
Photo by Pexels

Thesis Defense: Energy Conservation and Prediction in Smart Cities

Evaluating BiLSTM-GRU vs. RNN for Time Series Prediction

By

Paul Ebikina Ifidi
22101769
(MSc. Cmputer Engineering)

Supervisor: Dr. Parvaneh Esmaili

Table of Contents

1

Introduction

Objective & Motivation

- The objective is to evaluate the performance of LSTM-GRU versus RNN models. Specifically, we focus on time series prediction.
- This evaluation is crucial for understanding energy consumption patterns and making accurate predictions in smart cities.
- Accurate predictions enable better energy management, reducing costs and improving sustainability in urban environments.
- Through our analysis, we aim to highlight the strengths and weaknesses of each model, helping to guide future implementations.

Problem Statement

Growing Demand for Energy-Efficient Solutions in Smart Homes

•**Urbanization and Energy Challenges:**
- Rapid urbanization has increased energy consumption.
- Traditional energy systems are struggling to meet rising urban demands.

•**Role of AI in Energy Management:**
- AI offers innovative solutions for energy optimization and conservation.
- Predictive models are essential for smart energy management in homes.

Key Research Challenges

•**Model Selection:**
- Comparing deep learning models (LSTM-GRU) vs. traditional RNN machine learning models.
- Evaluating their effectiveness in forecasting energy usage.

•**Data Processing and Feature Selection:**
- Importance of preprocessing and selecting relevant features for accurate predictions.

•**System Integration:**
- Integrating predictive models into smart home energy management systems.
- Ensuring models are scalable and efficient for real-time applications.

Problem Statement contd.

Objectives of the Study

- Comparative Analysis:
  - Evaluate LSTM-GRU and RNN models in predicting energy consumption patterns.
  - Assess the models' ability to capture short and mid-term usage fluctuations.

- Feature Importance and Model Performance:
  - Investigate the impact of different input features on energy consumption predictions.
  - Balance between model interpretability and predictive accuracy.

- Practical Implications:
  - Provide recommendations for model selection and deployment.
  - Inform stakeholders on integrating AI-driven models for efficient energy management in smart homes.

Literature Review
(AI in Smart Home Energy Management)

- Role of AI & ML:
- Enhance energy efficiency.
- Predict and optimize usage patterns.

Key Findings:
- Deep Learning Models:
  - LSTM & GRU:
    - Effective for sequential data.
    - Superior in capturing energy trends.
- RNN:
  - Suitable for time series prediction.
  - Challenges with long sequences.

Literature Review contd.

<u>**Comparative Analyses**</u>

- Performance Metrics:
  - Deep learning often outperforms traditional models.
  - Metrics: MAE, RMSE, R-squared.

- Feature Engineering:
  - Crucial for model performance.
  - Relevant features: temperature, humidity, appliance usage, etc.

<u>**Research Gaps**</u>

- Comparative Studies:
  - Need for standardized benchmarks.

- Scalability & Real-Time:
  - Optimizing models for real-world applications.

- Interpretability:
  - Ensuring transparency and user trust.

Methodology

Dataset Overview

- Source: UCI Machine Learning Repository
- Details:
  - Energy consumption data.
  - Features: temperature, humidity, appliance usage, etc.

Preprocessing Steps

- Data Cleaning:
  - Handle missing values.
  - Normalize data for consistency.
- Feature Selection:
  - Identify key variables affecting energy use.
- Split Dataset:
  - Training set: 80%
  - Testing set: 20%

Methodology contd.

