Democratic and Popular Republic of Algeria
Ministry of Higher Education and Scientific Research
Abderrahmane Mira University of Béjaïa

Faculty of Exact Sciences
Department of Operational Research

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Data Science and Decision Support

Thesis Title

Deep Learning-Based Classification of Households for Domestic Consumption Balancing

Submitted by:

Ayad Oussama
Ghaoui Mohammed Riad

Chair	**Dr. M. Soufit**	**Associate Professor**	**A/Mira University Béjaia.**
Supervisor	**Dr. L. Asli**	**Associate Professor**	**A/Mira University Béjaia.**
Co-Supervisor	**Mrs. Z. Bouzeria**	**PhD Candidate**	**A/Mira University Béjaia.**
Examiner	**Dr. A. Laour**	**Associate Professor**	**A/Mira University Béjaia.**
Examiner	**Dr. R. Djabri**	**Associate Professor**	**A/Mira University Béjaia.**

Class of 2023/2024

Appreciations , Consecrations

We would like to extend our heartfelt gratitude and deepest appreciation to those who have supported and guided us throughout the journey of our master's thesis.

First and foremost, we are immensely grateful to Mr. Asli, our supervisor, for his invaluable guidance, continuous support, and unwavering belief in our abilities. His expertise, insightful feedback, and encouragement have been instrumental in shaping this research. Without his mentorship, this work would not have been possible.

We also wish to express our sincere thanks to Miss Zahra Bouzeria, our co-supervisor, for her constant support, constructive feedback, and valuable insights. Her dedication and commitment have significantly contributed to the success of our research.

Our deepest appreciation goes to our families— Ayad, Ghaoui, and Mehfadia— whose unwavering love, patience, and encouragement have been our source of strength and inspiration. Their constant support and understanding have been pivotal in helping us overcome challenges and stay motivated throughout this journey.

To all of you, we are profoundly grateful for your contributions, support, and belief in us. Thank you for being an integral part of this accomplishment.

Table of Content

List of Figures

List of Tables

Abreviations list

AMI Advanced metering infrastructure

ANN Artificial Neural Networks

ATL Appliance Transfer Learning

CNN Convolutional Neural Network

CTL Cross-Domain Transfer Learning

DDPG Deep Deterministic Policy Gradient

DL Deep Learning

EDA Exploratory Data Analysis

EPD Energy Performance Descriptor

GRU Gated Recurrent Unit

LSTM Long Short Term Memory

MAE Mean Absolute Error

MIT Massachusetts Institute of Technology

ML Machine Learning

NAG Nesterov Accelerated Gradient

NDE Non-Destructive Examination

NILM Non-Intrusive Load Monitoring

NN Neural Network

REDD Reference Energy Disaggregation Dataset

REFIT Reducing Energy and Fuel In The home

RNN Recurrent Neural Network

SAE Sum of Absolute Errors

seq2point Sequence-to-Point Learning

seq2seq Sequence-to-Sequence Learning

UK-DALE UK Domestic Appliance Level Electricity

General Introduction

The household sector stands at the crossroads of energy consumption, environmental impact, and societal well-being. As we navigate the complexities of a rapidly changing world, understanding how households utilize energy becomes paramount.

Energy consumption in Algeria has increased significantly over the last decade, rising by approximately 35%. The residential sector represents more than a third of final energy consumption. The expected increase in population growth and housing, along with stable gas and oil reserves and production, which are the main sources of energy in Algeria, threatens the government's ability to maintain a balance between supply and demand for energy consumption.

Globally, energy consumption has been on a rapid rise in Algeria in recent decades, with the household sector accounting for more than a third of this global energy consumption. This increase has been offset since 2005 by stable oil and gas reserves, a worrying sign of potential supply-demand imbalances. With an expected increase in population and housing, the main drivers of household sector consumption, this supply-demand imbalance could become even more critical.

In the literature, due to the importance of household classification for energy balancing, several authors have contributed to the advancement of this field over the years. Some have focused on load forecasting and electricity load profiling [1–3], specifically using deep learning in domestic energy realms [4–9]. Identifying the determinants of household electricity use is a key element in facilitating efficient energy use. Moreover, segmenting households into well-resolved and characterized groups enables exploration of electricity use trends at disaggregated levels, revealing consumption patterns and opportunities for reduction across different consumer groups. By considering such groups, the drivers and implications of consumption trends can be better understood, yielding new insights into electricity use and providing opportunities to target policies and interventions that meet the needs of population sub-groups.

This thesis explores the application of deep learning methodologies for the classification of households based on their energy consumption patterns. The primary objective is to develop models capable of accurately categorizing households according to their energy usage profiles. Such classification is crucial for achieving balance in domestic energy consumption, optimizing resource allocation, and promoting energy efficiency. As we delve into this multifaceted field, challenges abound. Data privacy, model interpretability, and scalability pose hurdles. Yet, the opportunities are equally compelling. It aims for policy frameworks that adapt dynamically to household needs, energy-aware smart homes that optimize usage, and equitable access to energy services for all. Through this thesis, we aim to contribute to the advancement of deep learning applications in energy management, specifically in the context of household energy consumption classification. By leveraging advanced neural network architectures and comprehensive datasets, this research endeavors to provide insights and solutions that enhance the efficiency and sustainability of domestic energy usage.

Our study leaps into the realm of artificial intelligence. Specifically, we focus on applying deep learning techniques, particularly Convolutional Neural Networks (CNNs), to classify households visually. The work is structured as follows:

Chapter 1 provides an in-depth exploration of deep learning, beginning with an introduction to its fundamental concepts and various learning methods. It delves into the architecture and workings of artificial neural networks (ANNs), including perceptrons, sigmoid neurons, and their applications in feedforward and convolutional neural networks (CNNs). Additionally, the chapter covers recurrent neural networks (RNNs) and their specialized variant, Long Short-Term Memory (LSTM), highlighting their significance in sequential data analysis.

Chapter 2 focuses on the context of energy classification, starting with an overview of electrical energy fundamentals and the evolution of domestic energy consumption patterns. It discusses the concept of energy disaggregation and explores existing methodologies for monitoring and classifying household energy usage. Furthermore, the chapter reviews the current literature on deep learning applications in domestic energy management, emphasizing recent advancements and challenges.

Chapter 3 details the methodology employed in this study, encompassing the collection and preprocessing of domestic energy consumption datasets. It describes the experimental setup, datasets used (such as REFIT Electrical Load Measurements and UK Domestic Appliance-Level Electricity), challenges encountered, and trends observed from the collected data.

Chapter 4 presents the experimental study conducted to classify households based on their energy consumption patterns. It provides an overview of the model architecture, discusses the application of transfer learning techniques, outlines evaluation metrics used to assess model performance, and presents the results obtained from testing on various datasets, including REFIT and UK-DALE.

1
Deep Learning

Introduction

In the chapters ahead, we'll uncover the burgeoning field of deep learning, which is set to gain even more attention in the years to come. This method has proven itself capable of tackling complex problems across a range of areas like computer vision, language processing, and speech recognition. Here, we'll delve into the different approaches to deep learning, with a special focus on artificial neural networks (ANNs). We'll take a closer look at various deep learning models, including feedforward neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory networks (LSTMs). Throughout our exploration, we'll weigh the pros and cons of each model and discuss their potential applications in different fields.

1.1 Introduction to Deep Learning

Deep Learning is a branch of machine learning that is based on artificial neural networks. It is capable of learning complex patterns and relationships within data. In deep learning, there is no need to explicitly program everything. It has become increasingly popular in recent years due to advancements in processing power and the availability of large datasets. Because it is based on artificial neural networks (ANN), also known as deep neural networks (DNN). These neural networks are inspired by the structure and function of biological neurons in the human brain, and they are designed to learn from large amounts of data [10].

1.2 The various learning methods

Deep learning can be used for supervised learning, unsupervised learning, as well as reinforcement learning. It employs a variety of methods to process this data [10].

1. **Supervised learning:**

   Supervised learning is the machine learning technique in which the neural network learns to make predictions or classify data based on labeled datasets. Here, we provide both the input features as well as the target variables. The neural network learns to make predictions based on the cost or error that arises from the difference between the predicted target and the actual target, this process is known as back-propagation. Deep learning algorithms like convolutional neural networks, and recurrent neural networks are used for many supervised tasks such as image classification and recognition, sentiment analysis, language translations, etc.

2. **Unsupervised learning:**

   Unsupervised learning is the technique in which the neural network learns to discover patterns or cluster the dataset based on unlabeled datasets. Here, there are no target variables. Instead, the machine must determine hidden patterns or relationships within the datasets. Algorithms like auto-encoders and generative models are used for unsupervised tasks such as clustering, dimensionality reduction, and anomaly detection.

3. **Reinforcement learning:**

   Reinforcement learning in machine learning is the technique in which an agent learns to make decisions in an environment to maximize a reward signal. The agent interacts with the environment by taking action and observing the resulting rewards. Deep learning can be used to learn policies, or a set of actions, that maximize the cumulative reward over time. Reinforcement learning algorithms like Deep Q Networks and Deep Deterministic Policy Gradient (DDPG) are used for tasks such as robotics and games, etc.

1.3 Artificial Neural Networks

Artificial neural networks are built on the principles of the structure and functioning of human neurons. They are also known as neural networks or networks of neurons. The input layer of an artificial neural network, which is the first layer, receives inputs from external sources and passes them to the hidden layer, which is the second layer. Each neuron in the hidden layer receives information from neurons in the previous layer, computes the weighted sum, and then transfers it to neurons in the next layer. These connections are weighted, meaning that the impacts of inputs from the previous layer are more or less optimized by assigning each input a distinct weight. These weights are then adjusted during the training process to improve the model's performance [10].

1.3.1 Artificial Neurons

Artificial neurons, also called units, are found in artificial neural networks. The entire artificial neural network is composed of these artificial neurons, which are arranged in a series of layers. The complexities of neural networks will depend on the complexities of underlying patterns in the dataset, whether a layer has a dozen units or millions of units. Generally, an artificial neural network consists of an input layer, an output layer, as well as hidden layers. The input layer receives data from the outside world that the neural network must analyze or learn from [10].

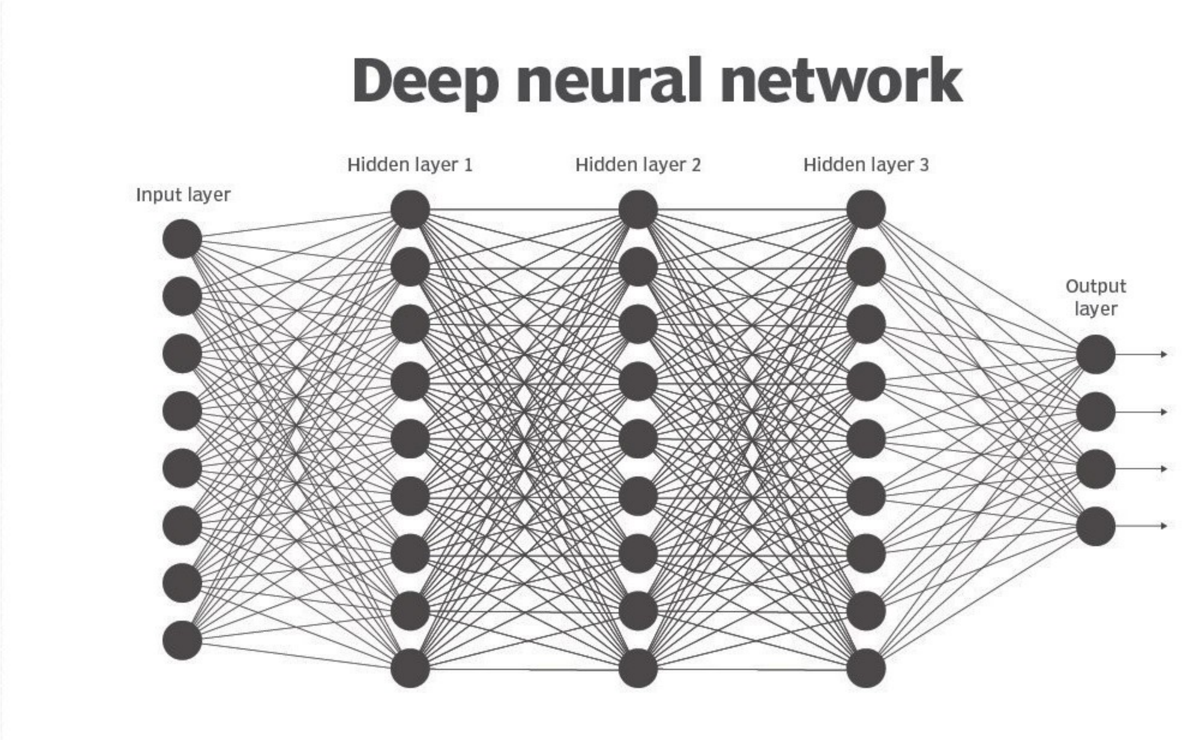

Figure 1.1: Artificial Neurons

1.3.2 Perceptrons

Perceptrons are one of the fundamental elements of machine learning, especially in the field of supervised learning and binary classification. Perceptrons were developed in the 1950s and 1960s by the scientist Frank Rosenblatt, inspired by the earlier work of Warren McCulloch and Walter Pitts. A perceptron is a learning model based on the simplified functioning of a biological neuron [11].

Here is a simplified explanation of its operation :

1. **Inputs :** The perceptron takes a set of inputs $x_1, x_2, ..., x_n$, each being weighted by a corresponding weight $w_1, w_2, ..., w_n$.

2. **Weighted Sum :** The inputs are multiplied by their respective weights and then summed:

$$z = \sum_{i=1}^{n} w_i \cdot x_i + b \quad (1.1)$$

   Where b is the bias (an adjustable constant that shifts the activation function) and z is the weighted sum.

3. **Activation Function :** The weighted sum is then passed through an activation function. The most commonly used activation function for perceptrons is the step function, which outputs 1 if the weighted sum exceeds a certain threshold and 0 otherwise. Mathematically, this can be expressed as follows:

$$\text{output} = \begin{cases} 1 & \text{if } z \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

4. **Training :** The process of training the perceptron involves iteratively adjusting the weights and bias to minimize a loss function that measures the difference between the predicted output and the actual output.

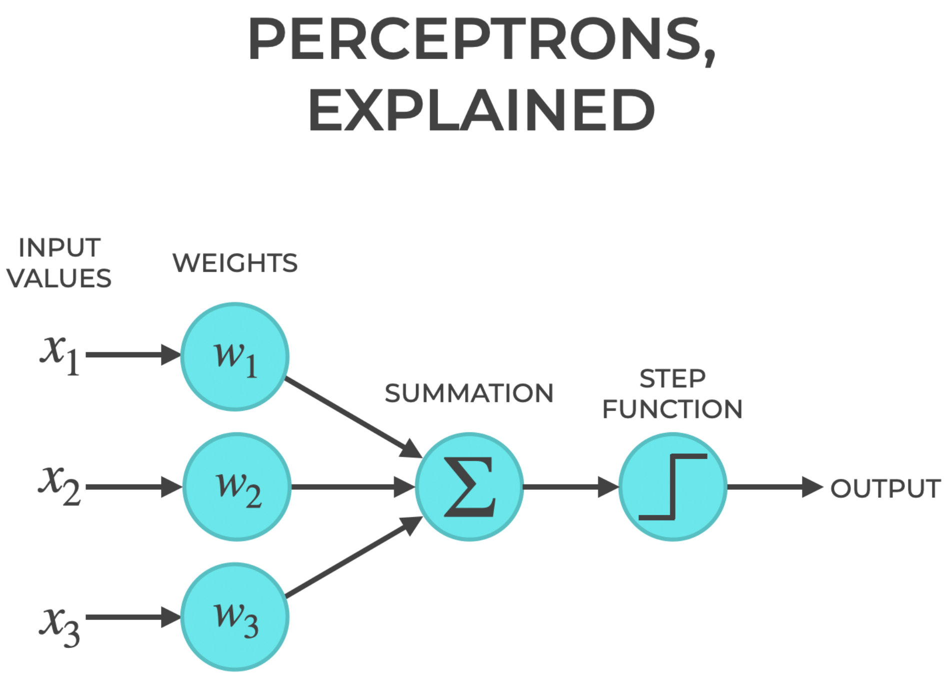

Figure 1.2: Basic Components of the Perceptron

Example of Perceptron

Let's consider this example of a perceptron to help us decide whether to go to the cinema or not, based on the following factors: the weather (good or bad), whether my friend will accompany me or not, and the quality of the movie (bad or good). We will also add random weights to these factors.

Let's proceed with the training and decision-making process with randomly initialized weights for our perceptron example.

Given the randomly initialized weights and bias:

- Weight for weather (w_1): 0.4
- Weight for movie quality (w_2): -0.7
- Weight for friend's availability (w_3): 0.2
- Bias (b): 0.1

We will use this perceptron to make a decision on whether to go to the cinema or not based on the input factors.
Here's how the process works:

1. **Training Data:**

   We have labeled training data based on past experiences. For example:

   - Good weather, good movie, friend present: Go to the cinema (1)
   - Bad weather, good movie, friend present: Go to the cinema (1)
   - Good weather, bad movie, friend present: Do not go to the cinema (0)
   - Bad weather, bad movie, friend present: Do not go to the cinema (0)

   and so on...

2. **Perceptron Training:**

   We adjust the weights and biases during training using a learning algorithm like the perceptron learning rule. The perceptron learns to classify input combinations into "Go to the cinema" or "Do not go to the cinema" based on the given factors. Decision Making:

   Once trained, the perceptron predicts whether to go to the cinema or not based on new input combinations of weather, movie quality, and friend availability. Suppose we have a new situation where:

   - Weather is good (1)
   - Movie quality is good (1)
   - Friend is not available (0)

   We will calculate the activation of the perceptron and make a decision based on the result.

3. **Calculating Activation:**

$$\text{Activation} = w_1 \times \text{Weather} + w_2 \times \text{Movie quality} + w_3 \times \text{Friend availability} + b$$

$$\text{Activation} = 0.4 \times 1 + (-0.7) \times 1 + 0.2 \times 0 + 0.1$$

$$\text{Activation} = 0.4 - 0.7 + 0.1 = -0.2$$

   As the activation is negative, the perceptron predicts "Do not go to the cinema" (0).

   This is how the perceptron makes a decision based on input features and weights learned during training.

4. **Note:**

   - If the weather weight is 0 for you, it could be different for someone else. A higher weight means that the weather is more important to them.
   - If the threshold value is 1.5 for you, it may be different for someone else. A lower threshold means they are more inclined to go to any concert.

1.3.3 Sigmoid Neurons

Sigmoid neurons are similar to perceptrons, but they are adjusted slightly to make their output much smoother than that of the perceptron, which is step-like. The sigmoid neuron uses the sigmoid activation function on the weighted sum of its inputs and bias, yielding an output between 0 and 1. This output can be interpreted as the probability of a binary event occurring or as the neuron's activation level [11].

The sigmoid function is defined as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1.2}$$

Where:

- $\sigma(z)$ is the output of the sigmoid function.
- e is the base of the natural logarithm (Euler's number).
- z is the input to the function.

Now, let's examine the sigmoid function and its properties:

1. **Range:** The output of the sigmoid function $\sigma(z)$ lies between 0 and 1. As z becomes large (positive or negative), the output tends towards 1 or 0, respectively. This property makes the sigmoid function suitable for binary classification tasks or representing probabilities.

2. **S-Shaped Curve:** The sigmoid function has an S-shaped curve, meaning it starts at 0 for very negative inputs, rises rapidly towards 1 as the input increases, and flattens out at 1 for large positive inputs. This gradual transition allows for smooth, continuous changes in output.

3. **Differentiability:** The sigmoid function is differentiable everywhere, which is important for training neural networks using gradient-based optimization algorithms like backpropagation.

   The derivative of the sigmoid function with respect to its input z is given by:

   $$\frac{d\sigma(z)}{dz} = \sigma(z) \times (1 - \sigma(z)) \tag{1.3}$$

   This derivative is often used during neural network training to compute gradients for updating the network parameters.

1.3.4 Difference Between Perceptron and Sigmoid Neurons

According to [11], the behavior of a network of sigmoid neurons is exactly the same as that of a network of perceptrons when the limit $c \to \infty$ and $w \cdot x + b \neq 0$ for the input x of any particular perceptron in the network. We need to demonstrate that the behavior of the sigmoid neuron approaches that of a perceptron as c becomes very large.

Recall that the output of a sigmoid neuron is given by the sigmoid function (1.2)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

where $z = w \cdot x + b$.

If we multiply both w and b by a positive constant $c > 0$, the new output of the sigmoid neuron becomes:

$$\sigma(cz) = \frac{1}{1 + e^{-cz}}$$

Now, as $c \to \infty$, the term $-cz$ inside the exponential function will become very negative for any input x where $w \cdot x + b \neq 0$. Consequently, $e^{-cz} \to 0$, and thus $\sigma(cz) \to 1$ (since the denominator tends towards 1).

However, when $w \cdot x + b = 0$ for one of the perceptrons, we have $z = 0$, and $cz = 0$ for any value of c. In this case, the behavior of the sigmoid neuron will differ from that of the perceptron as the sigmoid neuron will always produce $\sigma(0) = \frac{1}{2}$, which is different from the behavior of a perceptron (which outputs 1 or 0).

In summary, when $w \cdot x + b \neq 0$ for all perceptrons in the network, the behavior of the network of sigmoid neurons approaches that of the network of perceptrons as $c \to \infty$. However, when $w \cdot x + b = 0$ for one of the perceptrons, this property fails as the sigmoid neuron will always produce $\frac{1}{2}$ in this case, which is different from the behavior of a perceptron.

1.3.5 How Artificial Neural Networks Work

In a fully connected artificial neural network, there is an input layer and one or more hidden layers, interconnected with each other. Each neuron receives information from neurons in the previous layer or the input layer. The output of one neuron becomes the input for other neurons in the next layer of the network, and this process repeats until the last layer produces the output of the network. After passing through one or more hidden layers, this data is transformed into meaningful information for the output layer. Finally, the output layer provides a response from the artificial neural network to the data entering it.

Neurons are connected from one layer to another in most neural networks. Each link has weights that determine the importance of one unit's influence on another. The neural network gradually learns from the data as it passes from one unit to another, ultimately producing an output from the output layer [12].

1.4 Gradient-Based Learning

Gradient-based learning in neural networks involves using gradient descent optimization algorithms to update the parameters (weights and biases) of the network to minimize a predefined loss function. This process is based on the chain rule of differential calculus and involves computing the gradients of the loss function with respect to the network parameters during back-propagation [13].

Let's break down the process step by step using mathematical formulas:

Forward Pass:

- Given an input x to the neural network, the output y is computed through a series of transformations in each layer of the network.
- Mathematically, the output of a neuron j in layer l (excluding the input layer) is calculated as follows:

$$z_j^{(l)} = \sum_{i=1}^{n^{(l-1)}} w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)} \tag{1.4}$$

$$a_j^{(l)} = g(z_j^{(l)}) \tag{1.5}$$

  Where:

  1. $w_{ji}^{(l)}$ is the weight connecting neuron i in layer $(l-1)$ to neuron j in layer l.
  2. $a_i^{(l-1)}$ is the activation of neuron i in layer $(l-1)$.
  3. $b_j^{(l)}$ is the bias of neuron j in layer l.
  4. $g(\cdot)$ is the activation function applied element-wise to the weighted sum $z_j^{(l)}$.

Loss Calculation:

- After obtaining the output of the neural network y, a loss function $L(y, \hat{y})$ is calculated to quantify the difference between the predicted output y and the true target values $\hat{y}$.

- The choice of the loss function depends on the task, such as mean squared error (MSE) for regression or cross-entropy loss for classification.

Back-propagation:

- The gradients of the loss function with respect to the network parameters are computed using the chain rule of differential calculus.
- For each parameter $w_{ji}^{(l)}$ and $b_j^{(l)}$, the gradient is computed as follows:

$$\frac{\partial L}{\partial w_{ji}^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial w_{ji}^{(l)}} \tag{1.6}$$

$$\frac{\partial L}{\partial b_j^{(l)}} = \frac{\partial L}{\partial z_j^{(l)}} \cdot \frac{\partial z_j^{(l)}}{\partial b_j^{(l)}} \tag{1.7}$$

Where:

1. $\frac{\partial L}{\partial z_j^{(l)}}$ is the gradient of the loss function with respect to the weighted sum $z_j^{(l)}$ of neuron j in layer l.
2. The second term represents the derivative of the activation function applied to the weighted sum.

Gradient Descent:

- The computed gradients are used to update the network parameters in the opposite direction of the gradient to minimize the loss function.
- Parameters are updated according to:

$$w_{ji}^{(l)} \leftarrow w_{ji}^{(l)} - \eta \frac{\partial L}{\partial w_{ji}^{(l)}} \tag{1.8}$$

$$b_j^{(l)} \leftarrow b_j^{(l)} - \eta \frac{\partial L}{\partial b_j^{(l)}} \tag{1.9}$$

Where η is the learning rate, controlling the size of parameter updates.

