Prediction of Individual Household Electric Power Consumption

Catherine Hua
New York University
jh6780@nyu.edu

Lyric Li
New York University
rl3754@nyu.edu

Anni Zheng
New York University
az1932@nyu.edu

Abstract

Accurate forecasting of energy consumption is critical for optimizing power distribution, reducing costs, and promoting sustainable development. This study analyzes household energy usage patterns using Autoregressive Integrated Moving Average (ARIMA), eXtreme Gradient Boosting (XGBoost), and Long Short-Term Memory (LSTM). Performance was evaluated using metrics such as RMSE and R-squared, supported by visualizations comparing predictions with actual values. The best-performing model was LSTM, applied to the comprehensive dataset without the data aggregation step. This demonstrated its ability to incorporate big data and capture complex relationships, providing insights into real-world energy demand applications, though the issue of overfitting may arise.

1 Introduction

1.1 Background

Accurate energy prediction is crucial for effective energy management and sustainability. This project aims to identify the best model to capture the patterns and seasonal impacts of household energy consumption. Specifically, three models — ARIMA for statistical modeling, XGBoost for machine learning, and LSTM for deep learning — were selected for their popularity, reputation, and ability to address time-series problems.

To tailor the data to the specific structures, functionalities, and capabilities of all models for the purpose of uniform comparison, techniques such as the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF) were used to determine the appropriate time lags for creating lag features in the statistical model. The performance of the model was evaluated using a range of metrics, including root mean squared error (RMSE) and R-squared, which provided a thorough assessment of each model's ability to accurately capture seasonal and periodic trends in energy consumption. In addition, to support data exploration and facilitate the interpretation of results, visualizations such as box plots and line plots were generated to compare predicted values, true values, and absolute errors.

Finally, tuning techniques were employed, and the final best-performing models were used to generate predictions which were tested against the test dataset to assess their ability to forecast future energy consumption. This comprehensive approach enabled the assessment of not only the predictive accuracy of the models but also their suitability for real-world applications in energy demand forecasting.

1.2 Related Work

With the rise of locally distributed energy sources, such as small-scale renewable generation and storage, energy prosumers play an increasingly significant role in the energy landscape. As Pirbazari mentioned, this decentralized activity introduces instability in the power grid due to fluctuating supply and demand at the local level[4]. Therefore, accurate forecasting of energy usage is essential for optimizing energy distribution, reducing costs, and supporting sustainability at the local level through the incorporation of new methods of energy distribution.

Machine learning techniques have gained popularity for time series forecasting, offering a variety of methods ranging from statistical models to neural networks. Kontopoulou et al. compared ARIMA, SVM, Random Forest, LSTM, and other methods, finding one-fit-all model for capturing all kinds of complex relationship within data[2]. For example, ARIMA performed well for weather and traffic, while LSTM was superior in financial and utility forecasting. These mixed results suggest that real-world datasets demand domain-specific approaches.

On the other hand, algorithms like XGBoost have emerged as efficient alternatives. As Zhang et al. describe, XGBoost combines boosting techniques with feature engineering to deliver strong predictive performance [6]. This study extends the comparison to include ARIMA, LSTM, and XGBoost, providing insights specific to energy forecasting.

1.3 Work Distribution

Data Preprocessing and Cleaning: Catherine was responsible for preparing the dataset for analysis, including aggregating data into hourly time grain and reducing it to two-year range, ensuring the all models used in this study received clean and consistent input data for fair comparison.

Statistical Modeling (ARIMA): Anni conducted the statistical modeling using the seasonal ARIMA method. This included identifying appropriate parameters and seasonality trend through techniques like the ACF and PACF, as well as evaluating the model's performance on the dataset.

Machine Learning Modeling (XGBoost): Lyric implemented and optimized the XGBoost model. This involved engineering lag features, tuning hyperparameters, and analyzing the model's ability to capture complex temporal data relations with and without features.

Neural Networks (LSTM): Catherine took charge of developing and training the LSTM model. This process required designing the network architecture, selecting appropriate activation functions and features, and fine-tuning the model to effectively capture temporal patterns.

