Predicting Energy Consumption

Ebenezer Daniel, Emil Saliakhov, Sreeja Reddy Yeluru, Krishnarjun Lakshminarayanan
edanie21@depaul.edu, esaliakh@depaul.edu, syeluru@depaul.edu, klakshmi@depaul.edu

June 14, 2025

Abstract

The energy grid in the U.S. has trouble keeping up when demand spikes or during extreme weather, and this sometimes leads to blackouts. To help with this, we wanted to find a way to predict electricity use ahead of time, especially in the ComEd region of Northern Illinois. We compiled a dataset that includes hourly electricity usage and weather information from several stations in that area. Then we trained a few different deep learning models - RNN, LSTM, and hybrid approrach of CNN-LSTM - to see which one could do the best job at forecasting demand. The goal was to determine whether including weather data could improve predictions and help energy providers plan better.

1 Introduction

In 2025, the American Society of Civil Engineers[5] owered its grade of the US energy sector to D+. The main arguments in favor of this decision were the shortage of distribution transformers, the increased frequency of severe weather events, and the lack of transmission capacity. The weather in the Midwest region of Chicago and the surrounding places is always changing, so better forecasting can help energy providers, such as ComEd, to keep up with demand in these places. This problem is interesting and important because we see that weather strongly influences energy consumption, tied to climate change, optimizing energy resources, and reducing outages. Some of the key challenges and complexities are missing data for some parameters from a few weather stations, time lags, different consumption patterns, and spatial complexity. We need models that can handle both time-based and location based variation. That's why we used three types of deep learning models in our project RNN, LSTM, and CNN-LSTM and compared their performance. One of the main contributions of our project is that we built our own dataset by merging energy usage and weather data for this region.

2 Related Work

In this project, we looked at several research papers to understand how others have utilized machine learning and deep learning for energy forecasting. Most of the papers used models such as LSTM, CNN-LSTM, or other hybrid models to predict electricity usage. While many of them showed good results, we noticed that very few used regional energy data with multi-station weather inputs - which became one of the main objectives of our project.

Alhussein et al. (2020)[1] used a hybrid CNN-LSTM model to predict energy usage at the household level. They used CNN to identify local features and LSTM to understand time-based trends. Their model worked well for individual homes, but they did not use any weather data. In our project, we focused on regional-level data and added weather features from several stations to improve the model's predictions.

Melo et al. (2022)[4] also used LSTM, but with weather data such as temperature and humidity. They showed that including weather features helped make the forecasts more accurate. However, their weather data came from only one station. In our case, we used weather data from multiple stations across the ComEd region, which provides us with better spatial coverage.

Kang and Reiner (2022)[2] used simple regression models to find relationships between weather factors like temperature, rainfall, and sunlight, and hourly electricity use. They found strong correlations, but regression models are limited when it comes to capturing more complex and changing patterns. We used LSTM and CNN-LSTM because they are better at handling those kinds of relationships.

Qu et al. (2025)[6] used machine learning models like Random Forest and Artificial Neural Networks to determine which weather features had the most impact on energy use. While their results were useful, their models did not consider how energy usage changes over time. Our project focuses more on time-series models like LSTM that can learn from past patterns.

Maarif et al. (2023)[3] combined LSTM with explainable AI (XAI) techniques. This helped them not only predict future energy usage but also explain which input features (like temperature or humidity) had the biggest influence on the prediction. We found this idea useful and plan to explore similar methods for interpreting our model's results.

Sachin et al. (2020)[7] compared traditional models like ARIMA with deep learning models like RNN and LSTM using electricity usage data from London. They found that RNN and LSTM gave better results than ARIMA, especially for mid- and long-term predictions. However, they did not use any weather data in their models. Our project builds on this by using weather as an important feature, and also by comparing RNN, LSTM, and CNN-LSTM models on the same dataset.

In our project, we did not just stick to one model - we actually implemented and compared three different models: RNN, LSTM, and CNN-LSTM. This helped us understand how each one performs with the same input features and data. While RNN was simpler, it did not perform as well as LSTM. CNN-LSTM gave slightly better results than regular LSTM in some cases, but it was also more complex to train. We found that LSTM gave a good balance between accuracy and training time.