Model Development

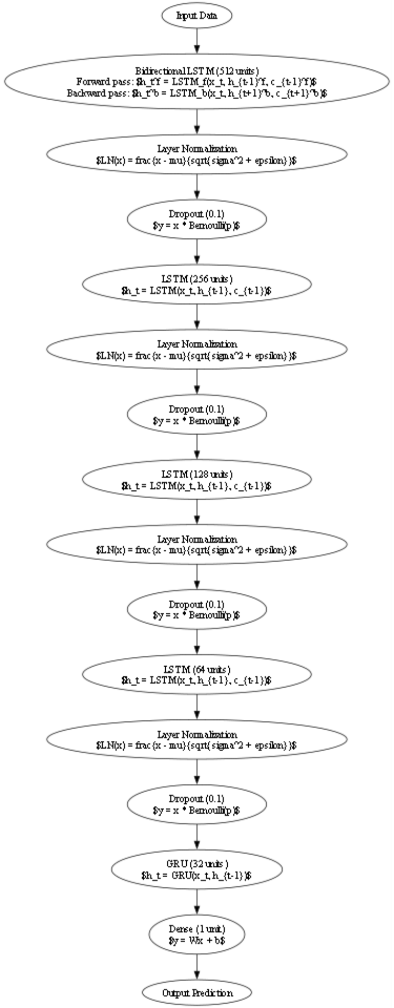
- Algorithms Used:
- LSTM
- GRU
- RNN

- Hyperparameter Tuning:
- Grid search for optimal parameters. (Epochs: 50, 100, 150 ; Batch: 32, 64, 128)
- Focus on improving accuracy and reducing error.

Evaluation Metrics

- Metrics:
- Mean Absolute Error (MAE)
- Root Mean Square Error (MSE/RMSE)
- R-squared

- Validation:
- Cross-validation to ensure model robustness.



Proposed Model Architecture

Objective & Motivation

- The objective is to evaluate the performance of LSTM-GRU versus RNN models. Specifically, we focus on time series prediction.
- This evaluation is crucial for understanding energy consumption patterns and making accurate predictions in smart cities.
- Accurate predictions enable better energy management, reducing costs and improving sustainability in urban environments.
- Through our analysis, we aim to highlight the strengths and weaknesses of each model, helping to guide future implementations.

Key Components of Proposed Model

- **Bidirectional LSTM:** Data is processed in both forward and backward direction thereby allowing model to capture information from both past and future states. The outputs from both the forward and backward LSTMs are concatenated at each time step, providing a richer representation for each position in the sequence.

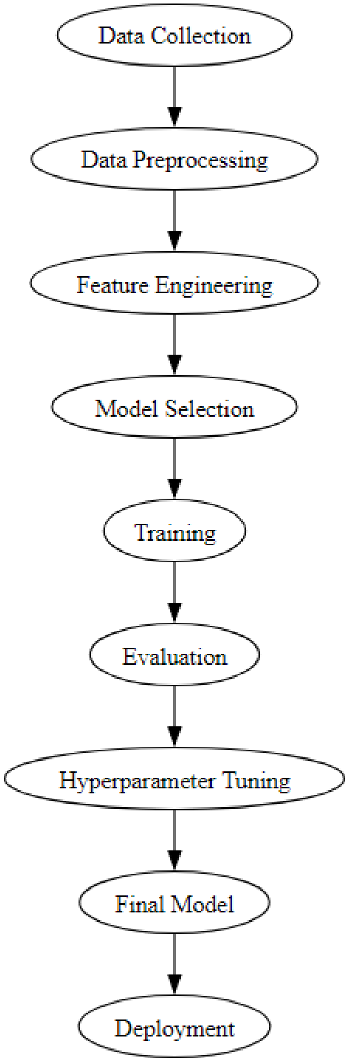
- **Dropout:** Regularization to avoid overfitting by dropping 0.1 fraction of neurons during training. Help model to learn more robust features without relying on specific neurons. Generalization is improved

- **Layer Normalization:** To stabilize and accelerate training across features for each data point unlike batch normalization which uses sequences