2 Dataset

The dataset used for this project is the Individual Household Electric Power Consumption dataset. It contains measurements of electric power consumption from a single household located in Sceaux, France (approximately 7 km from Paris), recorded over a 47-month period from December 2006 to November 2010. Despite the presence of 1.25% missing values across more than 2 million records, the dataset provides sufficient data for modeling and estimation due to its extensive time span and high-frequency measurements. The dataset consists of nine columns: 1.Date, 2.Time, 3.Global Active Power, 4.Global Reactive Power, 5.Voltage, 6.Global Intensity, 7.Sub_metering_1: Cooking devices (e.g., dishwasher, oven, microwave), 8.Sub_metering_2: Household chores (e.g., washing machine, refrigerator, lighting), 9.Sub_metering_3: Climate control (e.g., electric water heater, air conditioner).

To reduce overfitting issues among models and to enable ARIMA to handle large data more seamlessly, we used only half of the original dataset (17,849 data points), which encompasses two full years of household electric power consumption data spanning from December 2006 to December 2008. Additionally, we performed a train-test split, ensuring all models uniformly input 18 months of data while being expected to forecast 4 months, compared against the processed dataset.

Data dimensionality reduction was also performed by aggregating the minute-level time grain of the energy consumption measurements in the original dataset into hourly intervals, providing a manageable dataset size for model training and evaluation while preserving the key trends and seasonal patterns present in the original data.

2.1 Exploratory Data Analysis

To better understand the dataset and its characteristics, a thorough exploratory data analysis was conducted, focusing on trends, seasonality, and anomalies in household energy consumption.

The data reveal distinct periodicity at both daily and yearly levels, reflecting the household's energy usage routines and seasonal variations. Peaks in energy consumption are observed during the winter months, likely due to increased heating requirements, while the summer months show relatively lower energy usage. Additionally, a significant anomaly is present at the beginning of the test period (August 2008), where energy consumption drops sharply to an abnormally low level compared to the preceding data. This unexpected drop deviates from established patterns and poses challenges for predictive modeling. Models trained on the earlier data, which exhibit consistent trends, struggle to adapt to this sudden change, leading to inaccurate predictions in the initial stages of the test period.

The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots were also graphed to understand the temporal dependencies and identify the appropriate lags for modeling. The ACF plot reveals gradually decreasing correlations by lags, indicating relatively strong persistence in energy usage patterns over time. Meanwhile, the PACF plot shows a sharp drop after the first lag, followed by smaller significant correlations at higher lags. These observations suggest that energy consumption is influenced by immediate past values, as well as periodic effects over longer intervals.

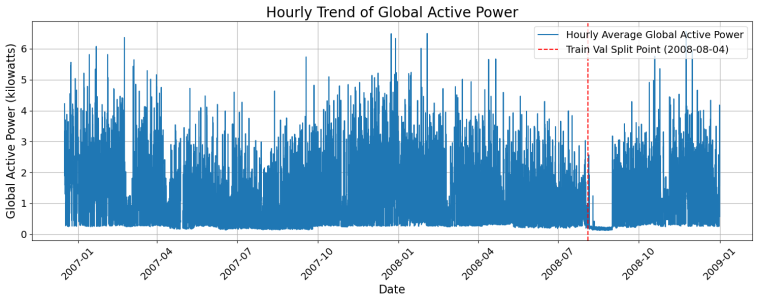
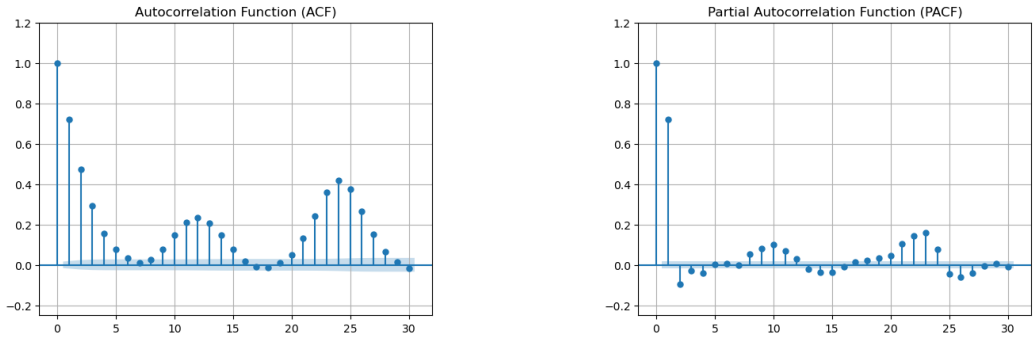