What makes our project distinct is that we are using real hourly energy consumption data from the ComEd region and combining it with weather data from several local stations. We did not find any open-source project that does this, so our work adds something new to this area of research.

3 Preliminary/Background

In our task of predicting energy consumption based on weather data, time dependencies play a role in addition to the weather parameters themselves. It is important to take into account local trends, as well as temporary weather changes. Classical neural networks (MLPs) are not able to capture such patterns in the data. Therefore, in our project we chose three architectures specifically focused on working with sequences: Simple RNN, LSTM, and hybrid CNN-LSTM. Below we will briefly describe their principles, limitations, and the reasons why we decided to apply these models to our task.

3.1 Recurrent Neural Networks (RNNs)

We chose RNN as a basic simple model that is able to take into account temporal dynamics (but in a short time). Based on the results of this model, we judged the validity of using other more advanced models.

At each time step (in our case, 1 hour), the model receives the input x_t (current weather) and stores the "memory" of the previous state in the form of the vector h_t. This mechanism allows to take into account the previous context when predicting the next step. This was the reason for choosing RNN for time series analysis.

3.2 The Vanishing Gradient Problem

In the context of forecasting energy consumption, RNN is able to detect patterns such as increased load after a cold snap. However, this algorithm has limitations related to the vanishing gradient, due to which the model loses the ability to "remember" information over long intervals. We also encountered this problem in our project when working with a 24-hour prediction window. The model was good at detecting changes in trend direction, but it was quite wrong in the values while maintaining the trend. On the backpropagation stage, gradients are repeatedly multiplied by the same weight matrix. If these weights are small, the gradient signal can shrink exponentially, preventing the network from learning long-range dependencies. This was a key motivation for developing more advanced architectures.

3.3 LSTM: Long Short-Term Memory

This model was chosen primarily for its advantages such as long-term memory and resistance to gradient attenuation. To overcome the limitations of RNN, we implemented Long Short-Term Memory (LSTM). This neural network architecture adds a separate long-term memory channel C_t, which is controlled by three "gates":

- the forget gate filters out irrelevant information from the past,
- the input gate controls what new information is stored in the memory,

- and the output gate determines how much of the memory is used in the current prediction.

This gating structure enables the model to preserve important information over long sequences and discard noise, making it well-suited for our use case where weather events influence load patterns with a delay. In our task of forecasting electricity consumption, this is important, since, for example, the influence of a strong cold snap can affect during the day. LSTM effectively captures such long-term dependencies. This was confirmed by the results of our study. Compared with RNN, LSTM provides a lower prediction error on a delayed sample.

3.4 CNN: local patterns and stability

In addition to recurrent models, we have considered a convolutional neural network (1D CNN). This model aims to recognize local patterns in the input time series. For example, a typical "pattern" of temperature changes preceding an increase in consumption may take 3-6 hours. CNN made it possible to efficiently notice such fragments without needing to store the state. However, the disadvantage of CNN is that such an architecture does not know how to memorize a sequence of patterns, it is able to see only a fixed entry window.

3.5 CNN-LSTM hybrid model: combining local and global context

We decided to use the strengths of the two networks and therefore used a hybrid CNN-LSTM. First, CNN extracts local features from weather data, and LSTM then interprets these features in a temporal context. This approach allowed: Reduce the size of the input for LSTM (which accelerated learning), Focus on informative patterns (thanks to CNN), Retain memory of previous states (via LSTM). As a result, we get a model that can simultaneously take into account local weather signals and their long-term effects on the load.

4 The Methodology

In this section, we will describe in detail the steps we followed when training and testing our models to forecast energy consumption based on weather data.

To predict electricity consumption in the Illinois North American region (ComEd coverage area), we implemented and compared three sequential neural networks architectures: a simple RNN, LSTM, and a CNN-LSTM hybrid model. All models were implemented using the PyTorch library. The models were trained based on constructed 24-hour time windows and received weather characteristics and historical energy consumption values as input.

4.1 Dataset Construction and Preprocessing

Initially, we downloaded data on hourly electricity consumption from ComEd (Commonwealth Edison) supplier c Kaggle. The data contains information about the hourly electricity consumption in the Northern Illinois region, including the city of Chicago and surrounding areas. In order to find weather information relevant to the assessed region, we determined the geographical boundaries of the ComEd coverage using the supplier's official sources. Based on these boundaries, weather data from all weather stations available in the region was collected using the meteostat library. This approach allowed us to obtain an average and more accurate picture of weather conditions than if data from only one station had been used.