Iteration:

Steps 1-4 are repeated iteratively for multiple epochs until convergence or until a predefined stopping criterion is met.

1.5 Feedforward Neural Network

A feedforward neural network represents one of the most basic structures among artificial neural networks. In this network, information flows in only one direction, starting from input nodes, possibly passing through hidden layers, and ending at output nodes. Unlike recurrent or convolutional networks, there is no feedback or loop in this type of network. Feedforward neural networks were among the earliest developed in this field and stand out for their simplicity. Feedforward neural networks are used in a variety of machine learning tasks, including pattern recognition, classification, regression analysis, image recognition, and time series prediction [14].

1.5.1 Challenges and Limitations

Despite the power of feedforward neural networks, they face their challenges and limitations. Among these challenges are the selection of the number of hidden layers and the number of neurons in each layer, parameters that can have a significant effect on the network's performance. Overfitting is also a common problem, where the network learns the training data too precisely, including noise, leading to poor performance when faced with new data [15].

1.6 Convolutional Neural Network

A Convolutional Neural Network (CNN) in deep learning is a specialized type of neural network designed to process and analyze visual data, such as images. It is particularly effective for tasks involving pattern recognition and image classification. CNNs have revolutionized the field of computer vision and are widely used in various applications, including image recognition, object detection, facial recognition, medical image analysis, and autonomous vehicles [13].

1.6.1 Architecture of a CNN

A typical CNN consists of several layers:

Convolutional Layer:

The central building block of a CNN is the convolutional layer. In this layer, convolution operations are performed on the input image using learnable filters or kernels to extract features. Each filter is applied over the entire input image, and the result is a feature map.

Mathematically, the output feature map $\mathbf{Y}$ of a convolutional layer can be calculated as follows:

$$Y[i,j] = \sum_{m}\sum_{n} X[i+m, j+n] \times W[m,n] + b \tag{1.10}$$

where:

- $Y[i,j]$ is the value of the feature map at position (i,j).
- X is the input image.
- W is the filter (weight matrix).

- b is the bias.
- m and n are the filter indices.

Activation Function:

Typically, an activation function like ReLU (Rectified Linear Unit) is applied element-wise on the feature map to introduce non-linearity. This helps the CNN learn complex patterns and relationships in the data.

$$Z = \text{ReLU}(Y)$$

Pooling Layer:

After convolution, pooling layers are often used to reduce the spatial dimensions of feature maps while retaining important information. Max pooling is a common pooling technique that extracts the maximum value from a local neighborhood.

$$Z' = \text{MaxPooling}(Z)$$

where Z' is the output of the pooling layer.

Fully Connected Layer:

Finally, the flattened output from the last convolutional or pooling layer is fed into one or more fully connected layers, followed by a softmax layer for classification or a regression layer for regression tasks.

1.6.2 Advantages of a CNN

CNNs are special because they have the following properties:

- **Parameter Sharing:** CNNs use parameter sharing, where the same set of weights is applied to multiple locations in the input image. This significantly reduces the number of parameters compared to fully connected networks, making CNNs more efficient and less prone to overfitting.
- **Translation Invariance:** CNNs are capable of capturing translation-invariant features in the input data. This means they can recognize patterns regardless of their position in the input image, making them robust to translation.

1.6.3 Recent Developments

CNNs continue to evolve with constant improvements. More advanced architectures like ResNet, Inception, and EfficientNet have been introduced to enhance the performance and efficiency of CNNs. Additionally, the use of techniques such as transfer learning and data augmentation has led to even better performance in various computer vision tasks [16].

1.7 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) in deep learning are a specific type of neural network suited for processing sequential data, such as texts, sounds, or time series. Unlike feedforward networks, RNNs are capable of maintaining long-term memory through their ability to retain hidden states that are updated at each step based on current inputs and previous states. This ability to contextualize sequential data by referring to past contexts makes RNNs effective tools for various tasks such as prediction, classification, and sequence generation [17].

1.7.1 Challenges and Recent Developments

While RNNs are effective at processing sequences of data, they encounter certain challenges such as the vanishing and exploding gradient problems. To overcome these difficulties, several variants of RNNs have been developed, including Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), which are specifically designed to better handle long-term dependencies in sequences [18].

1.8 Long Short-Term Memory Network

Long Short-Term Memory (LSTM) networks represent a specialized version of Recurrent Neural Networks (RNNs), created to overcome the limitations of traditional RNN models in capturing long-term dependencies in sequential data sequences. LSTMs are special because they can retain information about long temporal sequences, making them particularly well-suited for tasks such as natural language processing (NLP), sequence modeling, and time series prediction [13].

1.8.1 Challenges and Recent Advances

Although LSTMs have significantly enhanced the capabilities of sequence processing models, they are not exempt from inherent challenges, such as computational complexity and the need for substantial training data volumes. Additionally, recent research efforts have explored different versions of LSTMs, such as attention-based LSTMs, which further boost model performance in specific contexts [19].

1.9 Optimization Methods

Optimization methods are the heart of deep learning, Optimization methods in the context of machine learning and deep learning are the minimization of an objective function known as the loss function in order to improve the performance of the algorithm. They help researchers and deep learning engineers decide how well a model learns and how successful it becomes. There are different methods, like the classic Gradient Descent, and newer ones like Adam and Nesterov Accelerated Gradient.

1.9.1 Gradient Descent

Gradient Descent is a fundamental optimization algorithm employed in various machine learning and optimization tasks, including deep learning. Its goal is to iteratively minimize a given objective function, typically denoted as $J(\theta)$, with respect to a set of parameters θ [20].

Mathematically, the algorithm operates by first computing the gradient of the objective function $J(\theta)$ with respect to the parameters θ. This gradient, denoted as $\nabla J(\theta)$, provides information about the direction of the steepest ascent of the function at a specific point in the parameter space.

Subsequently, the algorithm updates the parameters θ by subtracting a fraction of the gradient from the current parameter values, controlled by a parameter known as the learning rate (α). The update rule can be represented as:

$$\theta = \theta - \alpha \cdot \nabla J(\theta) \tag{1.11}$$

The learning rate α plays a crucial role in the convergence behavior of Gradient Descent. Selecting an appropriate learning rate is essential to ensure convergence to the optimal solution without oscillations or divergence.

The iterative process of computing gradients and updating parameters is repeated until a stopping criterion is met, such as a predefined number of iterations or when the change in the objective function falls below a specified threshold.

1.9.2 Adam Optimizer

Adam optimizer is a widely used optimization algorithm in machine learning, particularly in deep learning. It combines ideas from both momentum and RMSProp methods to adaptively adjust the learning rates for different parameters during training [21].

The goal of Adam is to iteratively minimize a given objective function, typically denoted as $J(\theta)$, with respect to a set of parameters θ.

Mathematically, the algorithm computes the gradient of the objective function $J(\theta)$ with respect to the parameters θ, denoted as $\nabla J(\theta)$.

Adam maintains two moving averages: the first moment m, which is the exponential moving average of the gradients, and the second moment v, which is the exponential moving average of the squared gradients.

These moving averages are calculated as follows:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla J(\theta_t) \tag{1.12}$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla J(\theta_t))^2 \tag{1.13}$$

where β_1 and β_2 are hyperparameters that control the exponential decay rates of the moving averages.

The parameters θ are then updated using the following rule:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_t \tag{1.14}$$

where α is the learning rate, ϵ is a small constant to prevent division by zero, and $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected estimates of the first and second moments, respectively.

The Adam optimizer offers several advantages, including fast convergence, robustness to noisy gradients, and adaptive learning rates.

1.9.3 Nesterov Accelerated Gradient

Nesterov Accelerated Gradient (NAG) is an optimization algorithm commonly used in machine learning and deep learning. It is a modification of the classic momentum method that aims to improve convergence speed and stability [20].

Similar to other optimization algorithms, the goal of NAG is to iteratively minimize a given objective function $J(\theta)$ with respect to a set of parameters θ.

The key idea behind NAG is to adjust the parameter updates to account for the momentum. Instead of naively using the gradient at the current position to compute the next position, NAG computes an intermediate position based on the momentum-adjusted gradient.

Mathematically, the algorithm computes the gradient of the objective function $J(\theta)$ at the intermediate position $\theta - \beta \cdot v$, where β is the momentum coefficient and v is the momentum term. This intermediate position is denoted as $\tilde{\theta}$, and the gradient is evaluated at this position:

$$\nabla J(\tilde{\theta})$$

Then, the parameters are updated using this momentum-adjusted gradient:

$$\theta_{t+1} = \tilde{\theta} - \alpha \cdot \nabla J(\tilde{\theta}) \tag{1.15}$$

where α is the learning rate.

Nesterov Accelerated Gradient offers several advantages, including faster convergence compared to standard momentum, and it is less likely to oscillate around the optimal solution. However, choosing an appropriate momentum coefficient is crucial for optimal performance.

Conclusion

In conclusion, we've covered a lot of ground in this chapter, learning about different types of deep learning models and how they work in different situations. Each model has its own good points and not-so-good points, and which one to use depends on what you're trying to do and what data you have. As we move forward, we'll now shift our focus to the next chapter, which dives into the world of classification in the field of energy. We'll explore how deep learning techniques can be used to categorize and understand energy-related data, opening up new possibilities for solving energy-related challenges.

2

Classification in the Field of Energy

Introduction

This chapter serves as a comprehensive overview of electrical energy, encompassing its production, consumption, monitoring, challenges, and advancements in research and technology. It commences with an exploration of the fundamental concepts of electrical energy and the diverse methods employed for its production, spanning from traditional fossil fuels to nuclear energy and renewable sources.

The evolution of electrical energy consumption, particularly within domestic environments, is then examined, shedding light on the changing consumption patterns and the challenges they entail. Additionally, the chapter delves into the intricacies of energy disaggregation and the array of monitoring methods utilized to comprehend and optimize energy utilization.

Furthermore, the significance of the principle of classification in the energy domain is underscored, illustrating its role in categorizing energy-related data to enhance understanding and management. The chapter also explores the burgeoning applications of deep learning in domestic energy management, offering promising avenues for addressing challenges and optimizing energy efficiency.

Finally, the chapter provides a succinct overview of the current state of research in this field, laying the groundwork for further exploration and innovation in the realm of electrical energy management.

2.1 General Concepts of Electrical Energy

Electrical energy is a fundamental form of energy in our modern society, ubiquitous in our daily lives. At the core of this notion lie several key concepts.

Firstly, electric charge is the intrinsic property of subatomic particles, electrons, and protons, which carry negative and positive charges respectively. Electrical current, on the other hand, represents the flow of these charges through a conductor and can be either direct or alternating, depending on the context.

Electrical voltage, or potential difference, is the force that drives charges through a circuit, measured in volts. In parallel, electrical resistance represents the property of materials to oppose the flow of current, measured in ohms.

Finally, electrical power measures the amount of energy transferred per unit of time, an essential concept for assessing the usage and efficiency of electrical circuits. Together, these concepts form the basis of our understanding of electrical energy and its utilization in a wide range of applications, from electronic devices to electricity distribution systems.

2.1.1 Methods of Electrical Energy Production

According to [22], there are several methods of electrical energy production, each with its own advantages and disadvantages. Here are some of the most commonly used methods :

Electricity Production from Fossil Fuels :

This method involves the combustion of fossil fuels such as coal, oil, and natural gas to produce heat, which is then used to operate steam turbines or internal combustion engines, thereby generating electricity. While widely used due to its availability and relatively low cost, this method also emits significant amounts of greenhouse gases and contributes to climate change.

Nuclear Energy :

Electricity production from nuclear energy uses nuclear fission to generate heat, which is then converted into electricity through turbines and generators. Although this method produces few greenhouse gas emissions, it raises concerns regarding safety and the management of radioactive waste.

Hydroelectric Energy :

Hydroelectric energy is produced by harnessing the force of water to turn turbines, typically in hydroelectric dams. This method is renewable and relatively low in pollution, but it can have significant environmental impacts, such as disrupting aquatic ecosystems and submerging land.

Wind Energy :

Wind turbines use the wind to spin turbines, converting the kinetic energy of the wind into electricity. Wind energy is a renewable and clean source of electricity, although it is intermittent and dependent on weather conditions.

Solar Energy :

Solar panels convert sunlight into electricity using photovoltaic cells. Solar energy is an abundant and renewable source, but it is also intermittent and dependent on sunlight availability.

Geothermal Energy

Geothermal energy harnesses the natural heat of the Earth to produce electricity, typically by using steam or hot water underground to power turbines. This method is renewable and relatively low in pollution, but it is limited to certain geographical regions.

2.1.2 The Evolution of Electrical Energy Consumption

According to [23], global electricity production has significantly increased over the past three decades, rising from less than 12,000 terawatt-hours in 1990 to over 29,000 terawatt-hours in 2022. During this period, global electricity production experienced only two annual declines: in 2009 following the global financial crisis, and in 2020 amidst the coronavirus pandemic.

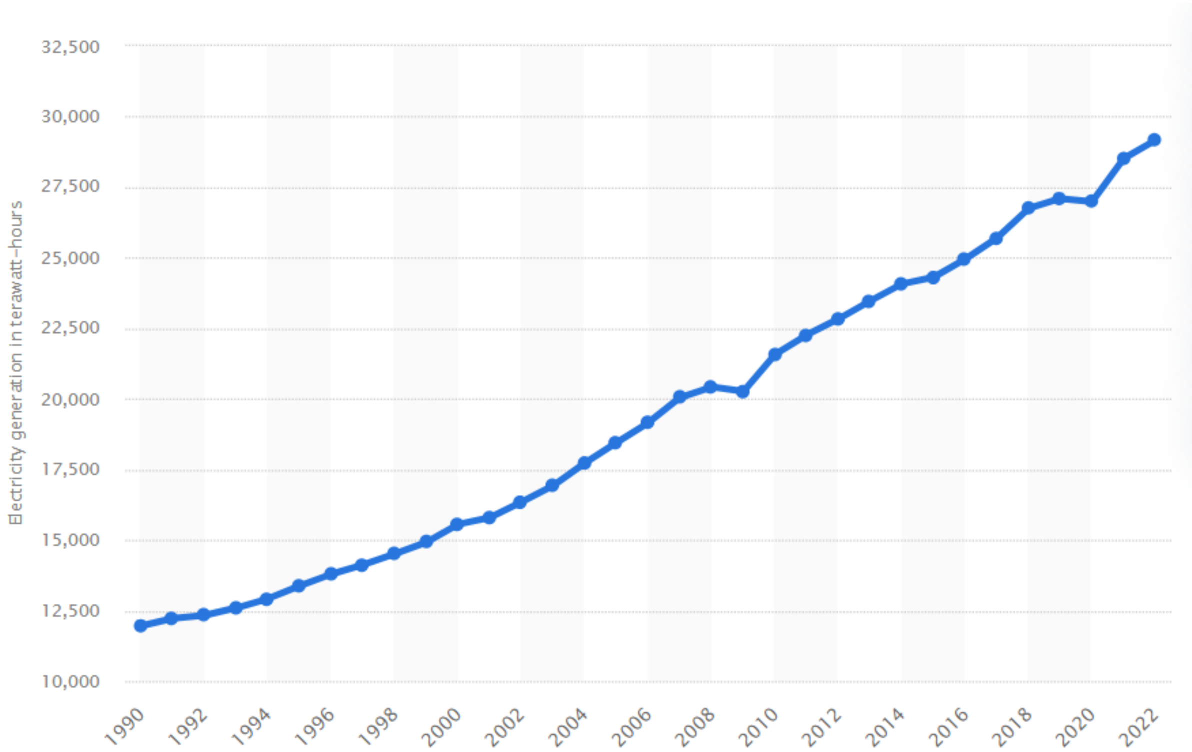

Figure 2.1: The Evolution of Electrical Energy Consumption

2.2 Domestic Energy Consumption

Domestic energy consumption represents a significant portion of the overall energy demand. How this energy is used within households can have a considerable impact on the environment, energy costs, and long-term sustainability. In this context, balancing domestic energy consumption becomes a major concern, requiring a proactive approach and innovative solutions. [24]

2.2.1 Understanding domestic energy consumption :

Domestic energy consumption, referring to the use of energy sources in homes, apartments, and other dwellings, is vital for local and global energy dynamics. It covers various sources like electricity, natural gas, oil, and renewables such as solar and wind power. Understanding it involves considering multiple factors.

Occupant behavior is a key determinant, as human activities within homes significantly affect energy usage. For instance, habits like leaving lights on or frequent appliance use impact consumption. Efficient appliances, like refrigerators and HVAC systems, also shape consumption patterns, with newer models typically using less energy. Additionally, proper thermal insulation in homes helps regulate indoor temperatures, reducing the need for excessive heating and cooling.

Weather conditions further influence energy consumption, with extreme temperatures prompting increased usage of heating or cooling systems. Seasonal variations also impact energy needs, such as longer daylight hours in summer affecting lighting usage. In summary, domestic energy consumption is complex, influenced by occupant behavior, appliance efficiency, insulation, and weather conditions [24].

2.2.2 Challenges of domestic energy consumption

The challenges of domestic energy consumption encompass various aspects that pose obstacles to achieving efficient and sustainable energy usage within households. Some of these challenges include [25] :

Peak demand and load management :

Peaks in electricity demand can strain the power grid, leading to the need for investments in infrastructure. Managing these peaks through load management strategies becomes essential to ensure the stability and reliability of the energy supply.

Technological limitations :

While advancements in technology offer solutions for optimizing energy usage, there may be limitations in accessibility and affordability for consumers. Ensuring that energy-efficient technologies are accessible to all households, regardless of socioeconomic status, is a challenge that needs to be addressed.

Behavioral barriers :

Changing consumer behavior towards more sustainable energy practices can be challenging. Encouraging individuals to adopt energy-saving habits and practices requires effective communication, education, and incentives.

Infrastructure constraints :

In some regions, inadequate infrastructure may limit access to reliable energy sources or hinder the implementation of renewable energy solutions. Addressing infrastructure gaps and promoting investment in energy infrastructure is essential for meeting domestic energy needs sustainably.

Data privacy and security :

With the increasing use of smart technologies and data-driven energy management systems, concerns about data privacy and security arise. Safeguarding consumer data and ensuring transparency in data collection and usage practices are essential to building trust and confidence in energy management solutions.

2.3 Energy disaggregation

Energy disaggregation relies on the principle that energy consumption can be broken down into several distinct components, each contributing in a specific way to the total consumption. This approach requires the use of technologies such as smart meters, energy consumption sensors, and machine learning algorithms to collect, analyze, and interpret energy consumption data at a granular level [26].

2.3.1 Monitoring Methods

Energy disaggregation can be achieved using different methods, each with its own advantages and limitations [27] :

NILM Monitoring (Non-Intrusive Load Monitoring) :

This method involves analyzing the characteristics of a building's overall energy consumption using machine learning techniques. While it is non-intrusive, it may require high-resolution data and may be limited by the accuracy of appliance modeling.

Load Monitoring :

This approach entails using sensors to directly measure the energy consumption of each appliance. While it provides accurate data, it can be costly to implement and may require intrusive installation.

Intrusive Monitoring :

In this method, special equipment is installed on each appliance to monitor its energy consumption. While this can provide very precise data, it can be expensive and may require modification of existing infrastructure.

Semi-Intrusive Monitoring :

This approach combines non-intrusive techniques with sensors installed on certain key appliances. While this may reduce costs and minimize disruptions, it may limit data coverage.

Non-Intrusive Monitoring :

This method relies on analyzing electricity meter data or other available sources without the need for additional sensor installation. While this approach is cost-effective and easy to implement, it may lack precision in disaggregating individual appliances.

2.4 Principle of Classification in the field of Energy

The principle of classification in the energy domain involves categorizing energy-related data into distinct groups or classes based on certain criteria or features. This classification is typically performed using machine learning or statistical techniques to analyze and interpret energy data effectively. The primary goal of classification in the energy domain is to gain insights, make predictions, or support decision-making processes related to energy management, consumption, or optimization [28].

Here are some key principles and considerations for classification in the energy domain:

1. **Feature Selection:** Identifying relevant features or attributes of energy data that can distinguish between different classes is crucial for effective classification. These features could include energy consumption patterns, usage profiles, environmental factors, or demographic information.

2. **Data Preprocessing:** Preprocessing steps such as data cleaning, normalization, and dimensionality reduction may be necessary to enhance the quality and usefulness of the data for classification tasks. This ensures that the data is in a suitable format and scale for the chosen classification algorithm.

3. **Choice of Classification Algorithm:** Selecting an appropriate classification algorithm depends on factors such as the nature of the data, the complexity of the classification task, and the desired outcomes. Commonly used algorithms include decision trees, support vector machines, k-nearest neighbors, and neural networks.

4. **Training and Evaluation:** The classification model is trained using labeled training data, where each data point is associated with a known class label. The model's performance is then evaluated using test data to assess its accuracy, precision, recall, and other relevant metrics.

5. **Validation and Optimization:** The trained model should be validated using independent validation data to ensure its generalizability and robustness. Additionally, model parameters may be optimized through techniques such as cross-validation or hyperparameter tuning to improve performance.

6. **Interpretability and Explainability:** Understanding how the classification model makes decisions is essential for interpreting its results and gaining insights from the data. Techniques for model interpretability, such as feature importance analysis or model visualization, can help explain the classification outcomes.

2.5 Applications of deep learning in domestic energy management

- **Load Forecasting:** Deep learning models can analyze historical energy consumption data to forecast future energy demand accurately. This enables homeowners to plan energy usage effectively, optimize appliance scheduling, and potentially reduce peak demand charges.

- **Energy Consumption Prediction:** Deep learning algorithms can predict household energy consumption patterns based on factors such as weather conditions, time of day, and occupancy. This information allows homeowners to adjust energy usage behaviors to minimize costs and improve efficiency.

- **Anomaly Detection:** Deep learning models can detect abnormal energy consumption patterns or equipment malfunctions in real time. By identifying anomalies, homeowners can address potential energy inefficiencies, equipment failures, or safety hazards promptly, thereby improving energy management and reducing maintenance costs.

- **Appliance Recognition:** Deep learning techniques, such as convolutional neural networks (CNNs), can analyze smart meter data or appliance signatures to identify individual appliances' energy consumption patterns. This allows homeowners to monitor and optimize energy usage for each appliance, leading to more efficient energy management.

- **Optimal Control Systems:** Deep reinforcement learning algorithms can optimize energy consumption by learning and adapting to dynamic environmental conditions, user preferences, and energy pricing signals. These systems can autonomously adjust smart home devices, such as thermostats, lighting, and appliances, to minimize energy costs while maintaining comfort levels.

- **Energy Demand Response:** Deep learning models can predict peak energy demand periods and automatically adjust energy usage in response to demand signals or price fluctuations. By participating in demand response programs, homeowners can earn incentives or reduce energy costs while supporting grid stability and reliability.

- **Energy Efficiency Recommendations:** Deep learning algorithms can analyze household energy data and provide personalized recommendations to improve energy efficiency. These recommendations may include upgrading to energy-efficient appliances, adjusting thermostat settings, or implementing energy-saving behaviors, helping homeowners reduce energy consumption and expenses over time [1].

2.6 On household classification for energy balancing: State of the art

Household classification for energy balancing involves categorizing households based on their energy consumption patterns, behaviors, and characteristics. This classification is essential for optimizing energy distribution, demand response, and load management strategies. Several studies have explored different methods for household classification in the context of energy balancing [2] :

- **Clustering Techniques:** Traditional clustering algorithms, such as K-means clustering and hierarchical clustering, have been applied to group households with similar energy consumption profiles. These techniques enable utilities to identify homogeneous clusters of households for targeted energy management interventions.

- **Feature Engineering:** Feature engineering plays a crucial role in household classification, where relevant features such as daily energy consumption, peak demand, occupancy patterns, and demographic information are extracted and used for classification. Feature selection techniques, dimensionality reduction, and data preprocessing methods enhance the effectiveness of classification algorithms.

- **Machine Learning Approaches:** Supervised machine learning algorithms, including decision trees, support vector machines, and random forests, have been employed for household classification tasks. These algorithms utilize labeled training data to predict household classifications based on input features.

- **Hybrid Methods:** Hybrid approaches combining multiple classification techniques, such as clustering with machine learning, have been proposed to improve the accuracy and robustness of household classification models. These methods leverage the strengths of different algorithms to achieve better energy-balancing outcomes.

- **Real-time Monitoring Systems:** Advanced metering infrastructure (AMI) and smart meter data analytics enable real-time monitoring of household energy consumption, facilitating dynamic classification and adaptation to changing energy demand patterns.

2.7 Deep Learning in Domestic Energy: Literature Review

Deep learning techniques have gained popularity in the field of domestic energy for their ability to extract complex patterns from large-scale energy data. Several studies have applied deep learning methods to various aspects of domestic energy management :

Deep Learning for Short-Term Load Forecasting: A Review and Future Directions

Wang et al [3]. conducted a review of deep learning methods specifically focusing on short-term load forecasting. They examined different deep learning architectures, data preprocessing techniques, and evaluation metrics used in short-term load forecasting tasks. The study also discusses future research directions and challenges in this area.

