FACULTY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Reducing Home Energy Consumption with Load Disaggregation

Prepared by

Nour Rabee' 1191035
Lojain Abdalrazaq 1190707

Under the Supervision of Dr. Abdalkarim Awad

**A graduation project submitted to the
Department of Electrical and Computer Engineering
in partial fulfillment of the requirements for the degree of B.Sc. in
Computer Systems Engineering**

BIRZEIT
JUNE 2024

Contents

List of Figures

List of Tables

List of Abbreviations

UKDALE: UK Domestic Appliance-Level Electricity dataset.

NILM: Non-Intrusive Load Monitoring.

ILM: Intrusive Load Monitoring.

Conv-BiLSTM: Convolutional Bidirectional Long Short-Term Memory.

PTP-Net: Principal Temporal Pooling Network.

OECD: Organization for Economic Cooperation and Development.

CNN: Convolutional Neural Network.

GHG: Greenhouse Gases.

LD: Load Disaggregation.

Seq2Seq: Sequence to Sequence.

ML: Machine Learning.

AI: Artificial Intelligence.

SVM: Support Vector Machine.

SVR: Support Vector Regression.

FSM: Finite State Machine.

RNN: Recurrent Neural Networks.

ANN: Artificial Neural Networks.

TP: True positives.

TN: True negatives.

FP: False positives.

FN: False negatives.

MCC: Matthews correlation coefficient.

MAE: Mean Absolute Error.

$\mathbf{R^2}$: R squared.

MSE: Mean Squared Error.

RMSE: Root Mean Square Error.

HDF5: Hierarchical Data Format version 5.

Abstract

The growing environmental impact and financial stress from power consumption in homes, along with the rising demand for electricity that increases carbon emissions and financial strain on households, prompted us to seek a smart solution. Our aim is not only to reduce power consumption but also to encourage responsible electricity use by promoting a change in people's behavior.

To address this challenge, the research employed the concept of load disaggregation as a key method. By examining the power usage of individual devices using the UKDALE dataset, which utilized both smart meters and other advanced monitoring technologies to collect information on residential energy use, the study aimed to provide comprehensive insights into power usage patterns. This approach, centered on load disaggregation, formed the foundation for empowering consumers with knowledge to optimize energy use, reduce costs, and mitigate environmental impact.

This study shows how our solution will help manage home energy use sustainably and cost-effectively by comparing two models, PTPNet and Conv-BiLSTM, and planning to enhance them in the future to further reduce energy consumption. This comparison helps identify the best ways to understand and optimize energy use, promoting more informed and responsible energy habits.

المستخلص

لمواجهة التحديات البيئية المتزايدة والضغط المالي المرتبط بالاستهلاك الكهربائي في المنازل، والذي يزيد من انبعاثات الكربون ويضع ضغطًا ماليًا على الأسر، قمنا بالبحث عن حلول ذكية. هدفنا ليس فقط تقليل استهلاك الطاقة، بل أيضًا تشجيع الاستخدام المسؤول للكهرباء من خلال تعزيز تغيير في سلوكات الاستهلاك لدى الأفراد.

اعتمدت الدراسة على مفهوم تفكيك الأحمال كطريقة رئيسية لمواجهة هذا التحدي. باستخدام مجموعة بيانات UKDALE ، التي تشمل العدادات الذكية وتقنيات المراقبة المتقدمة، جمعت الدراسة معلومات حول استخدام الطاقة السكنية، وتحليل مقدار الطاقة المستخدمة لكل جهاز في المنزل. هدفت الدراسة إلى توفير رؤى شاملة حول أنماط استخدام الطاقة، مما يمكّن المستهلكين من اتخاذ قرارات مستنيرة تؤدي إلى تحسين استخدام الطاقة، وتقليل التكاليف، والحد من الأثر البيئي.

تبرز هذه الدراسة كيف يمكن لحلنا أن يساهم في إدارة استخدام الطاقة في المنازل بشكل مستدام وفعّال من حيث التكلفة. قمنا بمقارنة نموذجين رئيسيين، PTPNet و Conv-BiLSTM، وأظهرت النتائج تفوق النماذج في توقع حالة الأجهزة واستهلاك الطاقة مقارنةً بالأداء المقدم في الأبحاث السابقة. تساعد هذه المقارنة في تحديد أفضل السبل لتحسين استخدام وفهم الطاقة، وتعزيز الوعي والمسؤولية نحو الطاقة.

Chapter 1: Introduction

1.1 Brief Overview of Smart Meters & Energy Disaggregation

Each appliance has its own distinctive power pattern. However, in many homes, multiple appliances run simultaneously, creating a combined pattern that looks like noise. Disaggregation is the method of sorting through this mixed energy signal from a household to identify individual devices. Through disaggregation, consumers can obtain clearer insights into how they are utilizing their energy budget effectively [1].

To illustrate this concept, consider a common cocktail party problem. Imagine walking down a busy street surrounded by sounds coming from all directions, and we need to separate each sound independently. For example, we separate the sound of people talking to each other, sound of cars passing in the road, and the sound of animals such as dogs barking. The load disaggregation is the same, where multiple appliances in homes contribute to a blended energy noisy signal, it seeks to tease apart the energy consumption patterns of individual appliances from the overall household usage [2].

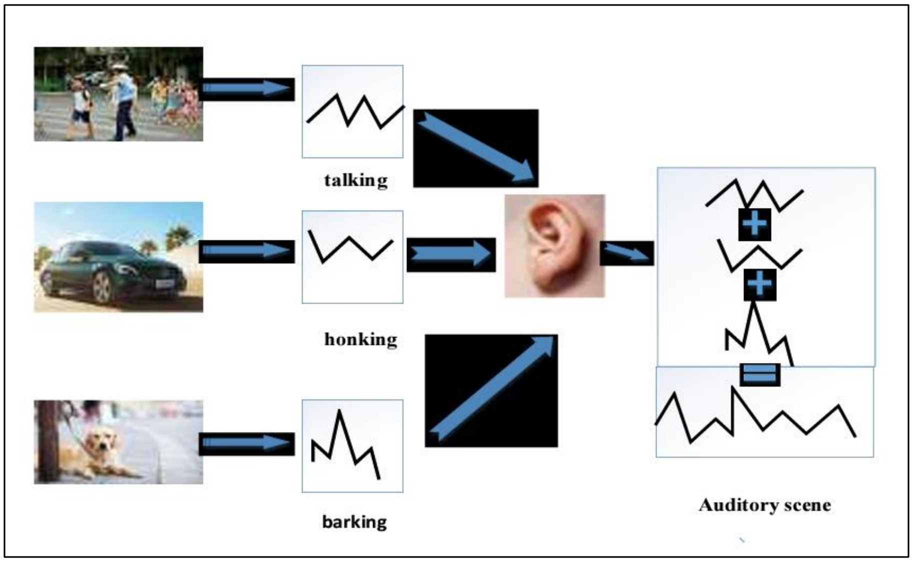

Figure 1: Cocktail problem diagram [2].

Energy disaggregation, alternatively termed as non-intrusive load monitoring (NILM), refers to a computational method used to estimate the power consumption of individual appliances. Employing a single smart meter, which measures the collective demand of numerous appliances, enables this process. An application of this technique is the creation of detailed electricity bills from a single smart meter that includes the entire household [1]. Regular meters, in contrast, only display the total energy consumption of the entire building. Without an upgrade to smart meters, it becomes difficult to pinpoint the energy usage of specific areas or devices within the building.

The overarching objective of NILM could be to assist users in reduction their energy consumption, aiding operators in grid management, identifying broken appliances, or studying appliance usage patterns [1]. One notable advantage of the NILM methodology is its independence from physical sensors at each monitored electrical appliance [3]. This feature is especially to individual users because installing special sensors for each device can be considered as an invasion of privacy and property. NILM's ability to provide detailed insights into energy consumption without the need for intrusive physical sensors contributes to its growing popularity among users.

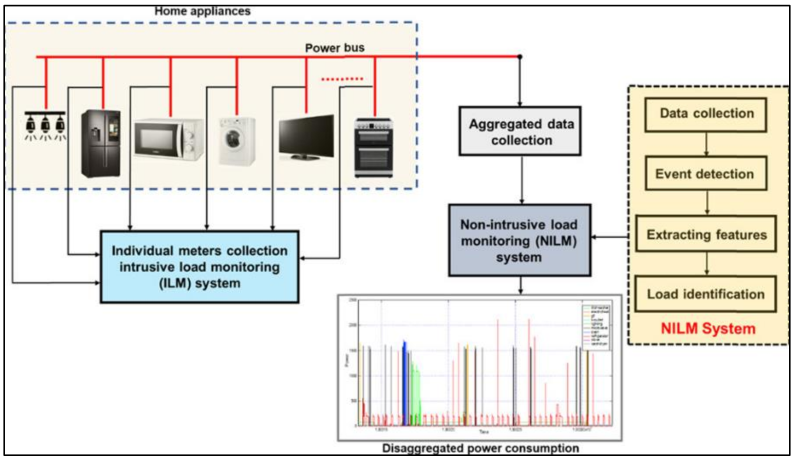

Figure 2: Nonintrusive load monitoring system architecture. [12]

Figure 2 shows the number of publications per year for the research topic "non-intrusive load monitoring" in the Science Direct database [4].

The renewed interest in NILM among researchers has been greatly influenced by the advancements in artificial intelligence and enhancements in computer performance.

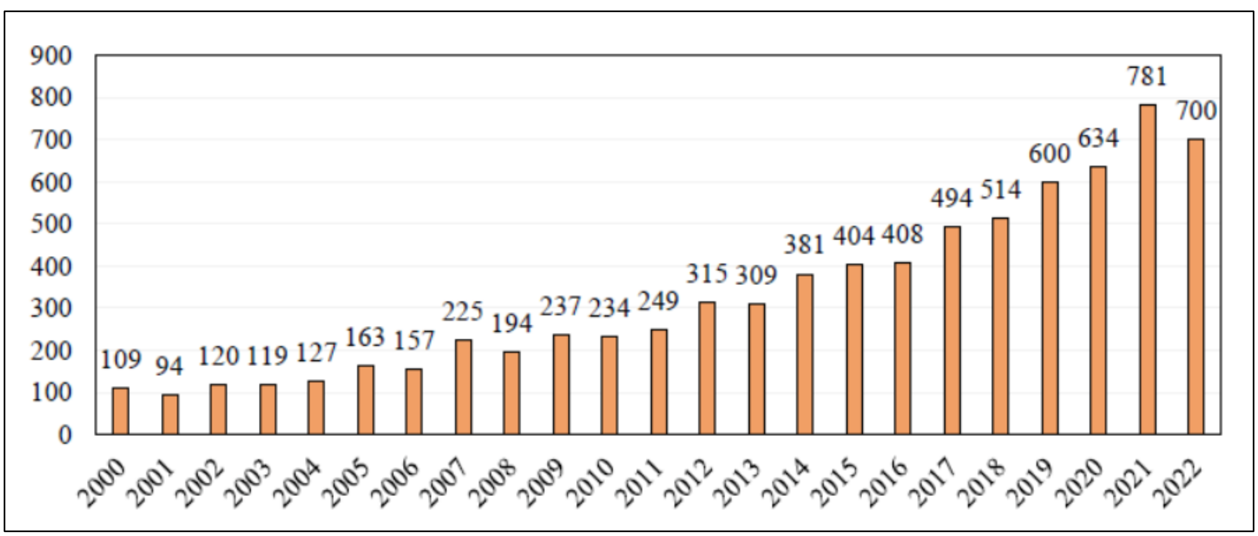

Figure 3: Number of publications on non-intrusive load monitoring from the Science Direct database Source [4].

1.2 Motivation

Energy plays a crucial role in driving the economic development of countries, enhancing people's standards and quality of life, and serving various purposes over the past few decades. As per the U.S. Energy Information Administration (EIA, 2019), global energy consumption is projected to rise by almost 50% from 2018 to 2050. Non-OECD countries are anticipated to account for 70% of this surge, with OECD countries contributing 15%. Nevertheless, the persistent growth in energy consumption raises concerns about future energy supply instability and poses serious environmental challenges [5].

The political situation in Palestine adds an extra layer of complexity to their challenges. A significant issue is the heavy reliance on Israeli sources for electricity, gas, and fuel. This dependence makes electricity very costly, putting a strain on the finances of households [6]. Moreover, higher energy consumption has led to greater CO_2 emission and greater environmental damage. These facts present challenges to Palestinian decision-makers in formulating and implementing the energy strategy for Palestine [7].

The overall objective is developing strategies that empower consumers to minimize power consumption, particularly in the context of home energy usage. Our study focuses on the integration of smart meters with the concept of load disaggregation. By implementing this innovative approach, consumers gain detailed insights into their power consumption patterns over varying timeframes, such as a day, a month or a year. Moreover, they can track and receive updates on the corresponding cost savings, creating a tangible connection between energy usage and financial benefits, by providing immediate feedback through notifications. Load monitoring is one of the ways to effectively manage electricity consumption in households, as it can result in significant energy saving. In fact, visualizing energy resources could save about 4–12% of electrical energy each year. It's important to note that our approach prioritizes consumer privacy, ensuring that sensitive information is securely managed within the local smart meter system. This transparency motivates consumers to adopt energy-efficient practices continuously, cultivating a sustainable and cost-effective approach to home energy management.

1.3 Problem Statements

<u>1.3.1 Financial Strain</u>

The high financial costs associated with importing electricity, gas, and fuel contribute to a significant economic burden on Palestinian territories, affecting both individual consumers and the overall economy.

