Predicting Household Energy Consumption Using  
Time Series Forecasting

ADTA 5560 Section 001 - Recurrent Neural Networks for Sequence Data

Team Name: Dynamic Neurons

|  |  |
| --- | --- |
| Sathvik Chava | 11752889 |
| Sumanjali Banjara | 11821870 |
| David Adeyemi | 11828970 |
| Andrew Gunason Paul | 11835138 |

Team Members:

# Abstract

The project focuses on predicting short-term energy consumption of household appliances using deep learning techniques on time series data. The dataset used contains real energy usage data collected at 10-minute intervals making it suitable for sequential modeling. Temporal features like hour of the day, day of the week and sinusoidal encodings are added to capture daily and weekly patterns in the usage of energy. The models that are build and tested are – Vanilla RNN, Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU), Bi-directional RNN and encoder-decoder.

Each model was tested and assessed using Mean Square Error (MSE) and Mean Absolute Error (MAE) by training them multiple times. Among these, the Encoder-Decoder model achieved the lowest average MSE, while the Vanilla RNN model demonstrated the best MAE performance.

The results suggest that AI -based forecasting can provide reliable, real- time insights to improve energy efficiency, reduce utility costs, and support high demand energy management. These insights help the homeowners, utility providers and smart grid developers to adopt predictive automation for sustainable energy solutions.

Table of Contents

[Abstract 2](#_Toc197117443)

[Chapter 1: Introduction and Business Understanding 6](#_Toc197117444)

[1.1 Background and Context 6](#_Toc197117445)

[1.2 Project Motivation 6](#_Toc197117446)

[1.3 Problem Statement 6](#_Toc197117447)

[1.4 Project Objective 7](#_Toc197117448)

[1.5 Scope of Work 7](#_Toc197117449)

[1.6 Significance and Implications 7](#_Toc197117450)

[Chapter 2: Data Understanding 8](#_Toc197117451)

[2.1 Dataset Overview 8](#_Toc197117452)

[2.2 Variable Description 8](#_Toc197117453)

[2.3 Target Variable 9](#_Toc197117454)

[2.4 Data Characteristics 9](#_Toc197117455)

[2.5 Key Considerations for Modeling 10](#_Toc197117456)

[2.6 Preliminary Summary Statistics 10](#_Toc197117457)

[2.7: Exploratory Data Analysis 11](#_Toc197117458)

[2.7.1 Distribution of Appliance Energy Consumption 11](#_Toc197117459)

[2.7.2 Energy Consumption Patterns Over Time 11](#_Toc197117460)

[2.7.3 Indoor Temperature Variations Over Time 12](#_Toc197117461)

[2.7.4 Indoor Temperatures and Energy Consumption Relationship 12](#_Toc197117462)

[2.7.5 Indoor Humidity and Energy Consumption Relationship 13](#_Toc197117463)

[Chapter 3: Data Preparation 14](#_Toc197117464)

[3.1 Overview 14](#_Toc197117465)

[3.2 Parsing Timestamps 14](#_Toc197117466)

[3.3 Time Feature Engineering 14](#_Toc197117467)

[3.4 Feature Selection 14](#_Toc197117468)

[3.5 Handling Missing Values 15](#_Toc197117469)

[3.6 Feature Scaling 16](#_Toc197117470)

[3.7 Train-Validation-Test Split 16](#_Toc197117471)

[3.8 Sequence Generation for RNN Input 16](#_Toc197117472)

[3.9 Summary of Preprocessing Steps 16](#_Toc197117473)

[Chapter 4: Model Development 18](#_Toc197117474)

[4.1 Overview 18](#_Toc197117475)

[4.2 Rationale for Using Recurrent Neural Networks 18](#_Toc197117476)

[4.3 Model 1: Baseline Simple RNN 18](#_Toc197117477)

[4.3.1 Architecture 18](#_Toc197117478)

[4.3.2 Compilation Settings 18](#_Toc197117479)

[4.3.3 Training Strategy 18](#_Toc197117480)

[4.3.4 Observations 19](#_Toc197117481)

[4.3.5 Training and Validation Performance 19](#_Toc197117482)

[4.3.6 Prediction Results on Test Set 19](#_Toc197117483)

[4.3.7 Evaluation Metrics 20](#_Toc197117484)

[4.4 Model 2: Bi-directional Recurrent Neural Network (Bi-RNN) 20](#_Toc197117485)

[4.4.1 Architecture 20](#_Toc197117486)

[4.4.2 Compilation Settings 20](#_Toc197117487)

[4.4.3 Training Strategy 21](#_Toc197117488)

[4.4.4 Training and Validation Performance 21](#_Toc197117489)

[4.4.5 Prediction Results on Test Set 21](#_Toc197117490)

[4.4.6 Evaluation Metrics 22](#_Toc197117491)

[4.5 Model 3: Long Short-Term Memory (LSTM) 22](#_Toc197117492)

[4.5.1 Architecture 22](#_Toc197117493)

[4.5.2 Compilation Settings 22](#_Toc197117494)

[4.5.3 Training Strategy 23](#_Toc197117495)

[4.5.4 Training and Validation Performance 23](#_Toc197117496)

[4.5.5 Prediction Results on Test Set 23](#_Toc197117497)

[4.4.6 Evaluation Metrics 24](#_Toc197117498)

[4.6 Model 4: Gated Recurrent Unit (GRU) 24](#_Toc197117499)

[4.6.1 Architecture 24](#_Toc197117500)

[4.6.2 Compilation Settings 24](#_Toc197117501)