Deep Learning Techniques for Residential Energy Disaggregation: A Review

Li et al [6]. reviewed deep learning techniques applied to residential energy disaggregation, which involves separating total household energy consumption into individual appliance-level consumption. The study discusses various deep learning models, datasets, evaluation metrics, and challenges in residential energy disaggregation.

Deep Reinforcement Learning for Home Energy Management: A Review

Zhang et al [7]. conducted a review of deep reinforcement learning (DRL) techniques applied to home energy management systems. The study discusses how DRL models can optimize energy consumption in smart homes by learning control policies based on environmental conditions, user preferences, and energy pricing signals. It examines various DRL architectures, applications, and challenges in home energy management.

Deep Learning-Based Energy Consumption Prediction in Smart Grids: A Review

Chen et al [5]. conducted a review of deep learning-based techniques for energy consumption prediction in smart grid environments. The study examines different deep learning models, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), and discusses their applications, advantages, and limitations in predicting energy consumption at various temporal and spatial scales.

"Deep Learning-Based Anomaly Detection for Smart Energy Systems: A Review

Wang et al [4]. reviewed deep learning-based anomaly detection methods for smart energy systems. The study explores various deep learning architectures, including autoencoders and recurrent neural networks, and evaluates their effectiveness in detecting abnormal energy consumption patterns, equipment malfunctions, and security threats in smart grid and smart home environments.

Deep Learning Techniques for Smart Home Energy Management: A Comprehensive Review

Liu et al [9]. conducted a comprehensive review of deep learning techniques for smart home energy management. The study covers various aspects of smart home energy management, including load forecasting, energy disaggregation, optimization, and demand response. It examines different deep learning models, datasets, evaluation metrics, and future research directions in this field.

Deep Learning for Smart Grids: A Comprehensive Review

Wu et al [8]. provided a comprehensive review of deep learning applications in smart grid systems. The study examines various aspects of smart grids, including demand response, energy optimization, fault detection, and security, and discusses how deep learning techniques can enhance the efficiency, reliability, and sustainability of smart grid operations.

2.8 Limitations and challenges encountered in the classification of energy homes

Limitations and challenges encountered in the classification of energy homes include [29] :

1. **Data Availability and Quality:** One of the primary challenges is the availability and quality of data. Obtaining comprehensive and accurate energy consumption data from households can be difficult, especially if the smart metering infrastructure is not widely deployed or if data collection methods are inconsistent.

2. **Data Variability:** Energy consumption patterns can vary significantly between households due to factors such as household size, occupancy patterns, lifestyle behaviors, and building characteristics. This variability makes it challenging to develop generalized classification models that accurately represent diverse households.

3. **Feature Selection:** Identifying relevant features or attributes for classification can be complex. While some features, such as daily energy consumption, may be readily available, others, such as occupant behavior or appliance usage patterns, may be more difficult to quantify and incorporate into classification models.

4. **Model Complexity:** Developing accurate classification models requires sophisticated algorithms and computational resources. As the complexity of the model increases, so does the risk of overfitting or model instability, particularly when dealing with limited or noisy data.

5. **Interpretability:** Deep learning models, while powerful, are often considered "black box" models, making it challenging to interpret how they make decisions. Understanding the reasoning behind classification outcomes is crucial for gaining insights and building trust in the classification results.

6. **Scalability:** Scaling classification models to large datasets or multiple households can pose logistical and computational challenges. Efficient algorithms and distributed computing systems may be required to process and analyze vast amounts of energy data in real time.

7. **Privacy and Security:** Collecting and analyzing household energy data raises privacy and security concerns. Ensuring the confidentiality and integrity of sensitive data while still extracting meaningful insights for classification purposes is a complex balancing act.

Conclusion

In conclusion, this chapter offers a comprehensive overview of electrical energy, covering its fundamental concepts, diverse production methods, and evolving consumption trends. Through examining domestic energy consumption and its associated challenges, we have identified critical issues such as peak demand management and the need for efficient energy disaggregation and monitoring methods. The principle of classification in the energy domain provides a framework for categorizing energy-related data, facilitating better understanding and management of energy resources.

Moreover, the integration of deep learning techniques into domestic energy management presents promising avenues for addressing these challenges and optimizing energy usage. By leveraging deep learning algorithms, we can enhance energy monitoring, prediction, and optimization processes, ultimately promoting sustainability and efficiency in energy consumption.

In light of the state-of-the-art research and previous studies using deep learning in domestic energy, it is evident that innovative solutions are essential for overcoming existing limitations and advancing energy management practices. This chapter lays the groundwork for further exploration and innovation in the field of electrical energy management, with the ultimate goal of addressing global energy challenges and fostering sustainable development.

3

Experimental Set Up

Introduction

In this chapter, we lay the foundation for our study by detailing the experimental setup and conducting an extensive exploratory analysis of the dataset. We begin by introducing the dataset used for our research, followed by a comprehensive exploration of energy consumption patterns in households. Through detailed comparative analysis, we uncover insightful observations about appliance energy consumption across different households. Additionally, we discuss the data processing techniques employed to prepare the dataset for model training and testing. This chapter serves as a crucial precursor to our subsequent chapters, providing valuable insights into the dataset and guiding our modeling approach.

3.1 Collection of domestic energy consumption datasets

3.1.1 Experimental Setup

Several open-source data sets are available for the purpose of energy disaggregation. These data were measured in household buildings from different countries. The sensors installed in these buildings read active power, but some sensors also read other information, for example, reactive power, current, and voltage. In NILM (non-intrusive load monitoring), the active power data are used. However, the main difference between the data sets is the sampling frequency. Due to this issue, pre-processing for aligning the readings needs to be done before NILM algorithms are applied to the data. In literature, five appliances are usually considered for disaggregation [30] , which are a kettle, microwave, fridge, dishwasher, and washing machine. In our experiments, three household electricity data sets will be used, which are REFIT [31], UK-DALE [32], and REDD [33].

3.1.2 The REFIT Electrical Load Measurements

The REFIT Electrical Load Measurements (REFIT) dataset is a comprehensive dataset designed to support research in energy consumption analysis and non-intrusive load monitoring (NILM). Collected from UK households, this dataset aims to provide valuable insights into residential energy usage, helping in the development and validation of energy disaggregation algorithms and other smart home technologies [31].

Data Collection : The REFIT dataset was gathered by Loughborough University and comprises detailed energy consumption data from multiple households. The collection methodology includes :

1. **Aggregate Metering :** Capturing the overall electricity usage of each household.
2. **Sub-metering :** Monitoring individual appliances to record their specific power consumption.

Sampling Rates : The REFIT dataset features :

- Low-Frequency Data: Power consumption data recorded at a sampling rate of every 8 seconds, which is suitable for long-term analysis and monitoring of energy use patterns.

Components of REFIT : REFIT encompasses several key elements :

1. Aggregate Power Consumption: Measurements of total household electricity usage over time.
2. Individual Appliance Consumption: Detailed power usage data for specific appliances, facilitating a breakdown of overall consumption.
3. Temporal Coverage: Long-term data collection over several months to capture variations in energy usage across different seasons and living conditions.

3.1.3 The UK Domestic Appliance-Level Electricity

The UK Domestic Appliance-Level Electricity (UK-DALE) dataset is a comprehensive dataset designed for research in energy disaggregation, also known as non-intrusive load monitoring (NILM). Collected from UK households, the UK-DALE dataset provides detailed appliance-level energy consumption data to aid in the development and validation of algorithms for decomposing aggregate energy usage into individual appliance contributions [32].

Data Collection The UK-DALE dataset was compiled by Jack Kelly and William Knottenbelt from Imperial College London. The data collection process involved:

- **Aggregate Metering**: Monitoring the total power consumption of entire households.
- **Sub-metering**: Recording the energy usage of individual appliances using dedicated sensors.

Sampling Rates The dataset features two types of data:

- **High-Frequency Data**: Recorded at 16 kHz for a detailed transient analysis of appliance signatures.
- **Low-Frequency Data**: Recorded at 1/6 Hz (one reading every 6 seconds), suitable for long-term monitoring and analysis.

Components of UK-DALE UK-DALE includes several key components:

- **Aggregate Power Consumption**: Total household energy usage data over time.
- **Individual Appliance Consumption**: Detailed power usage data for specific appliances, providing a granular view of energy consumption.
- **Multiple Houses**: Data collected from multiple households to capture a diverse range of usage patterns and appliance types.
- **Temporal Coverage**: Continuous data collection spanning several years, offering insights into long-term energy usage trends.

3.1.4 The Reference Energy Disaggregation DataSet

The Reference Energy Disaggregation Data Set (REDD) is a dataset specifically designed for the research community focusing on energy disaggregation and smart grid applications. It provides detailed electricity consumption data collected from a set of residential buildings. The primary aim of REDD is to facilitate the development and evaluation of energy disaggregation algorithms, which are methods for breaking down aggregate energy usage into individual appliance contributions [33] .

Data Collection REDD was compiled by MIT researchers and includes data from multiple households. The data collection process involved:

- **Main Metering**: Capturing the total power consumption of the entire house.
- **Sub-metering**: Recording the power usage of individual appliances or circuits within the house.

Sampling Rates The dataset features two types of data:

- **High-Frequency Data**: Recorded at a high sampling rate of approximately 15 kHz for short periods, useful for detailed transient analysis.
- **Low-Frequency Data**: Recorded at a lower sampling rate of 1 Hz for longer periods, suitable for long-term analysis of power usage patterns.

Components of REDD REDD comprises several key components:

- **Aggregate Power Consumption**: The total power consumed by the household over time.
- **Individual Appliance Consumption**: Power usage data for specific appliances or circuits, providing a breakdown of where energy is being used within the household.
- **Ground Truth Data**: Accurate, high-resolution measurements from individual devices serve as a reference for validating disaggregation algorithms.

3.1.5 Challenges

While each dataset—REDD, REFIT, and UK-DALE—provides invaluable resources for research in energy disaggregation and smart home technologies, they also present several challenges that researchers encounter :

Data Volume and Complexity:

- These datasets often contain large volumes of high-resolution data due to their detailed monitoring of energy consumption at both aggregate and appliance levels.
- Managing, storing, and processing such large datasets require robust computational infrastructure and efficient data management techniques.
- Analyzing the data to extract meaningful insights and develop accurate disaggregation algorithms can be computationally intensive and time-consuming.

Disaggregation Accuracy:

- Accurately disaggregating aggregate energy consumption into individual appliance contributions remains a challenging task.
- Appliance usage patterns may overlap, leading to ambiguities in disaggregation results.
- Noise in the data, arising from measurement errors or external factors, can further complicate the disaggregation process.
- Variations in appliance behavior, such as different operating modes and energy signatures, pose additional challenges to achieving high disaggregation accuracy.

Generalization and Transferability:

- Algorithms developed and validated using these datasets may exhibit limitations in generalization and transferability to different households or settings.
- Household-specific characteristics, such as appliance types, usage patterns, and user behaviors, may vary significantly across different regions and demographics.
- Ensuring the robustness and scalability of disaggregation algorithms across diverse contexts is essential for their real-world applicability.

3.2 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their main characteristics, often employing visual methods. Developed by John Tukey in the 1970s, EDA aims to understand the underlying structure, patterns, and relationships within the data before proceeding to more formal statistical modeling or hypothesis testing. It involves techniques for data visualization, data cleaning, and initial hypothesis generation.

Techniques Used in Exploratory Data Analysis:

- **Summary Statistics:** Computing measures such as mean, median, mode, variance, and standard deviation to describe the central tendency, dispersion, and shape of the data distribution.
- **Data Visualization:** Creating graphical representations such as histograms, box plots, scatter plots, and heatmaps to visualize the distribution of variables, identify outliers, and explore relationships between variables.
- **Data Cleaning:** Identifying and handling missing values, outliers, and inconsistencies in the dataset to ensure the reliability and quality of subsequent analyses.
- **Dimensionality Reduction:** Employing techniques like principal component analysis (PCA) or t-distributed stochastic neighbor embedding (t-SNE) to reduce the dimensionality of the dataset and visualize high-dimensional data in lower-dimensional space.
- **Clustering Analysis:** Using clustering algorithms such as k-means or hierarchical clustering to identify natural groupings or clusters within the data.
- **Correlation Analysis:** Calculating correlation coefficients to quantify the strength and direction of relationships between variables.

Benefits of Exploratory Data Analysis:

- **Data Understanding:** EDA helps researchers gain a deeper understanding of the dataset, its characteristics, and potential biases or limitations before performing more advanced analyses.
- **Pattern Discovery:** EDA enables the identification of patterns, trends, and irregularities within the data that may inform subsequent analyses or hypotheses.
- **Insight Generation:** By visually exploring the data, researchers can generate hypotheses and insights that may not be apparent through formal statistical tests alone.
- **Data Quality Assessment:** EDA facilitates the detection and handling of missing values, outliers, and inconsistencies, ensuring the reliability and validity of subsequent analyses.
- **Communication and Visualization:** Visual representations produced during EDA can effectively communicate complex patterns and relationships within the data to stakeholders and decision-makers.

- **Hypothesis Generation:** EDA often leads to the formulation of hypotheses that can be further tested using formal statistical methods, guiding the direction of subsequent research.

3.2.1 Observation of Trends and patterns from figures

Observation of monthly aggregate Electricity consumption of the houses (1-11-15-17) from the dataset including data from 20 households from the Loughborough area over the period 2013 – 2015 Collected under the support of " Engineering and Physical Sciences Research Council (EPSRC)" via the project entitled " Personalized Retrofit Decision Support Tools for UK Homes using Smart Home Technology (REFIT)" Starting from the monthly aggregate electricity consumption of the 4 selected houses (Figure 3.1), we notice that all the households exhibit variability in their energy use. Overall, there are distinct peaks and troughs, suggesting seasonal trends or specific events impacting energy usage.

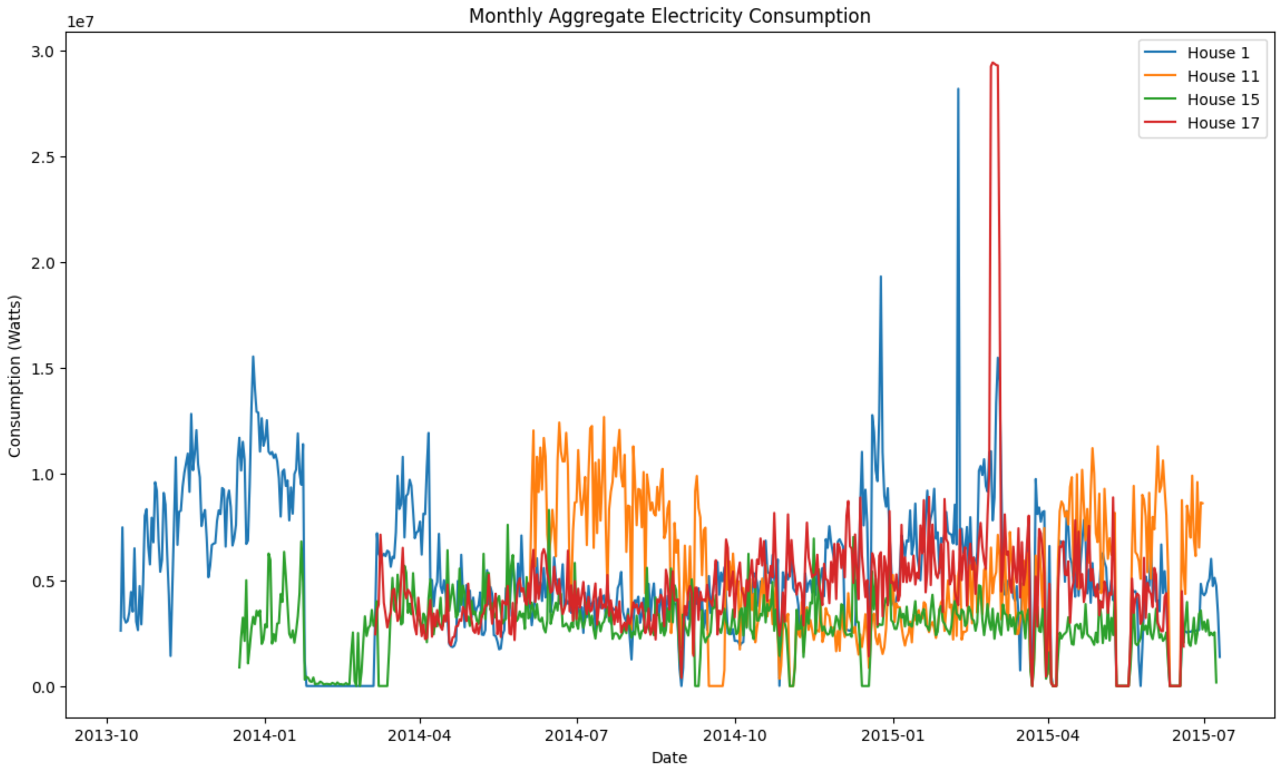

Figure 3.1: Monthly Aggregate Electricity Consumption

House 1 According to the daily and weekly consumption (Figures 3.2-3.3), there is an emphasis on power usage from October to January and an important rise in consumption from January to April. A specific inactivity period is registered for over a month. We can conclude that House 1 witnesses high aggregate electricity consumption mostly in winter.

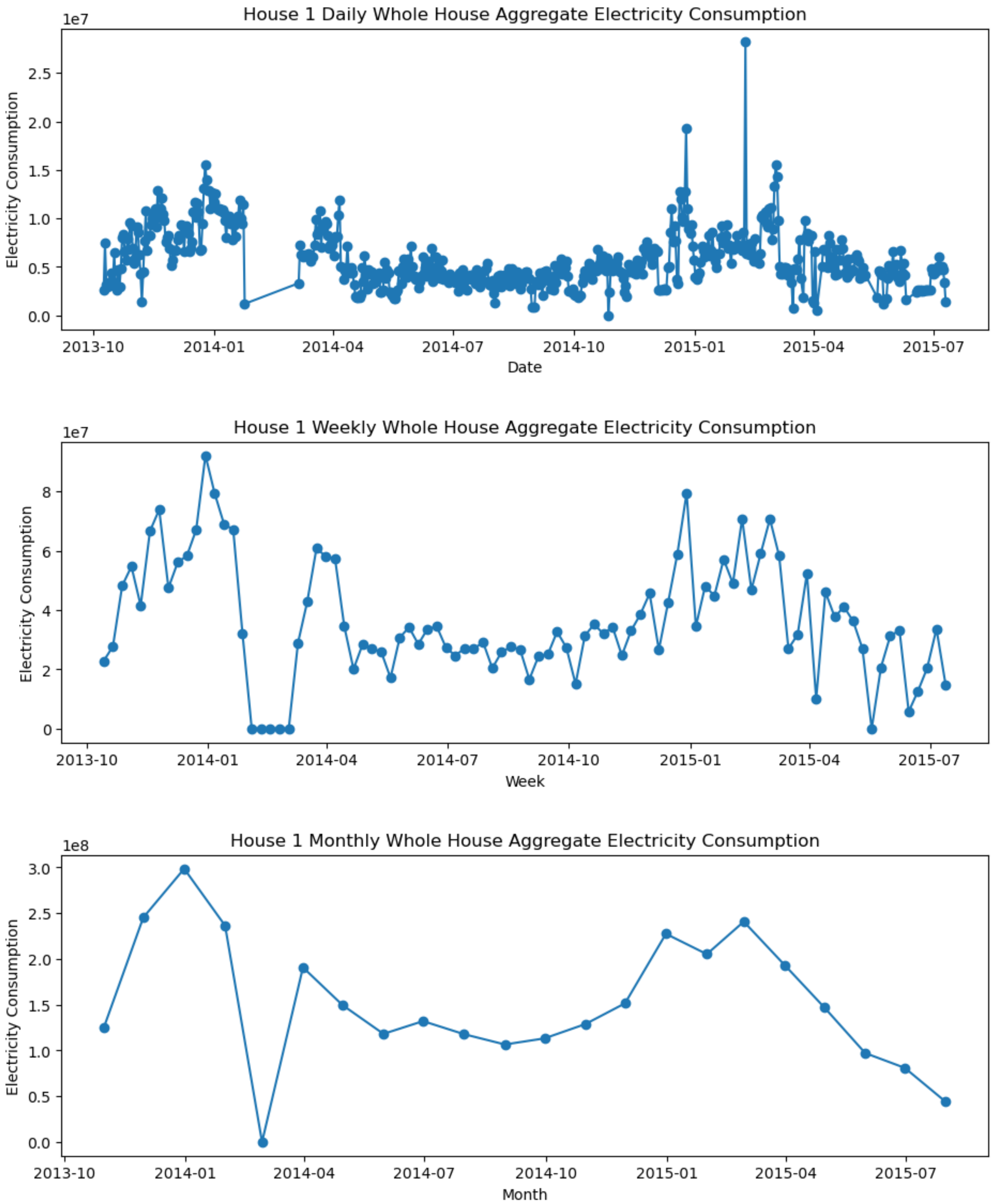

Figure 3.2: House 1 Whole House Aggregate Electricity Consumption

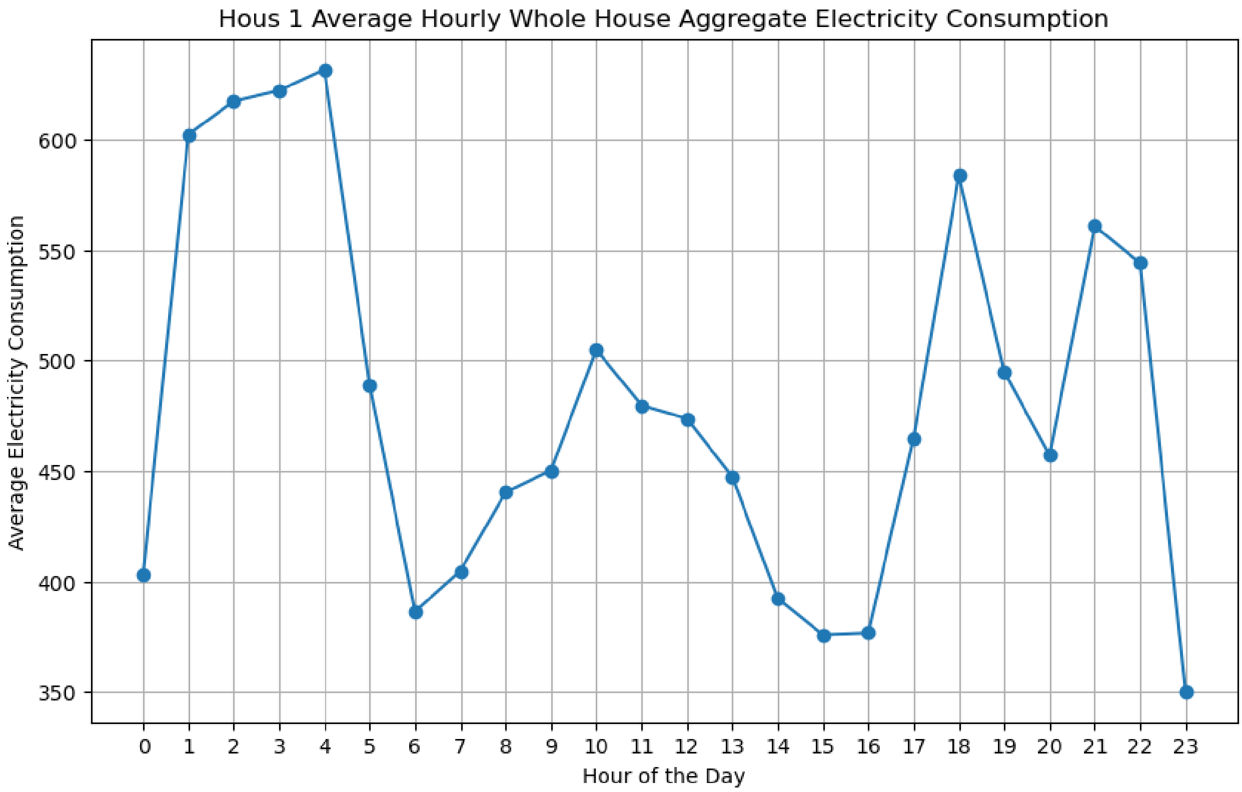

Figure 3.3: House 1 Average Hourly Whole House Aggregate Electricity Consumption

On a weekly basis (Figures 3.4), this average is stable on weekdays and shows a drop in consumption specifically on Friday, and the highest level is reached by the end of the weekend. From the hourly average consumption, we deduce that spikes are situated from 1 am till 4 am, and from 9 pm to 10 pm. This could be explained by the household habits and appliance use routines (Excessive use of overhead fan from midnight to 6 am, specific appliance use like kettle, Hi-fi, Toaster, and microwave from 1 to 4 am, and from 8 till 5 pm).

