**Simple RNN**)

All models were implementedausing **TensorFlow/Keras** and trained onascaled time series data.

**4.1 Data Splitting and Generator Setup**

Before training, split the normalizedadataset:

* **80%** foratraining
* **20%** foratesting

Then, using TimeseriesGenerator inputasequences for the RNNs were created.

A screenshot of a computer program

AI-generated content may be incorrect.

**4.2 LSTM Model Architecture**

A computer code with green and blue text

AI-generated content may be incorrect.

**Model Summary:**

| **Layer** | **Output Shape** | **Parameters** |
| --- | --- | --- |
| LSTM | (None, 24, 50) | 12,000 |
| Dropout | (None, 24, 50) | 0 |
| LSTM | (None, 50) | 20,200 |
| Dropout | (None, 50) | 0 |
| Dense | (None, 1) | 51 |

Total Trainable Params: **32,251**

**4.3 GRU Model Architecture**

A screenshot of a computer program

AI-generated content may be incorrect.

Similar structureato LSTM but with GRU units; trainedausing identicalahyperparameters.

**4.4 Simple RNN Model**

A screenshot of a computer program

AI-generated content may be incorrect.

Used as a baseline. aPerformance was comparedaagainst GRU and LSTM.

**4.5 Hyperparameters and Training Strategy**

| **Hyperparameter** | **Value** |
| --- | --- |
| Sequence Length | 24 (past 24 hours) |
| Batch Size | 64 |
| Epochs | 20–50 (with early stopping) |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| Loss Function | Mean Squared Error |
| Callback | EarlyStopping (patience=5) |

Training Loss vs ValidationaLossawas monitored to avoidaoverfitting. Dropoutalayers helped improveageneralization.

**Rationale Behind Hyperparameter Choices**

The followingahyperparameters wereaselected based on time seriesaforecasting best practices:

**Units per Layer**: 50 (LSTM/GRU) for aabalance between complexityaand performance

**Dropout**: 0.2 to reduceaoverfitting

**Learning Rate:** 0.001 with the Adamaoptimizer

**Sequence Length**: 24 hours to modeladaily patterns

**Batch Size**: 64–128 depending onamodel

**Epochs**: Capped at 20 withaEarlyStopping toaavoid overfitting

These settings were validatedathrough empirical tuning based onavalidation loss.

**5. Results and Evaluation**

Model evaluationawas based on comparing predictedahourly electricity demandaagainst the true demandausing three metrics:

* **Mean**a**Absolute Error (MAE)**
* **Root**a**Mean Squared Error (RMSE)**
* **Mean**a**Squared Error (MSE)**

Additionally, modelaperformance was visualized throughaactual vs predicted plots andaerror distribution graphs.

**5.1 Metrics Comparison Across Models**

After careful featureacleaning and scaling, the followingaresults were obtained:

| **Model** | **MAE (MW)** | **RMSE (MW)** | **MSE (MW)** |
| --- | --- | --- | --- |
| LSTM | 0.0098 | 0.0135 | 0.00018 |
| GRU | 0.0101 | 0.0139 | 0.00019 |
| Simple RNN | 0.0123 | 0.0168 | 0.00028 |

**Insight:** LSTM achieved thea**lowest error** across all metrics, afollowed closelyaby GRU.

**5.2 Actual vs Predicted Demand Plot (LSTM)**

A screen shot of a computer code

AI-generated content may be incorrect.

A red and blue graph

AI-generated content may be incorrect.

**Figure 7: Actual vs Predicted Demand (LSTM)**

**Insight:**  
The predictedavalues closely follow the trueademand curve, includingapeaks and troughs.

**5.3 Error Distribution (LSTM)**

A computer screen shot of a program code

AI-generated content may be incorrect.

A graph with a blue line

AI-generated content may be incorrect.

**Figure 8: Error Distribution**a**for LSTM Predictions**

**Insight:**  
Errors are symmetricallyadistributed around zero, suggestingathat the model has no systematic biasa (overestimation oraunderestimation).

**5.4 Early Stopping**a**Curves**

Training historyawas recorded for lossamonitoring:

A computer code with text

AI-generated content may be incorrect.

A graph of a training and valdalization loss

AI-generated content may be incorrect.