[4.6.3 Training Strategy 25](#_Toc197117502)

[4.6.4 Training and Validation Performance 25](#_Toc197117503)

[4.6.5 Prediction Results on Test Set 25](#_Toc197117504)

[4.6.6 Evaluation Metrics 26](#_Toc197117505)

[4.7 Model 5: Encoder-Decoder 26](#_Toc197117506)

[4.7.1 Architecture 26](#_Toc197117507)

[4.7.2 Compilation Settings 26](#_Toc197117508)

[4.7.3 Training Strategy 27](#_Toc197117509)

[4.7.4 Training and Validation Performance 27](#_Toc197117510)

[4.7.5 Prediction Results on Test Set 27](#_Toc197117511)

[4.7.6 Evaluation Metrics 28](#_Toc197117512)

[Chapter 5: Results and Model Evaluation 29](#_Toc197117513)

[5.1 Overview 29](#_Toc197117514)

[5.2 Performance Metrics 29](#_Toc197117515)

[5.3 Visual Comparison of Model Predictions 29](#_Toc197117516)

[5.4 Training Behavior and Convergence 30](#_Toc197117517)

[5.5 Model Selection Recommendation 30](#_Toc197117518)

[5.6 Limitations and Future Work 31](#_Toc197117519)

[Chapter 6: Conclusion and Final Recommendations 32](#_Toc197117520)

[6.1 Summary of Work 32](#_Toc197117521)

[6.2 Key Findings 32](#_Toc197117522)

[6.3 Final Recommendations 32](#_Toc197117523)

[6.4 Limitations 32](#_Toc197117524)

[6.5 Future Work 33](#_Toc197117525)

[Chapter 7: Bibliography 34](#_Toc197117526)

# Chapter 1: Introduction and Business Understanding

## Background and Context

As the world is rapidly shifting towards smart and sustainable lining, the necessity to monitor and predict household energy consumption has become essential. The usage of electricity in homes can change significantly due to multiple reasons like internal temperature, outdoor climate conditions, occupancy patterns and appliance usage. This uncertainty becomes a challenge to both homeowners and utility providers who wish to manage energy effectively.

There has been a rise in smart meters and IoT -enabled environments where large volumes of energy consumption data are now being recorded at frequent time intervals. This gives an opportunity for data- driven solutions, especially time series forecasting models, that provide insights for future usage. Advanced machine learning like recurrent neural network (RNNs), Long Short-Term Memory (LSTMs) and Gated Recurrent Units(GRUs) are well suited for time-series data and can effectively model these usage patterns.

This project draws from the work of Candanedo and Feldheim (2016), which shows how environmental factors like indoor temperature, humidity and outdoor weather conditions are useful for predicting household energy consumption. Their findings emphasized the value of data-driven predictive models for energy efficiency initiatives in residential settings.

## 1.2 Project Motivation

Accurate predictions of short-term energy can have several benefits:

* For homeowners: Helps owners reduce bills of electricity, allows them better plan and manage the appliance usage, especially during high tariffs hours.
* For utility providers: This project can help them to find balance between demand and supply, better load balance, and helps to plan integrating other energy sources.
* For environmental sustainability: It encourages a smarter energy pattern to cut down the carbon footprint.

Predicting energy usage is complex due to the changing weather conditions and varying indoor conditions and human behavior. In account of these challenges, we call for models that can adapt over time and learn from past data.

## 1.3 Problem Statement

The project focuses on using RNN-based models to predict hourly energy consumption in the households. The aim is to use both historical energy use data and environmental conditions both indoors and outdoors- to train models that can produce reliable short-term forecasts. These predictions can help the stakeholders to make better energy management decisions.

## 1.4 Project Objective

The primary objective of this project is to:

* Build forecasting time series models based on different RNN architectures including RNNs, LSTMs, and GRUs.
* Use the models then to predict the hourly appliance energy consumption for a household.
* Compare the performance of different RNN architectures using suitable metrics.
* Derive actionable insights based on forecast results to support energy optimization strategies.

## 1.5 Scope of Work

The project will encompass:

* Exploratory data analysis and preprocessing
* Time series data resampling and transformation
* Feature selection and scaling
* Building baseline and advanced RNN-based models (Simple RNN, LSTM, GRU)
* Model evaluation using appropriate regression metrics (e.g., RMSE, MAE)
* Visualization and interpretation of results
* Discussion of practical implications and future research directions

The forecasting focus will be on short-term (hour-ahead) energy usage predictions based on past hourly trends and environmental variables.

## 1.6 Significance and Implications

By accurately forecasting household energy consumption, the project offers:

* Help homeowners save money by managing energy use more effectively.
* Gives the energy providers with better planning and grid optimization.\
* It promotes energy sustainability and responsible usage in communities.
* A similar approach can also be applied to broader smart grid or small city energy initiatives.

# Chapter 2: Data Understanding

## 2.1 Dataset Overview

For this project, we have used the Appliances Energy Prediction Dataset sourced from the UCI Machine Learning Repository. The dataset contains measurements of energy consumption in a single residential household over a period of approximately four and a half months. Data were recorded at 10-minute intervals, resulting in about 19,735 observations across 29 variables.

The dataset was originally collected and analyzed in the study by Candanedo and Feldheim (2016), where data-driven models were built to predict household energy consumption based on environmental parameters. Their work serves as a foundational basis for this project, validating the relevance of the features captured in the dataset.