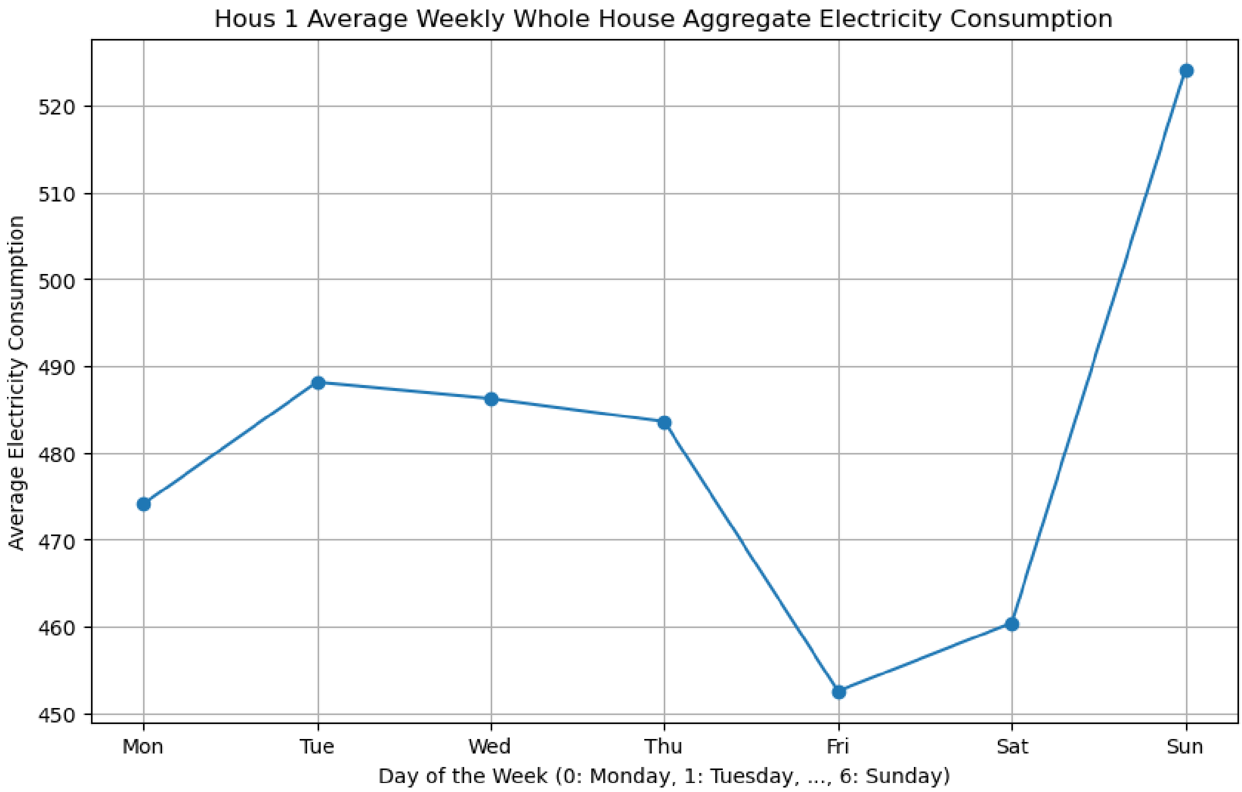

Figure 3.4: House 1 Average Weekly Whole House Aggregate Electricity Consumption

Appliances Consumption

- **Fridge-Freezer**: In the graph, the Fridge-Freezer's energy usage appears consistent throughout the day.
- **Washing Machine**: The Washing Machine shows spikes during specific hours. It's essential to check if these spikes align with typical laundry times.
- **Dishwasher**: The Dishwasher's energy use is relatively low but noticeable.
- **Television Set**: The TV's consumption varies, likely corresponding to viewing hours.
- **Microwave, Toaster, Hi-Fi, Kettle, and Overhead Fan**: These appliances contribute to the overall energy consumption.

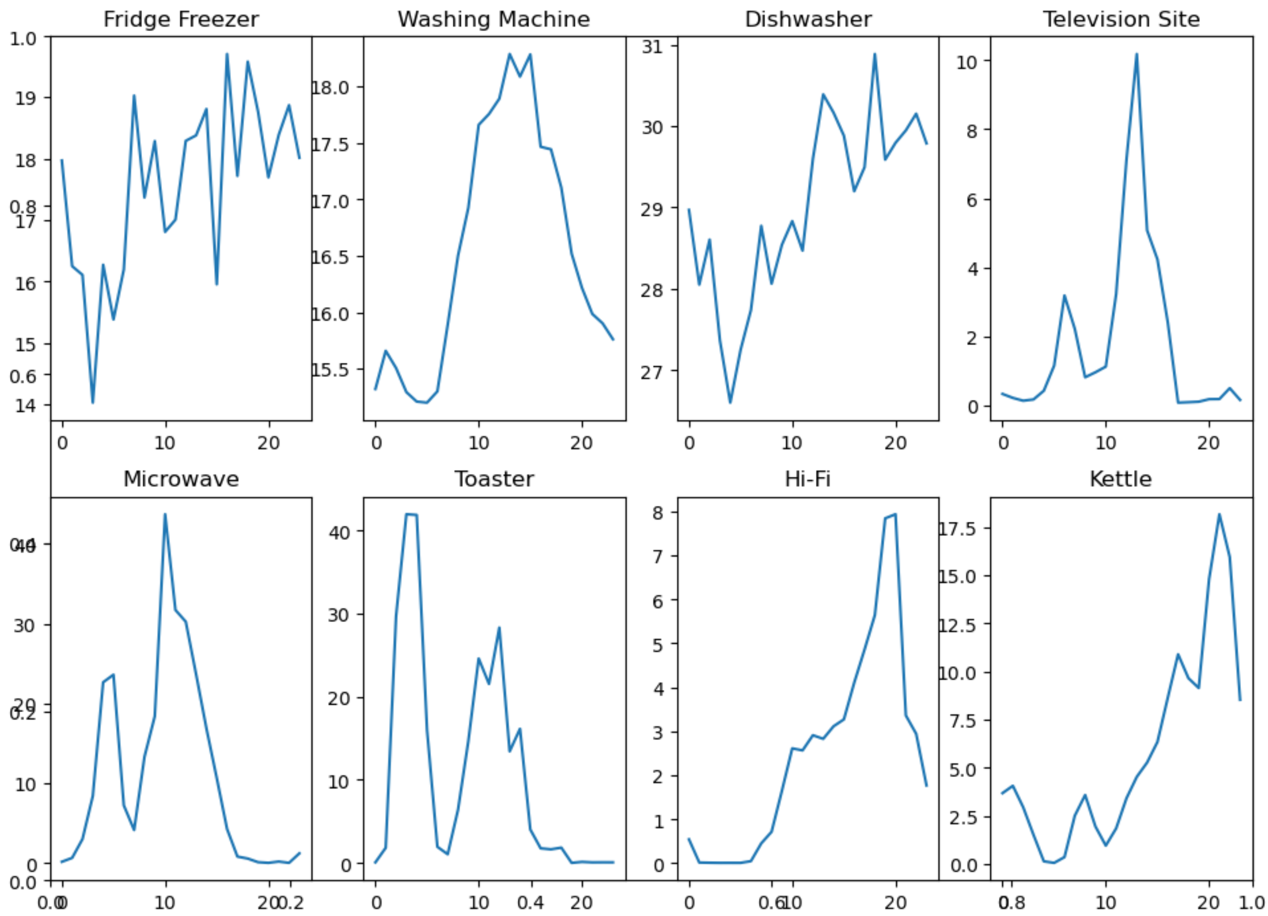

Figure 3.5: House 1 Appliances Consumption

House 11 The observation of the aggregate electricity consumption over time (daily, weekly, monthly) of the second household (Figures 3.6-3.8) shows specific summer peaks, from July to September, with a flagrant decrease in power use from September to March.

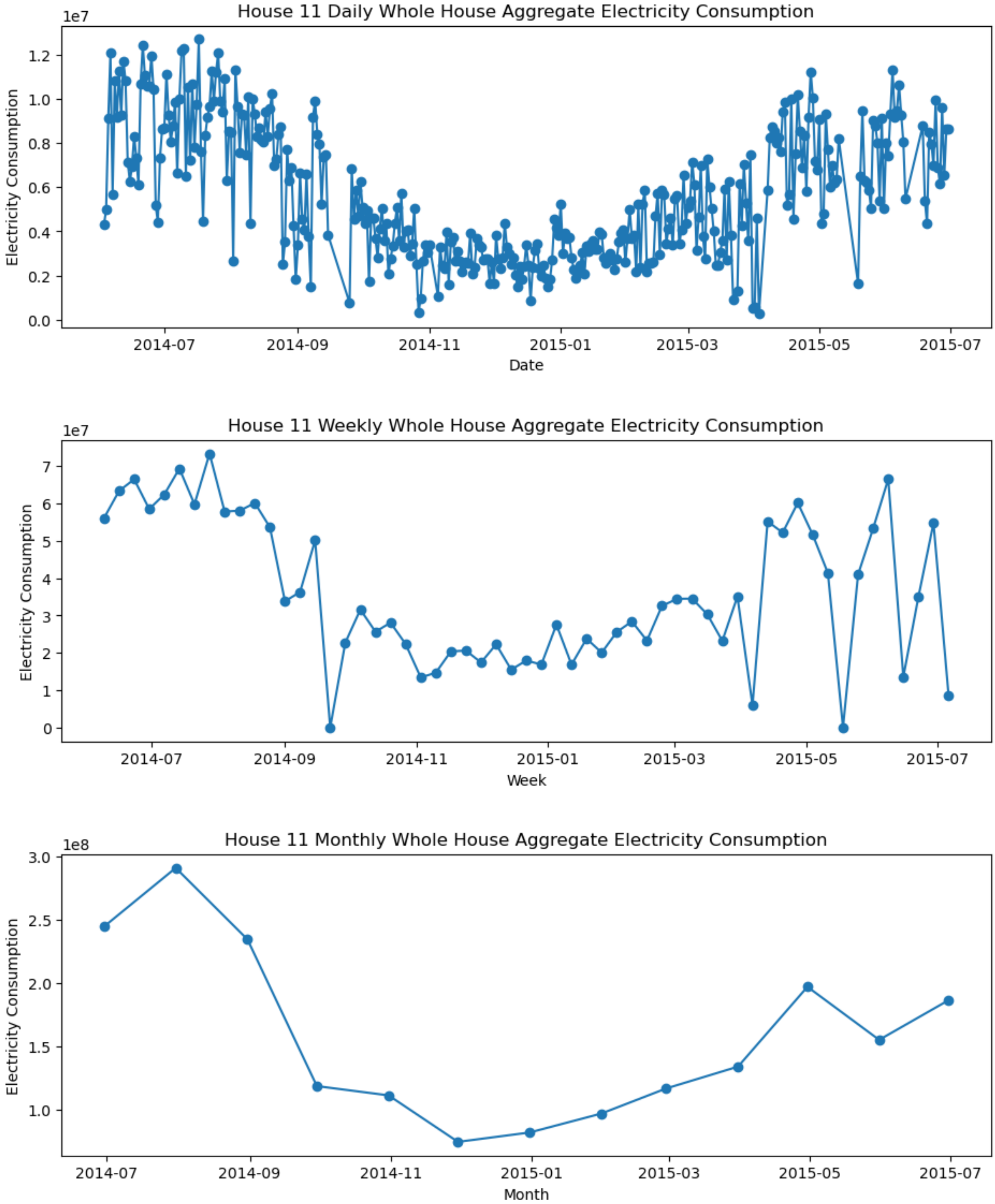

Figure 3.6: House 11 Whole House Aggregate Electricity Consumption

The consumption is witnessed highly at the beginning of the week, with a special peak on Tuesday, while the consumption of electricity is getting lower for the rest of the weekdays until reaching its minimum on Saturday. (Figure 3.7)

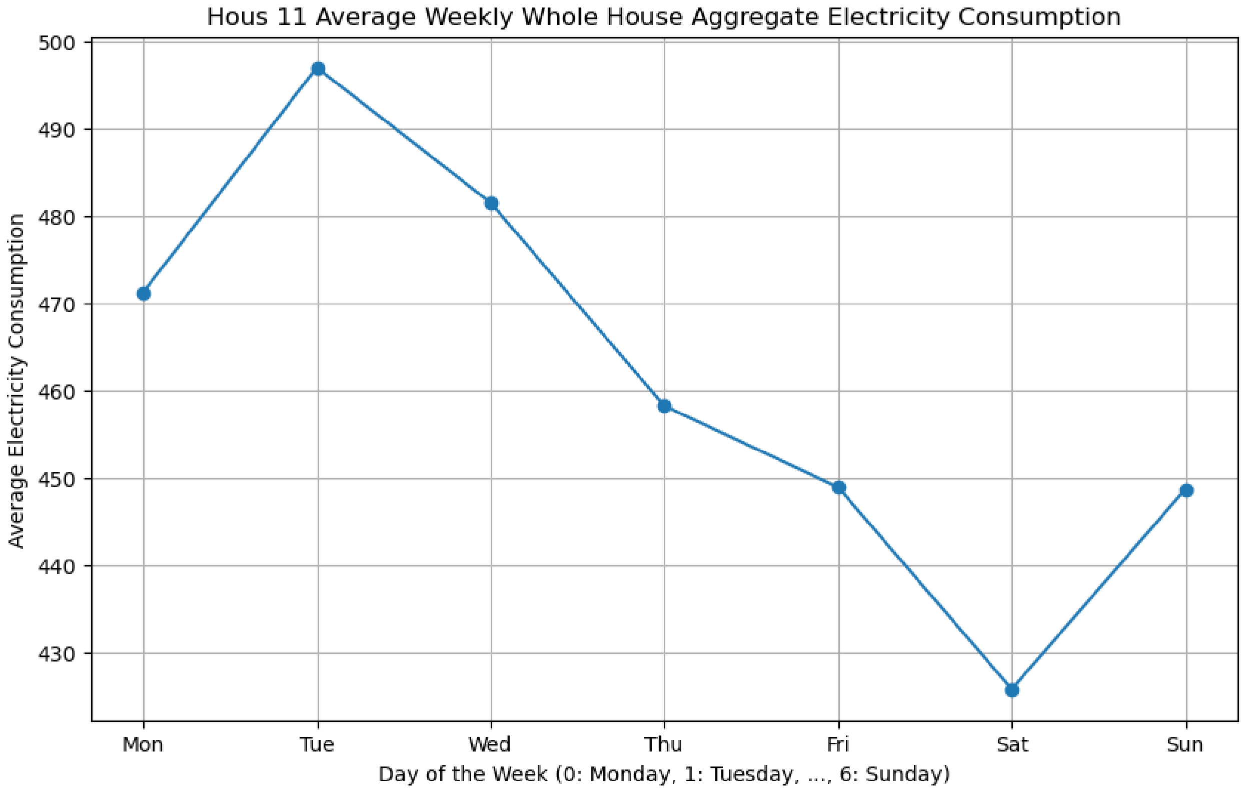

Figure 3.7: House 11 Average Weekly Whole House Aggregate Electricity Consumption

The average hourly whole house aggregate electricity consumption gives an interesting exponential growth from 6 am till midday, and an exponential drop follows directly from midday to 6. The consumption peaks during the day are explained by the intensive use of precise types of energy-consuming appliances (Dishwasher, Kettle, microwave, Hi-Fi, and router) at those specific time loops with an accentuation around midnight.

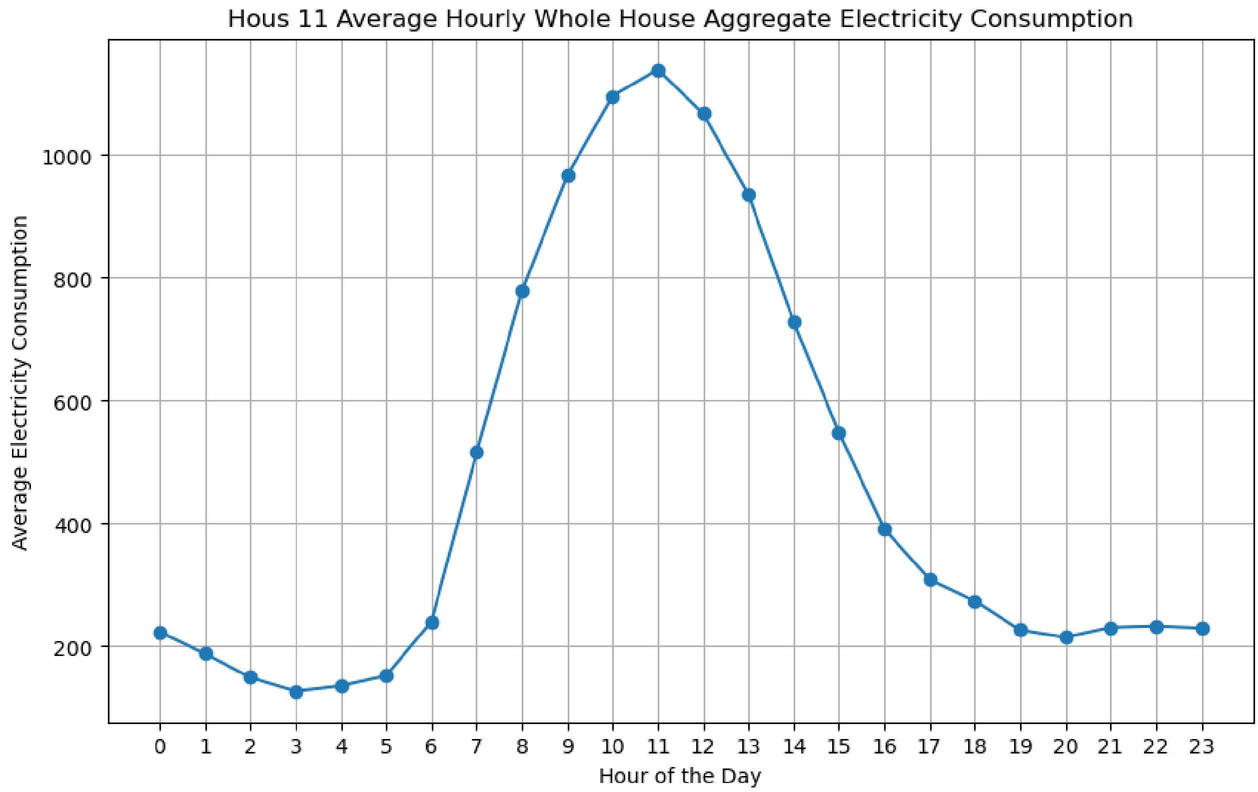

Figure 3.8: House 11 Average Hourly Whole House Aggregate Electricity Consumption

Appliances Consumption :

- **Fridge and Fridge-Freezer**: These appliances likely run continuously or intermittently. Their impact is relatively stable throughout the day.
- **Washing Machine and Dishwasher**: Spikes during laundry and dishwashing cycles.
- **Computer Site**: Specific usage patterns (e.g., work hours, leisure). Computers contribute to energy consumption.
- **Microwave and Kettle**: Peaks may occur during meal preparation. Short but intense energy bursts.
- **Hi-Fi**: Check for usage during leisure hours. Music systems can impact energy demand.

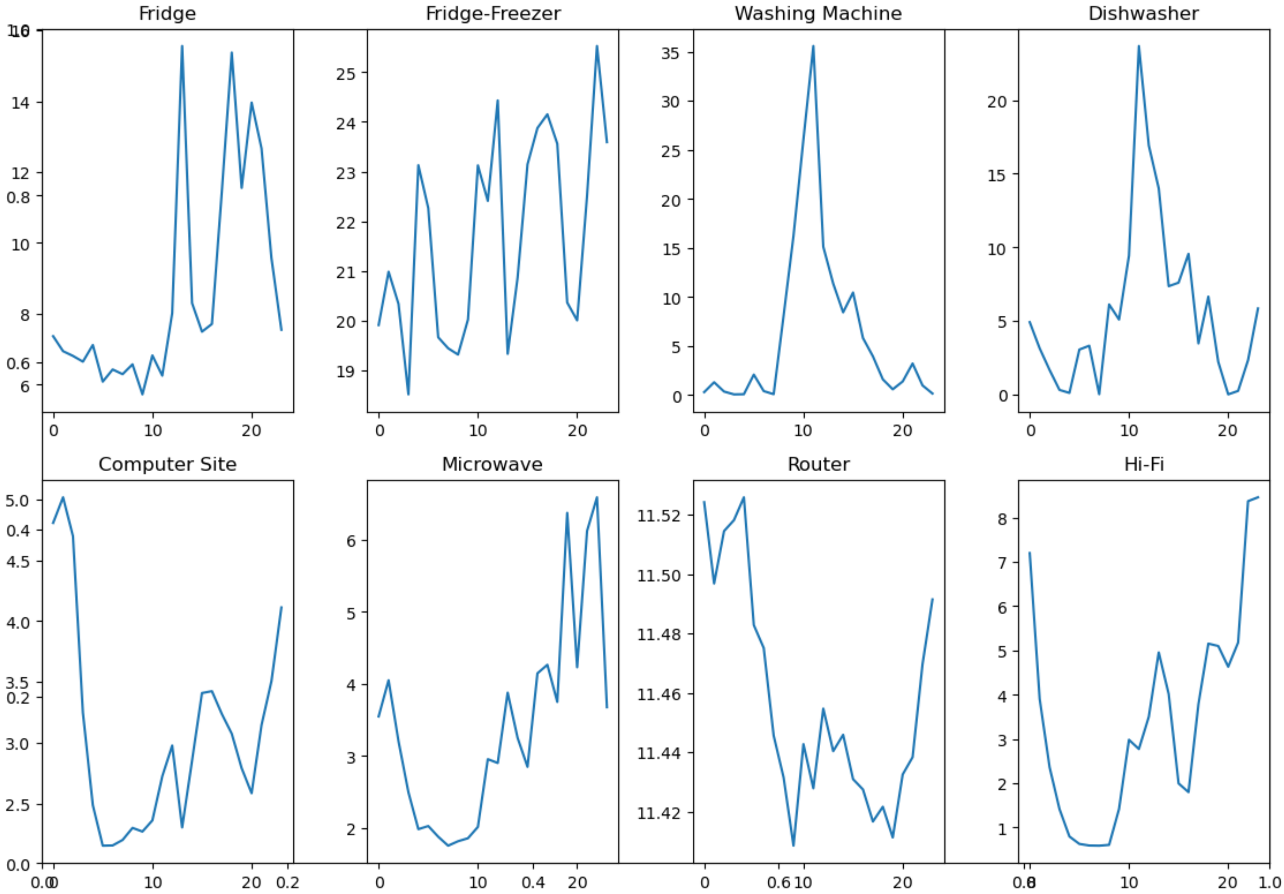

Figure 3.9: House 11 Appliances Consumption

House 15 The aggregate electricity consumption is approximately regular over the year from Mai 2014. We can denote an absence of consumption in January and June.

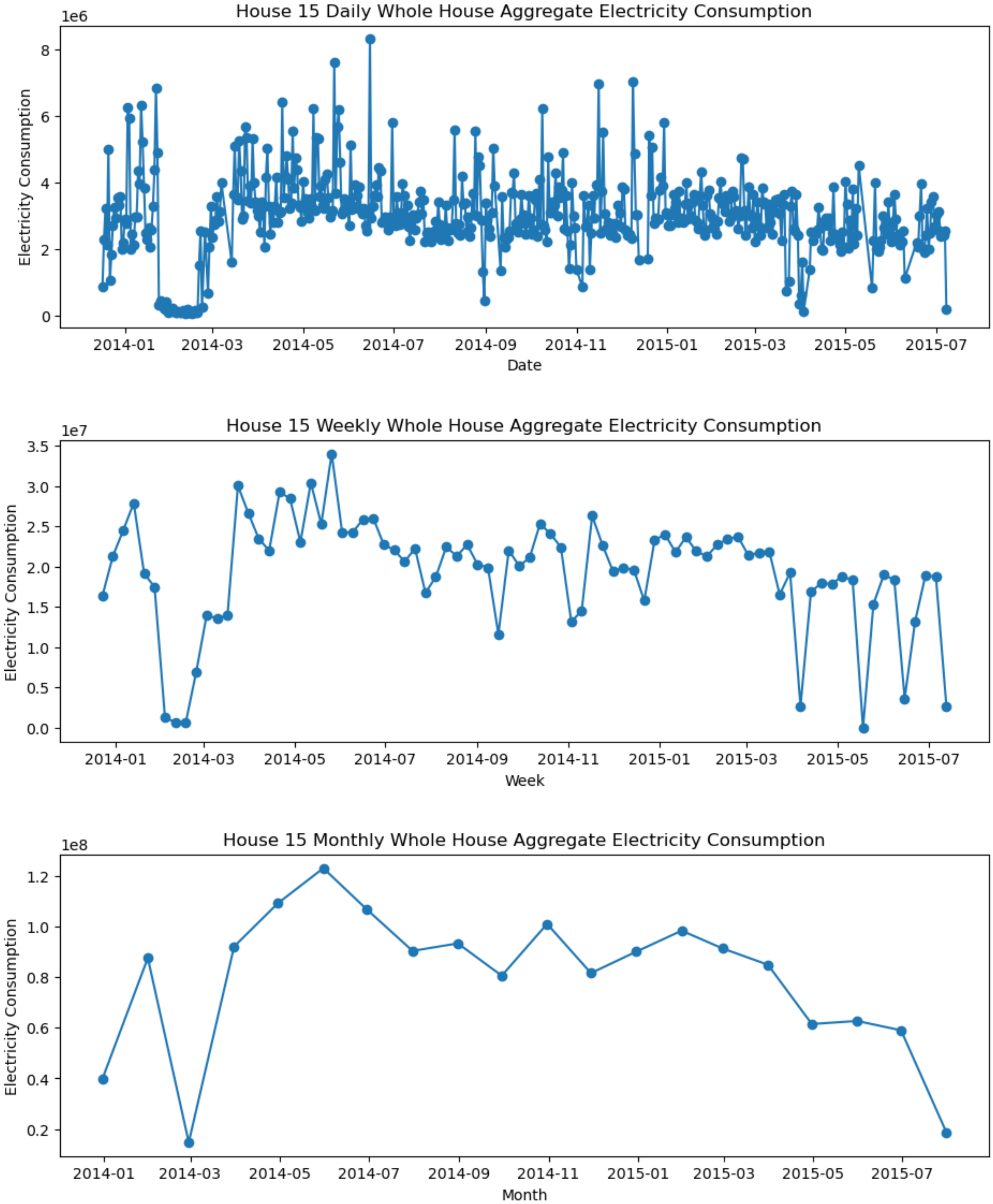

Figure 3.10: House 15 Whole House Aggregate Electricity Consumption

If we try to observe and analyze clearly the details of power use in the third household, starting from understanding the distribution of consumption along the day, we notice a gradual increase in the average hourly whole house aggregate electricity consumption. A peak appears from 6 to 8 pm directly continuing with a decrease of energy consumption until 11 pm. The consumption peaks, or the highest values during the daily period are due to the extra use of some appliances (washing machine, television site, and kettle).