10

Methodology Steps

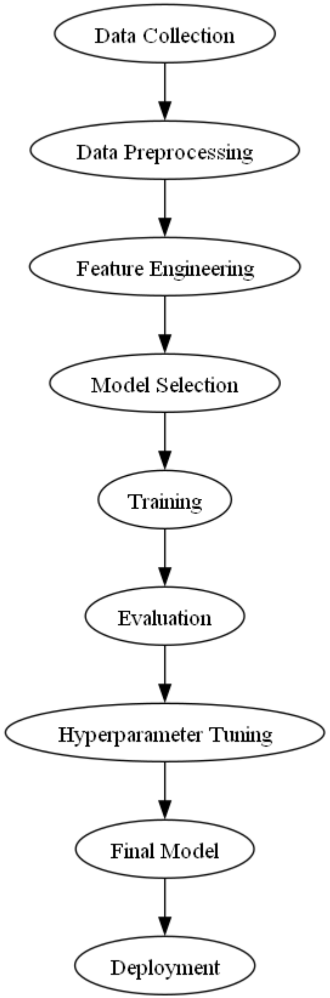
Flowchart of Implementation



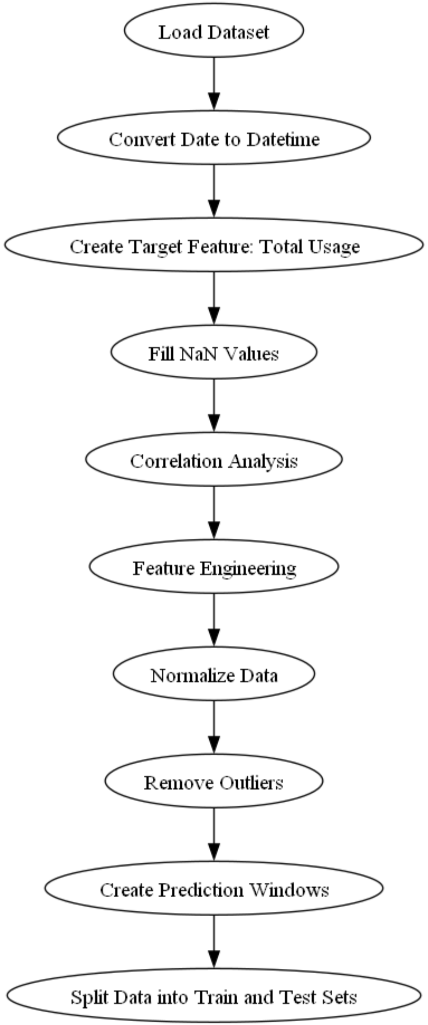
- **A detailed flowchart illustrates the entire model methodology, starting from data collection to final deployment.**

Modeling

Methodology



- The modeling methodology begins with data collection, cleaning, and preprocessing required for impactful model training.
- Model selection involves choosing between LSTM-GRU and RNN models based on their suitability for the given task.
- After models are trained and evaluated on their respective data, further tuning is performed to enhance their performance.
- Finally, the best-performing model undergoes thorough validation before being prepared for potential deployment.



Data Prep

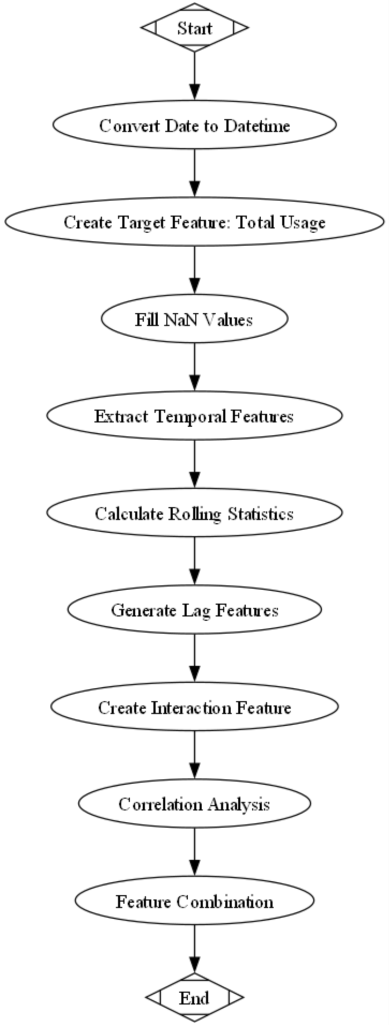
Preprocessing Steps

- The dataset is first loaded, and dates are converted to datetime format to ensure consistency during analysis.
- A target feature, total_usage, is created, and any NaN values in the dataset are filled to avoid calculation errors.
- Correlation analysis is performed, followed by feature engineering efforts to enhance model training accuracy.
- Finally, data is normalized, outliers are removed, and prediction windows are created before splitting data into training and test sets.