While the electricity prices in Palestine considered to be the highest prices in the MENA region, having a high portion of household income, the country currently faces the highest price in the region due to its complete reliance on energy imports [8].

<u>1.3.2 Environmental Impact</u>

Today, most of the electricity in the United States is made in power plants that use fossil fuels like coal and natural gas, biofuels, or nuclear energy. These power plants heat water to make steam, which spins a turbine to create electricity. This electricity goes into the power grid. When we burn fossil fuels for electricity, we make greenhouse gas emissions that add to climate change [9].

Most of the electricity made in big facilities in the US is from coal or natural gas. Some also comes from nuclear power plants. All this electricity-making creates about a third of the US's carbon dioxide emissions from energy. This is the biggest source of these emissions in the country. When you make your home more energy-efficient, you use less electricity, so you rely less on power plants that make a lot of carbon dioxide. This means your home needs less power from these plants, which helps the environment by lowering carbon dioxide emissions [9].

1.4 Methodology

The methodology outlines our working approach which is to implement and develop multiple load disaggregation deep learning-based techniques to disaggregate and estimate the power consumption from the total aggregated power for the target appliances. It is worth mentioning that, the load disaggregation method doesn't reduce the power consumption directly, but by breaking down the total aggregated power into appliance-level power, we are providing the user insights into their current appliances' power usage. As a result, the users could know the power consumed for each appliance, so they can manage their usage to reduce their power consumption and bills.

We begin by using the "UKDALE" dataset, which is an open-access dataset from the UK recording Domestic Appliance-Level Electricity to conduct research on various disaggregation algorithms, with data describing the aggregated power consumption and the ground truth power of individual appliances. It was built at a sample rate of 1/6 Hz for both the whole-house power consumption and individual appliances. The dataset consists of data from 5 houses, recorded over different periods [10].

Then, we preprocess and explore the dataset to gain insights into its structure and the behavior of various appliances in each household. We focus on training multiple approaches based on deep learning and neural networks on the UKDALE dataset, with the aim of leveraging their capabilities to learn from the data and accurately predict appliance-level energy consumption patterns. One of the models used and developed, is a Convolutional Neural Network (CNN) model based on temporal pooling (NILM-TP) for load disaggregation. Additionally, we used another model, the Convolutional LSTM (CONV-LSTM), which integrates a 1D convolutional layer with bidirectional LSTM layers, aiming to disaggregate the total load and effectively address the task of load disaggregation.

The trained deep learning models are tested to assess their ability to accurately estimate the power consumption and the state of each appliance, and make predictions. Our performance analysis includes comparing predicted energy consumption against ground truth data for several types of appliances used in the house, allowing us to identify the most effective approach based on their respective performances.

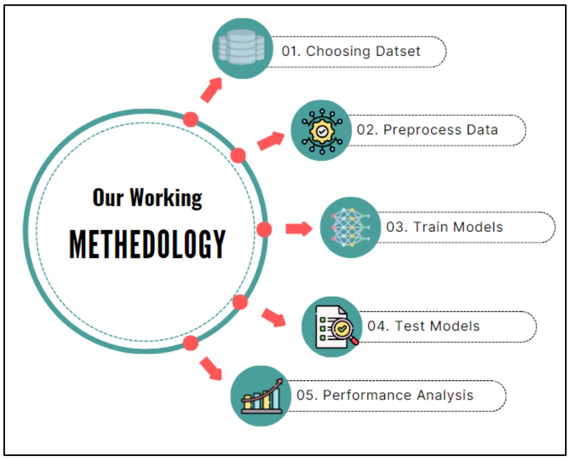

Figure 4: Methodology.

1.5 Organization of the Report

The report is structured into six chapters, each addressing different facets of the proposed work. The first chapter, the Introduction, offers an overview of the report, presenting background information on the problem, the methodology used in this project in an abstract manner, motivation, and the problem statement. Chapter 2, Literature Review, examines previous studies and research dealing with similar issues.

In chapter 3, we provide foundational knowledge on smart metering, discussing their definition, benefits, and practical use for saving energy and reducing environmental impact. We explore the environmental benefits of smart meters and their role in promoting energy efficiency.

Then, in chapter 4, we focus on the fundamentals of NILM technique, including background information, and types of appliances. Also, we gradually discuss the NILM algorithms, from traditional models to the most recent neural networks used in load disaggregation tasks. Throughout this discussion, we address the challenges and limitations that follow the usage of traditional models, and why we are shifting to the new deep learning models and architectures instead of using traditional ones. Moreover, we cover the deep-learning based models used in our project, which are integrated in the implementation and development our study. The chapter concludes with an evaluation of NILM algorithms, highlighting their performance and efficiency measures.

In Chapter 5, we discuss the practical implementation of selected NILM architectures and networks. We detail the process of reading, exploring, and constructing the UKDALE dataset, utilizing PyTorch as our primary training environment. This chapter also covers preprocessing steps, and dataset splitting into Training, Validation, and Testing sets, as well as the implementation of model architectures specific to our study. We thoroughly examine the training and testing procedures, discuss the final results, and compare our models' performance with other studies.

In Chapter 6, we discuss the challenges faced and outline future work, including data limitations and efforts to enhance models for more accurate predictions. We also introduce plans for a user-friendly application to help consumers adopt better energy-saving habits.

Chapter 2: Literature Review

This section of the report reviews related studies that discuss and introduce previous work conducted in the field of Non-Intrusive Load Monitoring (NILM). It highlights various methods for load disaggregation, focusing on the most utilized and effective algorithms for identifying appliances and determining their energy consumption based on aggregated usage data within households or buildings.

2.1 Load Disaggregation via Pattern Recognition: A Feasibility Study of a Novel Method in Residential Building

This paper presents a novel approach to non-intrusive load disaggregation using Hidden Markov Models (HMMs) for residential appliances. The method involves training the HMM with measured electric current data obtained from major appliances, such as laptops, refrigerators, TVs, and microwaves, during their operational phases. The training process includes estimating initial values for state transition probabilities and emission probabilities using the Baum–Welch algorithm. The trained HMM is then employed to predict the electric current for each appliance, and the results are compared with the actual measured data. The matching rate is used to assess the accuracy of the predictions, and the process continues iteratively until a predefined matching rate is achieved or a set number of repetitions is reached. The paper highlights the significance of considering various factors, such as appliance usage patterns and the need for modifying training data, to enhance the accuracy of load disaggregation, showcasing the potential of HMMs in addressing the challenges of appliance recognition and energy consumption analysis in residential settings [11].

2.2 A systematic approach in load disaggregation utilizing a multi-stage classification algorithm for consumer electrical appliances classification

This study presents a systematic approach for classifying individual electrical appliances using load disaggregation through V-I trajectory-based load signatures. The research employs a multi-stage classification algorithm, combining principal component analysis (PCA) and the k-nearest neighbor (k-NN) methodology. The PCA helps recognize patterns in images, and the k-NN decides which appliance category an image belongs to. What makes this method special is the use of the "k-value," which helps in efficient classification. The research successfully identifies different appliances using a supervised learning approach and suggests potential areas for future research, especially exploring unsupervised learning along with PCA [3].

2.3 Towards energy-efficient smart homes via precise nonintrusive load disaggregation based on hybrid ANN–PSO

This study aims to enhance the energy efficiency of smart homes by employing a method known as Nonintrusive Load Monitoring (NILM). The research introduces a sophisticated blend of two algorithms, namely Particle Swarm Optimization (PSO) and Artificial Neural Network (ANN), working in tandem to improve the precision of NILM. Notably, this approach can predict the energy consumption of each appliance by analyzing the overall power usage in the entire home.

This streamlined approach eliminates the need for a separate smart meter for each appliance, simplifying the system and reducing costs. The study demonstrates the superior performance of this method compared to alternative approaches, significantly reducing errors. Furthermore, the research delves into household energy usage patterns, offering insights into energy-saving practices, such as optimizing the timing of specific appliance usage. In summary, this research contributes to the development of smarter and more energy-efficient homes [12].

2.4 Improving Residential Load Disaggregation for Sustainable Development of Energy via Principal Component Analysis

In this paper, the aim is to enhance the sustainability and efficiency of energy systems through the method of non-intrusive load monitoring (NILM). The process lies in understanding the patterns of energy consumption by various household appliances. The proposed approach of this paper uses the unsupervised approach, employing principal component analysis (PCA) for dimensionality reduction, to identify and categorize the power consumption patterns of these appliances. The method of load disaggregation is tested using real-world data from the reference energy disaggregation dataset (REDD), showcasing its accuracy and effectiveness compared to traditional NILM methods. The results highlight the transparency and clarity of displaying power consumption patterns in a simplified, low-dimensional space, providing consumers with valuable insights into their energy usage [13].

2.5 An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges

This paper explores the concept of Non-Intrusive Load Monitoring (NILM), a crucial aspect of effective energy management. NILM, in contrast to intrusive methods, offers cost-effective and easily scalable solutions for obtaining detailed energy consumption insights. The authors provide a comprehensive survey covering the NILM system framework, advanced load disaggregation algorithms, load signature models, datasets, and performance metrics. They delve into practical applications, including demand response, energy efficiency, personalized energy management, and equipment diagnosis. The paper also addresses the challenges faced by NILM, emphasizing the need for a low-cost and robust framework to ensure scalability. Despite three decades of development, scalability remains a significant challenge for NILM, highlighting the importance of ongoing research and exploration of practical business models for large-scale deployment [14].

2.6 Non-Intrusive Electrical Load Monitoring in Commercial Buildings Based on Steady-State and Transient Load-Detection Algorithms

This paper focuses on non-intrusive electrical load monitoring (NILM) in commercial buildings, aiming to detect and analyze individual loads for improved energy management. In addition, it discussed the challenges in applying NILM to commercial spaces and presented their results, emphasizing the use of steady-state and transient load-detection algorithms. The paper highlighted the potential of using NILM for cost-effective data acquisition, and its relevance in many tasks such as shut-up detection, and calculation of energy consumption. Finally, the paper outlined future enhancements, such as improved parallel processing and real-world testing, reinforcing the significance of NILM in advancing energy efficiency in commercial buildings [15].

2.7 Power Disaggregation of Combined HVAC Loads Using Supervised Machine Learning Algorithms

This paper introduced a novel power disaggregation technique focused on the challenge of distinguishing power usage data from multiple identical HVAC compressors monitored by a single power meter in a commercial setting. This was done by using parameters, such as the one-minute interval real and reactive power of the compressors and air handlers, in addition to phase currents of both. The testing of the approach was done by using four supervised machine learning algorithms, which are Decision Trees (DT), Discriminant Analysis (DA), Support Vector Machine (SVM) and k-Nearest Neighbors (k-NN). The result showed that the K-NN algorithm emerges as the most efficient for solving the problem of aggregated power disaggregation. In addition, it presented the limitations of this work, including the need for constant monitoring, challenges in distinguishing similar load characteristics, and occasional discrepancies due to air handler units' post-compressor operation. Despite these limitations, the method extends beyond HVAC compressors, offering a practical approach for researchers dealing with similar disaggregation challenges in various device settings [16].

2.8 An Evaluation of NILM Approaches on Industrial Energy-Consumption Data

This paper investigates non-intrusive load monitoring (NILM) techniques for energy disaggregation, focusing on industrial data, which has been less explored compared to household settings. This work has the potential for significant energy savings in the industrial sector, responsible for a large portion of global energy consumption. The authors make three main contributions; firstly, they compare energy consumption data from two facilities and households; second, they assess the performance of seven load disaggregation algorithms on industrial data, contrasting it with household data; third, they provide a tool to convert an industrial dataset to a standard format for load disaggregation, promoting further research and benchmarking. The study finds differences in how energy is used in industrial settings, tests how well different methods work for different appliances, and shows the need for standards in this area. The research sets the stage for future work to improve energy monitoring methods and use industry knowledge to make them better [17].

2.9 Non-intrusive Load Monitoring based on Convolutional Neural Network with Differential Input

This paper introduces Non-intrusive Load Monitoring (NILM) as a cost-effective and flexible process for analyzing energy consumption in buildings by figuring out which appliances are running and how much energy they consume. The authors propose a system for managing energy usage using NILM and introduce a neural network model with a unique input method, demonstrating superior performance compared to existing models with raw input data. The experiments conducted validate the effectiveness of the proposed model. Overall, the research contributes to the enhancement of energy management in smart grids, with a specific focus on supporting energy-related industrial services [18].

2.10 Building Energy Consumption Prediction: An Extreme Deep Learning Approach

The paper discusses the prediction building energy consumption, a critical aspect of efficient energy management. It introduces an innovative method called extreme deep learning, combining stacked autoencoders (SAEs) and extreme learning machines (ELM). SAEs are responsible for extracting energy consumption features, while ELM serves as an accurate predictor. By comparing to traditional methods such as neural networks and linear regression, the efficacy of this approach is demonstrated. The study emphasizes the power of deep learning in accurately forecasting energy consumption, particularly when faced with cyclic changes. Future research aims to further improve the accuracy of predictions by integrating both data and periodic knowledge into the deep neural network [19].

2.11 Predicting Residential Energy Consumption Using CNN-LSTM Neural Networks

The paper discusses a new way to predict how much energy homes will use. It introduces a model called CNN-LSTM that combines two types of neural networks to analyze both the patterns and changes in energy use over time. The proposed model effectively combines spatial and temporal features, outperforming existing forecasting methods. It achieves almost perfect prediction performance and records the smallest root mean square error compared to conventional methods for household power consumption datasets. The model analyzes influential variables, such as electric water heaters and air conditioners, and demonstrates stable and efficient predictions, especially for irregular changes in electricity consumption. The study suggests that accurate energy predictions are important for managing power supply and reducing costs [20].