Figure 1: Hourly Trend of Global Active Power.



(a) ACF Graph

(b) PACF Graph

Figure 2: ACF and PACF Graphs

3 Methodology

3.1 ARIMA

The ARIMA model was constructed and evaluated using Python 3.11.8. The autocorrelation function (ACF) and partial autocorrelation function (PACF) were visualized using plot_acf and plot_pacf within 30 lags. The PACF graph showed a sharp drop starting from lag 1, suggesting an autoregressive order (p) of 1. The ACF graph exhibited smoother spikes, with oscillatory patterns beginning at lag 2, which is less obvious to determine the exact lag-order. Through iterative testing using the RMSE metric, the moving average order (q) was also determined to be 1.

On the other hand, to assess stationarity, the Augmented Dickey-Fuller (ADF) test was conducted. The test yielded a statistic of -12.0013 and a p-value of 3.34×10^{-22}, indicating strong confidence in rejecting the null hypothesis of a unit root. Thus, the differencing order (d) was determined to be 0.

Seasonality was examined based on oscillatory patterns observed in both ACF and PACF plots. The ACF plot showed periodic trends between lags 7 to 16 and 17 to 30, while the PACF plot displayed similar oscillations between lags 2 to 14 and 15 to 26. These trends suggested a seasonality period of approximately 12 hours. To refine the estimate, the model performance was tested with seasonality ranges of 12, 24, and up to 24×7 hours, finding the best fit with a 72-hour (3-day) seasonal period.

A subset of data from 2007-01-01 00:00:00 to 2007-01-03 23:00:00 was used to determine the seasonal order. The ADF test for this subset returned a statistic of -2.4016 and a p-value of 0.1413, indicating non-stationarity. Consequently, the seasonal differencing order (D) was set to 1. We have also tested the seasonal autoregressive (P) and moving average (Q) terms, which $P = 1$ and $Q = 1$ are the optimal choice. This means that the seasonal component does not depend on previous cycles.

The final model was specified as ARIMA(y_train, order=(1, 0, 1), seasonal_order=(1, 1, 1, 48)). Its performance was evaluated against the true data using RMSE and R-squared metrics, calculated as 0.8757 and 0.1510, respectively, via the ‘sklearn.metrics‘ library.

3.2 XGBoost

In contrast to ARIMA, XGBoost is a machine learning algorithm that constructs an ensemble of decision trees to enhance prediction accuracy. It operates by iteratively correcting errors from previous trees, optimizing a specified loss function, and efficiently handling large datasets. In this project, XGBoost is applied to predict energy usage based on historical data, capturing relationships between input features (e.g., time-based variables

such as year, month, day, and hour) and the target variable (power usage). By leveraging gradient boosting, XGBoost is designed to model complex patterns and trends in energy consumption.

The model parameters were set to 30 tree depth, 0.01 learning rate, and 42 random seed, with 1000 epochs. To improve model performance, we conducted feature engineering, introducing meaningful features such as 'is_weekend', 'is_holiday', and 'day_of_week' to provide temporal context. Additionally, we created combined features like 'hour_day' and categorized hours into 'hourly_mapping' based on patterns identified during exploratory data analysis (EDA). This improved the model's R-squared value from 0.17 to 0.25, reflecting its enhanced ability to capture variations in energy consumption across different temporal conditions.

