LUT School of Engineering Sciences
Computer Vision and Pattern Recognition
BM20A6100 - Advanced Data Analysis and Machine Learning
Professor – & Teaching Assistant: Lasse Lensu and Akseli Suutari

Forecasting the electric power consumption for a house

Week 3 - Data Pretreatment Baseline Models

Sunday 16th November, 2025
Umme Tanjuma Haque
Chamath Wijerathne
Nada Rahali

Contents

List of Figures

1 Introduction

The project tasks for this week focus on preparing the data set for RNNs sequence modeling. The goal was to prepare a clean, continuous dataset for forecasting, as LSTMs require regularly sampled, synchronized, and normalized data. Additionally, an initial baseline model was developed to serve as a comparative benchmark for future models in this project.

2 Data Pretreatment

The series is strictly sampled at a rate of one minute, and this was further verified by checking the datetime index for any gaps in the timestamp; it contains no gaps. Therefore, it already has a continuous and uniform sampling frequency, so there was no need to resample the series. All variables are synchronous, too, meaning that every feature was measured at the same timestamps.

It contains roughly 25,979 missing measurement entries, about 1.25% of all rows, represented as empty fields between semicolons in the raw text file. Since the timestamps themselves were intact, only the measurement values needed treatment. Linear interpolation was applied across all variables to restore continuity, after which no missing values remained, and no consecutive gaps persisted.

Because LSTM models can be sensitive to extreme values, STL decomposition was applied to the daily-averaged target series to find abnormal spikes (see Figure 1). It detected outliers using a three-standard-deviation threshold on the residual component; these were replaced with NaNs and afterward interpolated. This will remove sharp anomalies and keep the underlying trend and seasonal structure for later stages of modeling.

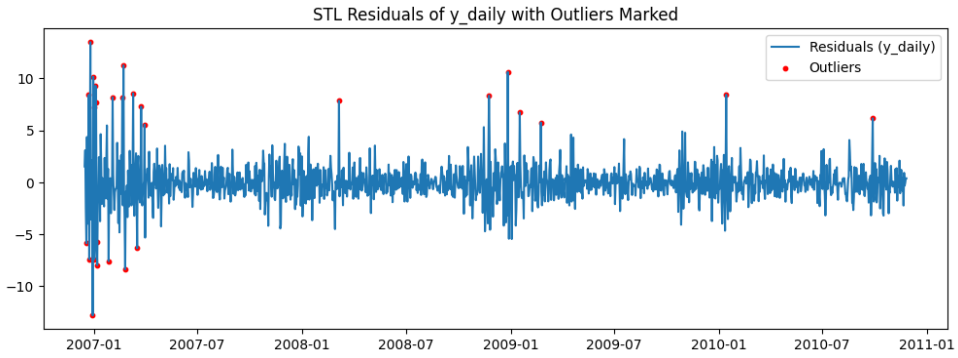

Figure 1: Detected Outliers in STL Residuals of the data

3 Mini Literature Review

Dividing long time series into smaller subsequences is considered a preprocessing step to be carried out on the inputs that will be fed into the RNN. Two usual methods for subsequence creation are described by Zhan and Kim [3]: the sliding window and tumbling window. Sliding window methods generate fixed-length overlapping sequences where a window shifts one time step at a time, meaning each P_1… P_n will predict P_n+1 and then it shifts forward by one step. This generates several training examples but is computationally heavier. Another approach is tumbling windows: they divide the series into blocks, and only when one full block of data is processed does it advance to the next. This approach reduces computational cost by not having overlap, but provides fewer training instances.

So far, seasonality and long-term trends are important factors in how subsequences need to be constructed. Wu et al. (2024) [2] remark that basic LSTMs lack the inductive bias necessary for capturing strict seasonal cycles or slowly evolving trends. In their words, the authors state that LSTMs "lack the necessary inductive bias to effectively capture seasonality and trends," which results in poor modeling of long-range context if these components dominate the signal. The proposed hybrid model in this work overcomes this limitation by decomposing the original time series into seasonal, trend, and remainder components using an STL-based procedure. Each of these components then passes through a different LSTM. By exposing the underlying structures explicitly through preprocessing, the model successfully learns seasonal patterns and trends that the basic LSTM fails to capture.

Another important consideration for LSTM-based forecasting is normalization. Passalis et al. (2019) [1] note that deep learning models require properly normalized inputs for stable training, and state that the most usual approach is still performed by z-score normalization, which subtracts the mean and then divides by the standard deviation. However, they stress that the use of fixed z-score statistics can be problematic for non-stationary time series since the statistics do not adapt once the distribution of data shifts. They discuss a number of alternative normalization strategies, such as min-max scaling, mean normalization, unit-length scaling, batch normalization, and instance normalization. The study further proves that models trained without any normalization perform extremely poorly, showing just how sensitive LSTMs are to the scale and distribution of their inputs.

4 Baseline Model

A simple autoregressive baseline was implemented to anchor the performance of the later models of recurrent neural networks. This baseline autoregressive model was implemented by using the previous seven days (T = 7) to predict the following day. This simple AR-style structure captures short-term linear dependencies and provides a transparent benchmark for comparison. This baseline captures low-order linear short-term relationships and provides a clear, transparent baseline with which to compare more complex models. As expected for a linear model, it provides adequate performance in stable or low-variation regions but struggles with sudden fluctuations, nonlinear dynamics, and seasonality irregularities. Providing a baseline in this regard is important, nonetheless, as it forms the minimum set of performance improvements expected from the LSTM models in later stages.