Next, we filtered the features in the weather dataset. We have removed columns with a large number of omissions, as well as correlated and overlapping weather characteristics. After that, both datasets (energy and weather) were brought to a single time standard (UTC), and combined by timestamp.

Categorical features have already been added to the combined dataset, reflecting the hour of the day, day of the week, and month. These signs should take into account regular consumption cycles: daily peaks, weekends, and seasonal changes.

Before using data in the models, all continuous features were standardized (StandardScaler), and the scaling parameters were calculated exclusively from the training set, which avoids information leakage when dividing data into training, validation, and test samples.

Finally, we applied a windowed approach to generating input data. For each time step (1 hour) we formed a window from previous observations of a fixed length (we used 24 hours), and assigned the target variable the value of electricity consumption for the next hour. This input format allows models to be trained on time dependencies, not just on specific weather values.

4.2 Splitting into training, validation, and test samples

To preserve the time dependence and prevent information leakage, a chronological data breakdown was used. The training set made up 70% of the sequential data, the validation set made up 15%, and the remaining 15% were used for testing. This partitioning simulates a real-world scenario in which the model must make predictions for the future without relying on future values.

4.3 Models Architecture

In all models, the input data was formed as fixed-length windows of weather and time features. For example, to predict consumption at time t, the model received a matrix of signs for the last 24 hours, where each row x_i contained weather parameters (temperature, humidity, pressure, etc.), as well as numerical signs of time (hour, day of the week, month).

The target variable (output) in all models was the power consumption for the next hour - that is, the scalar $y = consumption_{t+1}$.

The architectures of the three models handled this data in different ways:

- **Recurrent Neural Network (RNN)**

  The basic RNN model accepts as input a sequence of time steps (dimension: batch_size × seq_len × input_dim). At each step, it updates the hidden state, "remembering" short-term dependencies. The output is the last hidden state, which then passes through the linear layer and outputs a forecast.

  Advantages: the ability to take into account a consistent structure and short time dependencies.

  Limitations: limited memory, problems with gradients on long stretches.

- **Long Short-Term Memory (LSTM)**

  LSTM extends RNN by adding an internal memory state (cell state) and special "gates" that regulate information. At each step, the network decides what to save, forget, and transfer next.

  Input: the same sequence of X.

  Output: The last memory state is processed by a fully connected layer to calculate the forecast.

  This allows the LSTM to "remember" important weather values that affect energy consumption (for example, a cold snap a few hours ago) and ignore minor fluctuations.

- **Convolutional Neural Network + LSTM (CNN-LSTM)**

  This model combines two approaches:

  1D Convolution Layer: First, a convolutional layer (Conv1d) is used with several filters sliding along the time axis of the input. It extracts local patterns (for example, a series of cold hours), what effectively finds short-term patterns.

  LSTM Layer: The CNN outputs are then interpreted by the LSTM as a condensed sequence of features. This allows you to take into account both local (CNN) and long-term (LSTM) dependencies.

  Output: The last hidden state LSTM is used to predict energy consumption.

4.4 The learning process

All models were trained on the GPU (if available) using the Adam optimizer and a fixed learning rate of 0.001. The batch size was 64, the number of epochs was 20. At each epoch, the error was calculated on the training and validation sets. The MSELoss loss function was used for all models, since we were solving a regression problem.

5 Numerical Experiments

To validate our methodology, we conduct a series of numerical experiments, we built different deep learning models like Recurrent Neural Networks(RNN), Long Short Term Memory(LSTM) and a hybrid model of Convolution Neural Networks and LSTM (CNN-LSTM). This section represents a comparative analysis about the 3 different models and a qualitative assessment of the model's performance.

5.1 Data Gathering and Processing

The dataset used to in this project was compiled from integrating ComEd Hourly energy consumption with historical meteorological data from 18 regional weather stations(Northern Illinois) sourced from Metostat library. After pre-processing and cleaning the data, we also did feature engineering based on the time feature and generated new features like hour, day of the week, is weekend etc. These features were curial to provide context for our time series for our model prediction. This resulted in a rich dataset with 98 features for over 66,000 hourly time steps.