3.3 LSTM

LSTM is a gated RNN that has an input, output, and remember gate controlling writing, reading and storing information in the neuron. Suppose we have inputs $\mathbf{x}_t$ at time t, hidden state $\mathbf{h}_{t-1}$, and cell state $\mathbf{c}_{t-1}$ from the previous time step. First the forget gate decides what information to discard from the previous state through $\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{h}_{t-1} + \mathbf{b}_f)$, where W stands for weights. The input gate decides what information to store in this cell state with $\mathbf{i}_t = \sigma(\mathbf{W}_i + \mathbf{h}_{t-1} + \mathbf{b}_i)$ then the candidate cell state is therefore computed as $\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{h}_{t-1} + \mathbf{b}_c)$. Then the new cell state is updated with the candidate cell state $\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t$. Lastly the output gate determines the output from current cell and pass to the next cell state, $\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{h}_{t-1} + \mathbf{b}_o)$, also updates the new hidden state $\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$. These operations are performed recursively ensuring the quality of the flow of information and learn temporal dependencies.

We adjusted the dataset specifically to the need of LSTM using a standard input-output sequence windowing approach. N_STEPS_IN represents the lookback window, which is the number of time steps in the input sequence. N_STEPS_OUT represents the forecast horizon, which is the number of time steps to predict. In our case, both were set to 5 in consideration of the size of the training data.

The baseline LSTM has the structure as shown in Table 1 and we used 20 epochs, batch size 16 in proportionally to the size of the model and data.

Layer	Parameters	Description
Input	`input_shape=(N_STEPS_IN, 1)`	Input sequence of shape (2, 1).
LSTM	`units=64, activation='relu'`	First LSTM layer with 64 units, returns sequences.
Dropout	`rate=0.1`	Regularization with dropout rate 0.1.
LSTM	`units=32, activation='relu'`	Second LSTM layer with 32 units.
Dropout	`rate=0.1`	Regularization with dropout rate 0.1.
Dense	`units=N_STEPS_OUT`	Fully connected layer with 5 output units.

Table 1: LSTM Model Structure

3.3.1 LSTM - hourly active power

In accordance with other two models, baseline LSTM also used over $17,849$ hourly consumed energy data points. We set 64 and 32 units respectively for the hidden layers due to the relatively small amount of data, and applied ReLU to introduce non-linearity. Return sequences was set to True to ensure the layer outputs the entire sequence instead of just the last time step. For each hidden layer, we added a dropout layer to help prevent overfitting by randomly setting 10% to zero during training. Last end the structure with a fully connected layer with 5 units which produces the final predictions. We chose Adam optimizer that combines the benefits of momentum and adaptive learning rates. We also set the learning rate to 0.0001 that allowed for more stable convergence, as well as clipping gradients to prevent gradient explosion. Other techniques involve ReduceLROnPlateau to adjust learning rate during training, and EarlyStopping to improve efficiency.

3.3.2 LSTM - hourly active power with added features

We incorporated features such as 'is_weekend', 'is_holiday' to provide temporal context and investigate if extra information will help improving the performance.

3.3.3 LSTM - global active power

Another experiment was to use the original data with global active power that has $1,070,566$ data points. We used the same model with the exact structure and parameters, without additional feature engineering. The only adjustment was setting Dropout to 20% for each layer, considering the increased size of the input data.

4 Results and Analysis

4.1 ARIMA

The performance of the ARIMA model was evaluated by comparing its predictions with the original data. As shown in Figure 3, the ARIMA model struggled to capture the complexity of the time series. The predicted values displayed smooth and regular oscillations, which significantly deviated from the irregular and erratic patterns of the observed data. This limitation stems from ARIMA's assumption of linear relationships within the data, which fails to account for real-world complexities influenced by exogenous factors. For example, the spikes observed between 2008-11-05 and 2008-11-06 highlight the model's inability to predict the magnitude of sudden changes, as ARIMA naively focuses on regular patterns without considering external drivers.