13

Feature Engineering

Feature Engineering Steps



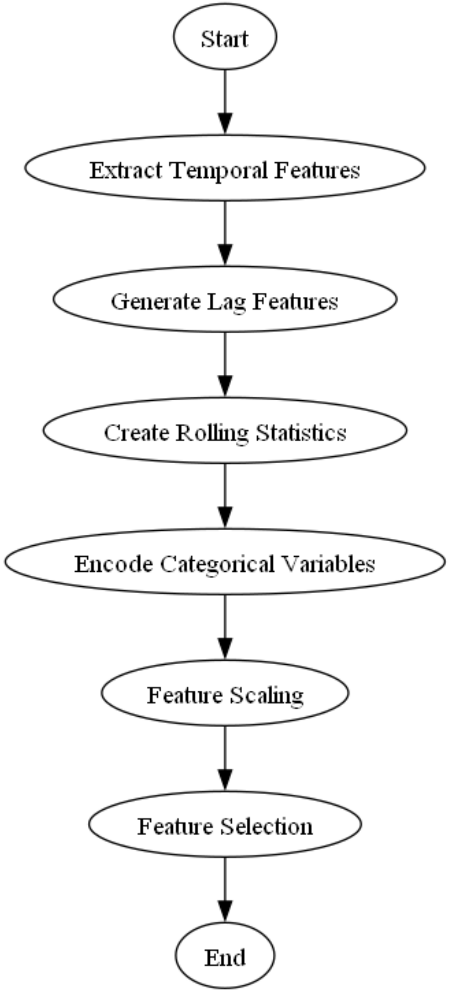
- **Start (start):** This node represents the beginning of the feature engineering process.

- **Convert Date to Datetime (A):** In this step, the date column in the dataset is converted to datetime format. This conversion ensures that the date information is properly recognized and can be used for further analysis.

- **Create Target Feature: Total Usage (B):** Here, a new target feature called "Total Usage" is created by summing the "Appliances" and "lights" columns. This combined feature represents the total energy usage for each observation.

- **Fill NaN Values (C):** This step involves filling any missing values (NaN) in the dataset. In this case, the missing values are filled using backward fill (bfill) method.

Feature Engineering contd.

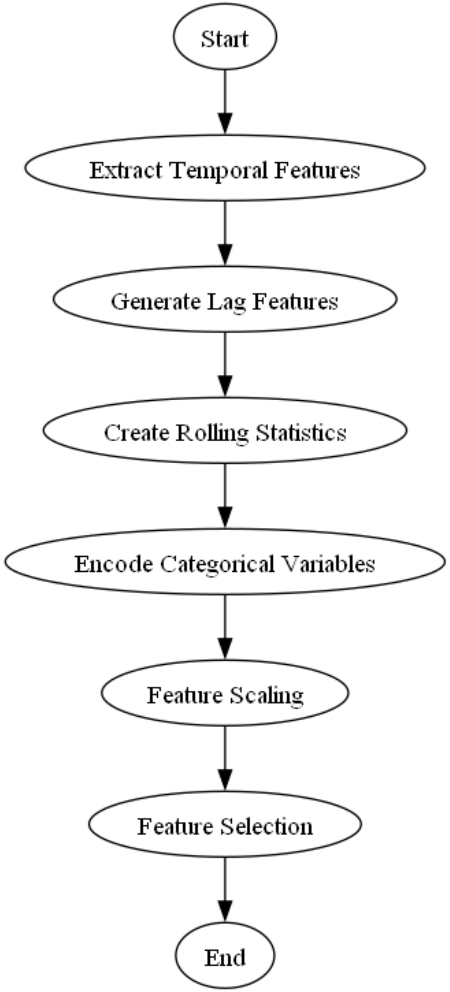


- **Extract Temporal Features (D):** Temporal features such as hour of the day and day of the week are extracted from the datetime column. These features capture time-related patterns in the data, which can be useful for modeling energy consumption behaviour.