2.12 Non-Intrusive Load Disaggregation by Convolutional Neural Network and Multilabel Classification

This paper introduced a novel methodology for Non-intrusive load monitoring (NILM) approach for monitoring the energy consumption of a residential building by developing a convolutional neural network model that uses a temporal pooling block to enhance the feature space without compromising temporal resolution. Their approach aimed to predict the status of target appliances and estimate their power consumption. By leveraging techniques from semantic image segmentation, their model effectively identifies the activation states of household appliances. This method demonstrated strong generalization and achieved good results compared to other neural network solutions for NILM, performing well on a reference dataset with unseen households [21].

2.13 Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a Transformer Based Attention Mechanism and Temporal Pooling

This paper proposed a deep learning model for non-intrusive load monitoring (NILM) using a transformer-based attention mechanism and temporal pooling. The model aims to disaggregate household energy consumption into individual appliance usage. Innovations such as attention mechanisms, temporal pooling, residual connections, and transformers enhance its accuracy and precision. Tested on UK-DALE, REDD, and REFIT datasets, the model outperformed existing methods. Future work suggests adding features like time and weather conditions, testing on diverse datasets, and extending to multitask learning for better energy management [22].

2.14 Neural NILM: Deep Neural Networks Applied to Energy Disaggregation

This paper explored the application of deep neural networks to non-intrusive load monitoring (NILM) to estimate appliance-level electricity consumption from aggregate household power data. The authors proposed three neural network architectures long short-term memory (LSTM), denoising autoencoders(DAE), and a regression-based network to this task. The models were tested on data from five appliances, with results indicating that all three neural networks achieved better F1 scores than traditional methods like combinatorial optimization and factorial hidden Markov models. It was notable, the models generalized well to houses not seen during training. The study highlighted the potential of deep neural networks for NILM but acknowledges the need for further research, particularly in handling multi-state appliances with long gaps between significant power events [23].

2.15 Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks

This paper presented a sliding window approach using artificial neural networks (ANNs) for real-time energy disaggregation, extracting appliance-level power consumption from aggregate data. The authors adapted two recurrent network architectures one with Gated Recurrent Unit (GRU) neurons and another with Long Short-Term Memory (LSTM) neurons that uses the sliding window for real-time energy disaggregation, and compared their performance using six metrics. They found that their approach performs better for multi-state devices. Additionally, the study compared ANNs using Gated Recurrent Unit (GRU) neurons and Long Short-Term Memory (LSTM) neurons, concluding that both perform equally well [24].

2.16 Energy Disaggregation Using Variational Autoencoder (VAE)

This paper introduced an energy disaggregation method using variational autoencoders (VAEs) for Non-Intrusive Load Monitoring (NILM). VAEs enhance the encoding of appliance consumption data by leveraging a probabilistic encoder to reconstruct complex load profiles, particularly benefiting multi-state appliances. The model evaluated on UK-DALE and REFIT datasets, the proposed approach demonstrated competitive performance compared to existing NILM methods [25].

Chapter 3: Smart Metering: Foundations and Context

In this chapter, we set the foundation for our study by exploring the fundamental aspects of smart metering. Before applying algorithms to UKDALE set and explain energy data, it's important to understand the core of the data source itself which are smart meters.

3.1 What are smart meters?

A Smart Meter is a device that electronically measures and records real-time data on electricity, gas, or water usage. Then, it sends this information to the utility provider, typically the electricity company, using connected sensors. This allows the utility company to access information directly from the meters, eliminating the need for manual checks of installations to bill customers and manage their service. The smart meter functions similarly to a traditional meter but has advanced features including two-way communication which means that smart meters can both send information to the utility provider (electricity company) and receive instructions or updates from them. In addition, smart meters gather detailed data about energy usage patterns over time. In contrast, the traditional meter can only measure the total amount of energy used and do not have the ability to communicate with the utility provider or collect detailed usage data [26].

Smart meters use either wireless or wired networks to send data directly to the utility provider, removing the necessity for manual meter readings [27]. These meters can capture real-time energy consumption details, such as voltage, phase angle, and frequency, and transmit this information securely. This allows users to track their usage patterns, adapt their behavior to lower consumption, and effectively manage their energy usage [28].

3.2 Benefits of smart meters

The use of smart meters offers a range of benefits for customers, they can be summarized as shown as follows:

- **Provide the customer an increased control over their electricity usage and expanding their options for pricing plans:** With the ability to measure usage in hourly increments, smart meters allow customers to make informed decisions about their electricity consumption and potentially save money through time-based rates. By taking advantage of these rates during peak periods, such as summer afternoons, customers can actively manage their electricity demand and make a positive impact on their monthly bills [29].

- **Give customers detailed information about** their energy consumption and expenses, enabling them to make well-informed choices. This knowledge enables individuals to efficiently manage their electricity usage and ultimately lower their utility bills [29].

  Nowadays, individuals who have Smart Meters at their disposal have the ability to retrieve detailed information about their electricity consumption from the previous day using their utility company's online platform. Also, people can control how much electricity they use and how much they spend by installing a special device at home or installing an app or website which acts as a simulator between the consumers and the smart meter. They can talk to the Smart Meter and shows how much electricity is being used right then and there. So, if electricity prices go up or there is a problem with the electricity system, people can see it right away and decide to use less electricity. They might choose to wait before using big appliances or turn them off completely. They can do this by pressing buttons on the device or setting it up to do it automatically. Also, if they're using too much electricity, they will get a message on their phone or email to let them know. This helps people keep track of their electricity use and make smart choices to save money [29].

- **Increasing privacy:** By enabling automatic transmission of electricity usage data to the utility for billing purposes, this process enhances privacy and eliminates the need for utility personnel to physically check the meter on-site. Not only does this reduce operational costs for the utility, but it also translates to savings for customers, as utility rates reflect their operating expenses. Also, as technology evolves and advances, customers can gain the benefits without needing the utility to replace the meter itself [29].

- **Integration with renewable energy sources:** As renewable energy sources gain popularity and widely being use, smart meters will play an increasingly vital role. They can monitor the output of renewable energy sources like solar panels and modify energy usage accordingly [29].

3.3 How to use a smart meter to save money on your energy bills?

While the installation of a smart meter alone won't automatically reduce your energy costs, it opens up opportunities for informed decision-making to potentially lower the energy bills. The key lies in leveraging the data provided by your smart meter, which tracks your energy usage throughout the day. This information could encourage you and your energy retailer to explore flexible pricing plans or time-of-use prices, an option unavailable with traditional meters [30].

Flexible energy plans typically charge higher rates during peak times, lower rates during less busy periods, and the lowest rates during times of lowest energy demand. By scheduling activities like dishwashing for these off-peak times, you can benefit from lower rates and alleviate strain on the energy system. However, if adjusting the timing of appliance usage isn't feasible, a fixed-rate plan might be more suitable. The main idea is that having access to detailed energy usage information enables you to make decisions that match your consumption patterns, potentially reducing your energy costs [31].

3.4 The environmental benefits of smart meters

Utilizing smart meters and implementing an intelligent grid technology can have a substantial positive impact on the environment. By decreasing reliance on non-renewable fossil fuels, these advancements can effectively decrease the release of damaging greenhouse gases (GHG) and other harmful pollutants into the air [32]. Overall, there are two key ways in which environmental benefits can be achieved through these advancements:

- **Reducing electricity consumption and increasing transmission and distribution efficiency** by promoting both a decrease in electricity usage and an increase in transmission and distribution efficiency, research shows that consumers are more likely to make eco-conscious decisions by frequently monitoring their energy usage and making changes such as turning off unneeded appliances, switching to energy-efficient lighting, and adjusting thermostats. This not only results in a decrease in necessary power production, but also potentially translates into improved air quality due to reduced emissions [32].

- **Reducing utility's vehicular needs:** By implementing smart meters, not only can we decrease utility demands, but we can also significantly lower the use of resources and the release of harmful greenhouse gases and pollutants that come with carrying out essential utility tasks such as connecting and disconnecting services, as well as taking meter readings. This can be achieved without the need for physical visits by utility trucks, as evidenced by CenterPoint Energy's success in avoiding more than 300,000 "trucks rolls" through electronic service orders alone, as of October 2010 [33].

There are so many advantages of using the smart meters on the environment by reducing emissions and harmful pollutants. They help minimize unnecessary appliance use and eliminate the need for utility vehicles, as they can be accessed remotely without manual checks or meter readings [32].

Chapter 4: Load Disaggregation: From Traditional Models to Deep Learning Approaches

This chapter illustrates the foundation for understanding the concept of load disaggregation, its definition, and types. Then, we go through introducing the various types of home appliances involved in this task. After that, we introduce and discuss the evolution of NILM models, following their development from traditional statistical methods to the latest deep learning approaches used in this context. In addition, we highlight the main deep learning models and networks implemented and developed in our study, drawing insights from the literature review.

4.1 Background and Problem Formulation

As mentioned before, Load Disaggregation (LD) is a technique used to disaggregate the total energy consumption of a household into individual appliance-level consumption [34]. In terms of load disaggregation technique, two primary methodologies exist; Intrusive Appliance Load Monitoring (IALM or ILM) and Non-Intrusive Appliance Load Monitoring (NIALM or NILM) [35].

The first methodology is the Intrusive Appliance Load Monitoring (IALM). In this method, we deploy a sensor on each appliance to monitor its energy consumption [36]. As a result, ILM is characterized by a complex hardware setup and a straightforward software process since each targeted appliance in the home is monitored through dedicated hardware for its specific type, and the collected data is centralized through separate data paths [37].

In simple terms, the Non-Intrusive Appliance Load Monitoring, represents a methodology marked by its simplicity in hardware configuration but complexity in software processes [37]. With this method, we analyze and get the power consumption for each appliance running or operating in the home by applying a specific model or algorithm to the total power consumption measured by smart meter. The problem of load disaggregation using NILM can be illustrated as follows: if **y(t)** represents the total active electrical power used in a system at the instant time $\boldsymbol{t}$, $y_i(t)$ represents the active power absorbed by specific appliance in the home, then the total load at a time $\boldsymbol{t}$ represents the sum of all appliances' power consumption as shown in the equation [21]:

$$y(t) = \sum_{i=1}^{N} y_i(t) + e(t), \qquad \forall t \in (0, T)$$

Where $\boldsymbol{N}$ the number of appliances is considered, and $\boldsymbol{e(t)}$ is the unidentified residual load. The problem in NILM is to get the values of $y_i(t)$ while we have only the total power measured at a specific time $\boldsymbol{t}$ by applying the Function F to the $y(t)$ [21]:

$$[y_1(t), y_2(t), y_3, \dots, y_i(t), \dots, y_N(t)] = F(y(t))$$

Where $\boldsymbol{F}$ is an operator or function that when we applied it to the total active power, it returns $\boldsymbol{N}$ distinct values (which is the number of appliances) that are the best estimate of the power absorbed by individual appliances [21].

4.2 Types of Appliances

In order to disaggregate a power consumption based on NILM algorithms, it is important to understand the type of home appliances that produce the total energy consumption signal and measured by the smart meter. Home appliances are categorized into four main types according to the features which are *On/Off*, *Finite State Machine (FSM)*, *Continuously Variable*, and *Always on* appliances [38].

The first type, called On/Off, includes devices like lights and toasters which have two power states: on and off. For example, when a light is turned on, its power level increases from 0 watts (off) to 12 watts (on). The second type, known as Finite State Machine, includes devices like lamps and fans, which have multiple power states. For instance, a lamp's power pattern, shown below, demonstrates varying power levels across different states, illustrating more complex behavior than the simple on/off states of the first type [38].

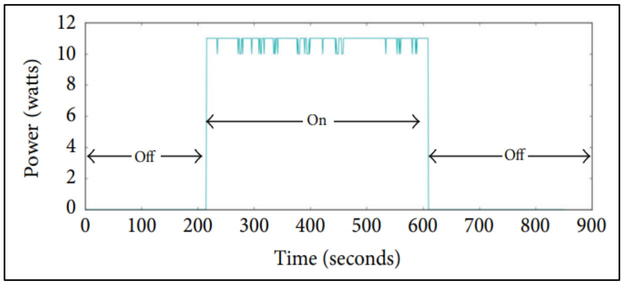

Figure 6: Type I (On/Off): light [38].

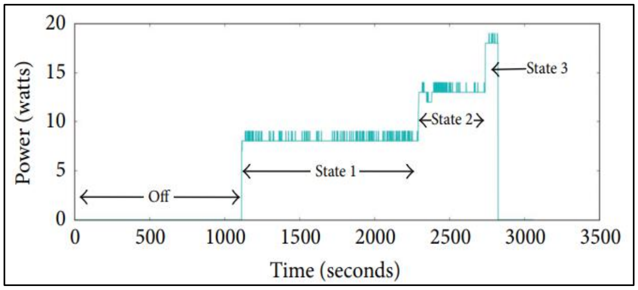

Figure 5: Type II (Finite State Machine - FSM): lamp [38].

The third type of devices called Continuously Variable (or Continuously Changeable) has a continuous changeable power pattern such as a washing machine. In this type, the amount of power changes, goes up and down where heating or drying. Finally, we introduce another type called "Always On." Appliances of this type are constantly running except under specific circumstances. For instance, a refrigerator is typically considered an "Always On" appliance [38].