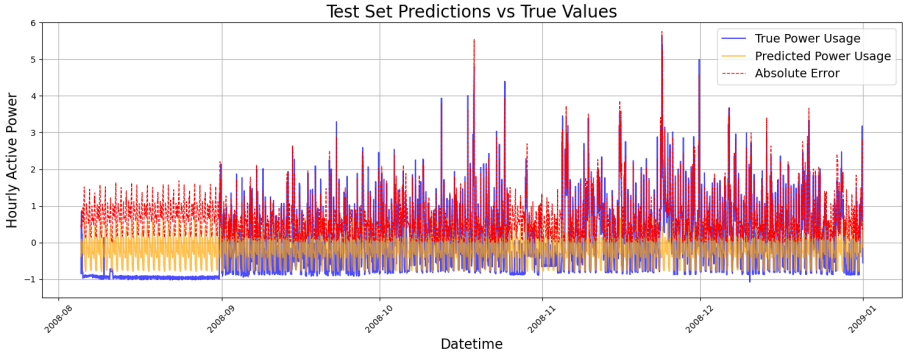


Figure 3: Comparison of True vs. Predicted Power Usage for ARIMA

The dataset also contains segments of low-frequency data, such as between 2008-11-03 and 2008-11-05 as shown in Figure 4, which further obscure the structure that ARIMA assumes. These periods of sparse activity can mislead the model's understanding of the underlying dynamics, reducing its effectiveness in capturing the nuances of the time series.

Moreover, the seasonality trend assumed by the model (72 hours) was not strictly observed in the data. Irregular spikes occurred, such as between 2008-11-04 and 2008-11-06, while other periods exhibited more regular patterns or no spikes at all, such as in the earlier days of the dataset. This variability demonstrates that the true complexity of the data extends beyond ARIMA's assumptions of consistent and regular seasonality.

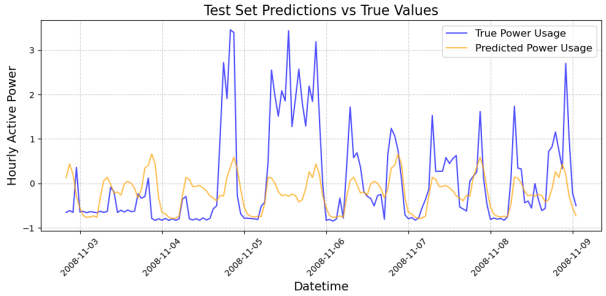


Figure 4: Comparison of True vs. Predicted Power Usage for ARIMA (Zoomed In)

4.2 XGBoost

One main reason for XGBoost to have such a unsatisfactory performance is it is constrained by the lack of ability to understand sequential information inherent in time series data. To address this, we attempted to incorporate sequential information into the model by introducing moving windows, such as the values one and two hours prior, as well as lagging statistics like the mean and standard deviation of the preceding two-hour window. While these features significantly resulted in seemingly perfect predictions, this was ultimately due to data leakage—since these features inadvertently gave the model access to the true values from two hours prior.

Additionally, we experimented with taking the model's own predicted values and using them as new features for predicting future energy usage, trying to make the prediction recurrently. We tested lag intervals from 2 to 10, but this method performed worse. A key issue was the significant energy drop at the beginning of the test period, which the model failed to account for. Once the predicted values captured this initial drop, subsequent predictions for energy usage remained unrealistically low throughout the test period. This behavior highlighted a cascading error effect, where incorrect early predictions adversely impacted the entire prediction sequence.

Despite these efforts, model performance plateaued at an RMSE of 0.822 and an R-squared value of 0.25. While the predicted power usage generally aligns with the overall trend of the true power consumption, it

shows significant deviations during periods of sharp changes, particularly in late 2008. Furthermore, the model fails to capture the unexpected drop in energy usage observed between August 2008 and September 2008. This limitation is likely due to the absence of similar patterns in the training data and XGBoost's inherent challenges in modeling sequential dependencies.