## 2.2 Variable Description

| Feature | Description |
| --- | --- |
| date | Timestamp of the measurement (in 10-minute intervals) |
| Appliances | Energy use of appliances (Wh) |
| lights | Energy consumption from lights (Wh) |
| T1 | Temperature in kitchen area (°C) |
| RH\_1 | Humidity in kitchen area (%) |
| T2 | Temperature in living room area (°C) |
| RH\_2 | Humidity in living room area (%) |
| T3 | Temperature in laundry room area (°C) |
| RH\_3 | Humidity in laundry room area (%) |
| T4 | Temperature in office room (°C) |
| RH\_4 | Humidity in office room (%) |
| T5 | Temperature in bathroom (°C) |
| RH\_5 | Humidity in bathroom (%) |
| T6 | Temperature outside the building (°C) |
| RH\_6 | Humidity outside the building (%) |
| T7 | Temperature in ironing room (°C) |
| RH\_7 | Humidity in ironing room (%) |
| T8 | Temperature in teenager’s room (°C) |
| RH\_8 | Humidity in teenager’s room (%) |
| T9 | Temperature in parents’ room (°C) |
| RH\_9 | Humidity in parents’ room (%) |
| To | Outdoor temperature from weather station (°C) |
| Pressure | Atmospheric pressure (millibars) |
| RH\_out | Outdoor humidity (%) |
| Windspeed | Wind speed (m/s) |
| Visibility | Visibility distance (km) |
| Tdewpoint | Dew point temperature (°C) |
| rv1, rv2 | Random variables (noise, no significant meaning) |

## 2.3 Target Variable

The primary target for prediction is:

* Appliances (Energy consumed in Wh at each time step)

The goal is to forecast appliance energy consumption using historical time series features.

## 2.4 Data Characteristics

* Time Granularity: 10-minute intervals (to be resampled to hourly for model simplicity)
* Data Type: Continuous numeric variables
* Missing Values:  
  Preliminary inspection reveals no missing values; a detailed check will follow during data preparation.
* Outliers:  
  Potential outliers in features like energy consumption and weather conditions will be explored during EDA.

## 2.5 Key Considerations for Modeling

* Timestamp Handling:  
  The date column is parsed into datetime format and used to generate additional informative features, but the raw timestamp itself is set as the index.
* Feature Engineering:
  + Extracted Features:
    - hour (of the day)
    - dayofweek (Monday–Sunday)
    - month (January–December)
    - is\_weekend (weekend indicator)
  + Sinusoidal encoding for hour and dayofweek to capture their cyclic nature.
* Scaling:  
  Neural network-based models require feature scaling to ensure faster convergence and better model performance.

## 2.6 Preliminary Summary Statistics

| Feature | Mean | Std. Dev | Min | Max |
| --- | --- | --- | --- | --- |
| Appliances | 97.69 Wh | 108.58 Wh | 10 Wh | 1080 Wh |
| Lights | 3.71 Wh | 17.22 Wh | 0 Wh | 70 Wh |
| T1 (Kitchen Temp) | 19.67°C | 1.99°C | 16.79°C | 24.50°C |
| RH\_1 (Kitchen Humidity) | 47.60% | 4.85% | 27.23% | 63.39% |
| To (Outside Temp) | 6.48°C | 6.02°C | -5.78°C | 26.26°C |
| Windspeed | 3.31 m/s | 1.41 m/s | 0 m/s | 7.23 m/s |
| Visibility | 40.49 km | 16.70 km | 5.00 km | 66.00 km |

## 2.7: Exploratory Data Analysis

### 2.7.1 Distribution of Appliance Energy Consumption

The distribution of energy consumption shows that it is right-skewed. While most of the values are below 200 Wh, there are some occasional spikes above 1000 Wh. These spikes can be interpreted as surges due to specific activities like cooking and laundry.

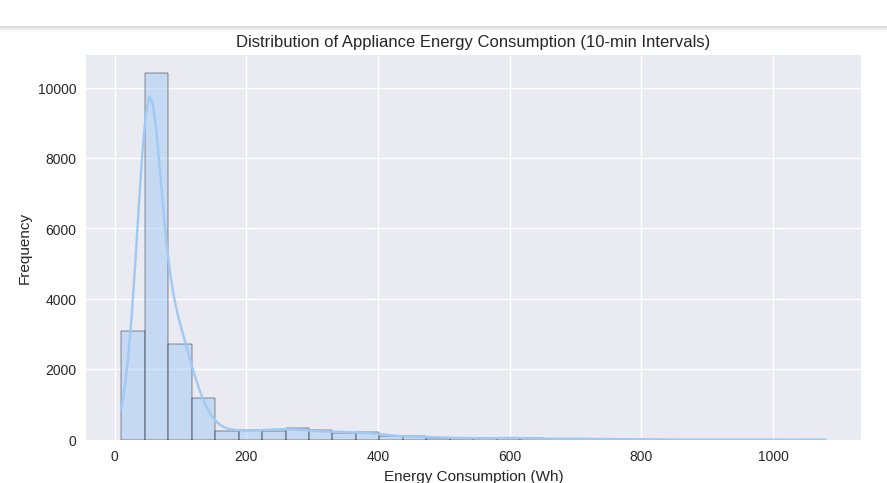


Figure 1 Distribution of Appliance Energy Consumption (10-minute intervals)

### 2.7.2 Energy Consumption Patterns Over Time

The time series line plot of appliance energy consumption shown in figure 2 suggests that there is a strong relation to the time-of-day. The daily and weekly cycles can be observed with the spikes.