- **Calculate Rolling Statistics (E):** Rolling statistics, including rolling mean and rolling standard deviation, are calculated for the "Total Usage" feature over a specified window size. These statistics provide information about the trend and variability in energy consumption over time.

- **Generate Lag Features (F):** Lag features are created by shifting the "Total Usage" feature by one and twenty-four time steps. These lag features capture the relationship between current and past energy consumption, which can help predict future consumption.

Feature Engineering contd.



Feature Engineering Steps

- **Create Interaction Feature (G):** An interaction feature is created by multiplying the *hour of the day* with the *day of the week*. This interaction captures potential joint effects between these two temporal features on energy consumption.
- **Correlation Analysis (H):** Correlation analysis is performed to measure the linear relationship between the features and the target variable ("Total Usage"). This analysis helps identify which features are most strongly correlated with energy consumption.
- **Feature Combination (I):** Finally, the selected features are combined into the final dataset for modeling. This dataset contains a combination of the original features, temporal features, rolling statistics, lag features, interaction features, and possibly other relevant features identified during the analysis.
- **End (end):** This node represents the end of the feature engineering process.

Photo by Pexels

Model Performance Analysis

Evaluating Time Series Models

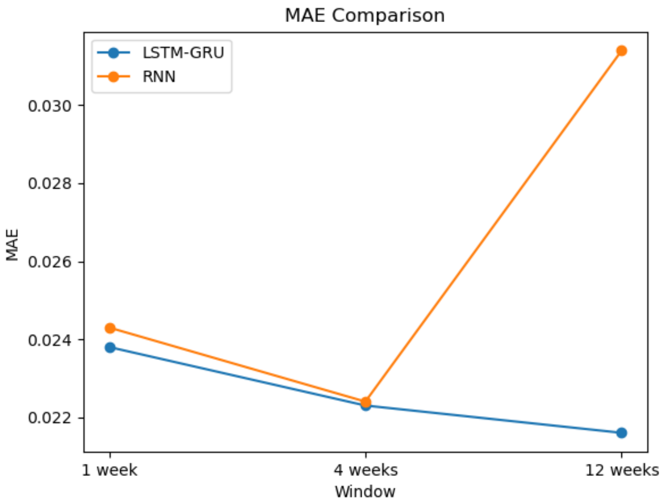

Photo by Pexels

Performance Metrics

Metrics Used

- For evaluating models, we consider several metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R^2 Score.

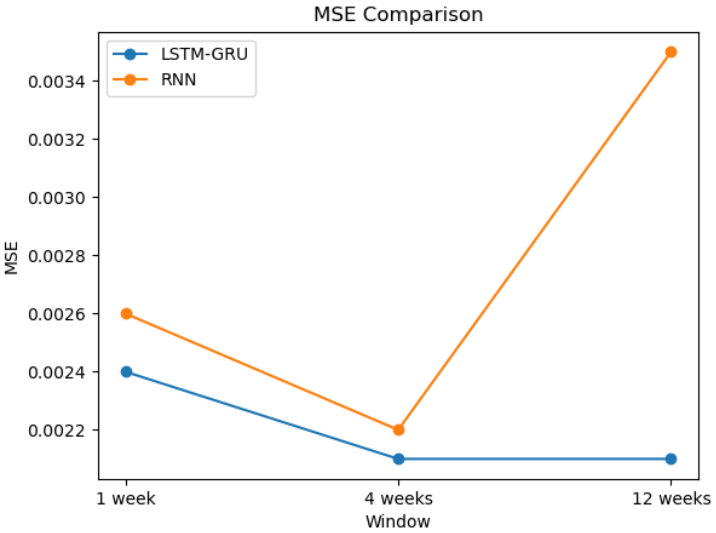
MAE Comparison



Error Magnitude

- The LSTM-GRU model shows consistently lower MAE values compared to the RNN model, suggesting better performance in error magnitude.
- This indicates that the LSTM-GRU model, on average, produces predictions closer to the actual values across various window sizes.
- The reduction in MAE implies that the LSTM-GRU model offers superior performance in minimizing the average prediction error.
- These insights highlight the model's effectiveness in achieving more accurate time series predictions.