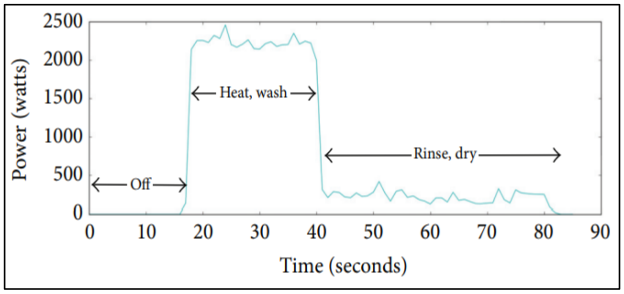

Figure 7: Type III (Continuously Variable): washing machine [38].

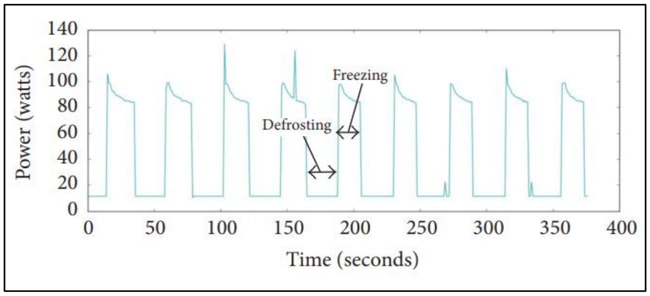

Figure 8: Type IV (Always On): refrigerator [38].

So, when multiple appliances are running at the same time in the home, and the total load generated by these appliances is measured by the smart meter; the load disaggregation techniques become an important operation for understanding and getting the distinct power patterns of individual appliances, enabling a more detailed analysis of energy usage within the household.

4.3 Evolution of NILM Models

Many algorithms have been created to solve the task of disaggregating energy data using the concept of NILM. As mentioned before, these algorithms typically take the total load from multiple appliances at each time point as their input, represented as $p(t) = p_1(t) + p_2(t) + \ldots + p_n(t)$ where $p_1(t)$, $p_2(t)$, ..., $p_n(t)$ are the individual loads of each appliance. The objective is to differentiate between the loads of each appliance, assuming they are additive. The behavior of the total load $p(t)$ varies depending on the type of the appliance being analyzed [39].

During our reading in the related works in this filed, studies have implemented and explored different methods and approaches for energy consumption forecasting which are categorized in three main approaches, statistical-based modeling, machine learning-based modeling, and deep learning-based modeling.

Statistical-based modeling, such as linear regression which predicts the value of unknown data by using another related and known data value, and multiple linear regression which estimates the relationship between a quantities dependent and two or more independent variables. For example, N. Fume in his work [40] used multiple linear regression in energy usage prediction residential in homes. While these approaches are straightforward, their accuracy can be impacted by several factors, including the correlation between the independent variables used for prediction. When independent variables are highly correlated, it can lead to issues such as multicollinearity, where the independent variables are not truly independent of each other. In such cases, the coefficients estimated by the regression model may become unstable or difficult to interpret accurately, which can affect the model's performance and accuracy in predicting the dependent variable.

On the side of machine learning-based modeling, many studies proposed solutions of NILM using many algorithms such as, Support Vector Machine (SVM), Support Vector Regression (SVR), and Random Forest. For instance, Y. Chen in his proposed study [41] used the support vector regression to predict power consumption in office building for a short term, typically within a day or a few days. Also, A. Bogomolov who attempted to predict energy consumption using a random forest regression method [42]. However, both SVR and random forest have limitations. One notable challenge is the potential for overfitting, especially when the model is fine-tuned to fit the training data too closely. Moreover, these algorithms may face challenges with the long-term predictions, where complex correlations between variables or an increase in data volume could lead to difficulties in generalization [43].

In the domain of deep learning-based modeling, Artificial Neural Networks (ANN) have shown a promising solution for the problem of load disaggregation by automatically extracting the complex power features for detailed appliance behavior understanding. Models such as sequence to sequence (Seq2Seq), which designed for sequence-based tasks, has achieved notable performance compared to the existing methods in the field of NILM tasks, although it has some challenges [44]. These kind of models needs substantial labeled data and computational resources during training.

In addition, it may face difficulties in modeling the spatial-temporal features of power consumption together. Spatial features refer to the distribution or arrangement of power consumption across different locations, while temporal features involve understanding how power consumption changes over time [45].

In addition, many researchers have used hybrid architectures within the artificial neural network (ANN) approach. They utilized multiple neural network approaches, such as recurrent neural networks (RNN) and convolutional neural networks (CNN). CNNs capture spatial patterns, while RNNs capture temporal patterns. By integrating these networks, researchers have improved the understanding of energy usage behavior and enhanced the performance of load disaggregation algorithms. For example, the study in [20] proposed a CNN-LSTM neural network that extracts both spatial and temporal features to effectively predict housing energy consumption.

4.4 Deep Learning Models used for NILM

Both deep learning and machine learning (ML) are types of the Artificial Intelligence (AI). Machine focuses mainly on enabling computers to learn from data and make predictions or decisions based on the input data. On the other hand, deep learning represents a subset of machine learning techniques that specifically involve neural networks with multiple layers, allowing for the extraction of complicated patterns from huge amounts of data. While deep learning is a part of machine learning, it deserves a separate category due to its unique architecture [46].

The different types of neural networks in deep learning, such as convolutional neural networks (CNN), recurrent neural networks (RNN), artificial neural networks (ANN), etc. are the core of the deep learning revolution, powering many applications and tasks [47]. In this section, we discuss the models and networks that have been employed and used in out implementation, illustrating their design and functionalities.

Convolutional Neural Network (CNN):

Convolutional neural networks (CNNs) distinguish themselves from other neural networks by their remarkable performance in handling image or signal inputs. CNNs are structured with various layers, including convolutional layers, pooling layers, and fully connected layers [48].

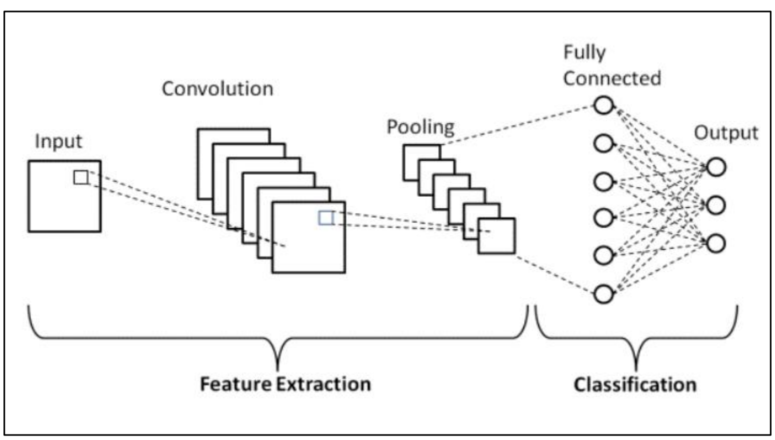

Figure 9: Convolutional Neural Network (CNN) architecture.

- **Convolutional layers**: These layers work by applying convolution operations to the input data, essentially using a filter (also called a kernel) to slide over the input and extract features “convolution operation”. Each filter is designed to detect specific patterns or features within the input data [47].
- **Pooling layers:** These layers are intended to reduce the spatial dimensions “down sampling” of the feature maps produced by the convolutional layers. This operation aids in reducing computational complexity and improves the network's capacity to handle variations in the input data [47].
- **Fully Connected Layers:** Also called Dense Layers, they are responsible for performing classification based on the features extracted in both convolutional and pooling layers [47].

Hence, the CNN consists of two main components: the feature extraction stage, which involves convolution and pooling layers, and the classification and recognition stage, carried out by fully connected layers [47].

In terms of Non-Intrusive Load Monitoring (NILM), CNNs have been effectively employed. The network’s filters can recognize specific short sequences, thereby detecting power variation sequences from the target appliance while filtering out those from non-target appliances. The filter parameters are learned automatically during CNN training. For appliances with complex power dynamics, such as multi-state or time-varying power, increasing the number of network filters helps to learn and extract more diverse power variation features. In the proposed work in [18], the Non-intrusive Load Monitoring based on convolutional neural network with differential input were discussed, and the experiments showed that the results of using the CNN to extract the input load measured by the smart meter outperformed the existing models with raw input.

Long Short-Term Memory Neural Network (LSTM):

Long Short-Term Memory (LSTM) networks are a specialized type of Recurrent Neural Network (RNN) designed to process sequential data, such as time series data. RNNs can retain information over time by feeding the output of each neuron back into the network as input for the next time step. This makes RNNs inherently suitable for tasks involving sequential data. However, traditional RNNs often struggle with the “vanishing gradient” problem, where gradient information diminishes during backpropagation through time, limiting the network’s ability to learn long-term dependencies [23].

LSTM networks address this issue with a unique architecture that includes memory cells and gates to control the flow of information. This structure allows LSTMs to maintain and update long-term memory, making them capable of learning patterns over extended sequences more effectively than traditional RNNs [49]. As a result, LSTMs have been successfully applied to various sequential tasks, such as automatic speech recognition and machine translation.

In the context of non-intrusive load monitoring (NILM), which involves disaggregating total energy consumption data into individual appliance usage, LSTMs are particularly well-suited. They can analyze sequential energy consumption data from smart meters, identifying patterns and disaggregating the total load into its components. This ability to handle long-term dependencies and learn from sequential data makes LSTMs ideal for NILM tasks.

There are many types of LSTM networks used in NILM. One commonly used type is Conv-LSTM, which integrates Convolutional layers for feature extraction, capturing both spatial and temporal features. Another effective variant is the Bidirectional LSTM (BiLSTM), which processes data in both forward and backward directions, allowing the network to have a more comprehensive understanding of the sequence [50]. These architectures have demonstrated superior performance in disaggregating energy consumption data, providing reliable results. For instance, in the work proposed [23]multiple neural network architectures, including Conv- BiLSTMs, were utilized, achieving good performance in NILM task.

4.5 Evaluation of NILM algorithms

In the last section of chapter 4, we introduce the used evaluation metrics and measures for assessing and comparing the performance of our models in the Non-Intrusive Load Monitoring (NILM) task. These metrics fall into two main dimensions: performance evaluation and efficiency evaluation.

4.5.1 Performance Evaluation: This dimension measures how well the model performs its intended NILM tasks, specifically predicting the power consumption of target appliances and their activation status (ON/OFF). Performance evaluation is categorized into two sections based on the task type [51]:

Classification Task: This involves predicting the ON/OFF status of appliances. A confusion matrix provides a more detailed assessment of the performance of various classification models applied to solve a specific task or application. The confusion matrix is a square matrix with dimensions N × N, where N represents the number of target classes. Its primary function is to compare the actual target values with the predicted output values generated by the employed machine learning model [51].

The matrix consists of the following four terms that are used in other accuracy measures:

- **True positives (TP):** These instances happen when both the actual value and the predicted value are positive [51].
- **True negatives (TN):** These instances arise when both the actual value and the predicted value are negative [51].
- **False positives (FP):** These cases occur when the actual value is negative, yet the predicted value is positive [51].
- **False negatives (FN):** These cases occur when the actual value is positive, but the predicted value is negative [51].

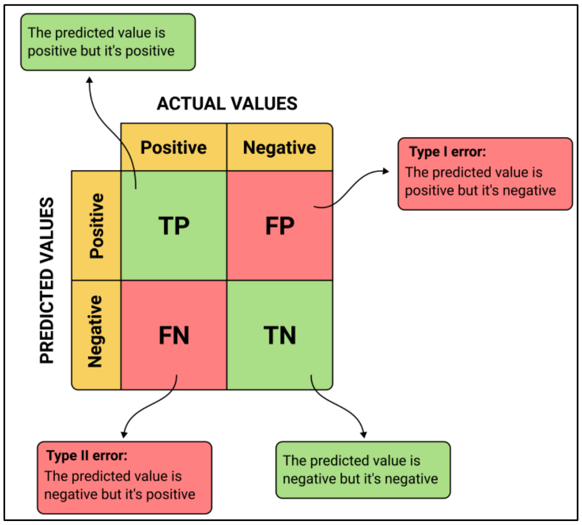

Figure 10: Confusion Matrix for binary classification.

- ***Accuracy***: frequently employed in classification tasks, evaluates the overall correctness of a classification model's predictions by comparing them to the true labels of the samples. It serves as a widely used metric to measure the model's performance [52]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}$$

- ***Precision***: performance measure that evaluates the accuracy of positive predictions made by a classification model. It is calculated using the following formula [52]:

$$Precision = \frac{(TP)}{(TP + FP)}$$

- ***F1-Score****:* this measure combines the precision and recall measures into a single value to provide a balanced assessment of a model's performance [52].

$$F1 - Score = \frac{2 \times (Precision \times Recall)}{Precision + Reccall}$$

- ***Matthews correlation coefficient (MCC)****:*it measures the correlation between the predicted and actual binary outcomes, considering all four elements of a confusion matrix [53].