To ensure a valid test to our model's forecasting, the data was split chronologically into training(70%), testing(15%), validation(15%). This helps the model to be trained on the past data to predict future events. We also used StandardScaler to normalize our data and only the training data was fitted.

5.2 Evaluation Metrics

As outlined in our project proposal, we evaluated the final models on the unseen test set using two key metrics:

- **Root Mean Squared Error (RMSE):** This is a standard metric for evaluating regression models, it is calculated by square root of the average of the squared distance between the predicted and actual values. It tells us the typical absolute error magnitude in Megawatts and heavily penalizes large, potentially costly mistakes.

- **Mean Absolute Percentage Error (MAPE):** It is the most intuitive and user friendly metric, as it measures the average error as a percentage of the actual value. This is more understandable and related in a business context while assessing the risk and financial impact. It tells us the relative error and is excellent for communicating the model's overall accuracy to a broader audience in an understandable way.

5.3 Comparative Analysis of Model Approaches

We trained and evaluated three different recurrent neural network architectures to determine the most effective approach: a standard Recurrent Neural Network (RNN), a Long Short-Term Memory (LSTM) network, and our proposed hybrid Convolutional Neural Network and LSTM (CNN-LSTM) model. The performance of each model on the test set is summarized in Table. 1

Table 1: Performance metric for the models

Model Architecture	**RMSE (MW)**	**MAPE (%)**
RNN	2164.78 MW	13.42%
LSTM	523.51 MW	3.37%
CNN-LSTM (Baseline, 2-layer LSTM)	1024.45 MW	7.14%
CNN-LSTM (Advanced, 3-layer LSTM)	205.76 MW	1.40%

By comparing all our models, the CNN-LSTM with 3 layers performs the best with 205.76 MW as RMSE value and 1.40% as MAPE. It achieves the lowest RMSE and remarkably low MAPE.

- The RNN provided the weakest forecast and it struggles with the vanishing gradient problem as it fails to capture pattern with the past 24 hours sequence and gives a bad prediction. This can be see in the plot between the actual and predicted values.

- The standard LSTM shows a dramatic improvement than the RNN model, as the gating mechanism with the LSTM cells gives it an advantage of remembering information for a longer period of time, thus it was able to achieve a RMSE value of 523.51 MW and a MAPE value of 3.37% which reduces the error rate in the prediction.

- The CNN-LSTM model with 64 filters followed by a 2-layer LSTM with 128 hidden units performed better than the RNN model but was less effective than the LSTM model. This suggests that we would need much more depth or optimization to extract features and give a better prediction.

- The CNN-LSTM model with CNN filters set to 128 and 128 hidden layers and a 3-layer stacked LSTM performed the best with a RMSE of 205.76 MW and a MAPE of 1.40%. The model was trained on 98 features including the energy consumption column from the previous 24 hours as an auto-regressive feature. This allows the model to learn from the series own momentum and recent trends, which is one of the strongest possible predictors. The powerful hierarchical feature extraction process then allows the CNN layer to identify nuanced spatial patterns among these features, which are passed to the deeper LSTM network to learn complex temporal dynamics. This combination of auto-regression and deep feature extraction is the primary reason for its superior predictive accuracy.

- An epoch represents one full pass over the entire training dataset. For all the models, the epoch was 20.

5.4 Plots for all the models

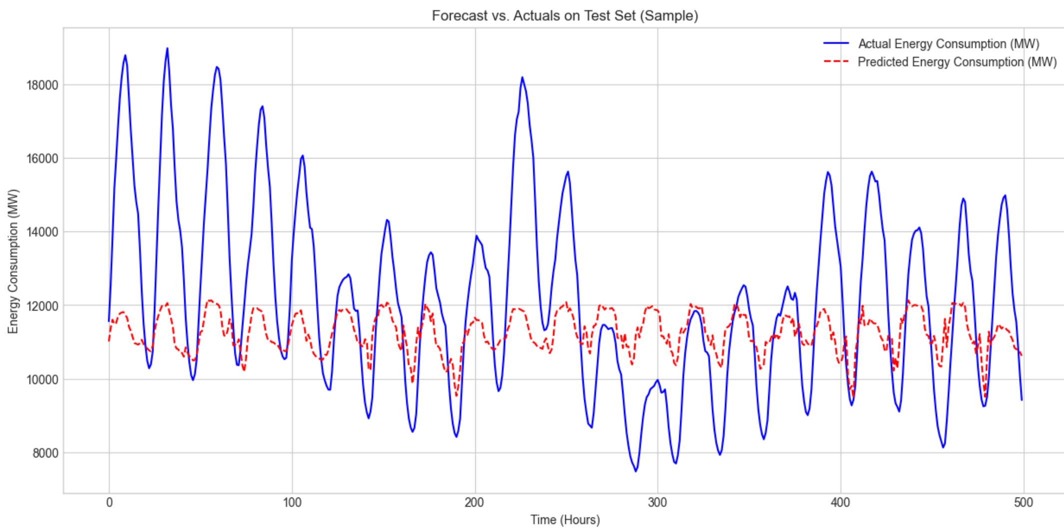

Figure 1: RNN prediction plot

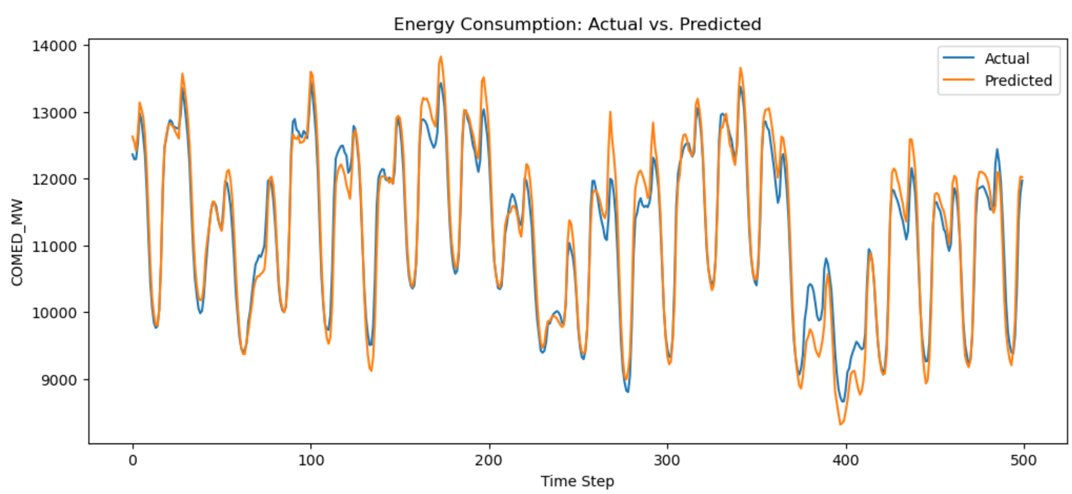

Figure 2: LSTM prediction plot

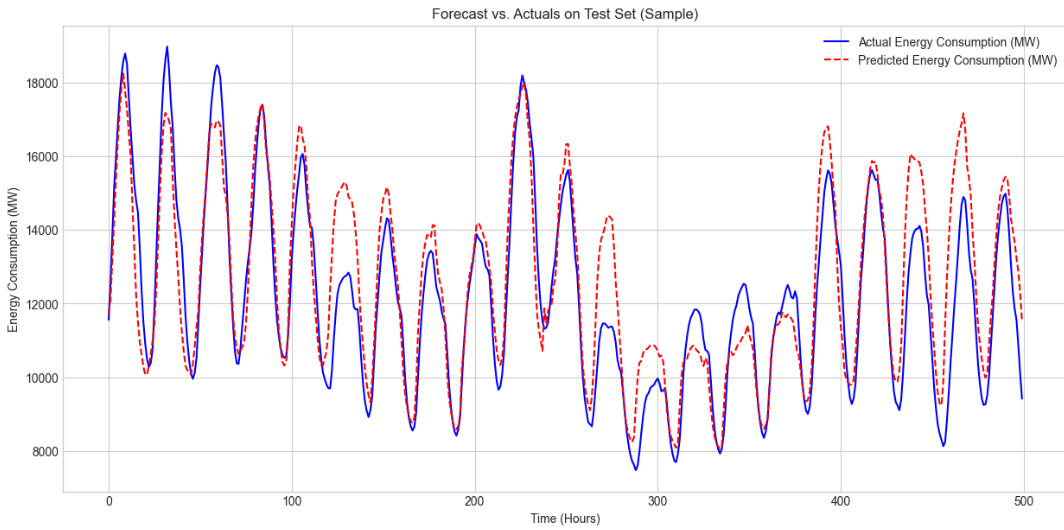

Figure 3: CNN-LSTM 2 layers prediction plot

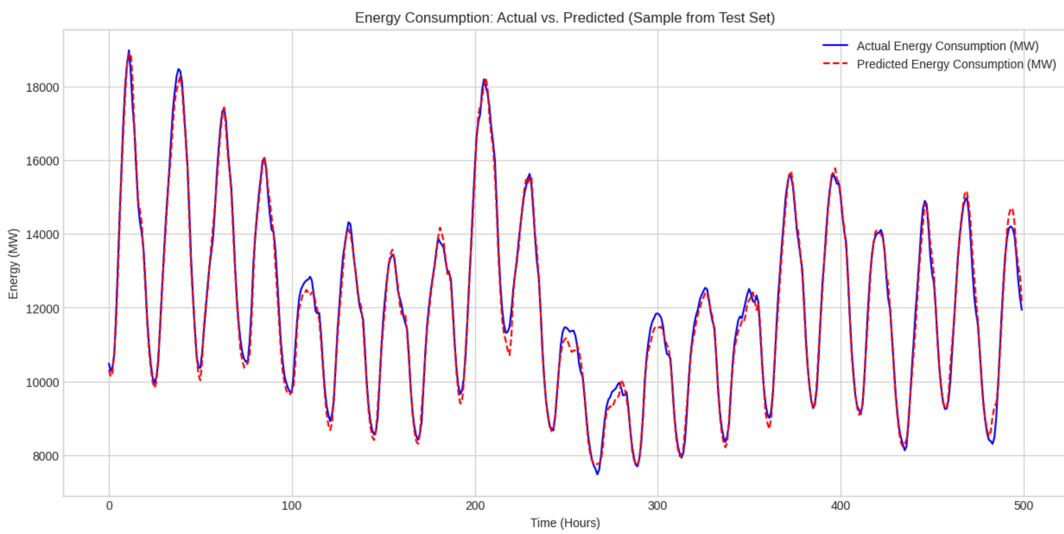

Figure 4: CNN-LSTM 3 layers prediction plot

5.5 Additional Findings

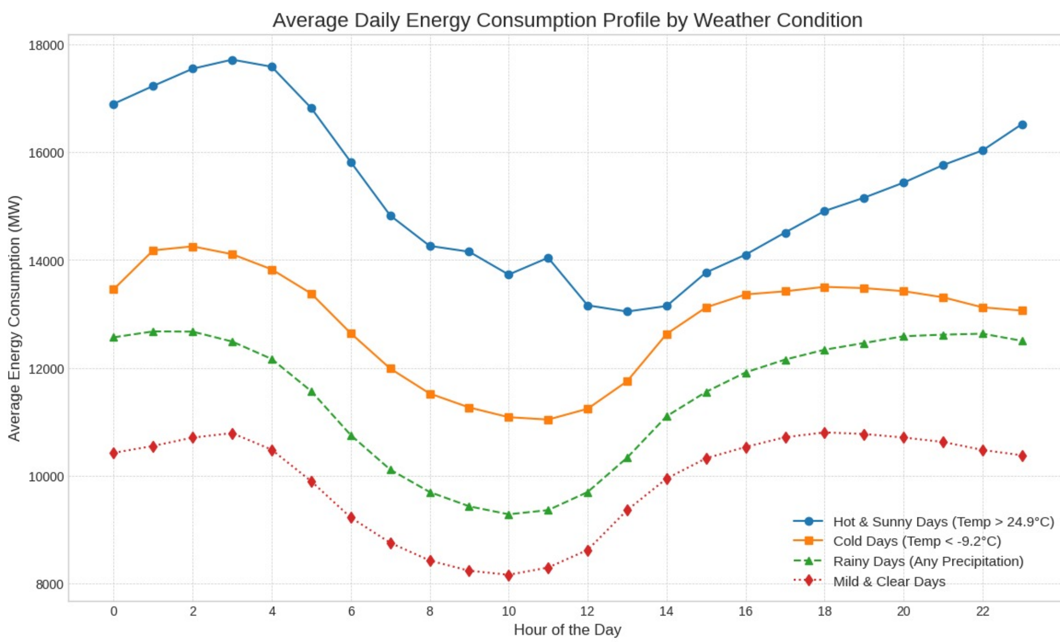

Figure 5: Average Daily Energy Consumption Profile by Weather Condition plot

6 Conclusion

This project successfully demonstrated the hybrid approach of CNN-LSTM architecture model for predicting energy consumption in the Northern Illinois ComEd's region. By creating a comprehensive data set that integrates the weather data from 18 weather station in that region, we were able to train our model to capture complex patterns of energy consumption. Our final model CNN-LSTM with 3 layers achieves a MAPE of 1.40% on a unseen test set and gives better prediction than the other models like RNN and LSTM etc..

The primary finding of our comparative analysis is that the hybrid CNN-LSTM architecture provides a superior approach to this problem. Our model learns the spatial relationship between a wide array of weather features and engineered time features through CNN and then processes the resulting sequence with LSTM to learn about temporal dependencies. The impact of our project is practical: it is a reliable forecasting tool and can help energy and grid providers to forecast needed to optimize resource allocation, enhance grid stability, reduce the risk of outages during peak load time, and lower the operational costs.

6.1 Limitations

Our model has a strong performance, but also it has limitations, our model only predicts historical energy consumption with the historical weather data. To predict future energy consumption, we would need future data. Also, our model does not take pandemics and weather anomalies into consideration.

6.2 Future Directions

By implementing Attention Mechanism on top of LSTM layers we could allow the model to dynamically weigh the importance of different past times when it makes a prediction, which would also improve the accuracy of the model.

We could also tune the hyper parameters and automate it by using techniques like Bayesian optimization or grid search and find the optimal learning rate, layer sizes and other parameters and improve the prediction.

We built a model that can power a simple web dashboard, allowing energy companies and grid managers to view upcoming energy forecasts and peak usage times in a clear and easy-to-use way, so decision-makers can act on the insights.

Finally, this project demonstrates that a thoughtfully designed deep learning model, fed with local data, can accurately predicts energy usage, helping to create a smoother and more dependable energy grid.

References

[1] Musaed Alhussein, Khursheed Aurangzeb, and Syed Irtaza Haider. Hybrid CNN-LSTM model for short-term individual household load forecasting. *IEEE Access*, 8:184162–184172, 2020. doi:10.1109/ACCESS.2020.3028281.

[2] Jieyi Kang and David M. Reiner. What is the effect of weather on household electricity consumption? empirical evidence from Ireland. *Energy Economics*, 111:106023, 2022. doi:10.1016/j.eneco.2022.106023.

[3] Muhammad Rifqi Maarif, Arif Rahman Saleh, Muhammad Habibi, Norma Latif Fitriyani, and Muhammad Syafruddin. Energy usage forecasting model based on long short-term memory (LSTM) and explainable artificial intelligence (XAI). *Information*, 14(5):265, 2023. doi:10.3390/info14050265.

[4] João Victor Jales Melo, George Rossany Soares Lira, Edson Guedes Costa, Antonio F. Leite Neto, and Iago B. Oliveira. Short-term load forecasting on individual consumers. *Energies*, 15(16):5856, 2022. doi:10.3390/en15165856.

[5] American Society of Civil Engineers. 2025 infrastructure report card: Energy, 2025. https://www.infrastructurereportcard.org.

[6] Rong Qu, Ruibing Kou, and Tianyi Zhang. The impact of weather variability on renewable energy consumption: Insights from explainable machine learning models. *Sustainability*, 17(1):87, 2025. doi:10.3390/su17010087.

[7] M. M. Sachin, M. Paily Baby, and A. Sudharson Ponraj. Analysis of energy consumption using rnn-lstm and arima model. *Journal of Physics: Conference Series*, 1716(1):012048, 2020. 10.1088/1742-6596/1716/1/012048.

A Appendix

You may include other additional sections here.