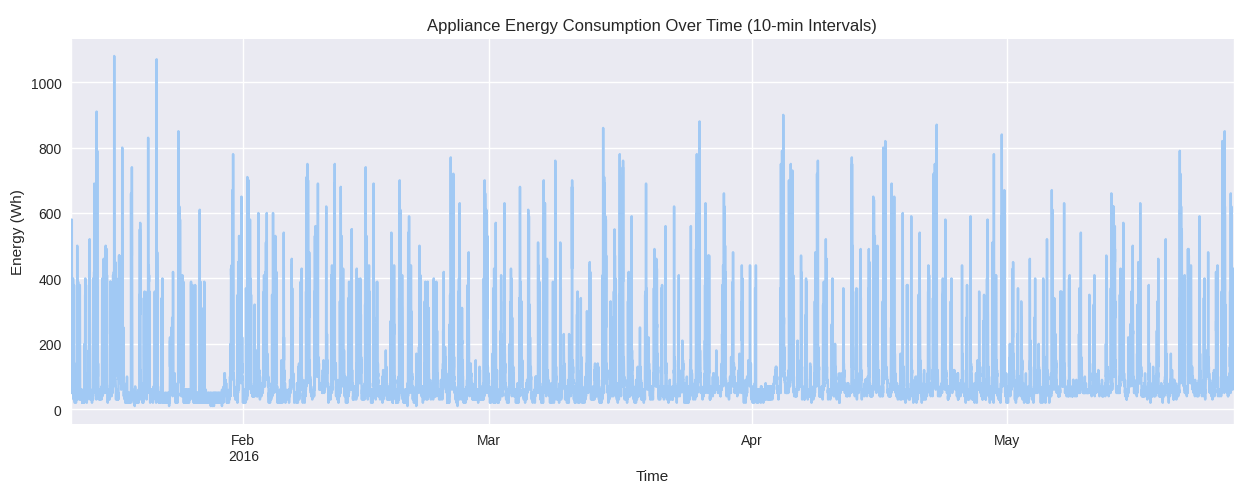


Figure 2 Appliance Energy Consumption Over Time (10-minute intervals)

### 2.7.3 Indoor Temperature Variations Over Time

Figure 3 shows the temperature variations across different rooms in the house. A seasonal pattern can be observed. The temperatures are low in January and February, the temperature rises in the month of May.

A graph showing different colored lines

AI-generated content may be incorrect.

Figure 3 Indoor Temperature Variations Over Time

### 2.7.4 Indoor Temperatures and Energy Consumption Relationship

Figure 4 shows the indoor temperatures and appliance energy usage in a single day

A graph showing different colored lines

AI-generated content may be incorrect.

Figure 4 Indoor Temperatures and Appliance Energy Consumption Over Time

### 2.7.5 Indoor Humidity and Energy Consumption Relationship

Figure 5 compares indoor humidity and appliance energy usage. The humidity spikes align with the energy usage spikes, there can also be possible human activities like cooking or bathing which result in an increase in humidity as well as energy consumption.

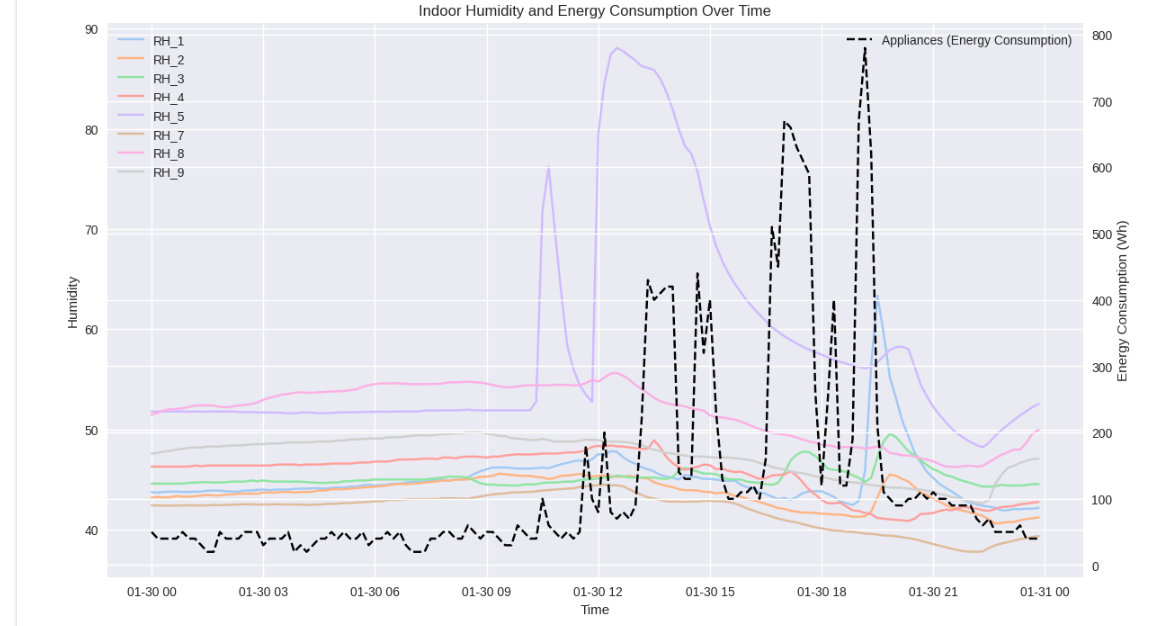


Figure 5 Indoor Humidity and Appliance Energy Consumption Over Time

# Chapter 3: Data Preparation

## 3.1 Overview

While working with time series data, data preparation is an essential step for accurate predictive modeling. In this project, several preprocessing techniques are applied, so that the input data is suitable for model training. This chapter focuses on outlying the preprocessing steps.