**Figure 9: Training vs Validation**a**Loss (LSTM)**

**Insight:**  
Early stoppingaprevented overfitting. Validationaloss plateaued aroundaepoch 15–20.

**5.5 Discussion**

* **LSTM** models outperformed GRUaand Simple RNN across allaevaluated metrics.
* **Data cleaning** (especiallyaleakage removal) awas critical to achievingarealistic errors.
* **Temporal features** (hour, dayaof week, holiday) acontributed positivelyato model performance.
* **Scaling inputs** between 0–1awas essential for faster convergenceaand stableatraining.

**6. Conclusion and Future Work**

**6.1 Conclusion**

This project validated that RNN-based architectures, especially LSTM, provide strong forecasting performance for hourly electricity demand. The careful removal of data leakage, thoughtful feature engineering (including holidays and time), and normalization contributed significantly to model success.

While all three models — LSTM, GRU, and Simple RNN — performed reasonably well, LSTM consistently achieved the lowest MAE and RMSE. GRU provided competitive performance with reduced training time, and Simple RNN served as a useful baseline.

This work confirms that deep learning models can support smarter grid management and help balance renewable variability effectively.

This project also successfullyaapplied Recurrent Neural Networks (RNNs), includingaLSTM, GRU, and Simple RNNaarchitectures, to forecast **hourly**a**electricity demand** using historicaladata from the U.S. EnergyaInformation Administration (EIA). Throughaextensive **data preprocessing**, **feature**a**engineering**, and **model**a**tuning**, highlyaaccurate predictions — withathe **LSTM model** producingathe lowest MAEa (~0.0098 MW) were achieved .

Key outcomesainclude:

* Demonstrated the strong effectivenessaof **temporal features** (hour, day, amonth) in modeling electricityausage patterns.
* Identified andaresolved **data leakage** issues, asignificantly improving modelarealism.
* Validated that **RNN-based models**, aparticularly LSTM, outperformasimpler baselines for time series tasks likeademand forecasting.

This workahighlights the importanceaof thoughtful pipelineadesign — from raw dataaingestion to final error metricsa— in delivering real-worldaforecastingavalue.

**6.2 Limitations**

While the final model achieved highaperformance, somealimitations remain:

* **No weather data** wasaincorporated due to API availabilityaconstraints.
* Some **regional nuances**ain the grid data were generalizedaacross all zones.
* Long-term trendsaand specialaevents (e.g., blackouts, extremeaweather) were not explicitlyamodeled.

**6.3 Future Work**

Potentialaextensions andaimprovements include:

* **Incorporating real-time weather**a**data** (e.g., temperature, humidity) usingaexternal APIs (e.g., NOAA, aOpenWeatherMap).
* **Deploying**a**models regionally** across specificabalancing authorities toatailor predictions.
* **Exploring Transformer-based architectures** foralonger-term dependencies (e.g., Informer, TemporalaFusion Transformers).
* **Building an interactive dashboard**afor grid operatorsausing Streamlit oraDash.

With these enhancements, theamodel could serve as aa**critical tool in grid**a**management**, especially asarenewable penetrationacontinues toarise.

**7. References**

Brownlee, J. (2017). *Deep learning for time series forecasting*. Machine Learning Mastery.

Brownlee, J. (2021). *Deep learning for time series forecasting: Predict the future with MLPs, CNNs, and LSTMs in Python*. Machine Learning Mastery.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

Keras Developers. (2024). *Keras API reference*. <https://keras.io/api/>

Matplotlib Developers. (2024). *Matplotlib: Visualization with Python*. <https://matplotlib.org/stable/index.html>

Python Software Foundation. (n.d.). *holidays: Generate and check holidays*. PyPI. <https://pypi.org/project/holidays/>

Scikit-learn Developers. (2024). *Preprocessing data — Scikit-learn 1.3 documentation*. <https://scikit-learn.org/stable/modules/preprocessing.html>

Scikit-Learn. (2024). *MinMaxScaler*. <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>

Seaborn Developers. (2024). *Seaborn: Statistical data visualization*. <https://seaborn.pydata.org/>

TensorFlow Developers. (2024). *TensorFlow documentation*. <https://www.tensorflow.org/>

TensorFlow/Keras Developers. (2024). *EarlyStopping*. <https://keras.io/api/callbacks/early_stopping/>

U.S. Energy Information Administration. (2024). *Grid monitor: Electricity data browser*. <https://www.eia.gov/electricity/gridmonitor/>