MSE Comparison



Robust Performance

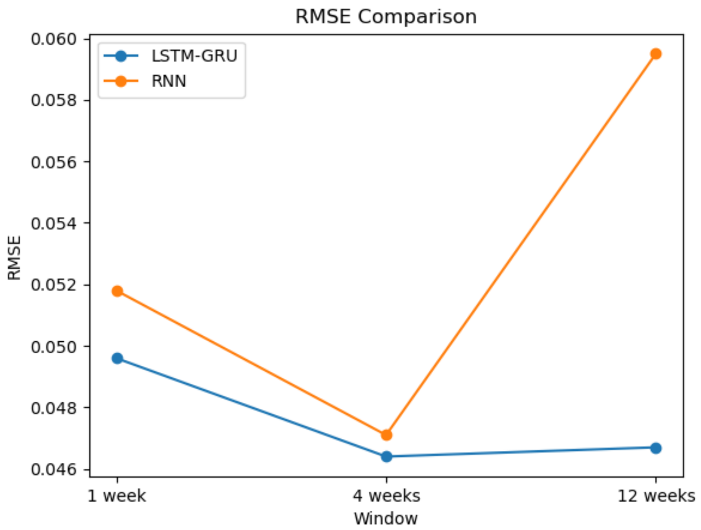
The LSTM-GRU model's lower MSE values indicate it is more robust against large prediction errors than the RNN model.

Lower MSE values imply fewer significant errors, suggesting reliable and consistent model performance across various scenarios.

This metric favors the LSTM-GRU model, demonstrating its ability to handle diverse datasets with minimal large prediction inaccuracies.

- Overall, the LSTM-GRU model's robust performance is evident from its reduced susceptibility to large error rates.

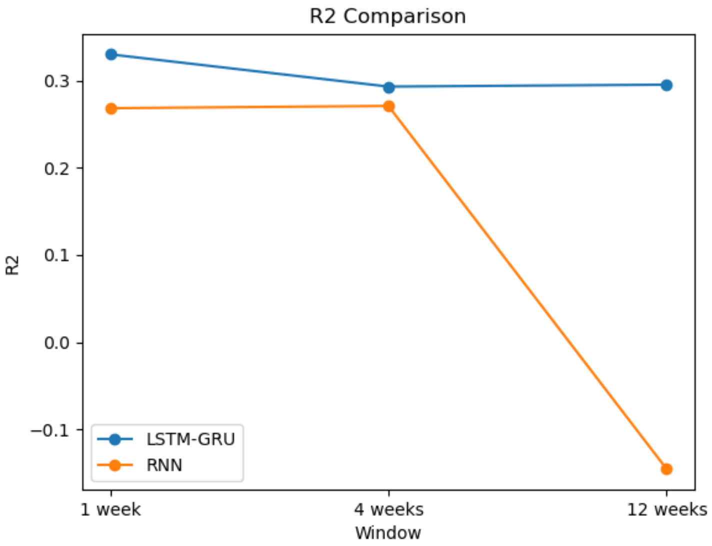
RMSE Comparison



Robust Performance

The Root Mean Squared Error (RMSE) comparison shows that the LSTM-GRU model consistently has lower RMSE values across all window sizes. Lower RMSE values suggest that the LSTM-GRU model provides better accuracy by effectively reducing the magnitude of errors.