## 3.2 Parsing Timestamps

The raw dataset contains a date column indicating the timestamp of each observation. The first preprocessing step involves:

* Converting the date column into a datetime object.
* Setting the date column as the index of the dataset for efficient time-based operations.
* Chronological order was enforced without random shuffling to preserve time series integrity.

## 3.3 Time Feature Engineering

From the timestamp, additional features were engineered:

* hour (0–23)
* dayofweek (0–6, Monday=0)
* month (1–12)
* is\_weekend (binary, 1 for Saturday/Sunday)
* Sinusoidal encoding for hour and dayofweek to represent cyclic behavior:
  + hour\_sin
  + hour\_cos
  + dayofweek\_sin
  + dayofweek\_cos

These engineered features allow the neural network to learn daily and weekly patterns more effectively.

## 3.4 Feature Selection

Not all features in the dataset contribute equally to predicting household energy consumption. Based on prior research (e.g., Candanedo and Feldheim, 2016) and domain knowledge, the following features are selected for modeling:

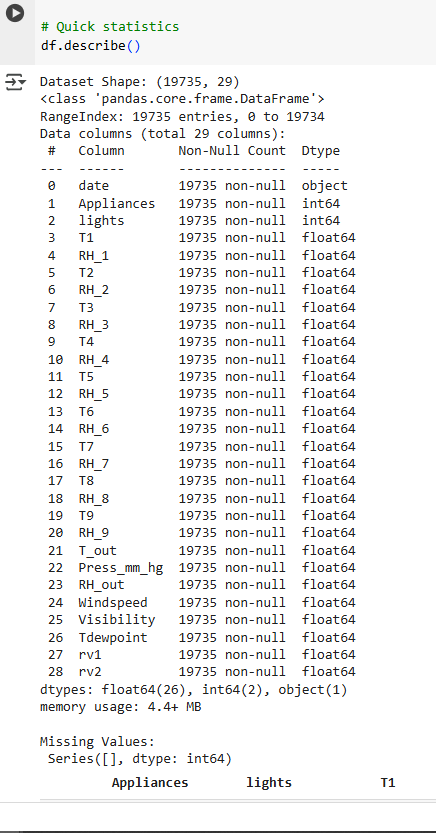
The following sets of features were retained:

* Indoor conditions: room temperatures and humidities (T1–T9, RH\_1–RH\_9)
* Outdoor weather data (To, Pressure, RH\_out, Windspeed, Visibility, Tdewpoint)
* Time-based features (hour, dayofweek, month, weekend indicators with sinusoidal encodings)

The lights feature is excluded from modeling due to its relatively low variation and contribution compared to total appliance energy usage.  
Additionally, the random variables rv1 and rv2 are excluded, as they have no clear physical meaning and are likely noise.

## 3.5 Handling Missing Values

Upon inspection, the dataset contains no missing values in any columns. Therefore, no imputation or deletion was necessary at this stage.



## 3.6 Feature Scaling

Neural network models, particularly RNN-based models, are sensitive to the scale of input data. To standardize the range of input features and improve training convergence:

* Min-Max Scaling is applied to all input features.
* Scaling is fit on the training set and then applied to the validation and test sets to prevent data leakage

## 3.7 Train-Validation-Test Split

To maintain the temporal integrity of the time series:

* The data is split sequentially into:
  + Training set (70%)
  + Validation set (15%)
  + Test set (15%)

There is no random shuffling, as future observations should not influence past predictions.

## 3.8 Sequence Generation for RNN Input

RNNs require input data formatted as sequences. For this project:

* A sliding window approach is used to generate input-output pairs:
  + Input (X): a sequence of past time steps (e.g., past 10 minutes).
  + Target (y): the appliance energy consumption at the next time step.

Example: Using the past 18-time steps (equivalent to 3 hours) to predict the next 10-minute energy consumption.

## 3.9 Summary of Preprocessing Steps

1. Parsed and indexed timestamps by converting the date column to datetime objects and setting it as the DataFrame index, ensuring chronological order was preserved.
2. Engineered time-based features such as hour, dayofweek, month, is\_weekend, and their sinusoidal encodings (hour\_sin, hour\_cos, dayofweek\_sin, dayofweek\_cos) to capture daily and weekly seasonality.
3. Selected relevant features including indoor temperatures and humidities (T1–T9, RH\_1–RH\_9), outdoor environmental conditions (To, Pressure, RH\_out, Windspeed, Visibility, Tdewpoint), and time-related features, while excluding 'lights', 'rv1', and 'rv2' as irrelevant.
4. Confirmed no missing values were present, thus no imputation or deletion was needed.
5. Applied Min-Max scaling to all input features, fitting only on the training set to avoid data leakage.
6. Chronologically split the data into training (70%), validation (15%), and testing (15%) sets without random shuffling to maintain temporal integrity.
7. Generated sequences using a sliding window approach, with past sequences (e.g., past 3 hours) used to predict the next time step's appliance energy consumption.

# Chapter 4: Model Development

## 4.1 Overview

This chapter outlines the development of Recurrent Neural Network (RNN)-based models for forecasting short-term household energy consumption. Three deep learning models were constructed and compared:

* A Baseline Simple RNN model,
* An enhanced Long Short-Term Memory (LSTM) model,
* A streamlined Gated Recurrent Unit (GRU) model.

Each model was trained to predict the appliance energy consumption at 10-minute intervals, based on historical indoor environmental conditions, outdoor weather, and engineered time features.