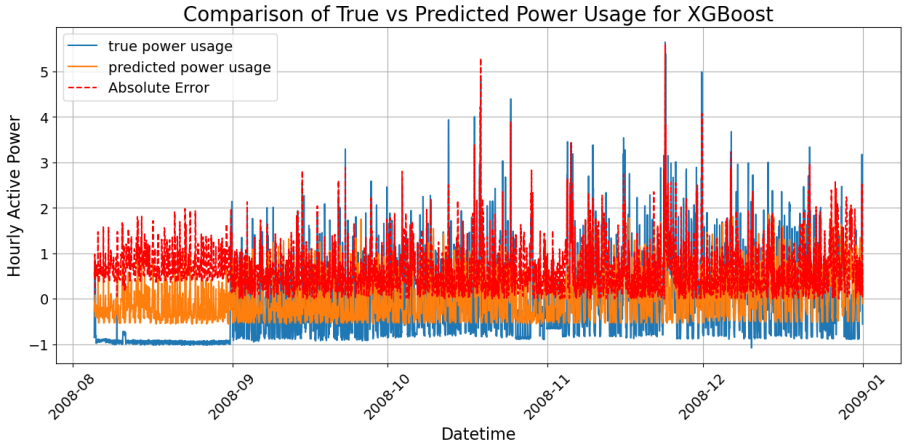


Figure 5: Comparison of True vs. Predicted Power Usage for XGBoost

4.3 LSTM

Among the three LSTM scenarios, the global active power model significantly outperformed the other two, achieving an R-squared score of 0.83 and a low RMSE of 0.38, which highlights its ability to effectively capture temporal dependencies within the dataset. In contrast, the baseline model reached only an R-squared score of 0.2 and RMSE of 0.85, likely due to the use of a much smaller dataset. Meanwhile, the model with feature engineering showed a slight improvement with an R-squared score of 0.21 and a RMSE of 0.84, but its overall performance remained relatively low.

Model	**R^2 Score**	**Test RMSE**
Baseline Model	0.20293	0.84870
With Feature Engineering	0.21751	0.84090
Global Active Power	0.84122	0.37446

Table 2: Performance metrics of different LSTM models.

One noteworthy issue was that the model trained on the 1 million data points experienced exploding gradients, despite explicitly clipping gradients to values below 1. The growing loss strongly indicated overfitting, as the model fit the training data too well but failed to generalize to other datasets. Another contributing factor was the uniformity among the three models: LSTM was constrained by limited choices of loss functions, hyperparameters, and data inputs. It also could not utilize a validation dataset to monitor training, as ARIMA does not support such functionality. Without validation monitoring, the training process relied solely on the training loss. Additionally, the loss function used was MSE, which did not perform well for skewed data due to its tendency to heavily penalize errors, leading to a biased optimization process. We expect the LSTM model with global active power data to perform better if these constraints are addressed.

Despite the differences, all three models struggled to fully capture the data's patterns. As shown in Figure 9, the models performed poorly in scenarios where there was no power usage. The figure is somewhat misleading, as the portions appear to have similar shapes, yet the R-squared scores reveal the models' inability to make precise predictions. Further investigation showed that the models tended to average out predictions rather than mapping the dynamic patterns in the data. As shown in Table 3, the mean predictions for all three models were around 0, confirming this observation. Moreover, the subsequent predictions reinforced this trend, as they were consistently much lower than the true values. Instead of generating meaningful predictions, the models appeared to be providing "placeholder" values, substituting for their lack of understanding of the underlying patterns.

Model	**Min Prediction**	**Max Prediction**	**Mean Prediction**
Baseline Model	-1.7704	2.6088	-0.0727
With Feature Engineering	-1.2457	2.2885	-0.1077
Global Active Power	-0.7577	15.1074	-0.1053

Table 3: Prediction statistics for different models.