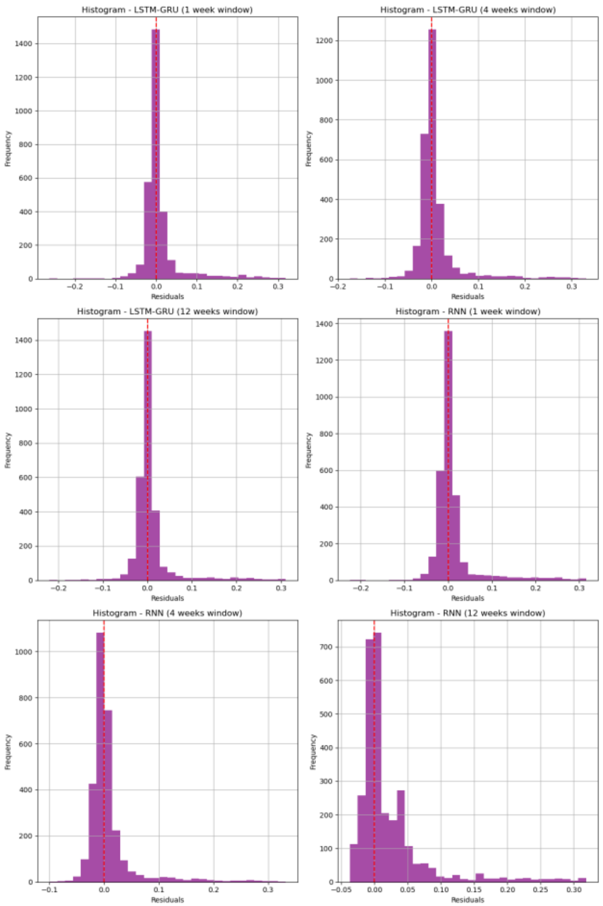
R2 Comparison



Model Fitness to Data

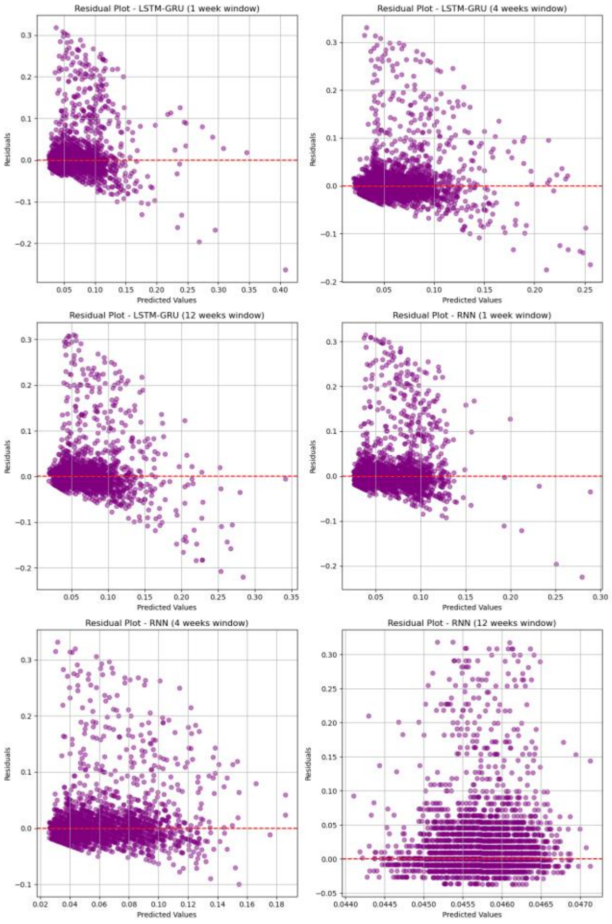
The R-squared (R^2) score comparison highlights that the LSTM-GRU model consistently achieves higher R^2 scores compared to the RNN model. Higher R^2 scores indicate a better fit to the data and a higher proportion of explained variance. This means that the LSTM-GRU model is more capable of capturing the underlying patterns in the data and explaining the variability in the target variable.