## 4.2 Rationale for Using Recurrent Neural Networks

The Recurrent Neural Networks are a good natural choice, for the task of time series forecasting. They can capture the sequential dependencies well, using the hidden state. The advantage with the RNN is the ability to remember the past steps, this is very helpful for the energy consumption pattern with daily and weekly cycles. The disadvantage with RNN is that they suffer from vanishing gradients problem on long sequencies.

## 4.3 Model 1: Baseline Simple RNN

### 4.3.1 Architecture

* Input Layer: 18 steps, each with 20 features
* RNN Layer: Simple RNN with 64 units and tanh activation
* Output Layer: Dense layer with a single neuron for regression output (Appliances energy consumption)

### 4.3.2 Compilation Settings

* Loss Function: Mean Squared Error (MSE)
* Optimizer: Adam
* Metrics: Mean Absolute Error (MAE)

### 4.3.3 Training Strategy

* Early stopping was implemented based on validation loss to prevent overfitting.
* Model was trained for a maximum of 50 epochs with a batch size of 32.

### 4.3.4 Observations

The baseline RNN model successfully captured general energy consumption patterns but showed limitations in accurately predicting sharp peaks in energy usage. This behavior is expected given the simple structure of a vanilla RNN without advanced memory mechanisms.

### 4.3.5 Training and Validation Performance

The training process for the baseline Simple RNN model is illustrated in Figure 6.  
The model exhibited good convergence behavior, with the validation loss consistently lower than the training loss after the first few epochs, indicating no serious overfitting.

A graph with a line graph

AI-generated content may be incorrect.

Figure 6 Training vs Validation Loss Curve for Simple RNN

### 4.3.6 Prediction Results on Test Set

The prediction performance on the test dataset is visualized in Figure 7.  
While the model was able to follow the general trend of appliance energy consumption, it struggled to capture sharp spikes and sudden increases in energy demand, which is expected given the limitations of standard RNN architectures.

A graph of energy consumption

AI-generated content may be incorrect.

Figure 7 Appliance Energy Consumption - Actual vs Predicted (Simple RNN Model)

### 4.3.7 Evaluation Metrics

The performance of the baseline Simple RNN model on the test which is ran 5 times and average MSE and average MAE set were summarized as follows:

* Average MSE: 0.007563
* Average MAE: 0.042867

These results provide a solid starting point for comparison against more advanced architectures (LSTM and GRU), which are expected to better handle sequential patterns and energy consumption volatility.

## 4.4 Model 2: Bi-directional Recurrent Neural Network (Bi-RNN)

### 4.4.1 Architecture

The second model in our code is a Bi-directional RNN, this model processes the input from both directions (forward and backward). It is able to capture the past and the future context.

Details of Architecture:

* Input: 36-time steps x 20 features
* Bidirectional RNN Layer: 64 units with tanh activation
* Output Layer: Dense layer (1 unit for regression)

### 4.4.2 Compilation Settings

* Loss Function: Mean Squared Error (MSE)
* Optimizer: Adam
* Metrics: Mean Absolute Error (MAE)

### 4.4.3 Training Strategy

* Early stopping was implemented based on validation loss to prevent overfitting.
* Model was trained for a maximum of 50 epochs with a batch size of 32

### 4.4.4 Training and Validation Performance

Figure 8 shows that the training loss is a steady decline over epochs. The validation loss has some fluctuations.

A graph showing a line graph

AI-generated content may be incorrect.

Figure 8 Bi-RNN Training vs Validation Loss

### 4.4.5 Prediction Results on Test Set

Figure 9 shows that model predicted values follow a general trend. The model struggles predicting spikes in energy consumption.

A graph showing a number of energy consumption

AI-generated content may be incorrect.

Figure 9: Bi-RNN Actual vs Predicted Energy Consumption

### 4.4.6 Evaluation Metrics

Based on 5 independent runs on the test data, the average MSE and MAE of the model are:  
Average MSE: 0.007748  
Average MAE: 0.06103

## 4.5 Model 3: Long Short-Term Memory (LSTM)

### 4.5.1 Architecture

The third model in our code uses Long Short-Term Memory. This model solves the vanishing gradient problem which a standard RNN has. The LSTM cells use memory gate, these memory gates help to retain useful information over a long sequence. This helps the model to learn better patterns in energy consumption.

Details of Architecture:

* Input: 18-time steps × 20 features
* LSTM Layer: 64 units with tanh activation
* Output Layer: Dense layer with 1 unit for energy prediction

### 4.5.2 Compilation Settings

* Loss Function: Mean Squared Error (MSE)
* Optimizer: Adam
* Evaluation Metric: Mean Absolute Error (MAE)

### 4.5.3 Training Strategy

The model was trained using early stopping, with a patience of 5 epochs and automatic restoration of the best weights. Training was conducted for a maximum of 50 epochs using a batch size of 32. Early stopping terminated training after 16 epochs.

### 4.5.4 Training and Validation Performance

Figure 10 shows the training and validation loss curves for the LSTM model.  
The validation loss remained consistently lower than the training loss, indicating good generalization and minimal overfitting throughout the training process.

A graph showing the difference between training and validation loss

AI-generated content may be incorrect.

Figure 10 Training vs Validation Loss Curve for LSTM

### 4.5.5 Prediction Results on Test Set

As shown in Figure 11, the LSTM model follows the broader energy consumption trend, although it still struggles with predicting sharp spikes. Nevertheless, the model is slightly better at handling fluctuations than the Simple RNN, likely due to its ability to retain longer-term patterns.