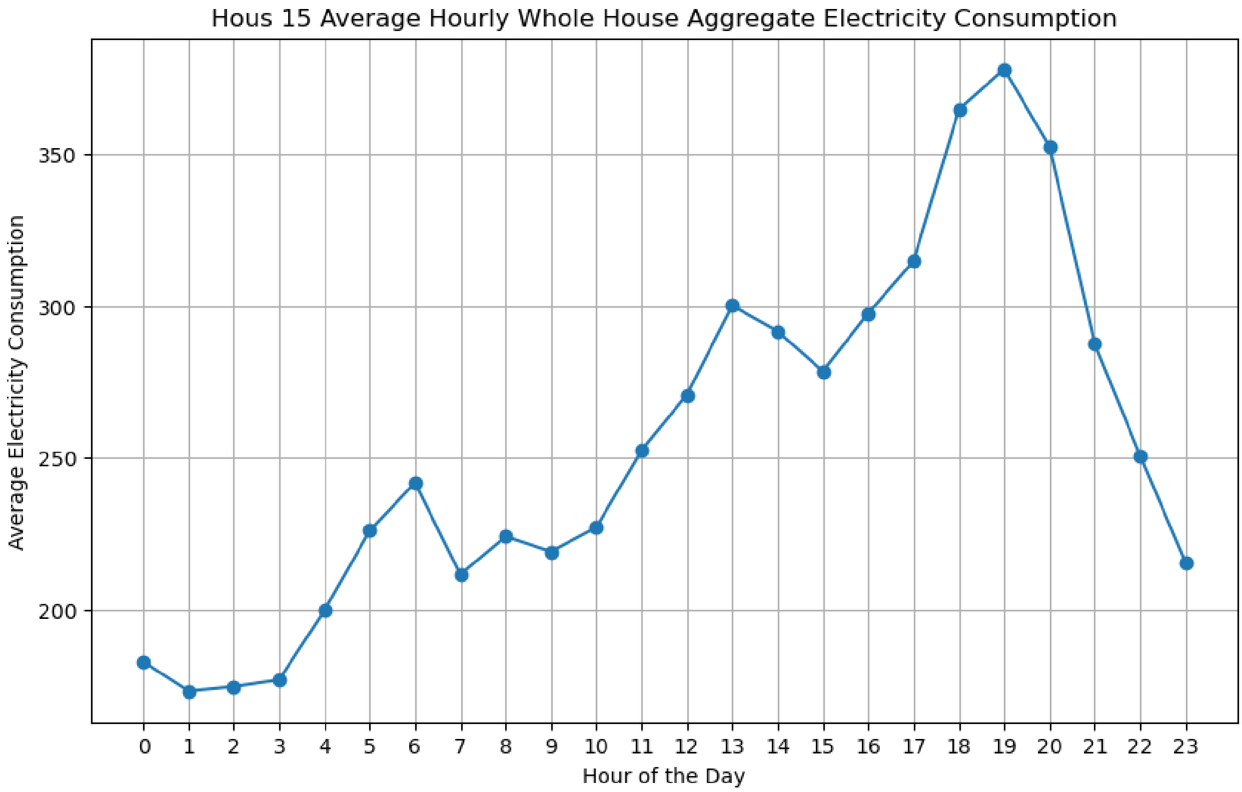

Figure 3.11: House 15 Average Hourly Whole House Aggregate Electricity Consumption

This consumption is trending on Monday, Thursday, and Saturday. Nevertheless, there is a considerable fall in the consumption level on Tuesday and Friday as some kind of rest period.

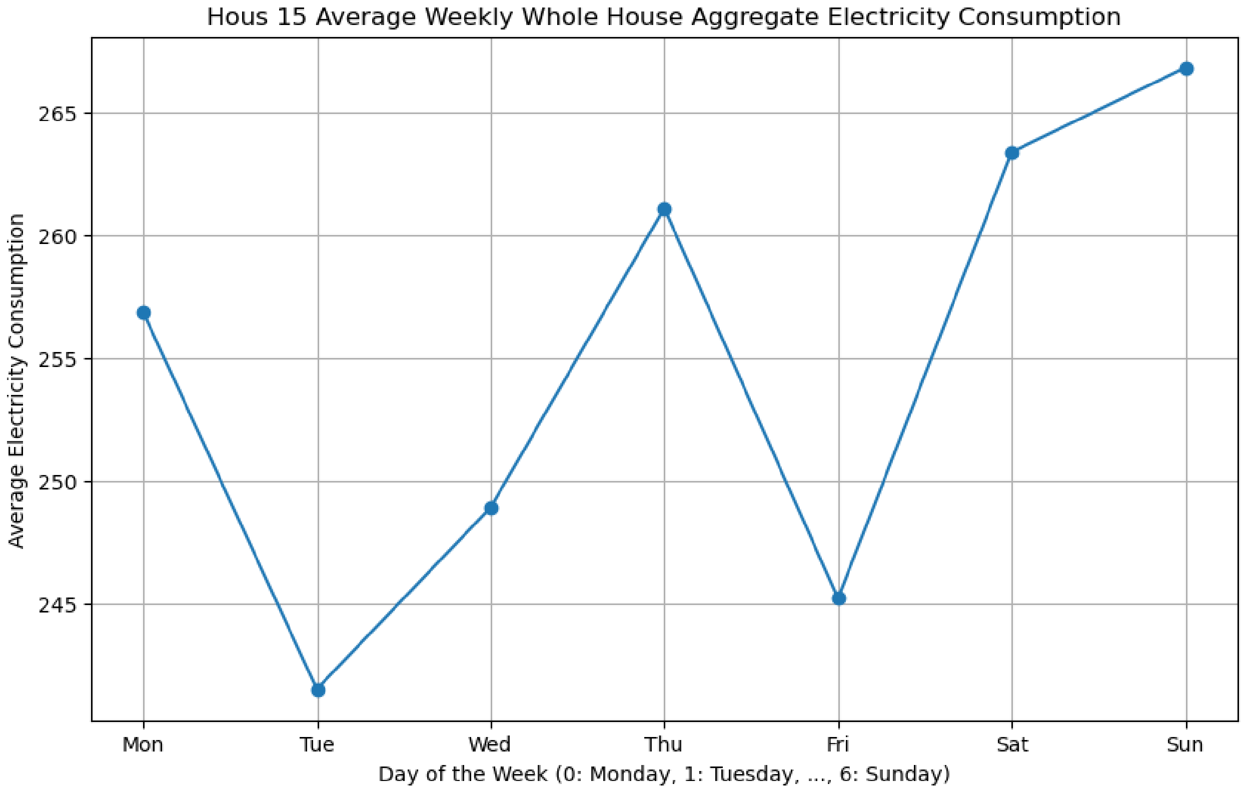

Figure 3.12: House 15 Average Weekly Whole House Aggregate Electricity Consumption

Appliances consumption

- **Fridge-Freezer**: Consistent energy usage throughout the day. Likely runs continuously or intermittently.
- **Tumble Dryer**: Peaks during specific hours (e.g., laundry cycles). A significant contributor to energy demand.
- **Washing Machine**: Similar to the tumble dryer, with usage spikes. Laundry activities drive energy consumption.
- **Dishwasher**: Peaks during dishwashing cycles. Consider optimizing usage times.
- **Microwave, Television, and Computer Sites**: Smaller fluctuations but still contribute. Monitor usage patterns for efficiency.
- **Kettle and Toaster**: Short bursts of high energy use. Occur during meal preparation.

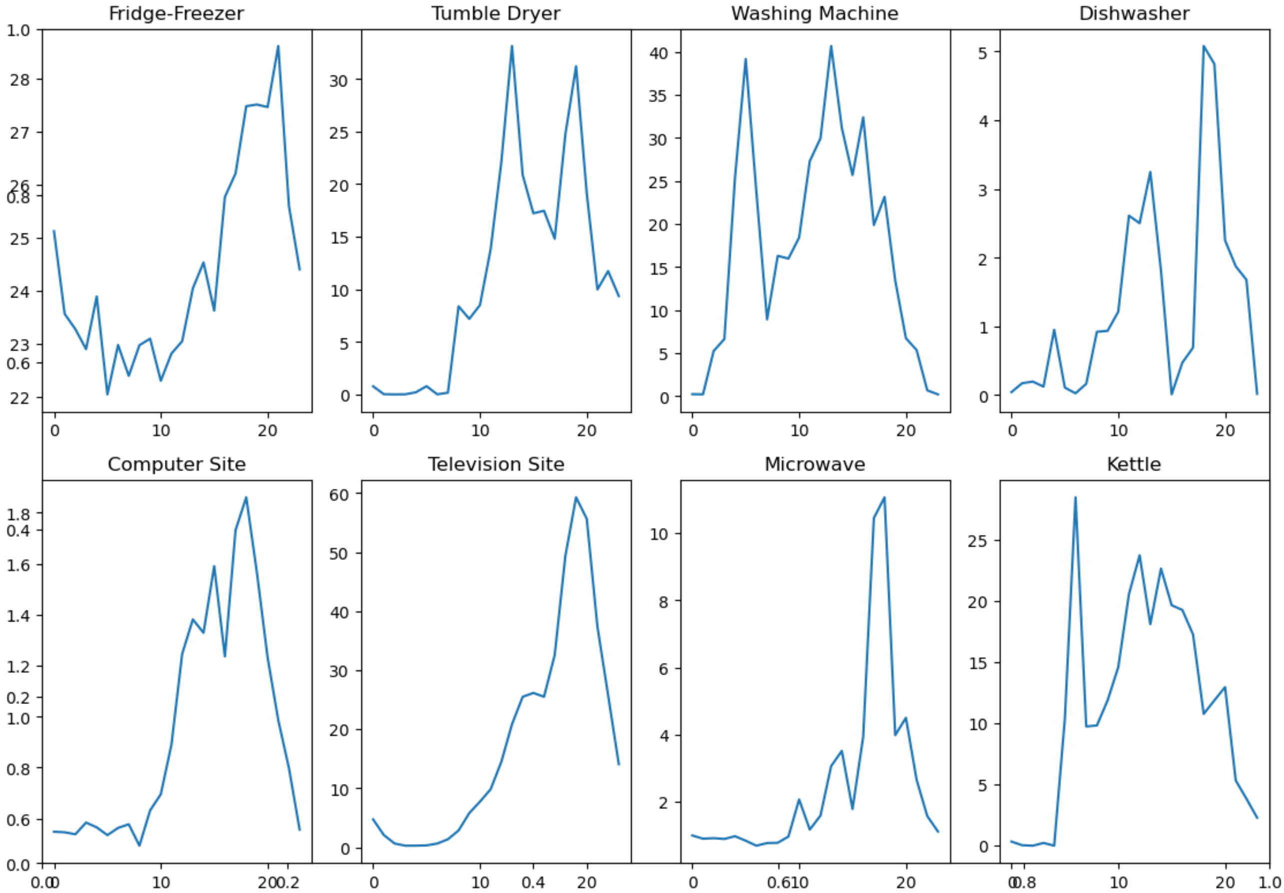

Figure 3.13: House 15 Appliances Consumption

House 17 The energy consumption for the last household is mainly stable, with a peak in March 2015.

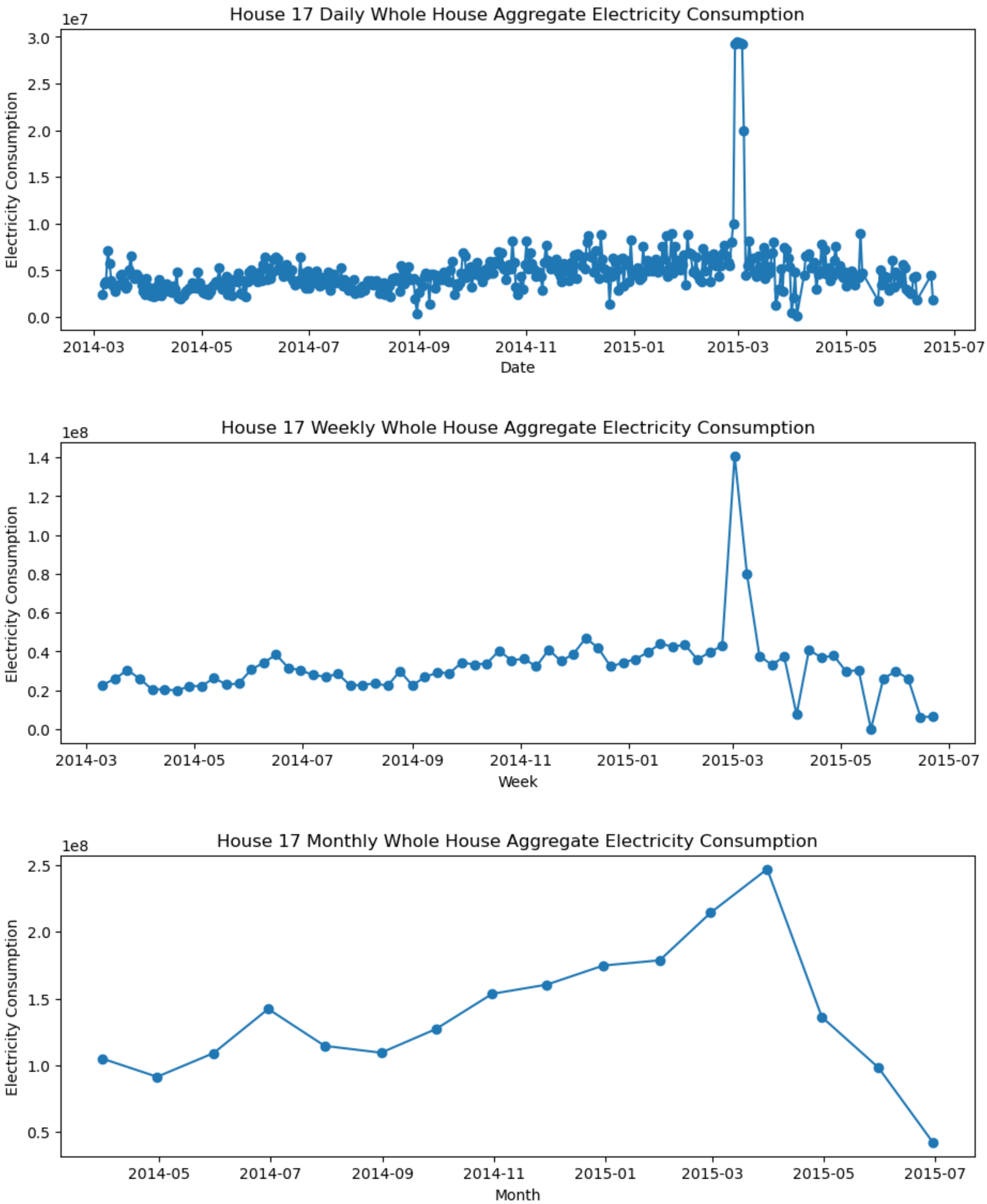

Figure 3.14: House 17 Whole House Aggregate Electricity Consumption

The average hourly whole house aggregate electricity consumption stays low during the night and increases around 8-9 am to register a peak of power usage at 6 pm. The daily important patterns followed by the average consumption growth are related to the intense use of specific appliances during the daily hours, from 7 am until 8 pm. We can cite the kettle, microwave, TV site, and computer site).

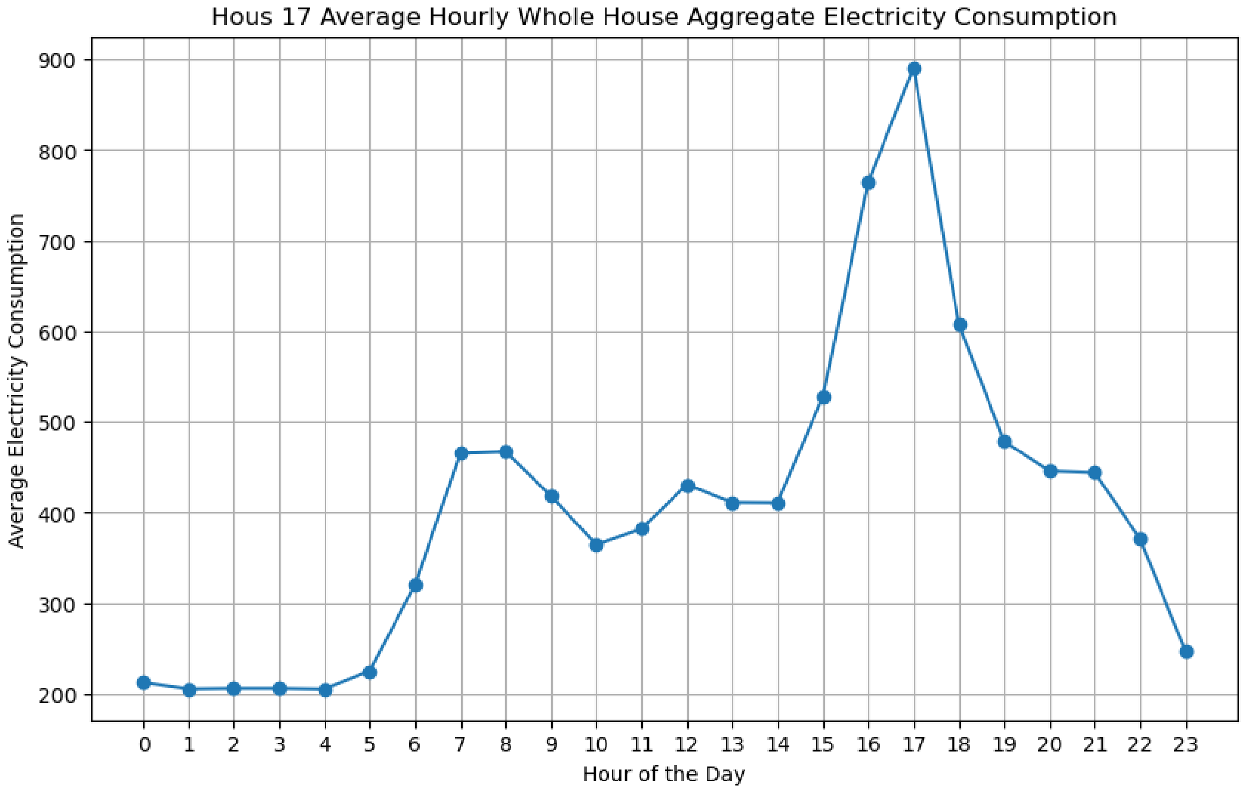

Figure 3.15: House 17 Average Hourly Whole House Aggregate Electricity Consumption

Weekly, the peaks of power consumption are seen during the weekends from Friday to Sunday with the highest peak on Saturday.

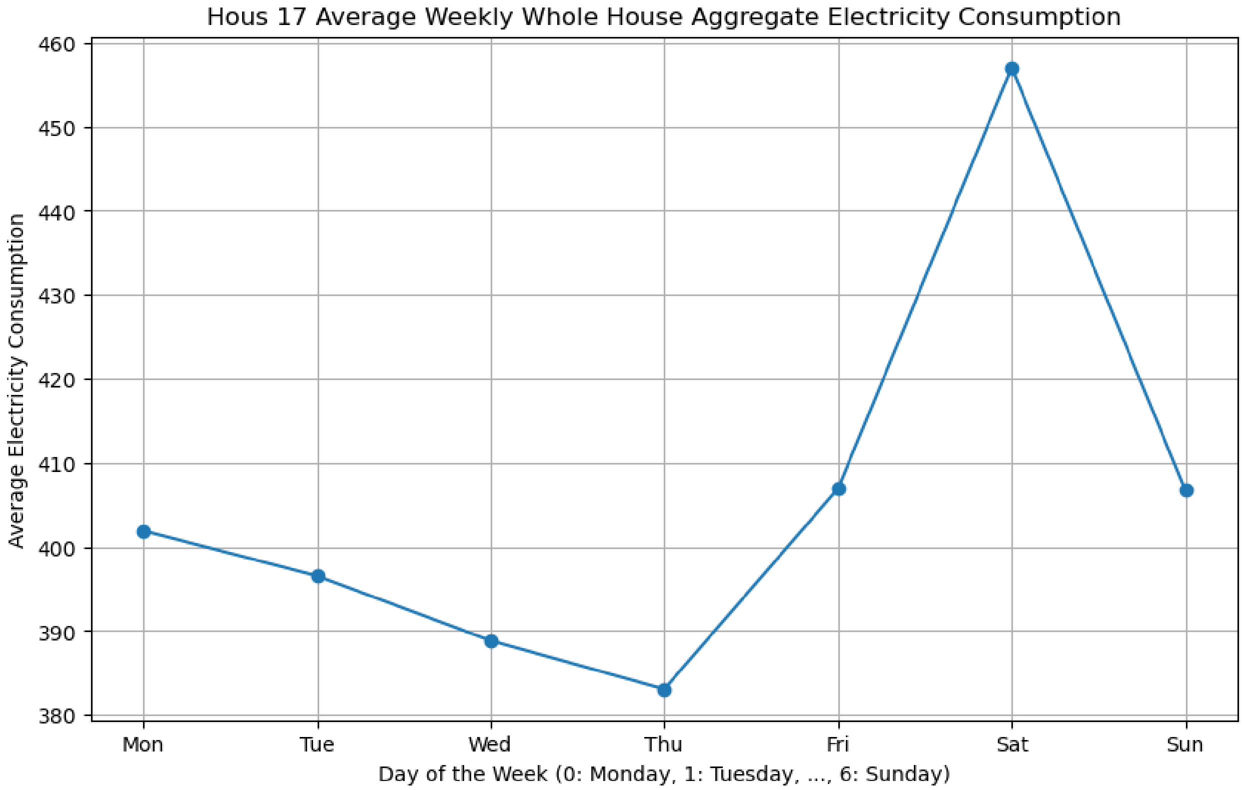

Figure 3.16: House 17 Average Weekly Whole House Aggregate Electricity Consumption

Appliances consumption

- **Freezer (Garage)**: Consistent energy usage throughout the day.
- **Fridge-Freezer**: Similar to the freezer, with stable usage. Refrigeration appliances contribute consistently.
- **Tumble Dryer (Garage)**: Peaks during specific hours (e.g., laundry cycles).
- **Washing Machine**: Similar to the tumble dryer, with usage spikes.
- **Microwave, Television, and Computer Sites**: Smaller fluctuations but still contribute.
- **Kettle and Plug Site (Bedroom)**: Short bursts of high energy use. Occur during meal preparation and device charging.