The training and validation loss curves (Figure 2) for the baseline autoregressive model indicate a smooth convergence: after the early epochs, both losses have more or less stabilized. There is a reasonable agreement between the predicted and actual values as seen in Figure 3, in that the model captures the short-term linear relationships fairly well, though there seem to be more mismatches at sharper spikes, likely because of abrupt fluctuations and influences from more complex seasonal effects.

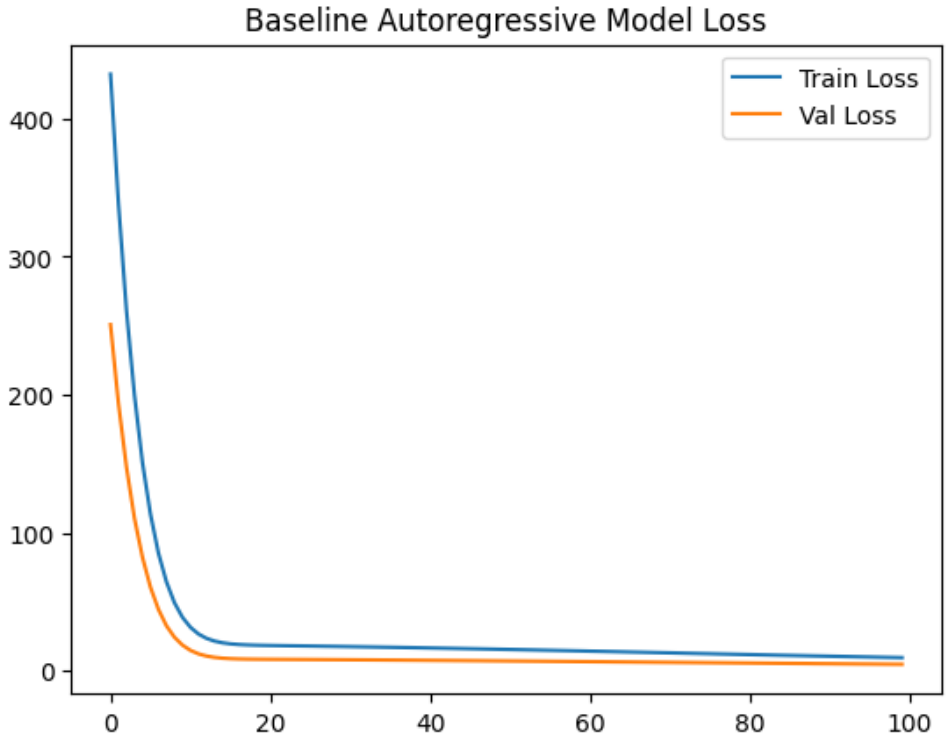

Figure 2: Baseline Autoregressive Model Loss

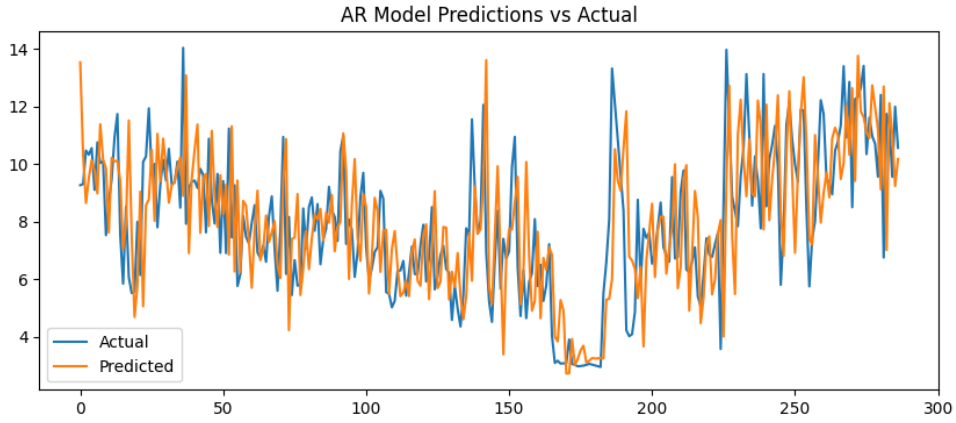

Figure 3: Autoregressive Model Predictions vs Actual Values

References

[1] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis. Deep adaptive input normalization for time series forecasting. https://arxiv.org/abs/1902.07892, 2019. [Online; accessed November 16, 2025].

[2] Y. Wu, X. Meng, J. Zhang, Y. He, J. Romo, Y. Dong, and D. Lu. Effective lstms with seasonal-trend decomposition and adaptive learning. https://www.sciencedirect.com/science/article/abs/pii/S0957417423017049, 2024. [Online; accessed November 16, 2025].

[3] Z. Zhan and S.-K. Kim. Versatile time-window sliding machine learning techniques for stock market forecasting. https://link.springer.com/article/10.1007/s10462-024-10851-x, 2024. [Online; accessed November 16, 2025].