A graph showing a line graph

AI-generated content may be incorrect.

Figure 11 Appliance Energy Consumption - Actual vs Predicted (LSTM Model)

### 4.4.6 Evaluation Metrics

The LSTM model's performance on the test with average values of MSE and MAE over 5 independent runs is summarized below:

* Average Loss (MSE): 0.007586
* Average MAE: 0.058501

These results are slightly weaker than the Simple RNN in terms of raw MSE/MAE, which may indicate the LSTM didn’t capture unique signals beyond what the simpler model learned

## 4.6 Model 4: Gated Recurrent Unit (GRU)

### 4.6.1 Architecture

The fourth and final model utilizes a Gated Recurrent Unit (GRU) layer, which is a lightweight alternative to LSTM. GRUs merge the forget and input gates into a single update gate and exclude the output gate entirely. This leads to a simpler and faster model while retaining the ability to capture sequential dependencies effectively.

Architecture Details:

* Input: 18-time steps × 20 features
* GRU Layer: 64 units with tanh activation
* Output Layer: Dense layer with 1 unit for regression output

### 4.6.2 Compilation Settings

* Loss Function: Mean Squared Error (MSE)
* Optimizer: Adam
* Metric: Mean Absolute Error (MAE)

### 4.6.3 Training Strategy

The GRU model was trained using the same approach as the previous models, with early stopping based on validation loss and a patience of 5 epochs. The training converged in 13 epochs, and the model achieved stable validation loss values, indicating a good balance between learning and generalization.

### 4.6.4 Training and Validation Performance

Figure 12 shows the training and validation loss curves.  
The GRU model displayed a steady decline in training loss, with some fluctuations in validation loss but no signs of overfitting.

A graph showing the difference between training and validation loss

AI-generated content may be incorrect.

Figure 12 Training vs Validation Loss Curve for GRU

### 4.6.5 Prediction Results on Test Set

Figure 13 illustrates the GRU model’s performance on unseen test data.  
The model effectively captured the general pattern of energy usage but, like the previous models, struggled with predicting sudden consumption spikes. However, it exhibited slightly better alignment with the actual values in baseline periods than the LSTM.

A graph of energy consumption

AI-generated content may be incorrect.

Figure 13 Appliance Energy Consumption - Actual vs Predicted (GRU Model)

### 4.6.6 Evaluation Metrics

The performance of the GRU model on the test set over 5 independent runs with average MSE and MAE is summarized below:

* Average (MSE): 0.007902
* Average MAE: 0.060999

Compared to the LSTM and Simple RNN, the GRU achieved the lowest MAE and competitive MSE, making it a strong candidate for time-efficient and accurate forecasting.

## 4.7 Model 5: Encoder-Decoder

### 4.7.1 Architecture

The fifth model in our code makes use of encoder-decoder architecture based on the LSTM layer. This model captures the temporal dependencies by encoding the input into a context vector and the decoding into an output sequence.

Architecture Details:

* Input: 36-times steps × 20 features
* Encoder LSTM Layer: 64 units with tanh activation
* Context Vector: RepeatVector(1) to pass encoded context to decoder
* Decoder LSTM Layer: 64 units with tanh activation \)
* Output Layer: TimeDistributed(Dense(1)) for single-step energy prediction

### 4.7.2 Compilation Settings

* Loss Function: Mean Squared Error (MSE)
* Optimizer: Adam
* Metric: Mean Absolute Error (MAE)

### 4.7.3 Training Strategy

The Encoder-Decoder model also uses early stopping based on the validation, with a patience level of 5. The training process ended at 10 epochs.

### 4.7.4 Training and Validation Performance

The figure 14 shows the training vs validation loss chart, the loss curve in the chart can be observed as smooth decline. This indicates a go fit without any complexity.

A graph showing the difference between a line and a line

AI-generated content may be incorrect.Figure 14 Encoder-Decoder LSTM: Training vs Validation Loss

### 4.7.5 Prediction Results on Test Set

Figure 15 shows the actual vs predicted values of energy consumption. The model performed well in base line predictions while struggling a bit for predicting the spikes.

A graph showing a line graph

AI-generated content may be incorrect.

Figure 15 Encoder-Decoder LSTM: Actual vs Predicted Energy Consumption

### 4.7.6 Evaluation Metrics

Based on 5 independent runs on the test sets, the average MSE and MAE are  
Average MSE: 0.007469  
Average MAE: 0.056027

# Chapter 5: Results and Model Evaluation

## 5.1 Overview

This chapter focuses on the comparison of all the deep learning models that are executed for the prediction. Each model was assessed based on the performance of test data. The models are compared using visual and quantitative measures. The goal is to identify the model with best accuracy and generalization.

## 5.2 Performance Metrics

The table below shows the Average test MSE and MAE for 5 independent runs on test sets.

|  |  |  |
| --- | --- | --- |
| **Model** | **Average Test MSE** | **Average Test MAE** |
| Simple RNN | 0.007563 | **0.04286** |
| Bi-RNN | 0.007748 | 0.06103 |
| LSTM | 0.007586 | 0.058501 |
| GRU | 0.007902 | 0.06099 |
| Encoder-Decoder | **0.007469** | 0.056027 |

Interpretation:

The best overall model is the Encoder-Decoder LSTM. It has the lowest MSE, competitive MAE and good stability. This model is a good trade-off between accuracy and generalization.