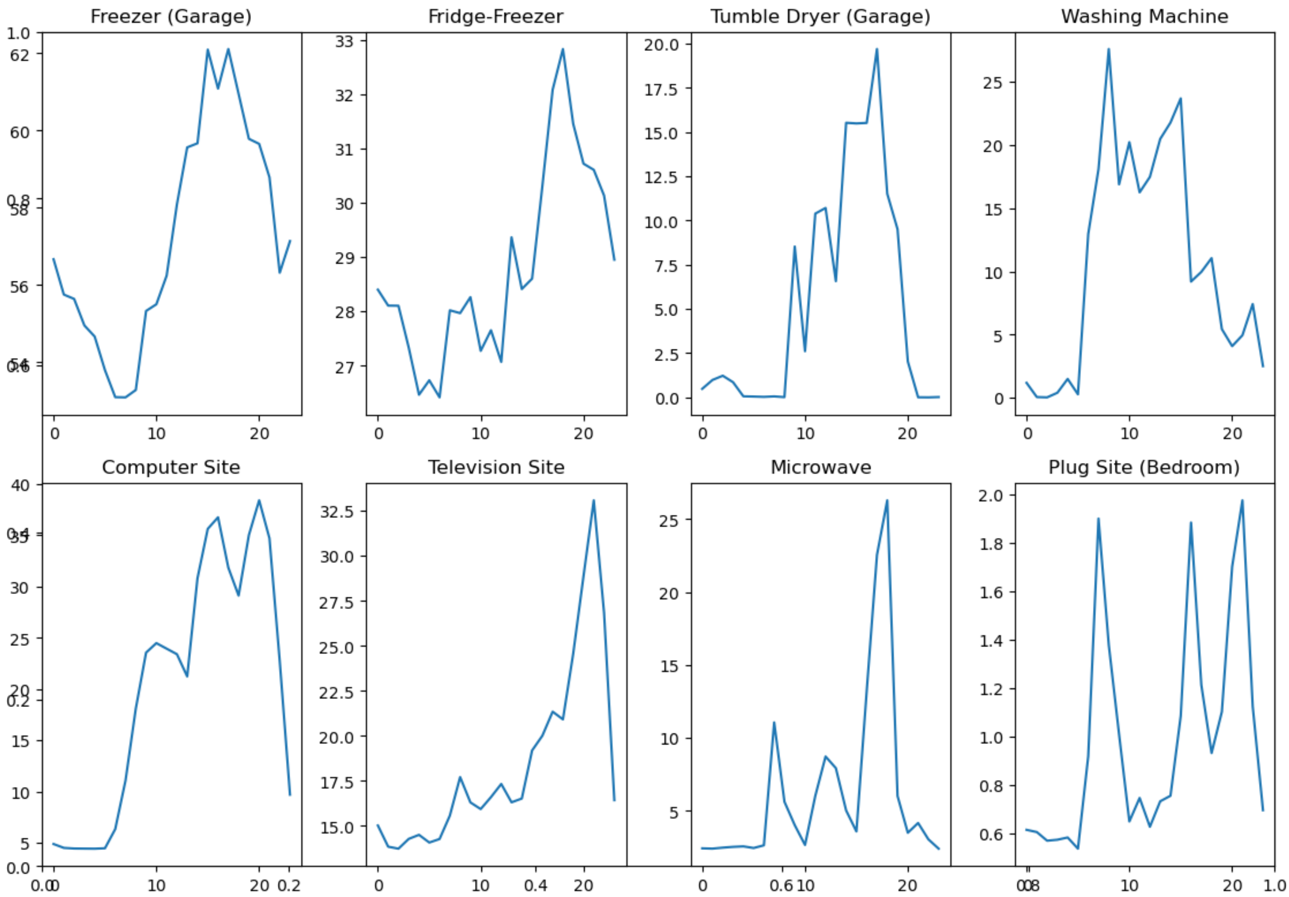

Figure 3.17: House 17 Appliances Consumption

- **Household consumption comparison**:
  1. **House 11**:
     - Consistent usage with minor fluctuations.
     - No extreme spikes.
     - Likely follows a stable routine.
  2. **House 1**:
     - Similar to House 11 but with slightly higher consumption.
     - Steady pattern without significant variations.
     - Efficient energy management.
  3. **House 15**:
     - Weekly variability with peaks on weekends.
     - Sundays consistently have the highest consumption.
     - Investigate reasons for weekend spikes.
  4. **House 17**:
     - Similar to House 15 in weekly trends.
     - Significant spike around the end of 2014.
     - Investigate reasons for this peak.

Overall Conclusions All households exhibit some level of consistency in their energy consumption. Weekdays generally have lower usage, while weekends show increased demand. Investigate specific events or behaviors causing spikes (e.g., laundry, cooking, and entertainment). After analyzing the respective observations on trends and patterns adopted by each of the households we can conclude that the average energy usage is directly impacted by peak hours depending on the household's habits and routines. The energy usage pattern changes considerably from weekdays to weekends. Seasonal variations are highlighted in the 4 houses' monthly average consumption during the whole observation period. Finally, appliance-specific patterns are the most visible in terms of energy consumption levels if we proceed to comparisons. To sum up, the usage of certain appliances contributes to distinct usage patterns. There are proper energy consumption profiles for each type of appliance (Heating, cooling, lighting, and cooking).

3.3 Data Processing

This section provides a comprehensive overview of the data processing steps involved in preparing the dataset for training a neural network aimed at energy disaggregation and household classification. The code for data processing has been implemented in the DatasetGenerator class, which handles the creation of training, testing, and validation datasets.

Purpose Data processing is a critical step to ensure that the raw data collected from households is clean, formatted, and suitable for training the neural network. Proper preprocessing helps in reducing noise, handling missing values, and normalizing the data for better model performance.

Steps

1. **Data Cleaning**: Removing any inconsistent or erroneous entries.
2. **Normalization**: Scaling the data to a standard range, typically [0, 1], helps in accelerating the training process and improving model convergence.
3. **Segmentation**: Dividing the continuous data into manageable segments to facilitate better learning of consumption patterns.
4. **Feature Extraction**: Extracting relevant features such as power consumption, appliance usage patterns, etc., which are essential for training the model.

Benefits

- Ensures data quality and consistency.
- Reduces the dimensionality of the data, making it easier to handle.
- Improves the model's ability to learn and generalize from the data.

3.3.1 DatasetGenerator Class

The `DatasetGenerator` class is designed to preprocess data from the REFIT dataset, specifically targeting appliance-level data. It includes methods for loading data, normalizing it, and saving the processed data into specific files for training, testing, and validation.

Parameters

- __appliance (`string`): The name of the target appliance. This determines which appliance's data will be processed.
- __directory (`string`): The location of the REFIT dataset. This is the directory where the raw data files are stored.
- __agg_mean (`int`): The mean value used to normalize the aggregate power data. Normalization helps in standardizing the data.
- __agg_std (`int`): The standard deviation used to normalize the aggregate power data.
- __training_set_length (`int`): The total number of rows in the appliance's training dataset. This is used to keep track of the amount of data processed.

Methods

- `__init__(self, appliance_name="kettle")`: Initializes the `DatasetGenerator` class with default values for the specified appliance. Sets default values for the directory, aggregate mean, and standard deviation.
- `digits_in_file_name(self, file_name)`: Extracts and returns the digits from a given file name. Useful for identifying house numbers in the dataset files.
- `load_file(self, house, channel)`: Loads data for a specified house and channel from the REFIT dataset. Returns the data as a pandas DataFrame.
  - `Parameters:`
    - `house (int)`: The house number.
    - `channel (int)`: The column from which data should be extracted.
- `generate_test_house(self)`: Normalizes the aggregate and appliance data for a specified house to be part of the testing set. Writes the normalized data to a testing file.
- `generate_validation_house(self)`: Normalizes the aggregate and appliance data for a specified house to be part of the validation set. Writes the normalized data to a validation file.
- `generate_train_house(self, file_name)`: Normalizes the aggregate and appliance data for a specified house to be part of the training set. Writes the normalized data to a training file.
  - `Parameters:`

    ∗ `file_name (string)`: The name of the file to be processed.

- `generate(self)`: Main method to generate normalized training, validation, and testing datasets from the REFIT dataset. It loops through the files in the specified directory, processes them, and saves the results. It also creates the necessary directories if they do not exist and prints the duration taken to generate the datasets.

Benefits

- **Automated Data Processing**: The `DatasetGenerator` class automates the process of loading, normalizing, and saving the data, which is essential for preparing large datasets.
- **Normalization**: By normalizing the data, the model training process becomes more efficient and stable. This helps in achieving faster convergence and better performance.
- **Separation of Data**: The class separates data into training, validation, and testing sets, which is crucial for building robust models and evaluating their performance accurately.
- **Error Handling**: The class includes basic error handling to skip files that cannot be processed, ensuring that the dataset generation process continues smoothly.

Example Usage On REFIT To use the `DatasetGenerator` class, instantiate it and call the `generate` method. This will process the dataset and generate the required files.

```
dsg = DatasetGenerator()
dsg.generate()
```

This example [34] initializes the dataset generator with default values and generates the datasets. We can customize the initialization parameters if we need to process data for a different appliance or use a different directory.

Appliance	**Mean**	**Standard Deviation**
Aggregate	522	814
Kettle	700	1000
Microwave	500	800
Fridge	200	400
Dishwasher	700	1000
Washing machine	400	700

Table 3.1: Parameters for normalising the data

3.3.2 Distribution of the REFIT dataset after the data processing

Appliance	Training			Validation			Test		
	house	samples (M)	time (Y)	house	samples (M)	time (Y)	house	samples (M)	time (Y)
Kettle	3, 4, 6, 7, 8, 9, 12, 13, 19, 20	59.19	15	5	7.43	1.9	2	5.73	1.5
Microwave	10, 12, 19	18.22	4.6	17	5.43	1.4	4	6.76	1.7
Fridge	2, 5, 9	19.33	4.9	12	5.86	1.5	15	6.23	1.6
Dish w.	5, 7, 9, 13, 16	30.82	9.8	18	5	1.3	20	5.17	1.3
Washing m.	2, 5, 7, 9, 15, 16, 17	43.47	11	18	5	1.3	8	6	1.5

Table 3.2: Distribution of the REFIT dataset. M: Millions; Y: Years. After the data processing

Summary The data processing step is crucial in any machine learning project.
The `DatasetGenerator` class provides a structured and efficient way to preprocess the REFIT dataset, ensuring that the data is clean, normalized, and properly formatted for training, validation, and testing. By automating these tasks, the class significantly reduces the manual effort required and ensures consistency in data preparation.

Conclusion

In conclusion, Chapter III serves as a pivotal stage in our research journey, providing crucial insights into household energy consumption dynamics. Through meticulous experimental setup, exploratory analysis, and comparative assessment, we gain a nuanced understanding of the dataset and its underlying patterns. Our findings lay the groundwork for subsequent chapters, guiding our modeling efforts and shaping our insights into energy disaggregation and consumption balance. Chapter III underscores the significance of thorough data exploration and preparation, in driving impactful research outcomes.

4

Experimental Results

Introduction

This chapter presents the experimental results of our study on energy disaggregation. We begin with an overview of the model architecture and the application of transfer learning techniques, including sequence-to-sequence and sequence-to-point learning, as well as CNN layers. Performance evaluation metrics are discussed, followed by detailed analyses of the training and testing phases using REFIT and UK-DALE datasets. Finally, we examine specific appliance results, compare cross-dataset performances, and address observed limitations and potential solutions.

4.1 Model Overview

The model is designed to predict the electricity consumption of individual household appliances from the aggregate household electricity usage data. This task involves energy disaggregation and sequence-to-point learning techniques, aiming to break down the total energy consumption into the specific consumption of each appliance.

4.1.1 Model Architecture

1. Input Layer:

   The model starts with an input layer that accepts a sequence of electricity usage data. This sequence has a fixed length, determined by the time window being analyzed.

2. Reshape Layer:

   The input sequence is reshaped to fit the format required for convolutional layers. This reshaping step adds additional dimensions, preparing the data for further processing.

3. Convolutional Layers:

   The model includes five convolutional layers. Each layer applies a series of filters to the input data to extract features.

   The first two convolutional layers use 30 filters each, with decreasing kernel sizes of 10 and 8 respectively.

   The third layer increases the number of filters to 40 and uses a kernel size of 6.

   The fourth and fifth layers use 50 filters but with kernel sizes of 5.

   These layers use ReLU activation functions to introduce non-linearity into the model, helping it learn complex patterns in the data.

   The convolutional layers use 'same' padding to ensure that the output size matches the input size, and they move across the input data one step at a time (stride of 1).

4. Flatten Layer:

   The output from the last convolutional layer is flattened into a single 1D vector. This transformation prepares the data for the dense layers, which require a 1D input.

5. Dense Layers:

   Following the convolutional layers, the model includes two dense (fully connected) layers.

   The first dense layer has 1024 units and uses a ReLU activation function. This layer further processes the features extracted by the convolutional layers.

   The second dense layer has a single unit with a linear activation function. This final layer outputs the predicted electricity consumption of the target appliance.

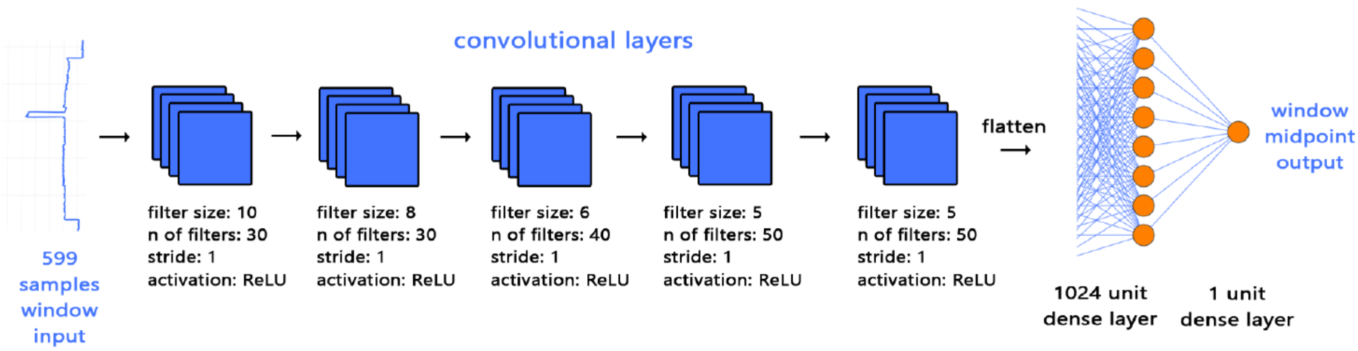

Figure 4.1: Modele Architecture

Summary

This convolutional neural network (CNN) is structured to efficiently process sequences of electricity usage data. The convolutional layers extract meaningful features from the input sequences, while the dense layers transform these features into a single prediction of an appliance's electricity consumption. The use of sequence-to-point learning allows the model to focus on predicting consumption at specific points in time, making it highly effective for energy disaggregation tasks. This approach enables the model to learn and recognize patterns in the aggregate data that correspond to the usage of individual appliances.

4.2 Transfer Learning in Energy Disaggregation

4.2.1 Sequence-to-Sequence Learning

Sequence-to-sequence (seq2seq) learning is a technique where an input sequence is mapped to an output sequence. In the context of energy disaggregation, this approach can be used to transform an aggregate sequence of household electricity consumption into separate sequences for each appliance. For example, given a time series of total power consumption, seq2seq learning can predict the power consumption time series for individual appliances [35].

Concept Explanation

Given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$ representing the total power consumption over T time steps, and an output sequence $Y = \{y_1, y_2, \ldots, y_{T'}\}$ for a specific appliance, seq2seq learning aims to find a function f such that:

$$Y = f(x)$$

In practice, seq2seq models for energy disaggregation use an encoder-decoder architecture. The encoder processes the input sequence and compresses it into a context vector, which captures the essential information. The decoder then uses this context vector to generate the output sequence.

Encoder

The encoder processes the input sequence using layers like Recurrent Neural Networks (RNNs) or Long Short-Term Memory Networks (LSTMs). The hidden states h_t at each time step t capture the temporal dependencies in the data:

$$h_t = \text{RNN}(x_t, h_{t-1})$$

The final hidden state h_T, or a combination of all hidden states, is used as the context vector c.

Decoder

The decoder generates the output sequence from the context vector. For each time step in the output sequence, the decoder predicts the appliance's power consumption:

$$s_t = \text{RNN}(y_{t-1}, s_{t-1}, c)$$

where s_t is the hidden state of the decoder at time step t, and y_t is the predicted power consumption. The probability of each output value is given by:

$$P(y_t \mid y_1, \ldots, y_{t-1}, c) = \text{softmax}(W s_t)$$

where W is a weight matrix.

By training on sequences of aggregate consumption and corresponding appliance consumption, the model learns to disaggregate the energy consumption patterns effectively.

4.2.2 Sequence-to-Point Learning in Energy Disaggregation

Sequence-to-point (seq2point) learning is a technique where an input sequence is mapped to a single-point prediction. In the context of energy disaggregation, this approach is used to predict the power consumption of a specific appliance from a window of aggregate power data. This method is particularly useful when we want to estimate the power usage at a specific time point for a particular appliance, given the overall household power consumption [36].

Concept Explanation

Given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, seq2point learning aims to find a function g such that:

$$y = g(x)$$

where y is the scalar output representing the power consumption of the appliance at a specific point.

Seq2point models typically use convolutional neural networks (CNNs) to extract features from the input sequence, followed by dense layers to produce the final point prediction.

Feature Extraction

Convolutional layers process the input sequence to extract meaningful features. For instance:

$$F = \text{CNN}(x)$$

where F represents the feature map extracted from the input sequence. These layers apply filters of different sizes to capture various patterns in the data, such as short-term spikes or long-term trends in power consumption.

Flattening and Dense Layers

The feature map F is flattened into a single vector:

$$\text{Flattened} = \text{flatten}(F)$$

This vector is then passed through fully connected (dense) layers to transform it into the final prediction:

$$y = W \cdot \text{Flattened} + b$$

where W and b are the weight matrix and bias vector of the dense layer.

Prediction

The final output y is a single scalar value, representing the predicted power consumption of the appliance at the given time point.

By focusing on point predictions, seq2point learning allows the model to pinpoint the power usage of specific appliances accurately, even in the presence of complex and overlapping usage patterns in the aggregate data.

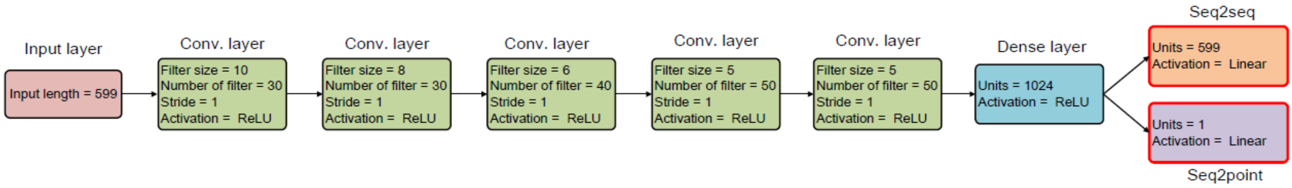

Figure 4.2: The architectures for sequence-to-point and sequence-to-sequence neural networks.

4.2.3 Appliance Transfer Learning with CNN Layers

Appliance transfer learning using CNN layers involves leveraging pre-trained convolutional neural network (CNN) models trained on one dataset to enhance the performance of energy disaggregation models on another dataset with different appliances, such as UK-DALE and REFIT.

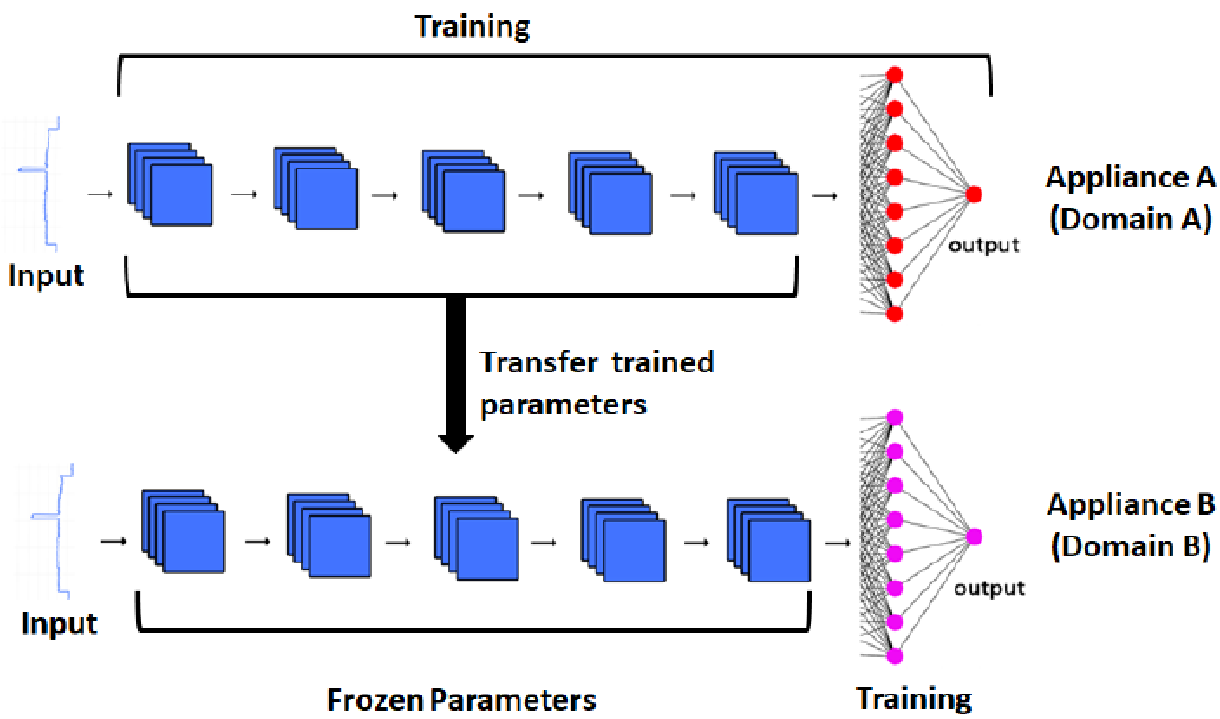

Figure 4.3: Frozen Parameters

Concept Explanation

Dataset Specificity:

- **UK-DALE:** Contains electricity consumption data primarily from UK households, featuring a specific set of appliances commonly found in UK homes.
- **REFIT:** Includes electricity usage data from households in the UK and Ireland, with appliances typical to these regions, potentially different from those in UK-DALE.

Utilizing CNN Layers:

- **Feature Extraction:** CNN layers are adept at capturing hierarchical patterns and features in sequential data, such as electricity consumption over time.
- **Transfer Learning:** pre-trained CNN models, originally trained on one dataset (e.g., UKdale), can extract generic features relevant to energy consumption patterns across different appliances.

Transfer Learning Approach:

- **CNN Feature Extractors:** The convolutional layers in a CNN model act as feature extractors, learning to detect spatial and temporal patterns in the input data.
- **Fine-Tuning:** Transfer learning involves adapting these pre-trained CNN layers to the target dataset (e.g., REFIT) by fine-tuning their weights based on the new dataset's specific characteristics.

Benefits in Energy Disaggregation:

- **Improved Accuracy:** Transfer learning with CNN layers helps in improving the accuracy of appliance-level energy disaggregation tasks by leveraging learned features from related datasets.
- **Generalization:** It enables models to generalize better across datasets with different appliance types, reducing overfitting and improving robustness in predicting appliance-specific energy consumption.

4.2.4 Cross Domain Transfer Learning with CNN Layers

Cross-domain transfer learning using CNN layers extends the application of transfer learning to scenarios where datasets not only differ in appliance types but also exhibit significant variations in data characteristics, such as geographic location or data collection methods.

Concept Explanation

Domain Differences:

- **Geographic:** Datasets like UK-DALE and REFIT may capture energy consumption patterns from different geographic regions, influencing appliance types and usage behaviors.
- **Data Characteristics:** Variations in sampling rates, data resolution, or sensor types can introduce differences in the structure and quality of the data.

CNN Adaptation Across Domains:

- **Domain-Adaptive CNN Layers:** Pretrained CNN layers are initially trained on a source domain (e.g., UK-DALE), learning to extract relevant features and patterns from the data.

- **Domain Shift Handling:** Techniques like domain adaptation or fine-tuning adjust these CNN layers to the characteristics of the target domain (e.g., REFIT), ensuring they capture domain-specific nuances effectively.

Enhancing Model Adaptability:

- **Feature Representation:** CNN layers excel in learning hierarchical representations of data, making them suitable for capturing complex patterns in energy consumption across diverse domains.
- **Efficient Knowledge Transfer:** Cross-domain transfer learning with CNN layers minimizes the need for extensive labeled data in each new domain, accelerating model adaptation and deployment.

Applications in Energy Disaggregation:

- **Robust Model Performance:** By adapting pre-trained CNN layers across different domains, models can achieve robust performance in disaggregating energy usage, effectively handling variations in appliance types and data characteristics.
- **Scalability:** These techniques support scalable deployment of energy disaggregation models across various regions and datasets, facilitating broader applications in energy management and sustainability.

4.3 Performance Evaluation Metrics

The Evaluation metrics provide a comprehensive evaluation of our CNN model's performance in energy disaggregation, these metrics ensure covering aspects of absolute error, cumulative error, daily usage patterns, and normalized performance. The following metrics are adapted to our model and initially used in [30].

$$\mathrm{MAE} = \frac{1}{T}\sum_{t=1}^{T} \left|\hat{x}_t - x_t\right|. \tag{4.1}$$

- $\hat{x}_t$ represents the predicted value at time t,
- x_t represents the actual (true) value at time t,
- T is the total number of samples or time points.

The second metric is the normalized signal aggregate error (SAE), which indicates the relative error of the total energy. Denote r as the total energy consumption of the appliance and $\hat{r}$ the predicted total energy, then SAE is defined :

$$SAE = \frac{|\hat{r} - r|}{r}. \tag{4.2}$$

The third metric is the energy per day (EPD):

$$EPD = \frac{1}{D}\sum_{n=1}^{D} \left|\hat{e} - e\right|, \tag{4.3}$$

Where $e = \sum_t x_t$ denotes the energy consumed in a day period and D is the total number of days. This metric indicates the absolute error of the predicted energy used in a day, which is typically useful when household users are interested in the total energy consumed in a period.

The fourth metric is the normalized disaggregation error (NDE):

$$NDE = \frac{\sum_{i,t} (x_{it} - \hat{x}_{it})^2}{\sum_{it} x_{it}^2} \tag{4.4}$$

This metric measures the normalized error of the squared difference between the prediction and the ground truth of the appliances.

4.3.1 Why These Metrics Were Chosen for Our CNN Model

MAE and SAE:

Both provide clear and interpretable measures of prediction accuracy and cumulative error, respectively. They help in understanding how well the CNN model performs in terms of absolute errors, which is crucial for practical applications like billing and appliance monitoring.

EPD:

Evaluating energy per day helps in understanding the model's performance in capturing daily energy usage patterns, which is important for user insights and energy management.

NDE:

Provides a normalized error measure, which is useful for comparing model performance across different scales of energy consumption and ensuring that the model performs well relative to the actual energy usage.

4.4 Training Phase Overview

The training phase of our CNN-based energy disaggregation model is a systematic process that involves preparing data, designing and compiling the model architecture, optimizing parameters through iterative training, and evaluating performance. This phase ensures that the model learns to accurately predict appliance-level energy consumption patterns, making it a valuable tool for energy management and conservation efforts.

Here's a general overview of what happens during this phase:

1. **Data Preparation**

    - **Data Loading:** Training begins with loading and preparing datasets. This typically involves preprocessing steps such as normalization and splitting data into training and validation sets.
    - **Data Feeding:** Data is fed into the model in batches to facilitate efficient computation and gradient updates. Techniques like sliding window generators are often used to manage large datasets effectively.

2. **Model Architecture and Compilation**
    - **Model Design:** The CNN architecture is designed based on the problem requirements and input data characteristics. CNNs are chosen for their ability to capture spatial dependencies in sequential data, making them suitable for time series like energy consumption.
    - **Compilation:** Once the model architecture is defined, it is compiled with an optimizer (such as Adam) and a loss function (like Mean Squared Error) that quantifies the difference between predicted and actual energy consumption.
3. **Training Execution**
    - **Parameter Optimization:** The core of training involves adjusting the model's parameters (weights and biases) iteratively to minimize the chosen loss function. This optimization process uses backpropagation to compute gradients and update parameters based on training data.
    - **Epochs and Iterations:** Training proceeds over a fixed number of epochs (complete passes through the training dataset). During each epoch, batches of data are processed, and after each batch or epoch, the model's performance on validation data may be evaluated to monitor generalization.
4. **Monitoring and Optimization**
    - **Early Stopping:** To prevent overfitting, techniques like early stopping may be employed. Early stopping halts training when the model's performance on a validation dataset fails to improve over a specified number of epochs.
5. **Model Evaluation and Saving**
    - **Performance Assessment:** Throughout the training, model performance is assessed based on various metrics that evaluate prediction accuracy and generalization. These assessments help in fine-tuning the model and ensuring it meets performance expectations.
    - **Model Saving:** Once training is complete and the model demonstrates satisfactory performance on validation data, it is typically saved for future use. Saved models can be deployed for inference on new, unseen data to make predictions about energy consumption.

Input window size (samples)	599
Number of maximum epochs	50
Batch size	1000
Minimum early-stopping epochs	5
The patience of early-stopping (in epochs)	5

Table 4.1: Hyper-parameters for training

Learning rate	0.001
Beta1	0.9
Beta2	0.999
Epsilon	10^{-8}

Table 4.2: Parameters used for Adam optimizer

4.4.1 Analysis of Training Parameters

The hyperparameters and optimizer settings chosen for this model training reflect a balanced approach to learning efficiency and stability. The input window size of 599 samples suggests a substantial context for time series analysis. The maximum of 50 epochs, combined with early-stopping measures (minimum 5 epochs, patience of 5), indicates a strategy to prevent overfitting while allowing sufficient training time. A large batch size of 1000 promotes stable gradient estimates and faster convergence. For the Adam optimizer, a relatively low learning rate of 0.001 is selected to ensure careful parameter updates. The Beta1 (0.9) and Beta2 (0.999) values are typical for Adam, providing a good balance between the speed of convergence and the stability of updates. The very small Epsilon (10^{-8}) helps prevent any division by zero in the optimizer's computations. These parameters collectively aim to achieve a robust training process that can effectively learn from the data while mitigating common issues like overfitting or unstable convergence.

4.5 Testing Phase Overview

The testing phase plays a pivotal role in ensuring the reliability and effectiveness of our CNN-based energy disaggregation model. By systematically evaluating its predictions against actual data, we gain insights into its performance and readiness for deployment, contributing to more efficient energy management strategies. Here's an overview of the testing phase, focusing on its key components and objectives:

1. **Initialization and Setup**
   - **Initialization:** The Tester class initializes with parameters such as the target appliance (___appliance), pruning algorithm (___algorithm), network architecture (___network_type), batch size (___batch_size), and others essential for testing.
   - **Data Setup:** The testing data is loaded from a specified directory (___test_directory) using the load_dataset method. This method retrieves a subset of data (___crop) for evaluation.

2. **Model Loading**

   - **Model Retrieval:** The pre-trained model is loaded using the create_model and load_model functions from model_structure.py. This step ensures that the model architecture matches the one used during training.

3. **Testing Execution**

   - **Testing Procedure:** The test_model method initiates the testing process:
     - **Data Generation:** Utilizes a TestSlidingWindowGenerator to generate input-output pairs for testing. This generator slides over the test dataset in Windows, preparing data batches for inference.
     - **Inference:** Calls model.predict() to perform inference on the test data. It measures the time taken for inference (test_time) and collects the predictions (testing_history).
     - **Evaluation:** Evaluates the model's performance metrics using the model.evaluate(), capturing metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE).

4. **Logging and Visualization**

   - **Logging Results:** Results such as inference time (test_time) and evaluation metrics (MSE, MAE) are logged into a specified file (___log_file) using Python's logging module.
   - **Plotting Results:** The plot_results method visualizes the testing outcomes:
     - **Plot Components:** Generates plots comparing the predicted appliance energy consumption (testing_history) against actual data (test_target) and aggregate energy consumption (test_input).
     - **Data Transformation:** Rescales the predictions and actual values using mean and standard deviation information (appliance_data and mains_data) to interpret them in meaningful units (Watts).

5. **Conclusion and Analysis**

   - **Result Interpretation:** The plots and logged metrics provide a comprehensive view of how well the model predicts appliance-level energy consumption. This evaluation is crucial for assessing the model's accuracy and performance on unseen data, essential for real-world deployment.

Importance of Testing Phase

- **Validation of Generalization:** Testing ensures that the model generalizes well to new, unseen data, validating its performance outside the training context.
- **Performance Evaluation:** Metrics like MSE and MAE quantify the prediction errors, guiding further improvements or adjustments to the model.
- **Decision Support:** Reliable testing results support decisions regarding model deployment or further refinement, enhancing the model's utility in practical applications.

4.5.1 Results Of the Training (REFIT)

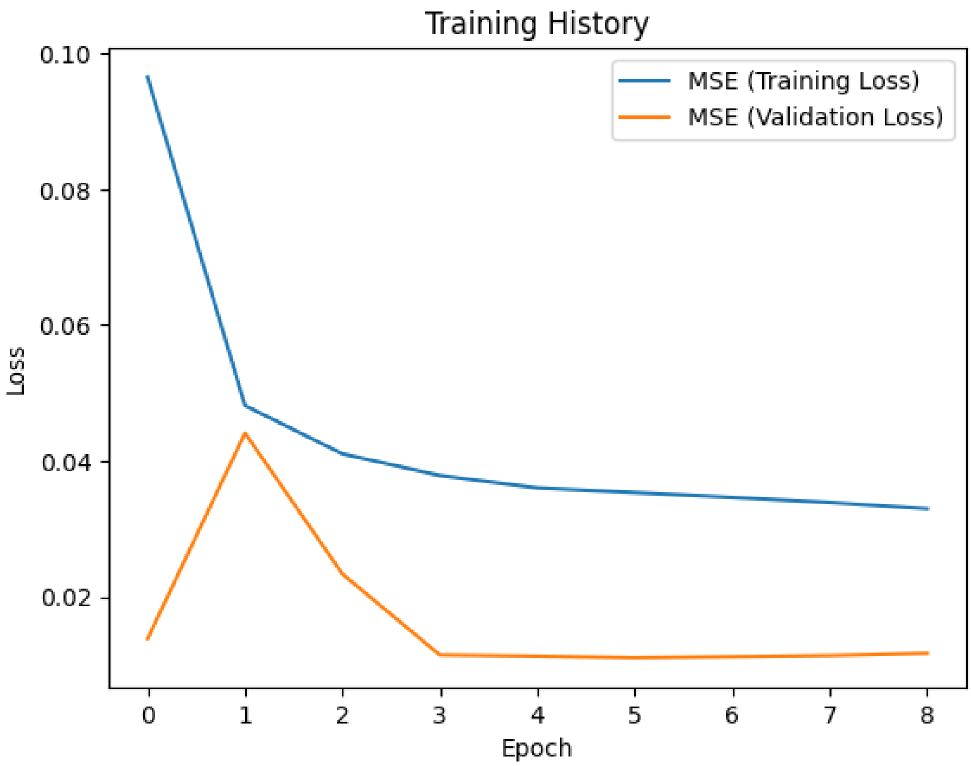

Figure 4.4: fridge Training and Validation loss curves

Fridge Training

Training History Comment The graph illustrates the training and validation loss curves for the fridge model trained on the REFIT dataset. The Mean Squared Error (MSE) is used as the loss function for both training and validation.

From the graph, we observe the following:

1. **Initial Training Phase**: The training loss starts relatively high at around 0.1 and decreases rapidly over the first few epochs. This indicates that the model is learning and improving its performance on the training data. 2. **Validation Loss Behavior**: The validation loss starts lower than the initial training loss, showing a sharp decline in the first epoch. It then fluctuates but stabilizes quickly around 0.02. 3. **Convergence**: By the 5th epoch, both training and validation losses stabilize, indicating that the model has converged. The training loss continues to decrease slightly, suggesting continued improvement, albeit at a slower rate.

Overall, the model shows a good fit to the training data with minimal overfitting, as indicated by the close values of training and validation losses in the latter epochs.

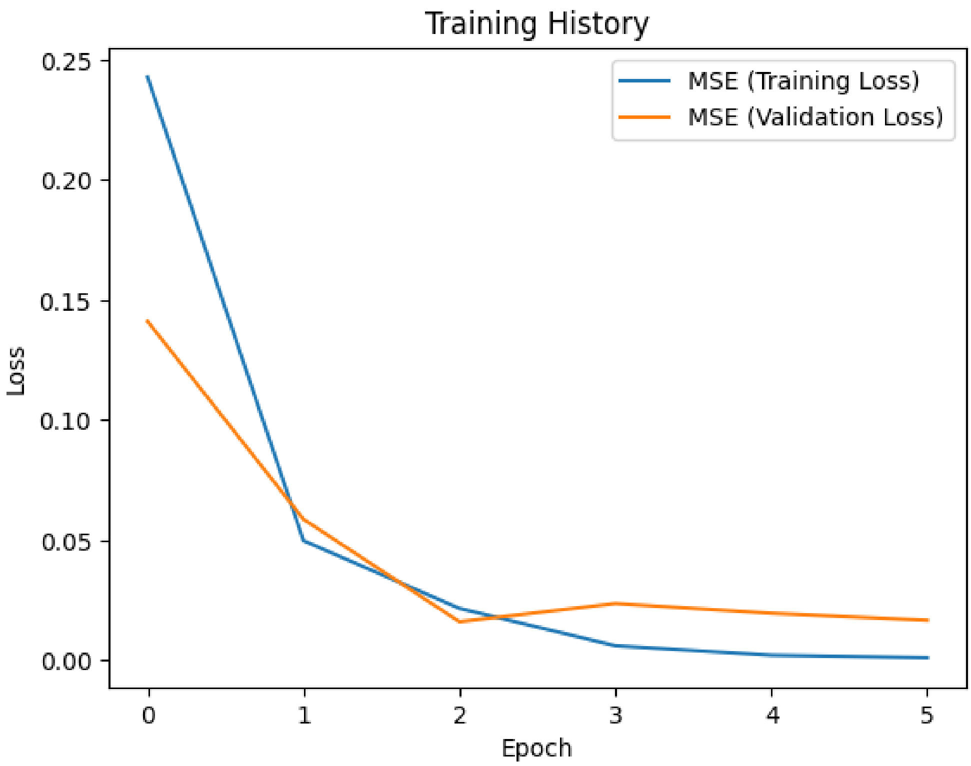

Figure 4.5: Kettle Training and Validation loss curves

Kettle Training

Training History Comment The graph for the kettle model reveals:

1. **Initial Training Phase**: The training loss begins at around 0.25, decreasing sharply in the first epoch, indicating significant initial learning. 2. **Validation Loss Behavior**: The validation loss follows a similar pattern, initially higher but dropping significantly to stabilize around 0.02 by epoch 3. 3. **Convergence**: By the 4th epoch, both training and validation losses have stabilized, suggesting that the model has effectively learned the kettle usage patterns.

The kettle model also shows good fitment with minimal overfitting, as evidenced by the closely aligned training and validation loss curves.

These results indicate that both models have been trained effectively on their respective datasets, achieving convergence with minimal overfitting.

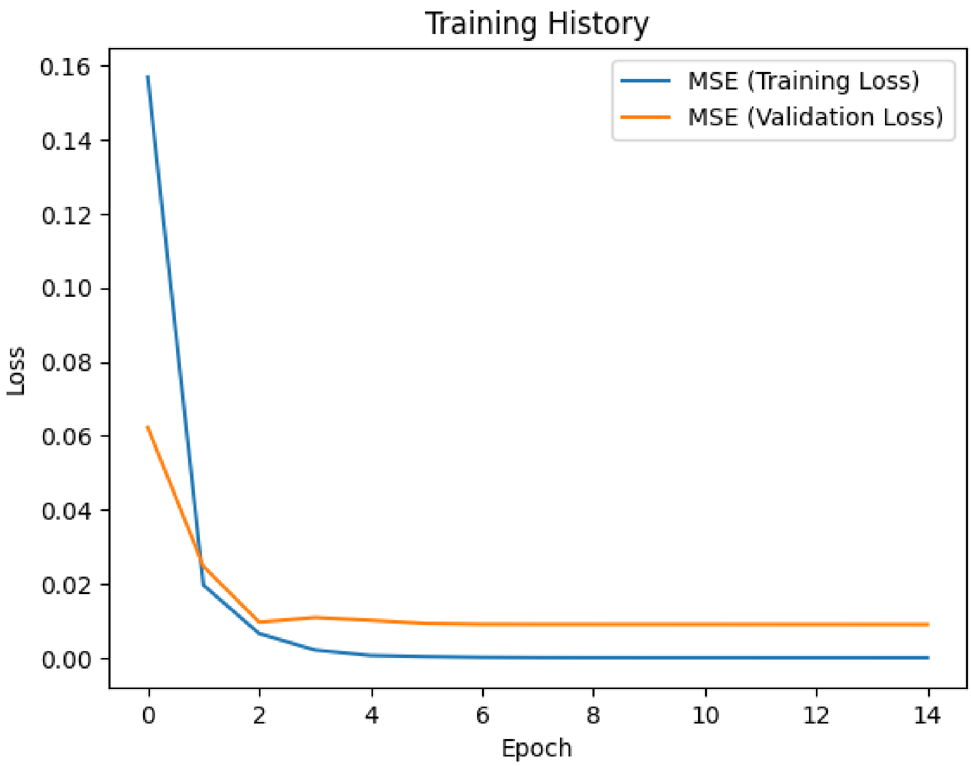

Figure 4.6: Microwave Training and Validation loss curves

Microwave Training

Training History Comment The graph for the microwave model shows:
1. **Initial Training Phase**: The training loss starts at approximately 0.16 and decreases rapidly over the first two epochs, indicating effective initial learning. 2. **Validation Loss Behavior**: The validation loss starts slightly lower than the initial training loss, rapidly decreasing and then stabilizing around 0.02 by epoch 4. 3. **Convergence**: By the 5th epoch, both the training and validation losses have stabilized, suggesting convergence. The training loss continues to decrease slightly, showing ongoing minor improvements.

Overall, the microwave model demonstrates a good fit with minimal overfitting, as indicated by the close training and validation loss values.

These results indicate that all three models have been trained effectively on their respective datasets, achieving convergence with minimal overfitting.

4.5.2 Results Of the Training (UK-DALE)

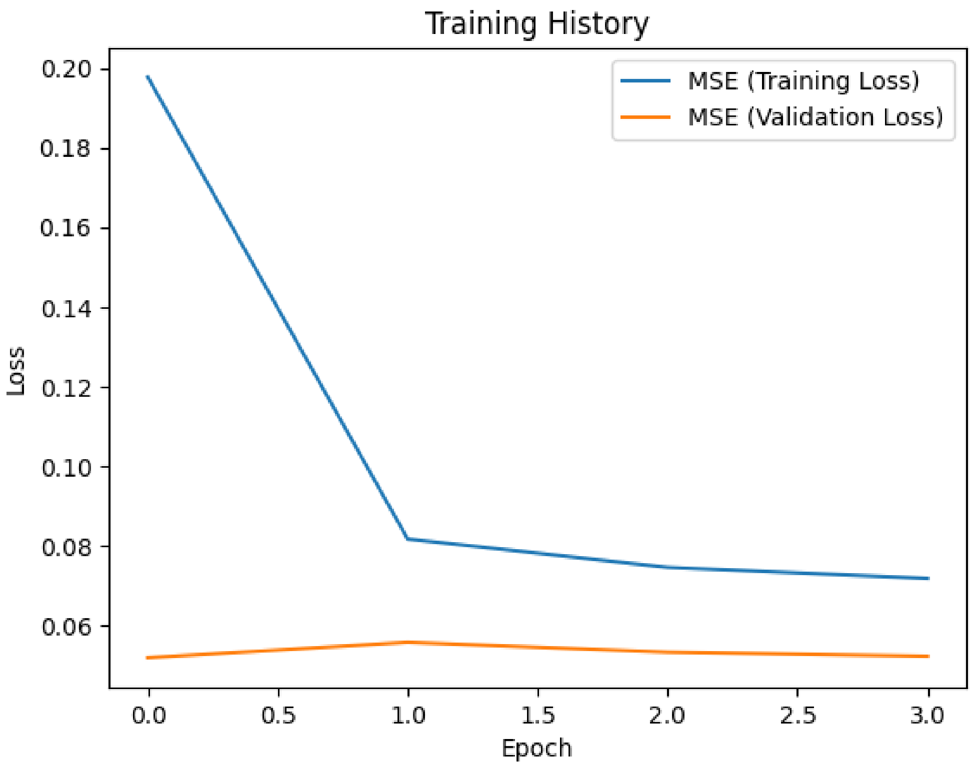

Figure 4.7: Kettle Training and Validation loss curves

Kettle training

Training History Comment The graph illustrates the training history of a model for kettle energy consumption prediction using the UK-DALE dataset. It depicts the Mean Squared Error (MSE) for both training and validation sets over 3 epochs.

- **Training Loss**: The blue line represents the MSE on the training set. It shows a rapid decrease from approximately 0.20 to 0.08 in the first epoch, followed by a more gradual decline to about 0.07 by the third epoch. This pattern indicates that the model quickly learns the main features of the data, with diminishing improvements in later epochs.
- **Validation Loss**: The orange line represents the MSE on the validation set. It starts at a much lower value (around 0.05) and remains relatively stable throughout the training process, with a slight increase to about 0.055 by the end of the third epoch.
- **Convergence**: The training and validation losses converge towards each other, suggesting that the model is learning generalizable patterns rather than overfitting the training data.
- **Early Stopping Potential**: Given the minimal improvement in both training and validation loss after the first epoch, early stopping could be considered to prevent potential overfitting and optimize computational resources.

- **Model Performance**: The low and stable validation loss indicates that the model performs well on unseen data, which is crucial for practical applications in energy consumption prediction.

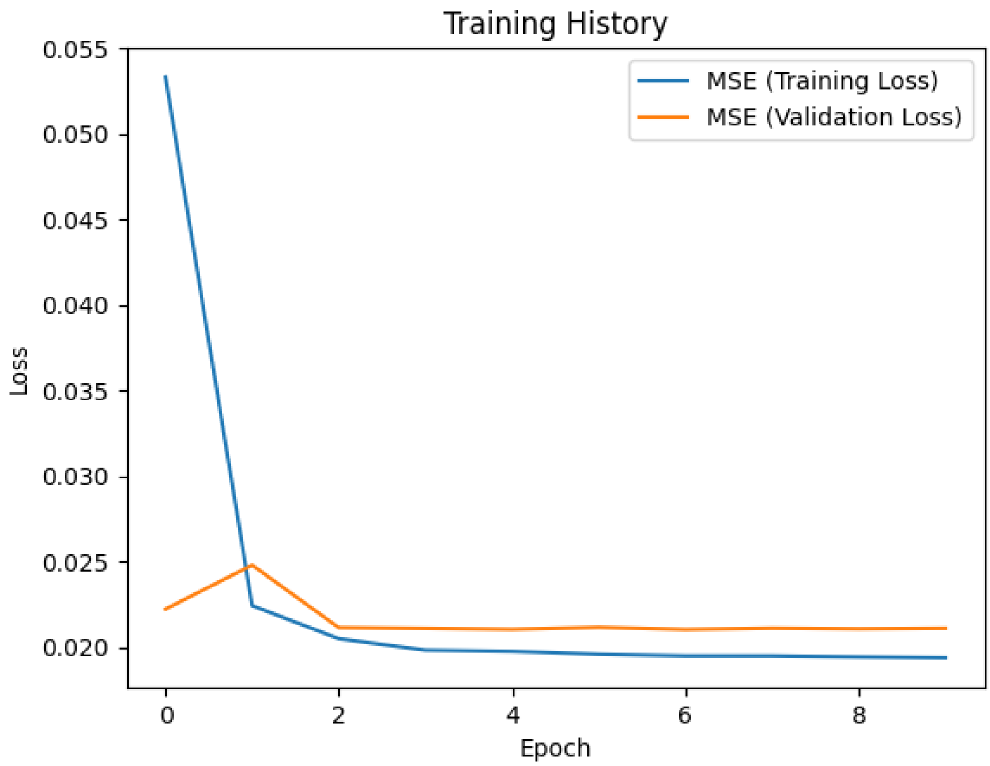

Figure 4.8: Fridge Training and Validation loss curves

Fridge Training

Training History Comment This graph illustrates the training progress of a model predicting fridge energy consumption using the UK-DALE dataset. It shows the Mean Squared Error (MSE) for both training and validation sets over 9 epochs.

- **Training Loss**: The blue line represents the MSE on the training set. It exhibits a dramatic decrease from about 0.054 to 0.023 in the first epoch, followed by a more gradual decline to approximately 0.020 by the ninth epoch. This pattern suggests rapid initial learning of the data's primary features, with incremental improvements thereafter.

- **Validation Loss**: The orange line depicts the MSE on the validation set. It starts at a lower value (around 0.023) compared to the training loss, shows a slight increase to about 0.025 in the first epoch, and then stabilizes around 0.021 for the remaining epochs.

- **Convergence**: The training and validation losses converge quickly, nearly meeting after the second epoch and maintaining a very close relationship thereafter. This convergence indicates that the model is learning generalizable patterns and not overfitting the training data.
- **Model Stability**: After the second epoch, both losses stabilize, showing minimal fluctuations. This stability suggests that the model has reached a robust state of performance on both seen and unseen data.
- **Early Stopping Consideration**: Given the minimal improvements after the second epoch, implementing early stopping could be beneficial to optimize computational resources without significantly sacrificing model performance.