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Regression Task: this involves evaluating the prediction of power consumption for target appliances using NILM model. The following measures will be used to compare model performance in predicting power consumption:

- ***Mean Absolute Error (MAE):*** short for mean absolute error, is a prevalent metric employed to assess the precision of regression models. It quantifies the average absolute difference between the predicted values and the actual values of the target variable.The MAE can be calculated using the following equation [54]:

$$MAE = (\frac{1}{n}) \times \sum_{i=1}^{n} |yi - yi\hat{}|$$

- ***Mean Squared Error (MSE):*** MSE is another common metric for evaluating regression models. It measures the average squared difference between the predicted values and the actual values. MSE penalizes larger errors more than MAE, making it useful for identifying significant deviations in predictions. The MSE can be calculated using the following equation [54]:

$$MSE = (\frac{1}{n}) \times \sum_{i=1}^{n} (yi - yi^{\wedge})^2$$

- ***R squared:*** also known as the coefficient of determination, indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. An R^2 value closer to 1 implies that the model explains a high proportion of the variance. It is calculated using the following formula [54]:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(yi - yi^{\wedge})^2}{\sum_{i=1}^{n}(yi - \bar{y})^2}$$

- ***Root Mean Square Error (RMSE):*** also known as Root Mean Square Deviation (RMSD), is the square root of the average of the squared differences between the predicted and actual values. It provides a measure of the magnitude of prediction errors, with larger errors having a more significant impact [54].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(yi - yi^{\wedge})^2}$$

<u>Most Suitable Metrics for Evaluation:</u>

Although all the previous metrics are good indicators of models performance, the **F1-Score** is the most suitable metric in our case and dataset as it balances precision and recall, making it particularly effective for handling class imbalances common in appliance usage data [55]. In addition, for the regression task, the **MAE** is the most suitable metric due to its straightforward interpretation and ability to measure the average absolute difference between predicted and actual values, providing a clear indication of the model’s prediction accuracy [56].

4.5.2 Efficiency Evaluation: this measures how effectively the NILM model uses computational resources in predicting power consumption and the status of target appliances [57]. The following efficiency measures will be used to compare NILM models:

- ***Training Time:***

  Training time is an important factor especially for models that need to be retrained frequently or in real-time systems. Models that can train quickly on large datasets are more efficient, as they accelerate the model development process [57].

- ***Prediction Time:***

  Prediction time is equally important, particularly for applications where quick predictions are required. A model with faster prediction times is preferable, especially in real-time applications and tasks [57].

In summary, achieving both accuracy and efficiency is vital in evaluating NILM models. Balancing high accuracy with manageable computational costs ensures practicality and reliability for load disaggregation tasks.

Chapter 5: Implementation and Experimentation of Deep Learning Approaches

In this chapter, we explore the implementation and experimentation of deep learning approaches for load disaggregation. We start with a CNN-based approach using a sequence-to-sequence (Seq2Seq) architecture. We explain the experimental setup, including how we built and preprocessed the UKDALE dataset. We then introduce the PTPNET (Principle Temporal Pooling) architecture, covering its training, testing, and performance analysis.

We also implement the Conv-LSTM architecture, a popular model that combines convolutional layers with bidirectional LSTM layers, known for its effectiveness in load disaggregation tasks. We compare the results of PTPNET with those of Conv-LSTM, using the same data and preprocessing methods. This comparison helps us evaluate the performance and effectiveness of these models in promoting energy-efficient appliance usage to reduce power consumption and costs.

We have implemented is inspired by the paper *"Non-Intrusive Load Disaggregation by Convolutional Neural Network and Multi-Label Classification"* which is a sequence-to-sequence (Seq2Seq) model based on CNN approach. The Seq2Seq model is an architecture designed for sequential data. It takes an input sequence of power consumption, processes it in the model layers, and generates an output sequence that represents the power consumption or the status (ON/OFF) of a specific target appliance depending on the effectiveness of the Convolutional Neural Network (CNN). The main architecture components involve encoder, decoder, attention block, and a temporal Pooling module.

Our implementation focuses on predicting the power consumption values and status of the target working appliance at each time. As a result, our problem is divided into two tasks, Binary Classification task, and Regression task. The classification task is for predicting the working status of the appliance whether it is ON or OFF, while the regression task is for predicting the continues power values for the target appliances. It is important to mention that the paper we were inspired by only focuses on the classification task, which is predicting whether the appliance is on or off. However, it does not delve into power prediction directly; instead, it multiplies the predicted status with a constant value to estimate power consumption. As a result, it might not provide the most accurate predictions of power usage.

Our implementation differs by incorporating both classification and regression tasks, providing a more comprehensive understanding of appliance behavior. Our focus appliances include the fridge, washing machine, and dishwasher for several reasons. First, the data for many other appliances was not sufficient to train the model effectively, making these appliances more suitable for machine learning tasks. For example, the iron was only used a few times in the first house. Additionally, we selected appliances that were present in all households, simplifying testing across different homes. Lastly, we chose a mix of appliances with distinct usage patterns, such as the fridge, dishwasher, and washing machine, to enhance the model's learning capabilities.

The outcomes of both tasks, offer several advantages, helping users reduce power usage. In terms of classification tasks, users can retrieve information about whether specific appliances are in use, ensuring they are switched off when they are not in use. This could help users to prevent accidents, and save money as well. For example, if an iron is left powered on while not in use, utilizing this model can prevent accidents and unnecessary expenses. Also, understanding an appliance's power consumption at a given time enables users to identify sources of energy usage and adjust their habits accordingly to lower consumption and bills. Moreover, it facilitates the detection of faults in appliances. For instance, if a user notices an increase in the fridge's power consumption compared to the average, it signals a potential issue or damage that can be addressed before additional energy is consumed.

5.1 PyTorch and Supporting Libraries

The implementation of the model was done by using PyTorch framework. PyTorch is an open source machine learning framework based on the Python programming language and Torch Library which is a Machine Learning library used for building and training deep neural networks [58].

PyTorch relies on tensors, which are n-dimensionally arrays used to store almost everything in deep learning such as: input data, weights, biases, predictions, etc. Also, other libraries were used such as scikit-learn for metrics computation**,** pandas for data manipulation, and matplotlib for data visualization, ensuring a robust and stable implementation [58].

5.2 Exploring and Building the UKDALE Dataset

The intial step in our implemntaiton is to read and explore the UKDALE dataset. This dataset was saved in a HDF5 file formemate with 3.1 GB data size. The h5 formate is a type of files that called "Hierarchical Data Format version 5", which used when we want to store high size of data. With panda library, we established a connection to the "ukdale.h5" file, providing access to its content.

```
# making  a connection and storing the data in the ukdale.h5 file in the store variable.
store = pd.HDFStore('ukdale.h5')
```

Figure 11: Reading the dataset file.

As mentioned in precvious sections, the UKDALE consists data of 5 different houses in UK, each house has different types and numbers of appliances. The dataset formate that we have read is organized hierarchcally, with unique keys are representing the building number and meter number. The building number is form 1 to 5 since the dataset measures the power for only 5 houses, while the meter number represents individual appaliance number in that building. For example, the key "/building5/elec/meter1" represents the meter 1 within the fifth building, while "/building5/elec/meter10" represents the meter 10 within the same building, and so on.

```python
# List all keys in the HDFStore
keys = store.keys()

# Display the keys
print("Keys in HDFStore:")
for key in keys:
    print(key)
```

```
Keys in HDFStore:
/building5/elec/meter1
/building5/elec/meter10
/building5/elec/meter11
/building5/elec/meter12
/building5/elec/meter13
/building5/elec/meter14
/building5/elec/meter15
/building5/elec/meter16
/building5/elec/meter17
```

Figure 12: Some key values in the dataset.

With these unique keys, we are accessing different datasets stored within the file, each representing the power consumption of specific appliances within the building, measured in watts along with corresponding timestamps. The power consumption of the individual appliances was recorded with a period of 6 seconds interval. Each house had a different number of appliances and different data recording periods:

- House 1: 52 appliances, data recorded from 2012-11-09 to 2015-01-05.
- House 2: 18 appliances, data recorded from 2013-02-17 to 2013-10-10.
- House 3: 4 appliances, data recorded from 2013-02-27 to 2013-04-08.
- House 4: 5 appliances, data recorded from 2013-03-09 to 2014-11-13.
- House 5: 24 appliances, data recorded from 2014-06-29 to 2014-11-13.

```python
for key in list(store.keys()):
    # Print the key (path)
    print("############### KEY -->", key,"###############")
    # Retrieve and print the data associated with the key
    print("------------>> DATA")
    print(store[key])  # Print the first few rows of the data
    print("The numebr of lines in key",key," is ----> ",len(store[key]))
    print("###############################################################################
```

```
############### KEY --> /building5/elec/meter10 ###############
------------>> DATA
physical_quantity          power
type                      active
2014-06-29 17:23:52+01:00   28.0
2014-06-29 17:23:58+01:00   28.0
2014-06-29 17:24:02+01:00   29.0
2014-06-29 17:24:08+01:00   28.0
2014-06-29 17:24:14+01:00   28.0
...                          ...
2014-09-07 15:00:04+01:00   27.0
2014-09-07 15:00:10+01:00   26.0
2014-09-07 15:00:16+01:00   27.0
2014-09-07 15:00:23+01:00   26.0
2014-09-07 15:00:29+01:00   27.0

[985784 rows x 1 columns]
The numebr of lines in key /building5/elec/meter10  is ---->  985784
```

Figure 13: Accessing the dataset using keys.

In addition to the dataset .h5 file, there are five associated '.dat' files, each corresponding to one of the five buildings. These '.dat' files serve as reference guides, linking meter numbers to their corresponding appliance names. For example, the meter 1 in building 1 represents the aggregated power of house 1, while subsequent numbers represent various appliances such as boilers, washing machines, dishwashers, TVs, toaster, and more. In other words, the file "house_1_labels.dat" conains the list of applainces in house 1, etc.

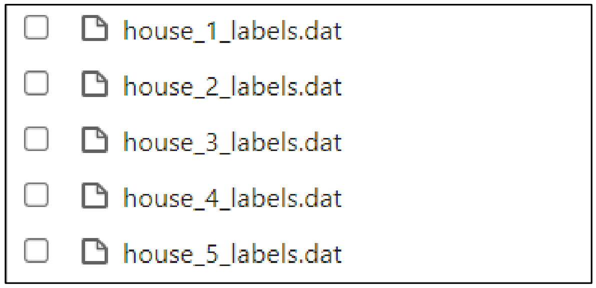

Figure 14: The 5 .dat files containing the list of appliances for house 1 to house 5.

In all five houses, the first appliance (numbered 1) represents the total aggregated power used in each house. The aggregated power in the UK-DALE dataset is apparent power, which is the total power consumed in an electrical system. Apparent power is the sum of two components:

- **Real Power (Active Power):** This is the actual power consumed by devices in the system that performs useful work, such as lighting a bulb. Real power is measured in watts (W) [59].
- **Reactive Power:** This is the power used to maintain the electric and magnetic fields in devices like motors and transformers. Reactive power does not perform useful work but is necessary for the operation of these devices. It is measured in reactive volt-amperes (VAR) [60].

Apparent power is measured in volt-amperes (VA) and represents the total power flow in the electrical system, combining both real and reactive power components.

5.3 UKDALE Dataset Preprocessing

In this section, our goal is to prepare the data for training and testing the model in both tasks, focusing on three key appliances: the Fridge, Washing Machine, and Dishwasher. We started by resampling the data into consistent one-minute intervals. Since the original measurements were recorded every 6 seconds, we averaged the values over each minute. Resampling the data into one-minute intervals allowed us to reduce the granularity of the data while still retaining important information about power consumption trends. Instead of dealing with measurements every 6 seconds, we now had an average value for each appliance for every minute. This not only made the dataset more manageable but also smoothed out any fluctuations in the data, making it easier to detect patterns and trends.

Additionally, we set maximum power limits for each target appliance as specified in the paper during the resampling process. For instance, the maximum power threshold for the Fridge is 300 Watts.

To address missing values in the dataset, we employed a forward fill method. This means that missing values are filled by carrying forward the last observed value. However, if more than 300 consecutive missing values occur, we filled them with a zero value to ensure the continuity of the time series data.

	Fridge	Dishwasher	Washing Machine
Max. power (W)	300	2500	2500

Figure 16: Maximum Power Consumption Specifications for Target Appliances.

```
def resample_meter(store=None, building=1, meter=1, period='1min', cutoff=1000.):
    key = '/building{}/elec/meter{}'.format(building,meter)
    m = store[key]
    v = m.values.flatten()
    t = m.index
    s = pd.Series(v, index=t).clip(0.,cutoff)
    s[s<10.] = 0.
    return s.resample('1s').ffill(limit=300).fillna(0.).resample(period).mean().tz_convert('UTC')
```

Figure 15: The method utilized for resampling the input data series.

We select the desired appliances, resample the data using the specified resampling function, and save the retrieved data in new dataframes for each house. Threshold values, indicating cutoff points, are adjusted according to the type of appliance for consistency across all houses.

```
house = 1
m = get_series(store, house, 'aggregate', 10000.)
m.name = 'aggregate'
a2 = get_series(store, house, 'fridge', 300.)
a2.name = 'fridge'
a3 = get_series(store, house, 'washing_machine', 2500.)
a3.name = 'washing_machine'
a5 = get_series(store, house, 'dishwasher', 2500.)
a5.name = 'dish_washer'

ds_1 = pd.concat([m, a2, a3, a5], axis=1)

ds_1.ffill(inplace=True)  # Forward fill missing values
```

Figure 17: resampling the data sequences of House 1.

For example, the threshold for the fridge is set at 300 watts, while for the dishwasher, it is 2500 watts. Similarly, the threshold for aggregated power is set at 10000 watts, representing the cumulative power consumption of the entire house.

Also Figures 19 and 20 show the sequence of aggregated power consumption before and after resampling process. It is noticed that before reampling the data was mesured every 6 seconds with integer power values, while when resmapling it to 1-min interval, the values of the power became float since we are taking the average power consumption of the whole 1-min interval.

```
dtype: float32
####################################
----Resampled Series:
2012-11-09 22:28:00+00:00     591.555542
2012-11-09 22:29:00+00:00     583.633362
2012-11-09 22:30:00+00:00     585.083313
2012-11-09 22:31:00+00:00     589.849976
2012-11-09 22:32:00+00:00     586.750000
2012-11-09 22:33:00+00:00     581.783325
2012-11-09 22:34:00+00:00     650.133362
2012-11-09 22:35:00+00:00     660.849976
2012-11-09 22:36:00+00:00     523.516663
2012-11-09 22:37:00+00:00     517.366638
2012-11-09 22:38:00+00:00     624.533325
2012-11-09 22:39:00+00:00     761.516663
2012-11-09 22:40:00+00:00     554.816650
2012-11-09 22:41:00+00:00     463.866669
2012-11-09 22:42:00+00:00     464.966675
2012-11-09 22:43:00+00:00     460.950012
```

Figure 19: Aggregate power sequence data for house 1 after resampling.

```
Filename: house_1_labels.dat
------------------------------------------
Label Matched: 1 aggregate
----Key: /building1/elec/meter1
####################################
----Original Meter Data:
physical_quantity               power
type                         apparent
2012-11-09 22:28:15+00:00       599.0
2012-11-09 22:28:21+00:00       582.0
2012-11-09 22:28:27+00:00       600.0
2012-11-09 22:28:33+00:00       586.0
2012-11-09 22:28:40+00:00       596.0
2012-11-09 22:28:52+00:00       581.0
2012-11-09 22:28:58+00:00       597.0
2012-11-09 22:29:04+00:00       592.0
2012-11-09 22:29:10+00:00       587.0
2012-11-09 22:29:16+00:00       574.0
```

Figure 18: Aggregate power sequence data for house 1 before resampling.

Figure 19 shows the concatenated power series ‘ds_1’ for various appliances along with the aggregated power consumption for house. The columns represent different appliances, including “aggregate,” “kettle,” “fridge,” “washing_machine,” “microwave,” and “dish_washer”. Each row corresponds to a specific datetime entry.

```
# printign the first dataset which is related for HOUSE 1
ds_1
```

datetime	aggregate	fridge	washing_machine	dish_washer
2012-11-09 22:28:00+00:00	591.555542	NaN	0.0	0.0
2012-11-09 22:29:00+00:00	583.633362	NaN	0.0	0.0
2012-11-09 22:30:00+00:00	585.083313	NaN	0.0	0.0
2012-11-09 22:31:00+00:00	589.849976	NaN	0.0	0.0
2012-11-09 22:32:00+00:00	586.750000	NaN	0.0	0.0
...	...	...	...	...
2015-01-05 06:21:00+00:00	202.127655	81.36364	0.0	0.0
2015-01-05 06:22:00+00:00	202.127655	81.36364	0.0	0.0
2015-01-05 06:23:00+00:00	202.127655	81.36364	0.0	0.0
2015-01-05 06:24:00+00:00	202.127655	81.36364	0.0	0.0
2015-01-05 06:25:00+00:00	202.127655	81.36364	0.0	0.0

Figure 20: Resampled data for House 1.

We plot the data to visulaize and explore the patterns of the target appliances. Figure 23 represents the aggregated power values as a signal over time, which is the input data series to the model, and Figure 20, 21, and 22 are the output power values for the target appliances.

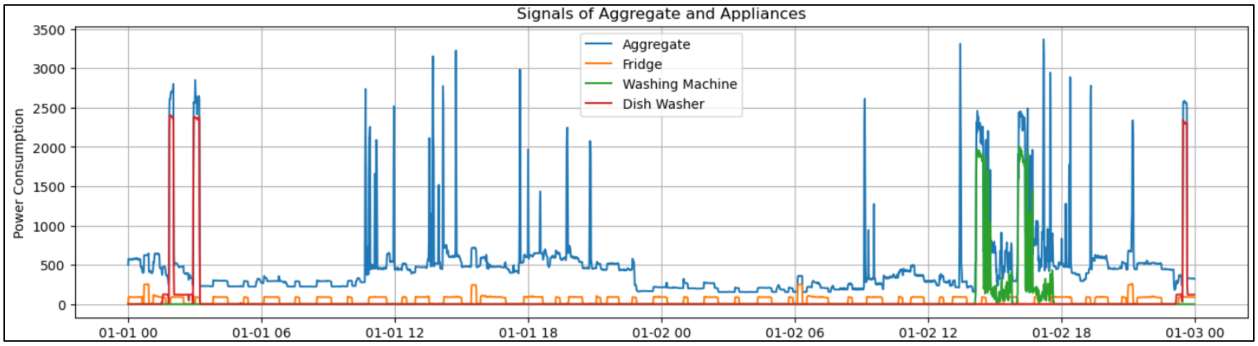

Figure 24: The aggregated power and the target appliances signals.

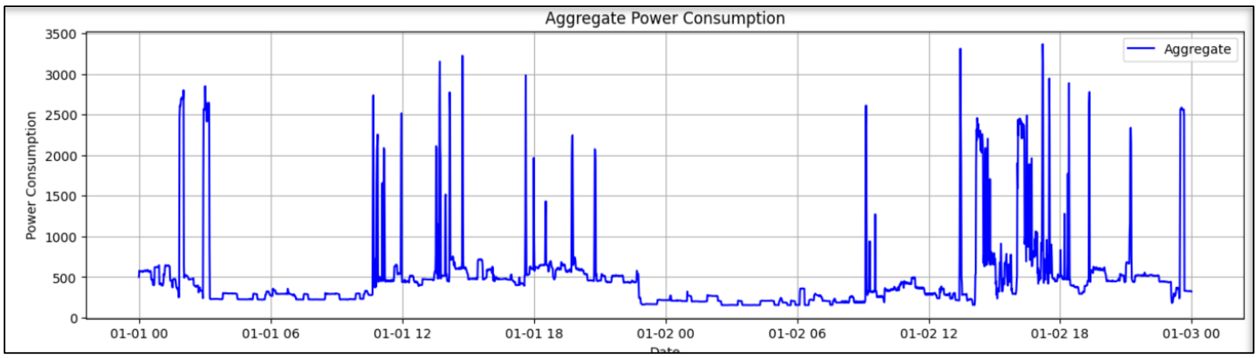

Figure 23: Aggregated power signal.

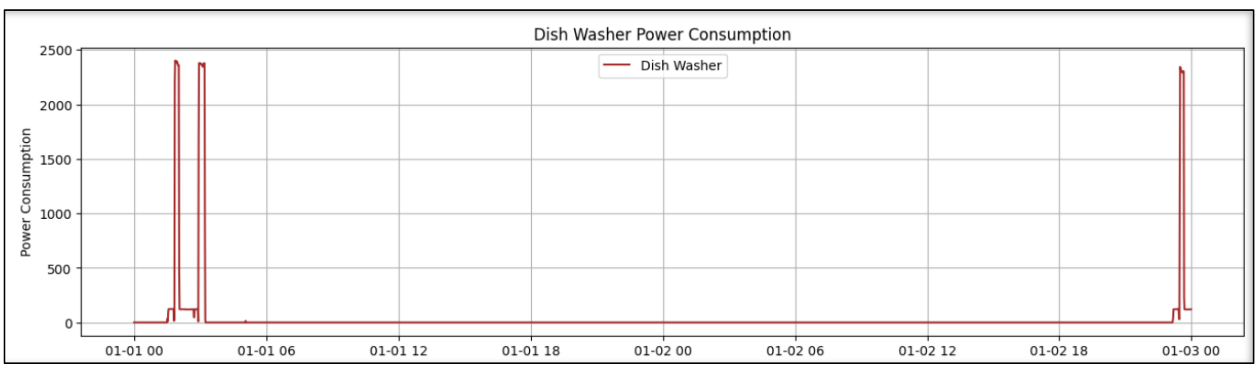

Figure 22: Dishwasher output signal.

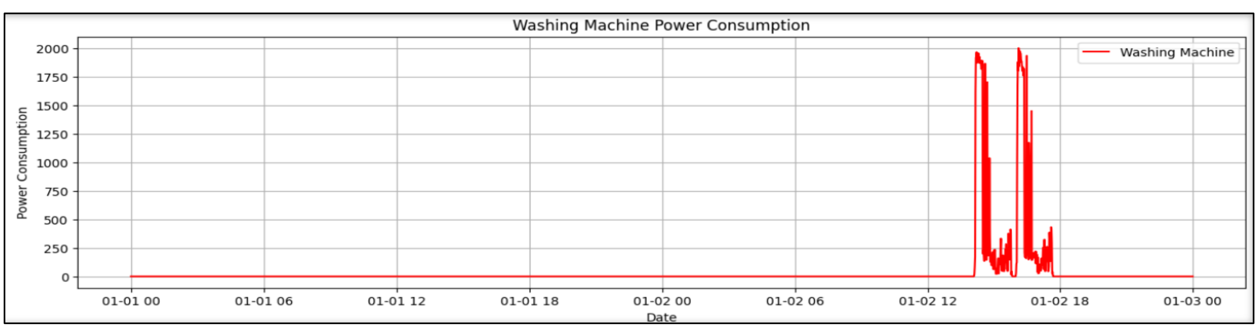

Figure 21: The washing machine output signal.

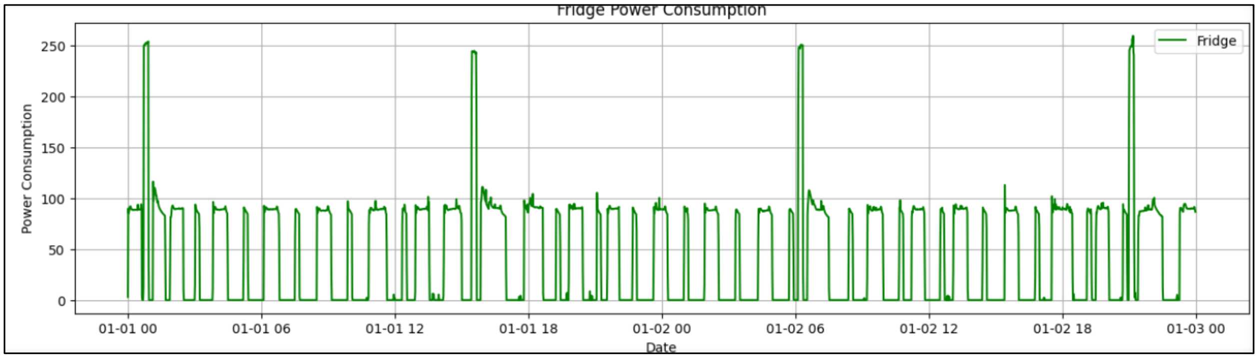

Figure 25: The fridge output signal.

So we have prepared five separate dataframes, each representing the resampled data series for aggregated power consumption and specific target appliances across the five houses. Since we have two separate tasks ahead, our objective now is to update the values of the target appliances in the datasets. While in both tasks the input series represent aggregated power values, the output differs. For classification tasks, the output will be 0/1 for the target appliances, while for regression tasks, the output will be continuous power consumption values.

Since the data is not yet ready for the classification task, we need to adjust the values of the power consumption in the target appliances into binary values instead of continuous power values, and this is done by using the concept of thresholding.

Thresholding involves applying specific thresholds to distinguish between ON and OFF states of the appliances based on their power consumption levels. The process is straightforward for appliances like the Refrigerator, where a specific threshold value thresholds to differentiate between ON and OFF states.

However, appliances like the Washing Machine and Dish Washer have more complex functioning compared to the refrigerator. In these cases, the power consumption values could be below the threshold but the appliances may still be functioning and operating. To address this complexity, a minimum time duration is set during which the power consumption must remain below the threshold to consider the appliance as OFF. Similarly, a minimum ON duration is established to consider the appliance as actually switched ON. The threshold, minimum OFF duration, and minimum ON duration are presented as follows:

	Fridge	**Dishwasher**	**Washing Machine**
Max. power (W)	300	2500	2500
Power threshold (W)	50	10	20
OFF duration (min)	0	3	30
ON duration (min)	1	30	30

Figure 26: Thresholding, and duration parameters.

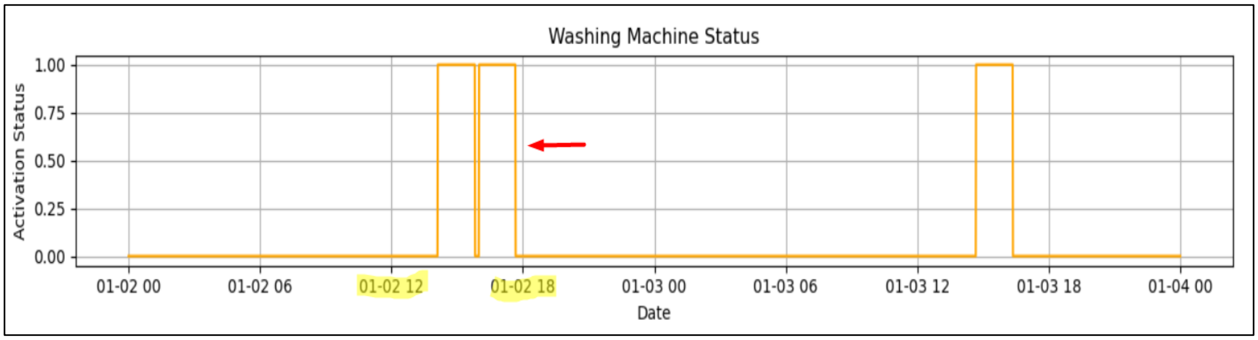

Figure 27: Washing Machine status.

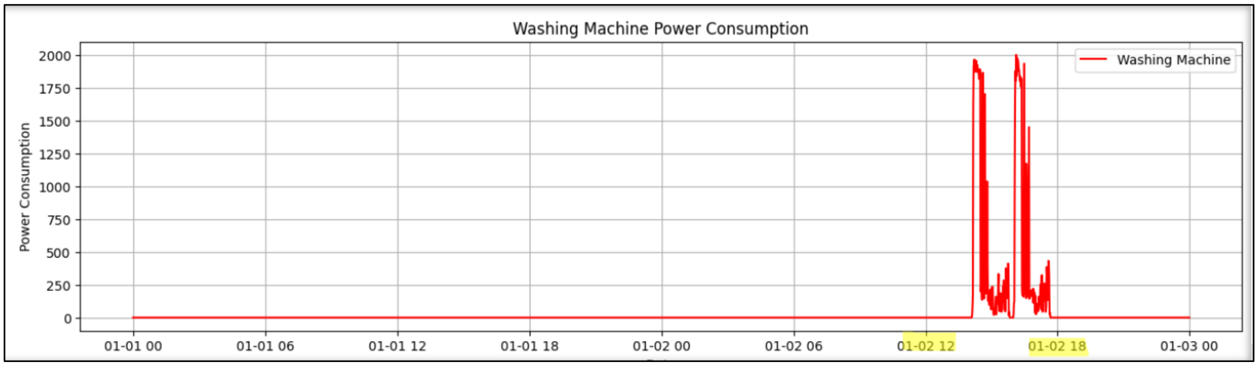

Figure 28: Power signal for washing machine.

5.4 Splitting the dataset into Training, Validation, and Testing

Considering that our model is a Seq2Seq model, which means that the data will be fed to the model as a sequence of data, and the output will also be a sequence. As a result, the current dataset is divided into batches each batch has equivalent sequence length to match the model architecture while splitting it into training, validation, and testing.

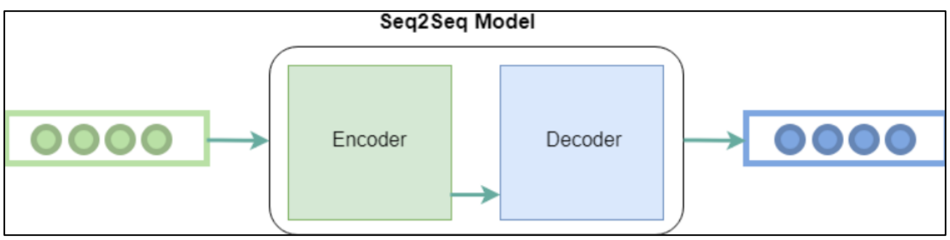

Figure 29: Sequence to Sequence model example.

To make sure our data is consistent, we created a class called "Power" that normalizes power loads. We divide each power value by a reference value of 2000 Watts. This ensures that all data points are on the same scale. We don't normalize the status since it is binary (on/off) and doesn't need scaling.

After normalization, we split our data into training, validation, and testing sets. Each set is made by using the Power class to process the input data sequences. Input sequences are 510 samples long (representing minutes), while output sequences (power consumption values or status) are 480 minutes long, which is 8 hours. The difference in length is because of the convolution filters used in the model.

The datasets are divided as follows:

- Training dataset: 80% of the total data.
- Validation dataset: the next 10% of the data.
- Testing dataset: the last 10% of the data.

```
##### The trianing dataset --> 80% #####
ds_house_train = [Power(ds_meter[i][:int(0.8*ds_len[i])],
                        ds_appliance[i][:int(0.8*ds_len[i])],
                        ds_status[i][:int(0.8*ds_len[i])],
                        SEQ_LEN, BORDER, MAX_POWER, True) for i in range(5)]

##### The validation dataset --> 10% #####
# Define the validation datasets for each house, where data from the next 10% of the total length is used
ds_house_valid = [Power(ds_meter[i][int(0.8*ds_len[i]):int(0.9*ds_len[i])],
                        ds_appliance[i][int(0.8*ds_len[i]):int(0.9*ds_len[i])],
                        ds_status[i][int(0.8*ds_len[i]):int(0.9*ds_len[i])],
                        SEQ_LEN, BORDER, MAX_POWER, False) for i in range(5)]

##### The testing dataset --> 10% #####
# Define the test datasets for each house, where data from the last 10% of the total length is used
ds_house_test  = [Power(ds_meter[i][int(0.9*ds_len[i]):],
                        ds_appliance[i][int(0.9*ds_len[i]):],
                        ds_status[i][int(0.9*ds_len[i]):],
                        SEQ_LEN, BORDER, MAX_POWER, False) for i in range(5)]
```

Figure 30: Training, validation, and testing datasets.

```
# Copy and normalize meter, appliance, and status data
self.meter = meter.copy() / self.max_power  # Normalize meter data
self.appliance = appliance.copy() / self.max_power  # Normalize appliance data
self.status = status.copy()  # Copy status data
```

Figure 31: Dataset Normalization.

The figures below show the data after it is been normalized. You will notice changes in the scale of the vertical axis due to the normalization process. This includes the total aggregated power signal, individual power signals for each appliance.

From the above figures, it is notable that the input window length is 510 points (or minutes), while the output status or power signal of the appliance is shorter, with a length of 480 minutes.

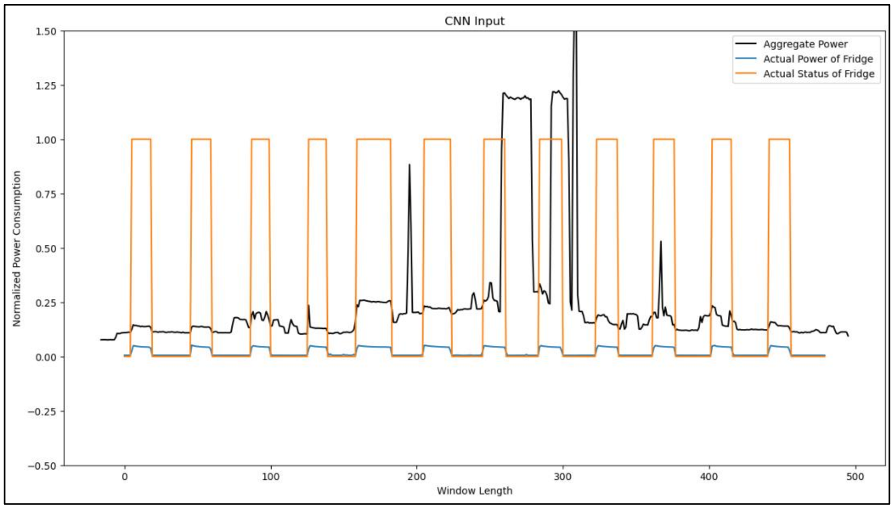

Figure 34: Normalization data for fridge.

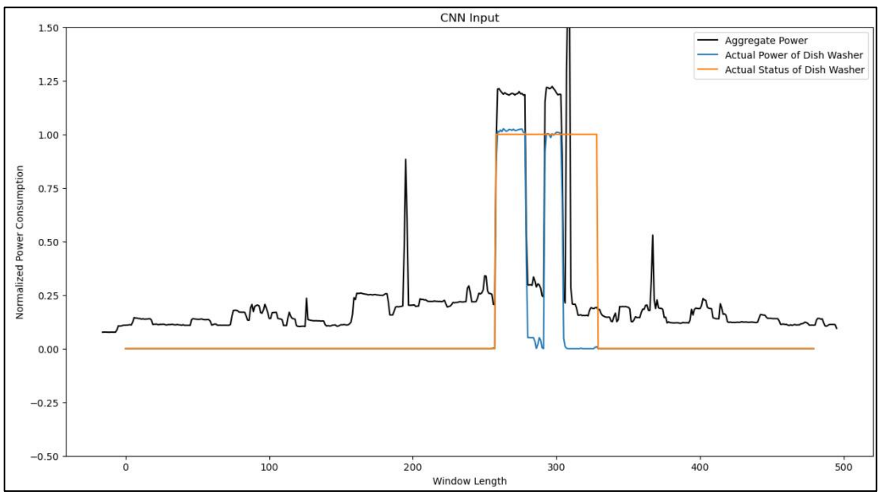

Figure 33: Normalization data for Dish Washer.

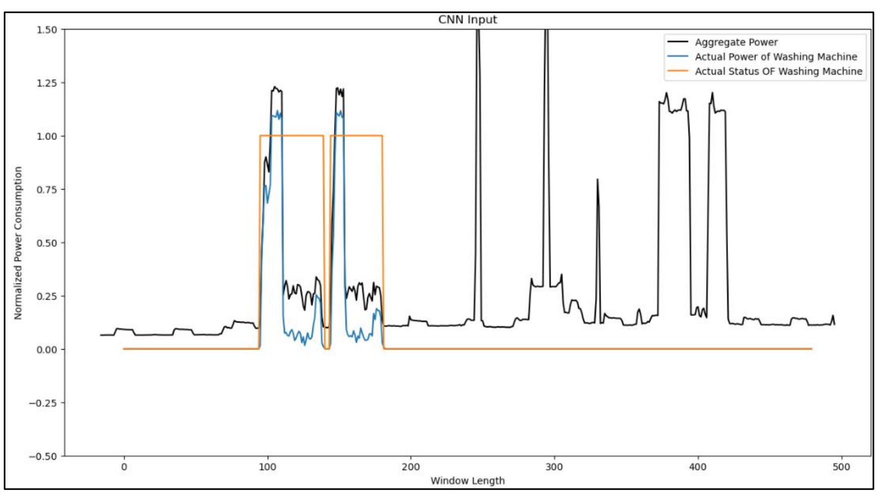

Figure 32: Normalization for Washing Machine.

5.5 SEEN and UNSEEN Use Cases

During the training and testing process of our Seq2Seq model, we consider two scenarios, the "SEEN" case and the "UNSEEN" case.

SEEN CASE: In this case, the model is trained and tested on data from the set of houses that it trained on. For example, the training process will be conducted on the data of the three houses (House 1, House 2, and House 5) and the testing will be on data from House 1.

UNSEEN CASE: In this case, the model will trained on data from houses and tested on data from different houses. For example, the model will be trained on a dataset of House 1 and House 3, and tested on the different data from House 2. The purpose of this case is to test the generalization ability of our model to predict the appliance's signatures (the power in the regression task, and the status in the classification task).

The following table shows the data porions used in both seen and unseen cases in the process of training and testing:

	Use Case: *seen*			Use Case: *unseen*		
	Training (%)	**Validation (%)**	**Testing (%)**	**Training (%)**	**Validation (%)**	**Testing (%)**
House 1	80	10	10	80	10	-
House 2	80	-	-	-	-	100
House 5	80	-	-	80	10	-

Figure 35: Training, Validation, and Testing Datasets in both cases: SEEN and UNSEEN.

5.6 Model Architecture Implementation

The model architecture consists of Four main components: the encoder, temporal pooling layer, attention block, and decoder. Each of these components consists of multiple layers and stages designed to process and transform the input data sequence, and predicting the output results.

<u>ENCODER:</u>

The encoder stage serves as the intial processing block in the model architecture. Its primary objective is to receive the input data, which represents the aggregated power consumption sequence of length 510 samples (510 min), and enhance the feature space while reducing the temporal resolution.

The stage consists of four encoder blocks, each followed with a max-pooling layer. Within each encoder block, there is a sequence of operations including a one-dimensional convolutional layer (1-D Conv), rectified linear unit (ReLU) activation function, batch normalization (Batch Norm), and dropout layer for regularization. The purpose of these operations is to extract meaningful features from the input data and enhance the complexity of the feature representation. It's important to note that our model deals with sequential data, hence the utilization of 1-D Convolutional layers.

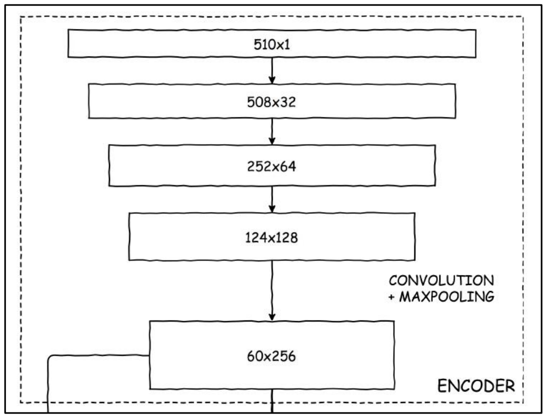

Figure 36: Encoder block diagram.

The max-pooling layer, which follows each encoder block, is responsible for reducing the dimensionality of the feature vector, enhancing robustness to variations, and promoting generalization. The code block that represents the building of the encoder layer in the model is as the following:

TEMPORAL POOLING:

The next stage in our model is the temporal pooling which acts as intermediary component between the encoder and decoder. It temporal plooing layer consists of the following layers:

- The encoder output passes through 4 avg pooling modules with different filter sizes.
- It followed by 1-D Conv filter with UNIT filter size in order to reduce the number of features to a quarter of input one.
- Linear upsampling is performed to obtain a temporal resolution at the output of the poling block t be equal to the output of the encoder.
- Finally the drop out regulairzation is applied to prevent the overfitting.

Also, the four layers of this block finally will be concatenated together to be the input of the next stage which is the decoder.