## 5.3 Visual Comparison of Model Predictions

* RNN – reasonable baseline performance but underestimates spikes
* Bi-RNN - reasonable baseline performance but underestimates spikes
* LSTM – slightly smoother but more lag in capturing rapid changes
* GRU – strong baseline alignment, modest improvements over LSTM on general trend
* Encoder-Decoder - strong baseline alignment, modest improvements over GRU on general trend

These comparisons highlight each model's ability to approximate consumption behavior, particularly during periods of low-to-moderate usage. However, all models struggled with capturing sharp peaks, which may be due to:

* Limited input sequence length (18-time steps)
* High volatility or randomness in energy usage behaviors (e.g., cooking, heating)

## 5.4 Training Behavior and Convergence

All models were trained using early stopping to avoid overfitting, and their loss curves were analyzed:

* RNN: fast convergence within 17 epochs
* Bi-RNN: fast convergence within 13 epochs
* LSTM: smoother and longer convergence (~16 epochs)
* GRU: Steady and consistent learning with minor fluctuations (~13 epochs)
* Encoder-Decoder: fast convergence within 10 epochs

Figure references:

A graph with different colored lines

AI-generated content may be incorrect.

Figure 16 Training and validation loss curves for all models

Observation: GRU demonstrated the most stable learning behavior with minimal overfitting, likely due to its efficient structure and fewer trainable parameters compared to LSTM.

## 5.5 Model Selection Recommendation

Based on both quantitative and qualitative analysis, the Encoder-Decoder model appears to offer the best trade-off between:

* Prediction accuracy (low MAE)
* Simplicity and efficiency
* Stability during training

Recommendation: For future implementations or real-time deployment in smart home systems, GRU is recommended as the default architecture, with the potential to improve further via:

* Longer input sequences
* Multivariate or multi-horizon forecasting
* Data augmentation or external feature integration (e.g., occupancy sensors)

## 5.6 Limitations and Future Work

While the results are promising, there are key limitations:

* All models struggle with predicting sharp spikes in energy use
* External behavioral factors (like user activity) are not included
* Only a single household dataset was used

Future improvements may include:

* Attention-based models or Transformer architectures
* Hybrid models combining CNN and RNN layers
* Testing on multiple households to improve generalization

# Chapter 6: Conclusion and Final Recommendations

## 6.1 Summary of Work

In this project, we developed and evaluated three recurrent neural network architectures — Simple RNN, Bi-RNN, LSTM, GRU, and Encoder-Decoder — for the purpose of forecasting household appliance energy consumption at 10-minute intervals.  
Through a rigorous process involving data preprocessing, feature engineering, model building, and comparative analysis, the study explored how sequential modeling techniques can capture short-term energy usage patterns using historical environmental conditions and engineered time features.

Key steps undertaken:

* Parsed and extracted meaningful features from timestamps (e.g., hour of day, day of week) including sinusoidal encodings.
* Scaled features for effective neural network learning.
* Structured time series data into supervised learning format through sliding windows.
* Built, trained, and evaluated three different RNN-based models using early stopping to avoid overfitting.

## 6.2 Key Findings

* The Encoder-Decoder LSTM achieved the best performance with the lowest MSE. This outperforms all other models.
* Simple RNN (vanilla RNN) followed the Encoder-Decoder, offering good performance with lowest MAE and simpler architecture.
* All the models exhibited difficulty in predicting the sudden spikes in energy consumption.

## 6.3 Final Recommendations

Based on the comparative results:

* Encoder-Decoder is recommended for short-horizon energy forecasting tasks where data is relatively stable and highly granular (e.g., smart meters).
* Feature engineering (particularly temporal features) played a critical role in enabling the models to learn daily and weekly consumption patterns effectively.

## 6.4 Limitations

* Single Household Data: Results are based on one residential environment and may not generalize across multiple homes with diverse usage patterns.
* Uncaptured Human Behavior: No direct behavioral or occupancy data was incorporated, limiting the ability to explain abrupt usage spikes.
* Limited Forecast Horizon: Only short-term (10-minute ahead) forecasting was performed; multi-step or multi-horizon forecasting was not explored.

## 6.5 Future Work

Future research directions could include:

* Extending the models to multi-step forecasts (e.g., predict energy consumption several hours ahead).
* Incorporating external behavioral signals such as motion sensors, appliance schedules, or calendar data.
* Experimenting with advanced architectures like Bidirectional RNNs, Seq2Seq models, or Transformer-based architectures that naturally handle longer sequence modeling and attention.
* Expanding the dataset across multiple households and varying climates to build a generalized energy forecasting system.

# Chapter 7: Bibliography

1. Barbato, A., Capone, A., Rodolfi, M., & Tagliaferri, D. (2011, October 1). Forecasting the usage of household appliances through power meter sensors for demand management in the smart grid. IEEE Xplore. <https://doi.org/10.1109/SmartGridComm.2011.6102356>
2. Candanedo, L. M., Feldheim, V., & Deramaix, D. (2017). Data driven prediction models of energy use of appliances in a low-energy house. Energy and Buildings, 140, 81–97. <https://doi.org/10.1016/j.enbuild.2017.01.083>
3. Basu, K., Hawarah, L., Arghira, N., Joumaa, H., & Ploix, S. (2013). A prediction system for home appliance usage. Energy and Buildings, 67, 668–679. <https://doi.org/10.1016/j.enbuild.2013.02.008>
4. Jones, R. V., & Lomas, K. J. (2016). Determinants of high electrical energy demand in UK homes: Appliance ownership and use. Energy and Buildings, 117(1), 71–82. <https://doi.org/10.1016/j.enbuild.2016.02.020>
5. UCI Machine Learning Repository. (n.d.). Archive.ics.uci.edu. <https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction>