

Time Series Forecasting of Appliance Energy Usage

Angel Diaz Carral

April 25, 2025

1 Introduction

This project focuses on forecasting energy consumption using multivariate time series data. It demonstrates both deterministic and probabilistic deep learning approaches and highlights the benefits of modeling uncertainty in forecasting tasks.

Deliverables submitted:

- **Report:** This PDF document explaining the dataset, models, methodology, and conclusions.
- **Jupyter Notebooks:** A modular pipeline with three standalone notebooks:
 1. `1_eda.ipynb` Exploratory Data Analysis.
 2. `2_training.ipynb` Training scripts for LSTM-VAE.
 3. `3_evaluation.ipynb` Evaluation and visualization of results.
- **Python Package:** A production-ready package that includes:
 - Trained model (`model.pkl`)
 - Inference and evaluation utilities
- **GitHub Repository:** The full source code, notebooks, Dockerfile, and documentation can be found at:

<https://github.com/adiazcarral/ADC-TestDataScience-2>

The following sections explain the methodology, data preprocessing, model training, and key insights derived from this work.

2 Dataset

The dataset consists of multivariate sensor readings recorded every 10 minutes from a household. It includes 28 input variables (e.g., lights, temperature, humidity) and one target variable: **Appliances**, representing the energy consumed.

3 Exploratory Data Analysis (EDA)

In the EDA notebook (`01_eda.ipynb`), we begin by analyzing the data's key features, understanding the distribution of variables, and identifying any patterns, missing values, or outliers. The focus is on gaining insights into the temporal behavior of the dataset, especially the energy consumption patterns.

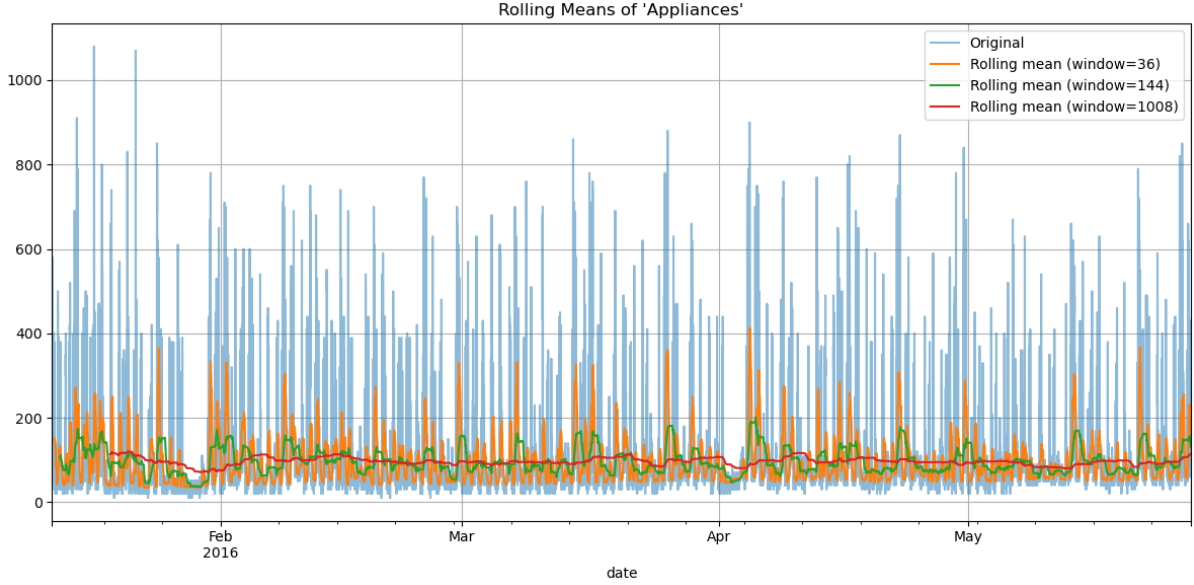


Figure 1: Visualization of target "Appliances"

3.1 Seasonality Analysis

One of the key components of the exploratory analysis is the study of seasonality in the energy consumption data. To detect seasonality, I performed the following tests and analyses:

- **Seasonal Decomposition of Time Series (STL):** The dataset was decomposed using STL decomposition to separate the time series into trend, seasonal, and residual components. This helped identify any repeating patterns in energy usage over different periods, such as daily or seasonal cycles.
- **Autocorrelation Function (ACF):** I plotted the ACF to identify periodic correlations in the data. Peaks in the ACF plot at specific lags indicate seasonality at those intervals (e.g., daily, weekly, yearly).
- **Fourier Transform:** A Fourier Transform was applied to the data to examine the dominant frequencies in the time series. Significant peaks in the frequency domain suggest the presence of seasonal cycles.
- **Heatmaps of Hourly and Daily Energy Consumption:** Heatmaps were created to visualize the variation in energy usage across different hours of the day and days of the week, further helping to confirm any seasonal patterns.

These methods provided a comprehensive understanding of the seasonal variations in energy consumption, allowing for better modeling decisions when handling time series forecasts.

3.2 Data Preprocessing

After studying seasonality, the data was cleaned and preprocessed to address any missing values, scaling issues, and potential outliers. A filter was applied to remove any blurry or unclear observations that could negatively impact model performance.

4 Forecasting Models

We compare two deep learning architectures:

4.1 RNN (Deterministic)

A classic recurrent neural network trained to produce point forecasts. It is simple, efficient, and suitable for baseline performance.

4.2 LSTM-VAE (Probabilistic)

The LSTM-VAE model is an enhanced version of the variational autoencoder (VAE) architecture, adapted for time series forecasting. This model combines an LSTM-based encoder-decoder framework with the probabilistic nature of a VAE to generate probabilistic forecasts.

Architecture: The model operates in a sequence-to-one (seq2one) setup, where a sequence of past observations is encoded into a latent space, which is then decoded to predict the future. The key components are:

- **Encoder:** The LSTM-based encoder processes the input sequence (100 time steps) and compresses the temporal information into a probabilistic latent space, represented by a mean and variance (from a Gaussian distribution).
- **Latent Space:** The latent space encodes the underlying dynamics of the time series, capturing the temporal correlations in the data. By sampling from this latent space, we can model uncertainty in the forecast.
- **Decoder:** The decoder takes the sampled latent variable and generates the predicted future values (100 time steps of the `Appliances` variable).
- **Probabilistic Forecasting:** The model outputs both the predicted mean and uncertainty intervals for each time step, providing a range of possible future scenarios rather than a single point forecast.

Training: The LSTM-VAE model is trained by minimizing the combination of the reconstruction loss (measuring the difference between the predicted and actual future values) and the Kullback-Leibler (KL) divergence (penalizing the difference between the learned latent distribution and the prior). This approach ensures the latent space is structured and the model can generalize well to unseen data.

Advantages:

- Captures temporal dependencies in the data through the LSTM architecture.
- Provides probabilistic forecasts, offering uncertainty estimates for future predictions.
- More robust to noisy or missing data, as it learns to generate plausible future values by sampling from the learned distribution.

5 Training Setup

Input-output windows: Each training sample consists of:

- **Input (X):** A sequence of historical multivariate sensor readings of length `input_window` (e.g., 100), **taken from the past**. To avoid data leakage, we introduce a **forecast horizon** — a gap between the end of the input and the prediction target. For instance, to predict y_{t+1} , the model receives input from $[t - 199, \dots, t - 100]$ — that is, the model learns to forecast values 100 steps ahead using earlier data.
- **Target (y):** The "Appliances" value at time step $t + 1$.

This results in a forecasting task with a fixed **lead time of 100 steps**, mimicking real-world scenarios where decisions must be made in advance without access to future measurements.

We explore two training regimes:

- **Direct forecasting:** Predict all 100 steps in a single forward pass.
- **One-step forecasting:** Predict one step at a time recursively.

6 Validation and Evaluation

We compare the performance of the models in two settings: direct forecasting and one-step autoregressive forecasting. The following figures illustrate the results for each of these two approaches. We split the data into training, validation, and test sets. Evaluation metrics (jupyter notebook 03) include:

- MAE (Mean Absolute Error)
- RMSE (Root Mean Squared Error)
- R^2 score
- CRPS (Continuous Ranked Probability Score) for probabilistic models

6.1 Direct Forecasting

In direct forecasting, the model predicts all 100 steps in one forward pass. This approach evaluates the model's ability to make long-term predictions in a single step.

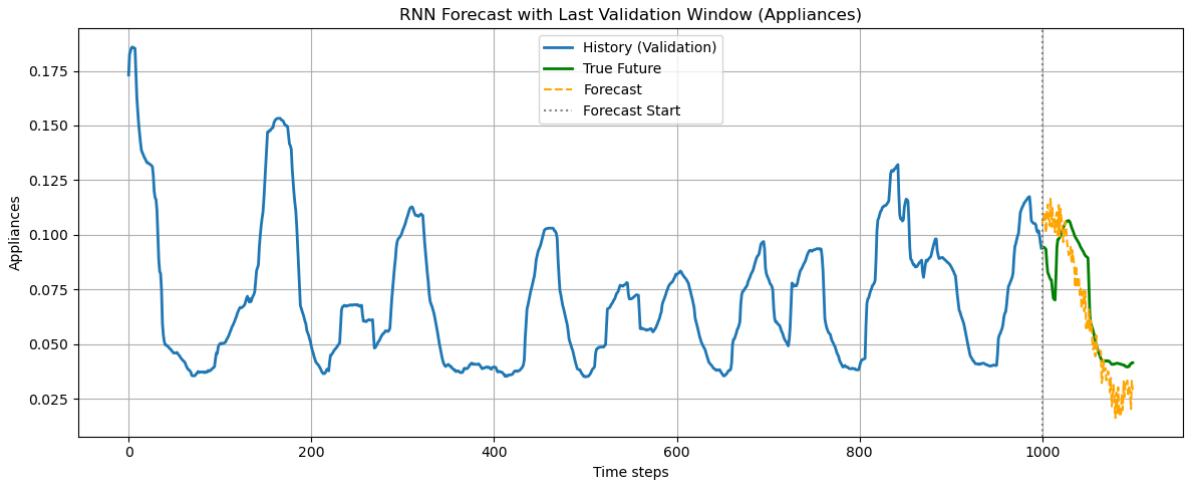


Figure 2: Performance comparison of RNN on the validation set for one-step forecasting.

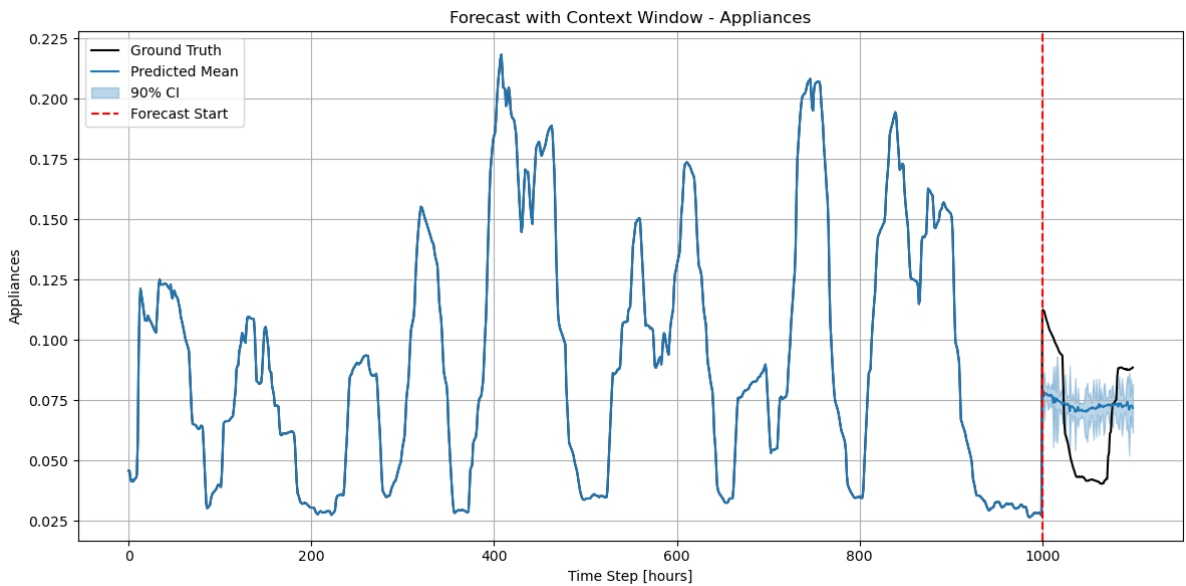


Figure 3: Performance comparison of LSTM-VAE on the validation set for one-step forecasting.

6.2 One-Step Forecasting

In the one-step forecasting approach, the model predicts one step at a time. This method is generally slower but may offer better performance for certain time series tasks.

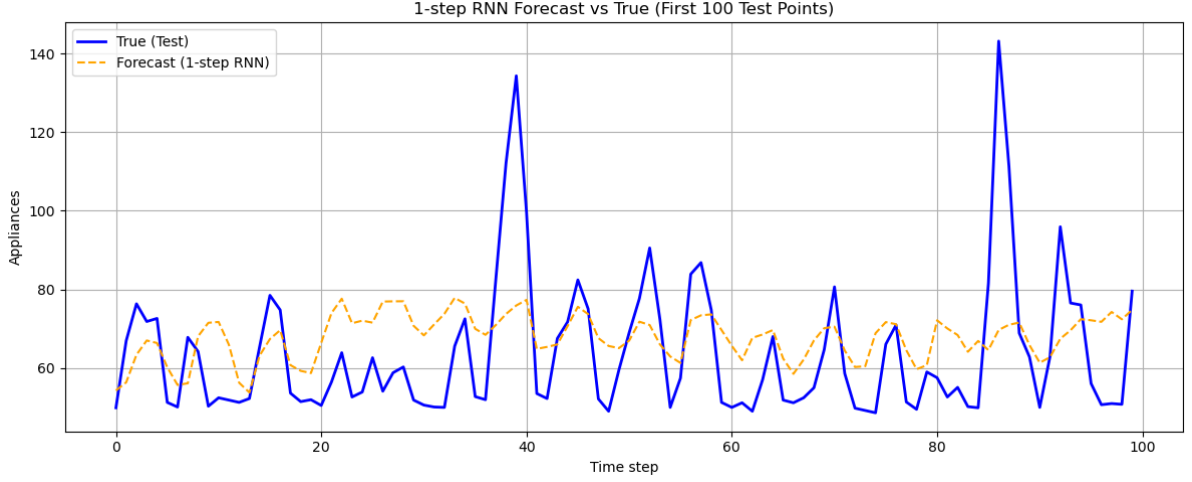


Figure 4: Performance comparison of RNN on the validation set for direct forecasting.

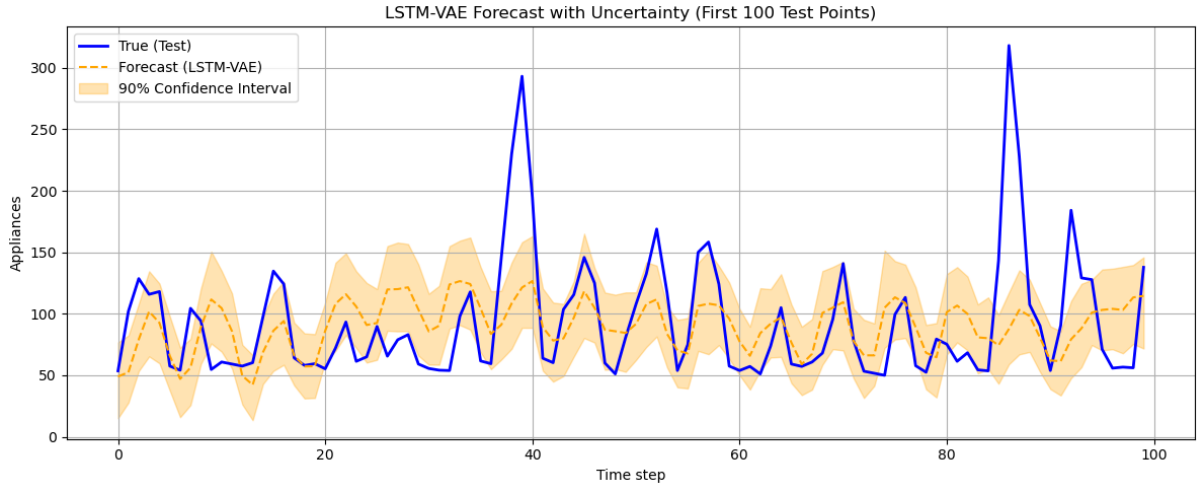


Figure 5: Performance comparison of LSTM-VAE on the validation set for direct forecasting.

The **LSTM-VAE** consistently provides more stable predictions with tighter confidence intervals and competitive RMSE/MAE, making it favorable for deployment.

7 Conclusions

The probabilistic approach using an LSTM-VAE offers significant advantages:

- Captures uncertainty in long-term forecasts.
- Produces more informative predictions (intervals, distributions).
- Robust to noisy or incomplete data.

Adapting an image-focused architecture (VAE) to time series required careful reshaping of data and replacing convolutional layers with LSTM cells. This allowed the latent representation to encode temporal dynamics rather than spatial patterns.

8 Final Remarks

Probabilistic forecasting is essential in energy modeling and many real-world domains where uncertainty matters. This project demonstrates how modern deep generative models can be adapted and applied effectively to sequential problems.