Residual Analysis



Model Validation

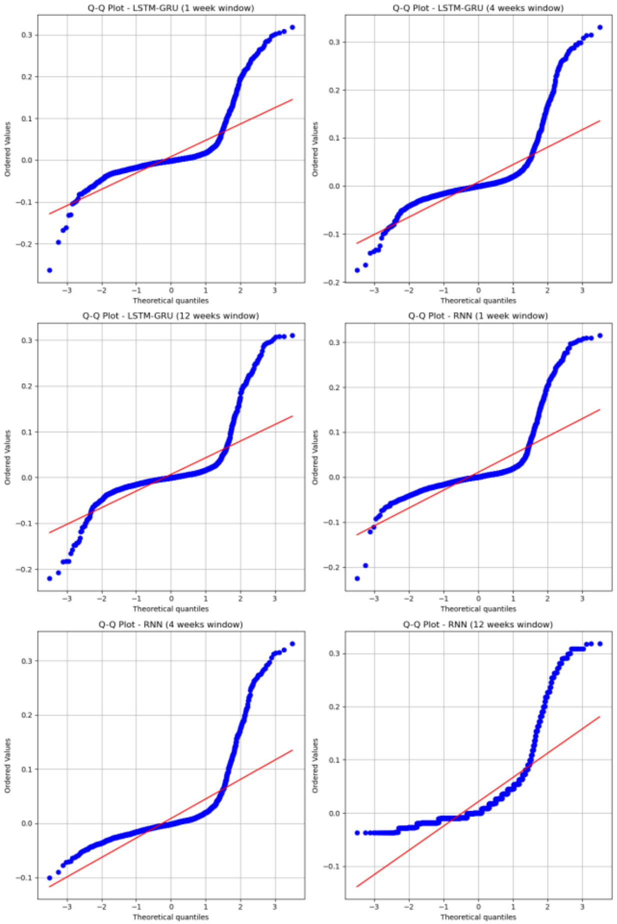
- Residual analysis is essential for validating model accuracy by examining differences between observed and predicted values.
- A histogram of residuals ideally follows a normal distribution, indicating random and unbiased errors from the model.



Residual Analysis

Residual Scatter Plot

- By studying residual scatter plots, we can identify systematic errors or potential areas for model improvement.



Residual Analysis

Q-Q Plots

- Q-Q plots help assess whether residuals conform to normal distribution, validating common assumptions in modeling approaches.
- The Q-Q plot suggest that the residuals are approximately normally distributed, with slight deviations at the tails.

Key Observations

LSTM-GRU Model:

•**Consistent Performance:** Across different window sizes, the LSTM-GRU model consistently outperforms the RNN model in terms of MAE, MSE, RMSE, and R-squared scores.

•**Optimal Parameters:** The best parameters for the LSTM-GRU model remain relatively stable across different window sizes, with a batch size of 32, epochs ranging from 100 to 150, and sigmoid activation function.

•**Improvement with Window Size:** There is a slight improvement in model performance with an increase in the window size, indicating that the LSTM-GRU model benefits from capturing longer-term dependencies in the data.

Key Observations contd.

RNN Model:

•**Decreasing Performance:** The RNN model's performance deteriorates as the window size increases, with higher MAE, MSE, RMSE, and lower R-squared scores observed for the 12-week window compared to the 1-week window.

•**Similar Parameters:** The optimal parameters for the RNN model remain consistent across different window sizes, with a batch size of 32, epochs of 150, and sigmoid activation function.

•**Challenges with Longer Windows:** The RNN model struggles to capture long-term dependencies effectively, leading to poorer performance and negative R-squared values for the 12-week window.

CONCLUSION

- The LSTM-GRU model demonstrates superior performance and stability compared to the RNN model across varying window sizes.
- Longer window sizes benefit the LSTM-GRU model by allowing it to capture more extended temporal patterns effectively.
- Further experimentation and optimization may be required to improve the RNN model's performance, especially for longer-term predictions.

Future Work

Next Steps

- Future studies should investigate advanced models like Transformers and explore hybrid modeling approaches for time series prediction.
- Experimenting with different features or alternative data sources could enhance the model's predictive power and accuracy.
- Developing strategies for real-time prediction can offer practical applications and improve dynamic model evaluation.
- These steps will contribute to further enhancing the effectiveness and reliability of time series models in various contexts.