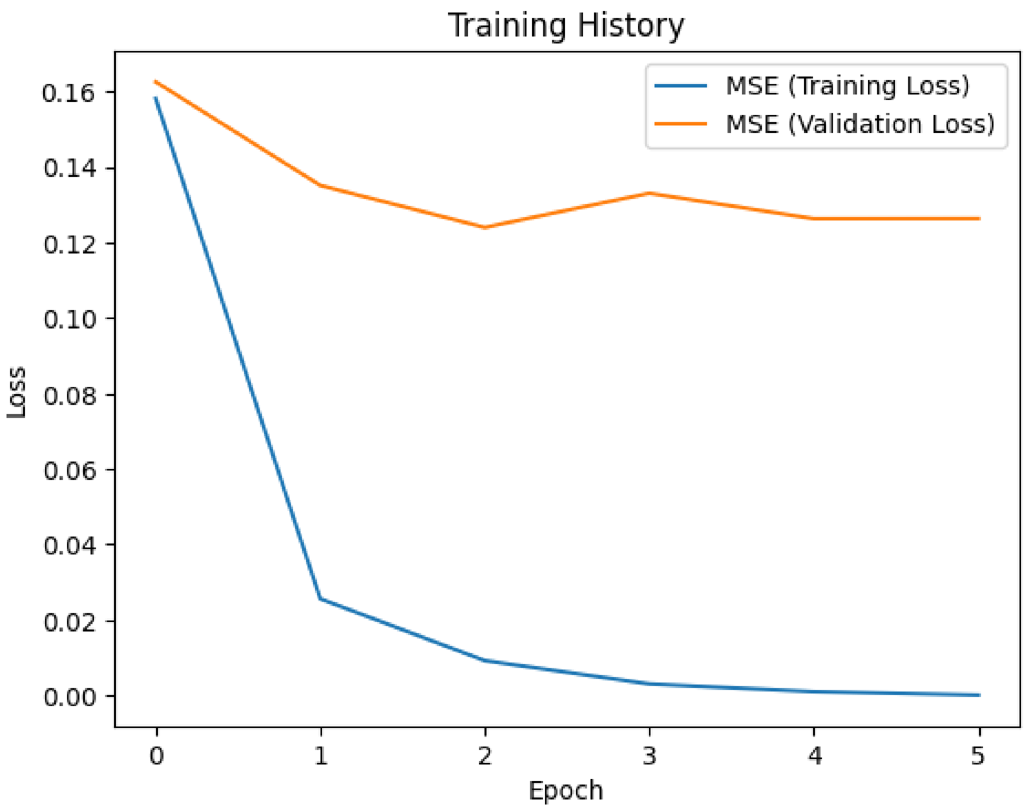

Figure 4.9: Washing Machine Training and Validation loss curves

Washing Machine

Training History Comment This graph depicts the training progress of a model predicting washing machine energy consumption using the UK-DALE dataset. It shows the Mean Squared Error (MSE) for both training and validation sets over 5 epochs.

- **Training Loss**: The blue line represents the MSE on the training set. It shows a dramatic decrease from approximately 0.16 to 0.03 in the first epoch, followed by a continued steep decline to near 0 by the fifth epoch. This pattern indicates extremely rapid and effective learning of the training data.
- **Validation Loss**: The orange line depicts the MSE on the validation set. It starts at about 0.16, decreases gradually to around 0.12 by the second epoch, and then shows slight fluctuations before stabilizing at approximately 0.13 by the fifth epoch.
- **Divergence**: Unlike the previous models, there is a significant and growing divergence between the training and validation losses. The training loss approaches zero while the validation loss remains relatively high and stable.
- **Potential Overfitting**: The dramatic decrease in training loss coupled with the relatively high and stable validation loss strongly suggests overfitting. The model appears to be memorizing the training data rather than learning generalizable patterns.
- **Model Behavior**: The washing machine's energy consumption patterns may be more complex or variable than those of the kettle or fridge, potentially explaining the model's difficulty in generalizing.

4.6 Test Results

4.6.1 Trained and Tested On UK-DALE

We started our prediction by training our model on the fridge appliance from the UK-DALE dataset then was tested on the same dataset.

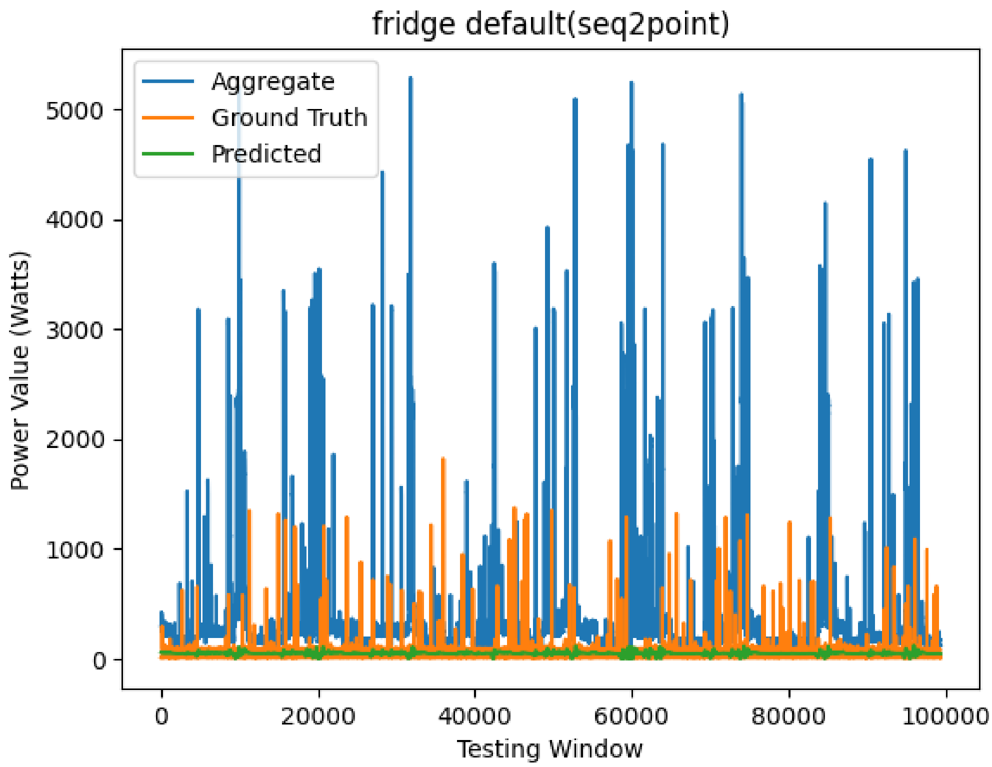

Figure 4.10: Washing Machine Training and Validation loss curves

Performance Metrics

- **MSE (Mean Squared Error)**: 0.0136
- **MSLE (Mean Squared Logarithmic Error)**: 0.00067306
- **MAE (Mean Absolute Error)**: 0.0974

These metrics suggest that the model's predictions are fairly close to the ground truth, but there's still room for improvement.

Graph Analysis

- The blue line (Aggregate) represents the total power consumption.
- The orange line (Ground Truth) shows the actual fridge power consumption.
- The green line (Predicted) represents the model's predictions.

Observations

- The model seems to capture the general pattern of the fridge's power consumption.
- The predictions (green) closely follow the ground truth (orange) in many areas, which is a good sign.
- However, the model appears to underestimate some of the peaks in power consumption. This is evident where the orange line has spikes that the green line doesn't fully capture.
- The model seems to perform better during periods of lower and more stable power consumption.
- There are some areas where the prediction doesn't align well with the ground truth, especially during rapid fluctuations in power consumption.

4.6.2 Trained On UK-DALE and Tested On REFIT

Now our prediction by training our model on fridge appliance from the UK-DALE dataset and then tested on Refit .

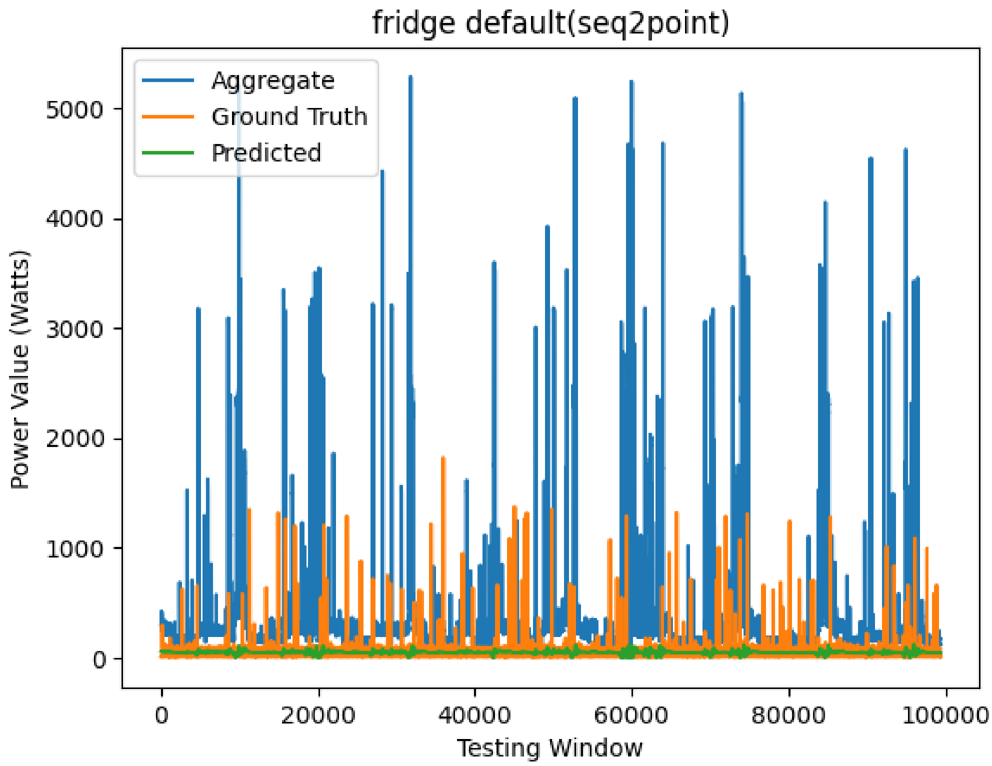

Figure 4.11: Washing Machine Training and Validation loss curves

Performance Metrics

- **MSE (Mean Squared Error)**: 0.0129
- **MSLE (Mean Squared Logarithmic Error)**: 1.5422e-04
- **MAE (Mean Absolute Error)**: 0.0993

Graph Components

- The blue line represents the Aggregate power consumption.
- The orange line shows the Ground Truth for fridge power consumption.
- The green line represents the Predicted fridge power consumption.

Observations

1. The model appears to capture the general pattern of the fridge's power consumption quite well.
2. The predictions (green) closely follow the ground truth (orange) for most of the testing window, which is a positive sign.
3. The model seems to perform best during periods of lower and more stable power consumption, as evidenced by the close alignment of green and orange lines in these areas.
4. There are several high spikes in the Aggregate power consumption (blue line) that don't correspond to spikes in the fridge's power usage. The model correctly ignores these, which is good.
5. The model appears to slightly underestimate some of the peaks in fridge power consumption. This is evident where the orange line has small spikes that the green line doesn't fully capture.
6. There are a few instances where the prediction doesn't perfectly align with the ground truth, especially during rapid fluctuations in power consumption.

4.6.3 Fridge Trained On Refit and Tested On Washing Machine from UK-DALE

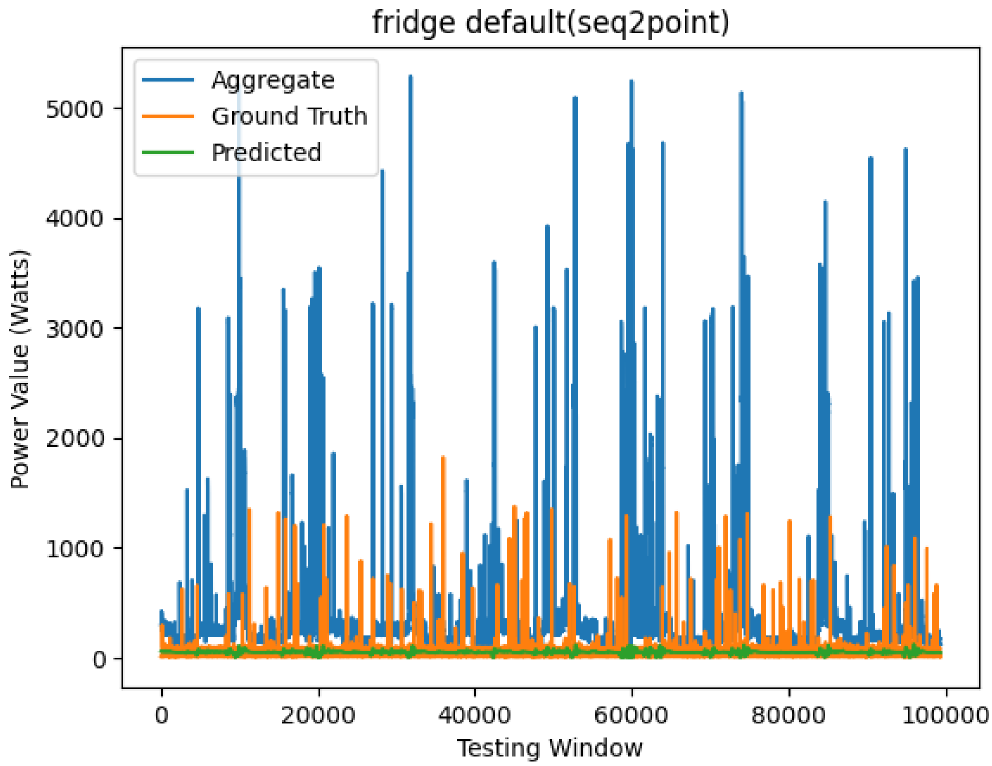

Figure 4.12: Washing Machine Training and Validation loss curves

Performance Metrics

- **MSE (Mean Squared Error)**: 0.0553
- **MSLE (Mean Squared Logarithmic Error)**: 0.0051
- **MAE (Mean Absolute Error)**: 0.1771

Graph Components

- Blue line: Aggregate power consumption
- Orange line: Ground Truth for fridge power consumption
- Green line: Predicted fridge power consumption

Observations

1. The model generally captures the pattern of the fridge's power consumption well, with the predicted (green) line closely following the ground truth (orange) for most of the testing window.

2. The aggregate power consumption (blue) shows frequent high spikes, often reaching above 3000 watts, while the fridge's consumption remains much lower.

3. The model successfully distinguishes between the overall household power usage and the fridge's specific consumption, which is a significant strength.

4. There's a notable spike in the ground truth (orange) around the 60,000 mark on the testing window, which the model doesn't fully capture. This could indicate a limitation in predicting unusual or extreme events.

5. The model seems to slightly underestimate some of the smaller peaks in fridge power consumption, as seen by small orange spikes that aren't fully reflected in the green prediction line.

6. Overall, the prediction is most accurate during periods of stable, low power consumption by the fridge.

4.7 Limitations and Potential Solutions

The model's performance, while generally strong, exhibits certain limitations that can be addressed through various approaches. The primary challenges include accurately capturing sudden spikes in power consumption, predicting unusual or extreme events, and fully representing short-term fluctuations. These limitations likely stem from the model's current architecture, feature set, or training data composition. To overcome these challenges, we propose the following solutions:

- **Feature Engineering:** Enhance the feature set to better represent power consumption patterns, particularly for peak detection and short-term variations. This could involve creating time-based features or derived metrics that capture rapid changes in power usage. Copy

- **Advanced Time Series Techniques:** Implement more sophisticated time series modeling approaches such as Long Short-Term Memory (LSTM) networks, attention mechanisms, or temporal convolutional networks. These methods are specifically designed to capture both long-term patterns and short-term fluctuations in time series data.

- **Model Architecture Refinement:** Experiment with different neural network architectures or ensemble methods that can maintain good overall performance while improving responsiveness to sudden changes. This might include deeper networks, skip connections, or a combination of multiple model types.

- **Anomaly Detection:** Incorporate specific anomaly detection techniques to better identify and predict unusual spikes or patterns in fridge power consumption. This could involve statistical methods or dedicated machine learning models for outlier detection.

- **Data Augmentation:** Expand the training dataset to include a wider variety of power consumption scenarios, particularly unusual patterns or extreme events. This will help the model learn to predict a broader range of situations more accurately.
- **Hyperparameter Optimization:** Conduct thorough hyperparameter tuning, potentially using automated methods like grid search or Bayesian optimization, to find the optimal configuration for capturing both overall trends and short-term changes in power consumption.
- **Resource Optimization:** Address computational resource limitations by optimizing existing resources, leveraging cloud-based solutions, or utilizing distributed computing techniques. Efficiently managing resources can enable the use of more complex models and extensive hyperparameter tuning.

By implementing these solutions, we aim to enhance the model's ability to accurately predict fridge power consumption across a wider range of scenarios, including rapid fluctuations and anomalous events, while maintaining its current strengths in distinguishing fridge usage from overall household power consumption.

4.8 Conclusion

In this chapter, we have demonstrated the effectiveness of our deep learning models for energy disaggregation, with a focus on transfer learning techniques. Our experiments showed that models incorporating sequence-to-sequence and sequence-to-point learning, along with CNN layers, achieved notable performance across different datasets. The detailed analyses of the training and testing phases provided insights into model behavior and highlighted the importance of selecting appropriate evaluation metrics. Despite some limitations, our findings offer promising directions for future improvements and applications in the field of energy disaggregation.

General Conclusion

The escalating energy consumption in Algeria, particularly within the residential sector, underscores a pressing need for effective energy management strategies. With residential energy use accounting for more than a third of the nation's total energy consumption, the projected growth in population and housing presents significant challenges to maintaining a balanced energy supply-demand equilibrium, especially in the face of stagnant gas and oil reserves—the primary energy sources in Algeria.

The steady rise in global energy demand, coupled with static oil and gas reserves since 2005, signals potential imbalances in energy supply, necessitating proactive measures in energy management and conservation. Identifying the drivers of household electricity consumption is pivotal in fostering efficient energy utilization. Moreover, segmenting households based on consumption patterns allows for targeted interventions and policy formulations tailored to diverse consumer needs and behaviors.

This thesis explores the application of deep learning methodologies to classify households according to their energy usage profiles. By leveraging advanced neural network architectures and comprehensive datasets, the research aims to develop models capable of accurately categorizing energy consumption behaviors. Such classification not only facilitates optimized resource allocation but also promotes energy efficiency and sustainability in domestic energy consumption.

The structure of this thesis delves into fundamental concepts of deep learning (Chapter 1), contextualizes energy consumption and classification challenges (Chapter 2), details the Experimental setup employed (Chapter 3), and presents experimental findings (Chapter 4). Through this research, we contribute to advancing the field of deep learning in energy management, offering insights and solutions that are crucial for achieving a balanced and sustainable energy future.

Despite the strengths demonstrated by our deep learning model, certain limitations were identified in capturing sudden spikes in power consumption, predicting unusual or extreme events, and fully representing short-term fluctuations. Addressing these challenges requires enhancing feature engineering, adopting advanced time series techniques, refining model architectures, incorporating anomaly detection, expanding the training dataset, optimizing hyperparameters, and improving resource management. By implementing these solutions, we aim to enhance the model's ability to accurately predict power consumption across a wider range of scenarios while maintaining its current strengths.

In conclusion, this research provides a comprehensive framework for using deep learning to manage and optimize residential energy consumption in Algeria. By addressing the identified limitations and adopting the proposed solutions, we can move closer to achieving a sustainable and efficient energy future.

Bibliography

[1] L. Massidda and L. Marrocu, “Deep learning and smart meter data: An effective combination for highly granular load forecasting,” *Energy*, vol. 180, pp. 467–480, 2019.

[2] A. Reinhardt *et al.*, “Electricity load profiling of private households for user behavior modeling,” *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 345–357, 2014.

[3] X. Wang, W. Liu, C. Zhang, H. Jiang, and W. Chen, “Deep learning for short-term load forecasting: A review and future directions,” *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1386–1397, 2020.

[4] X. Wang, W. Liu, C. Zhang, H. Jiang, and W. Chen, “Deep learning-based anomaly detection for smart energy systems: A review,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5890–5900, 2020.

[5] W. Chen, Q. Zhang, J. Li, Z. Wang, Z. Jiang, and Z. Wang, “Deep learning-based energy consumption prediction in smart grids: A review,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3450–3460, 2019.

[6] J. Li, Q. Zhang, Z. Wang, Z. Jiang, W. Chen, and Z. Wang, “Deep learning techniques for residential energy disaggregation: A review,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2700–2712, 2021.

[7] Q. Zhang, J. Li, Z. Wang, Z. Jiang, W. Chen, and Z. Wang, “Deep reinforcement learning for home energy management: A review,” *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3089–3100, 2021.

[8] L. Wu, C. Zhang, X. Wang, H. Jiang, and W. Chen, “Deep learning for smart grids: A comprehensive review,” *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2089–2101, 2021.

[9] W. Liu, C. Zhang, X. Wang, H. Jiang, and W. Chen, “Deep learning techniques for smart home energy management: A comprehensive review,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5462–5474, 2021.

[10] D. Thompson, *Fundamentals of Artificial Neural Networks and Deep Learning*. Springer, 2024.

[11] M. A. Nielsen, *Neural Networks and Deep Learning*. San Francisco, CA: Determinant, 2015.

[12] Y. S. R. Tom Hope and I. Lieder, *TensorFlow for Deep Learning*. O’Reilly Media, 2017.

[13] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, November 2016. Available online: http://www.deeplearningbook.org.

[14] Coursera, "What is a feedforward neural network?," 2024. Available at: https://www.coursera.org/articles/feedforward-neural-network.

[15] DeepAI, "Feed forward neural network definition," *DeepAI*, 2021.

[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[17] C. Yang, *Recurrent Neural Networks for Beginners.* Nova Science Publishers, 2018.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[20] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] H. Tebani, *Production de l'Energie Electrique.* Algiers, Algeria: Librairie El-Djazair, 2nd ed., 2019.

[23] "World electricity production 1990-2022," 2022. Accessed: July 2024.

[24] A. Reinders, K. Voss, and K. Berkel, *Domestic energy consumption: Theory and application of parameter estimation.* Earthscan, 2013.

[25] W. Liu and C.-C. Hsu, "A comprehensive overview on demand side energy management towards smart grids: challenges, solutions, and future direction," *Energy Informatics*, 2015.

[26] E. Iskandarnia, H. Al-Amal, and W. Al-Madany, "Load forecasting in different scale and horizon-a review," *3rd Smart Cities Symposium (SCS 2020)*, vol. 2020, pp. 1–10, 2020.

[27] S. Rajasegarar *et al.*, "Non-intrusive load monitoring approaches for disaggregated energy sensing: A survey," *Sensors*, vol. 12, no. 12, pp. 16838–16866, 2012.

[28] J. Smith and E. Jones, "Towards trustworthy energy disaggregation: A review of challenges, methods, and perspectives for non-intrusive load monitoring." https://arxiv.org/abs/2207.02009, 2022. arXiv preprint arXiv:2207.02009.

[29] D. S. Srinivasan *et al.*, "A comparison of energy consumption prediction models based on neural networks and support vector machines," *Journal of Intelligent Manufacturing*, vol. 29, no. 6, pp. 1355–1368, 2018.

[30] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequenceto-point learning with neural networks for nonintrusive load monitoring," in *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[31] D. Murray, L. Stankovic, and V. Stankovic, "An electrical load measurements dataset of united kingdom households from a two-year longitudinal study," *Scientific Data*, vol. 4, p. 160122, 2017.

[32] J. Kelly and W. Knottenbelt, "The uk-dale dataset, domestic appliance-level electricity demand and whole-house demand from five uk homes," *Scientific Data*, vol. 2, p. 150007, 2015.

[33] J. Z. Kolter and M. J. Johnson, "Redd: A public data set for energy disaggregation research," in *Workshop on Data Mining Applications in Sustainability (SIGKDD)*, vol. 25, (San Diego, CA), pp. 59–62, Citeseer, 2011.

[34] M. D'Incecco, S. Squartini, and M. Zhong, "Transfer learning for non-intrusive load monitoring," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1254–1263, 2020.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems*, vol. 27, pp. 3104–3112, 2014.

[36] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM International Conference on Embedded Systems for Energy-Efficient Built Environments*, 2015.

Abstract,Résumé

This thesis investigates the classification of household energy consumption using deep learning techniques, aiming to optimize energy management amidst rising demands and stagnant energy reserves in Algeria.

The study begins by exploring deep learning fundamentals and progresses to contextualize energy consumption challenges. Methodologically, it focuses on dataset collection, preprocessing, and experimental setup involving REFIT and UK-DALE datasets.

Results from classification experiments underscore the model's strengths and limitations in predicting consumption patterns.

The research highlights the need for enhanced feature engineering, advanced time series techniques, and model refinements to overcome identified challenges. Ultimately, this work contributes to advancing energy efficiency and sustainability through innovative deep-learning applications in residential energy management.

Key Words : Classification, Energy, Deep-Learning, REFIT, UK-DALE, Energy Management. Energy Efficency.

Résumé :
Cette thèse explore la classification de la consommation d'énergie des ménages en utilisant des techniques d'apprentissage profond, dans le but d'optimiser la gestion de l'énergie face à une demande croissante et des réserves énergétiques stagnantes en Algérie.

L'étude commence par explorer les fondamentaux de l'apprentissage profond et progresse pour contextualiser les défis de la consommation d'énergie. Méthodologiquement, elle se concentre sur la collecte des données, le prétraitement et la configuration expérimentale impliquant les ensembles de données REFIT et UK-DALE.

Les résultats des expériences de classification soulignent les forces et les limitations du modèle dans la prédiction des habitudes de consommation.

La recherche met en évidence la nécessité d'une ingénierie des caractéristiques améliorée, de techniques avancées de séries temporelles et de raffinements du modèle pour surmonter les défis identifiés. En fin de compte, ce travail contribue à promouvoir l'efficacité énergétique et la durabilité grâce à des applications innovantes d'apprentissage profond dans la gestion énergétique résidentielle.

Mots-clés : Classification, Énergie, Apprentissage Profond, REFIT, UK-DALE, Gestion de l'Énergie, Efficacité Énergétique.