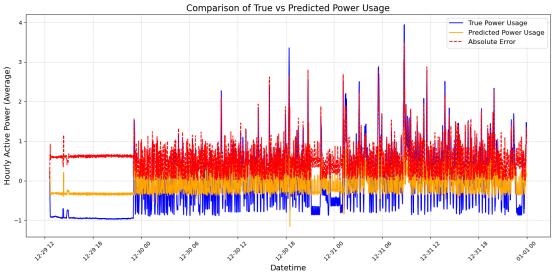


Figure 6: Hourly active power

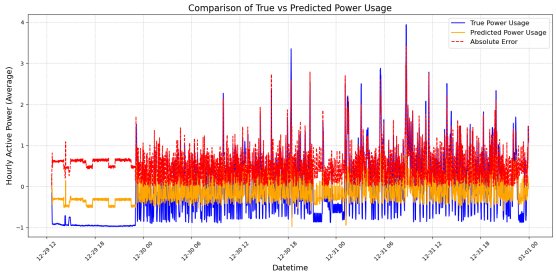


Figure 7: With feature engineering

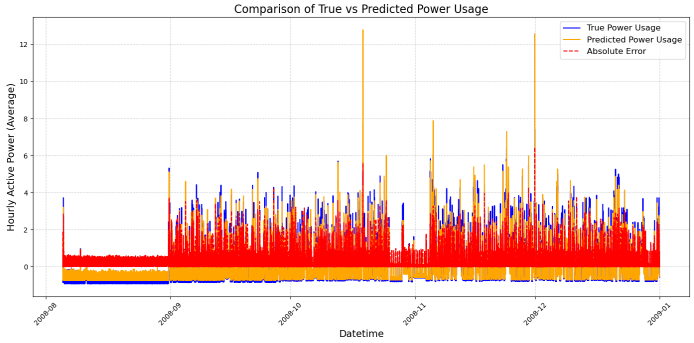


Figure 8: Global active power

Figure 9: Comparison of True vs. Predicted Power Usage for various LSTM approaches

5 Conclusion and Future Work

This study compared ARIMA, XGBoost, and LSTM for forecasting household energy consumption, revealing distinct strengths and limitations. ARIMA struggled with the data's complexity due to its linear assumptions. XGBoost showed improvement after feature engineering but continued to struggle with its limited ability to model sequential dependencies. LSTM, on the other hand, demonstrated potential for capturing temporal relationships with big data input, particularly when provided with richer datasets and well-incorporated features.

In the future, since both XGBoost and LSTM perform better with larger datasets, using a broader dataset with delicate preprocessing and tailored training strategy could improve the scalability of their forecasts. To address the failure of all three models in handling anomalies and unexpected events, future work will focus on integrating anomaly detection techniques during preprocessing. For instance, augmenting the training dataset with synthetic anomalies could improve model robustness and adaptability to irregular trends. This study focused on energy data from a single household, but future work could explore other types of data, like web traffic or weather patterns, to better understand how different data characteristics affect model performance.

References

[1] Hebrail, G.& Berard, A. (2006) Individual household electric power consumption. *UCI Machine Learning Repository*. https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption

[2] Kontopoulou, V.I., Panagopoulos, A.D., Kakkos, I.& Matsopoulos, G.K. (2023, July 30) A review of ARIMA vs. Machine Learning Approaches for time series forecasting in Data Driven Networks. *MDPI*. https://www.mdpi.com/1999-5903/15/8/255

[3] Liao, J.-M., Chang, M.-J.& Chang, L.-M. (2020, April 10) Prediction of air-conditioning energy consumption in R&D building using multiple machine learning techniques. *MDPI*. https://www.mdpi.com/1996-1073/13/7/1847

[4] Pirbazari, A.M. (2021, May 20) Predictive analytics for maintaining power system stability in Smart Energy Communities. *UiS Scholarly Publishing Services*. https://ebooks.uis.no/index.php/USPS/catalog/book/83

[5] Sharma, V. (2022, January 1) Exploring the predictive power of machine learning for energy consumption in buildings. *Journal of Technological Innovations*. https://jtipublishing.com/jti/article/view/41/150

[6] Zhang, L., Bian, W., Qu, W., Tuo, L. & Wang, Y. (2021, April) Time Series forecast of sales volume based on XGBoost. *IOP Conference Series: Materials Science and Engineering*.https://iopscience.iop.org/article/10.1088/1742-6596/1873/1/012067/pdf