ATTENTION BLOCK:

To enhance our model's performance, we added an attention block. This block is important for focusing and pay attention on the most relevant parts of the input sequence when predicting each time step in the output sequence. By integrating this attention mechanism, the model can weigh the importance of different parts of the input data, allowing it to better capture long-term dependencies and improve accuracy in predicting appliance-level power consumption and status. It is worth noting that this attention block is exclusively added to the model responsible for power prediction.

The model for status prediction consists solely of an encoder with temporal pooling and a decoder.

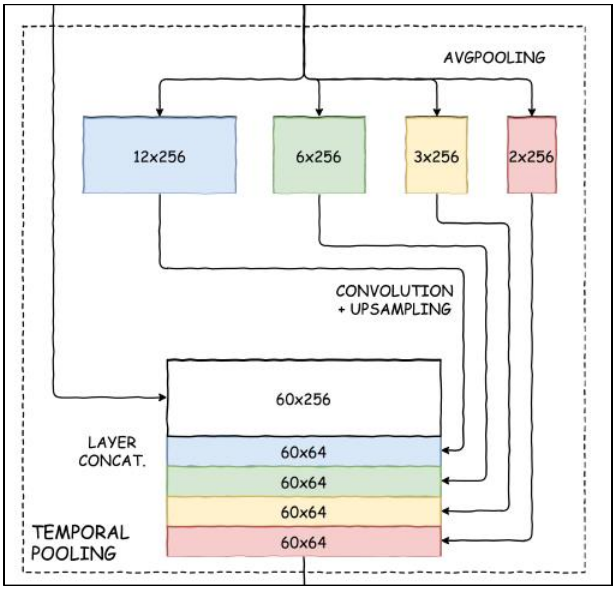

Figure 37: Temporal Pooling block diagram.

DECODER:

The Decoder stage serves as the final step in our model architecture, responsible for reconstructing the original input data using the features learned by the Encoder. It consists of a single layer known as a Transposed Convolutional Layer, which expands the extracted features back to their original shape. Following this layer, batch normalization is applied to stabilize the learning process and improve model performance. Also, another convolutional layer follows the Transposed Convolutional Layer to maintain the spatial dimensions of the input while adjusting the number of channels to align with the target appliances which is 3.

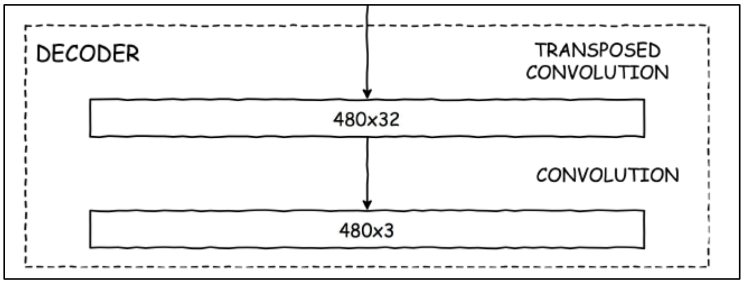

Figure 38: Decoder block diagram.

5.7 Training and Testing the Model

Both the regression and binary classification tasks were trained and tested on both seen and unseen data. For the binary classification task, the last layer used is the sigmoid activation function, which maps the output to a value between 0 and 1, representing the probability of the appliance being on or off (0 for off, 1 for on). The model's loss is calculated using Binary Cross Entropy Loss by comparing its predictions to the actual status.

For the regression task, the last layer directly takes the model's output, as it predicts continuous power values. We evaluated both Mean Absolute Error (MAE) and Mean Squared Error (MSE) as loss functions to calculate the difference between the model's predicted power values and the actual power consumption of the target appliances. MAE performed better due to its lower sensitivity to outliers and noise in the data.

In our experiments, we trained 20 models for each scenario (seen and unseen data) to address the inherent randomness in weight initialization. Each model starts with a unique set of weights, which can significantly impact the training process and final results. By training multiple models, we increased the likelihood of obtaining optimal weight configurations, leading to better performance.

We also compared our results using PTPNet with a model called Conv-BiLSTM, which consists of one Conv1D layer, two bidirectional LSTM layers, and two dense layers. This comparison was conducted on the same environment, utilizing identical dataset and preprocessing techniques. Convolutional layers detect local patterns in data, while LSTMs capture sequences over time by reading data in both forward and backward directions. This combination effectively handles spatiotemporal data, making it well-suited for non-intrusive load monitoring (NILM) tasks. The convolutional component identifies usage patterns, and the LSTM component analyzes their temporal evolution, enhancing prediction accuracy. After many trials, we found that MSE loss gives better results in power prediction, and binary cross-entropy was used for status prediction. This hybrid approach provided a robust comparison to our primary PTPNet model by better modeling the complex dependencies in power consumption data [23].

BINARY CLASSIFICATION TASK (STATUS PREDICTION)

As previously mentioned, the objective of this task is to predict the activation status of the specified appliances (fridge, dishwasher, and washing machine). Given that it is a classification task, we employed evaluation metrics such as accuracy, recall, F1 score, MCC, and precision to assess the performance of the model under examination. We will compare the results of this task between the PTPNet model and the Conv-BiLSTM model in each case, both for seen and unseen scenarios.

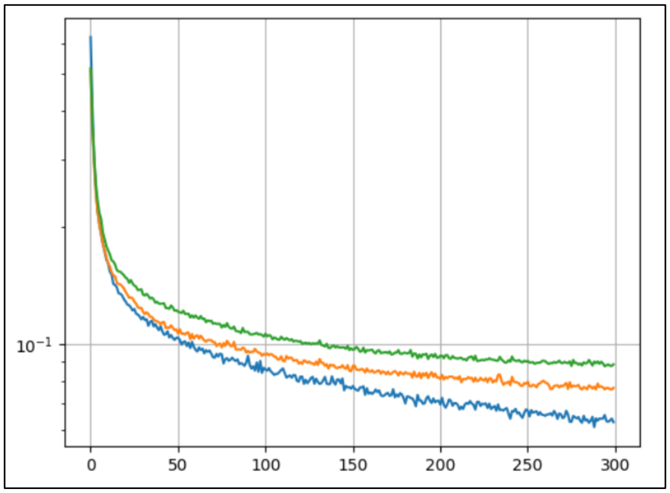

Figure 40: Loss curves of PTPNET in the seen case - status prediction

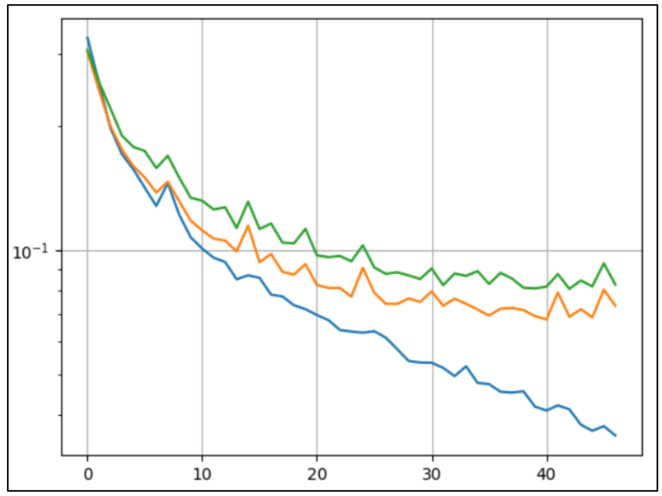

Figure 39: Loss curves of CONV-BiLSTM in the seen case - status prediction

Figures 39 and 40 are the loss curves for PTPNet and Conv-BiLSTM in the seen case during the training of 20 models. It's evident that the loss curves for PTPNet are more synchronized and smoother compared to Conv-BiLSTM, where we observe more fluctuations and poorer convergence. We also can notice that PTPNet stops training in the seen case at 300 epochs, a value determined through hyperparameter tuning as the optimal point. However, Conv-BiLSTM stops training around 47 epochs via the early stopping criteria which is a form of regularization used to avoid overfitting when training.

The blue one is the training loss, the orange is the validation loss, and the green is the testing loss.

It is important to clarify that the term "seen case" does not imply that the model trains and tests on the exact same data. Instead, it means that the model is trained and tested on data from the same house, where the dataset is partitioned into training, validation, and testing sets.

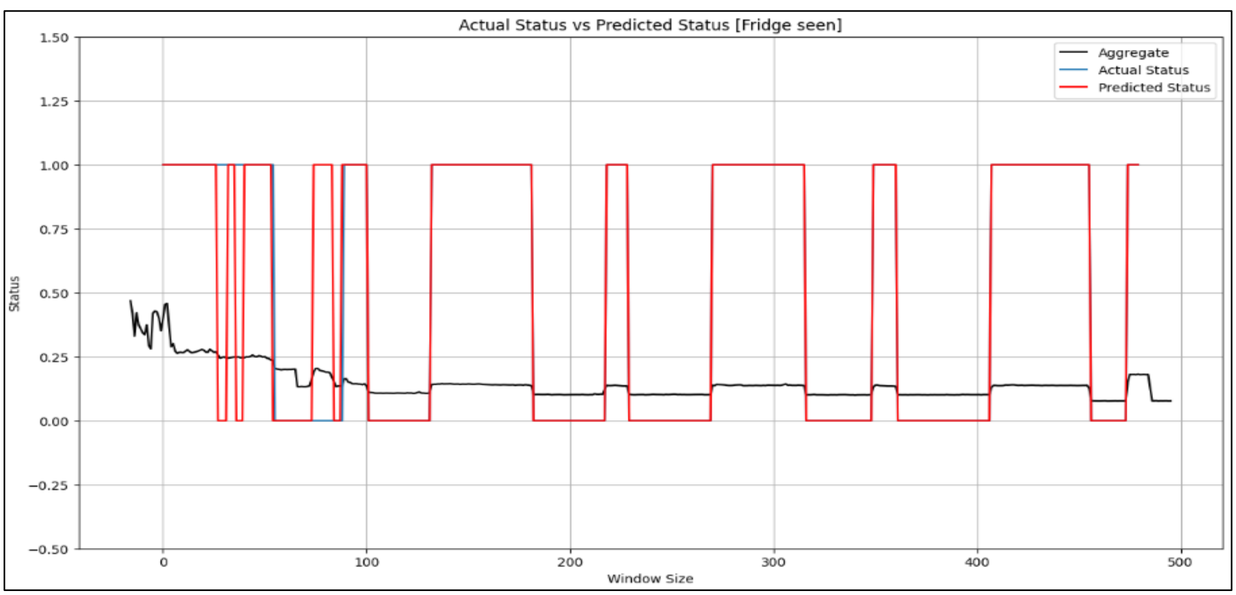

Figure 41: Status Fridge Prediction in the seen case for PTPNET.

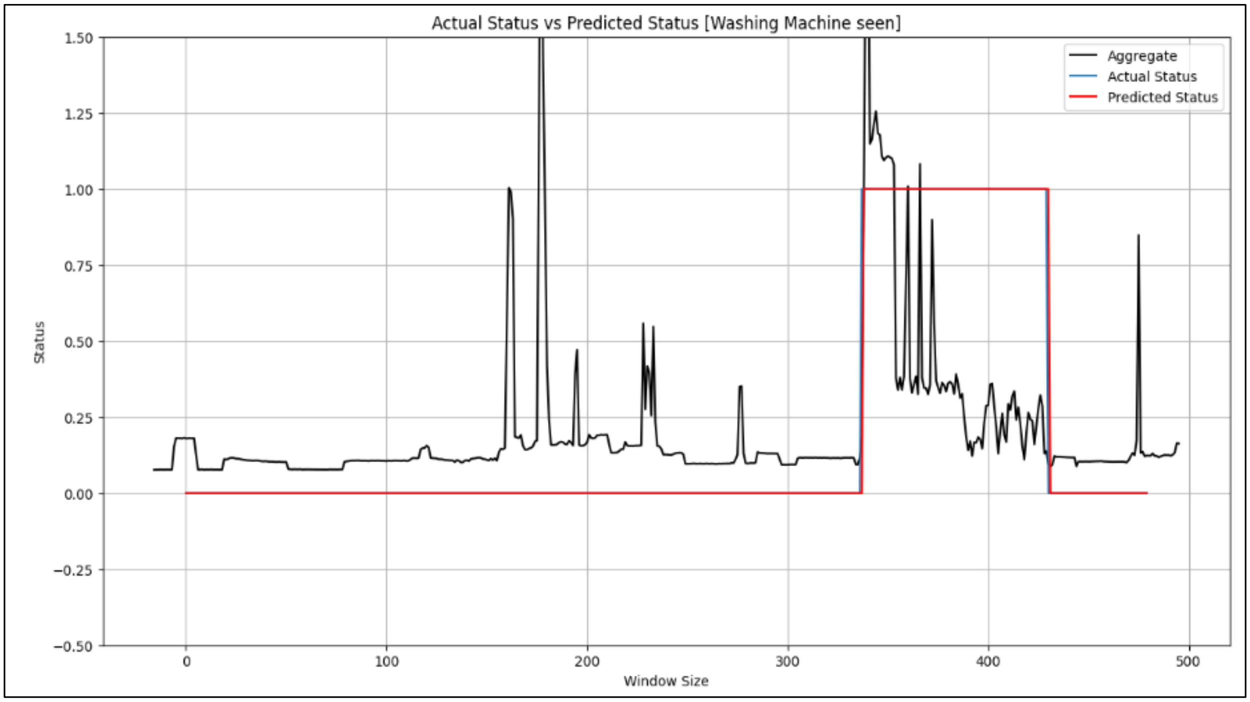

Figure 43: Status Washing Machine Prediction in the seen case for PTPNET.

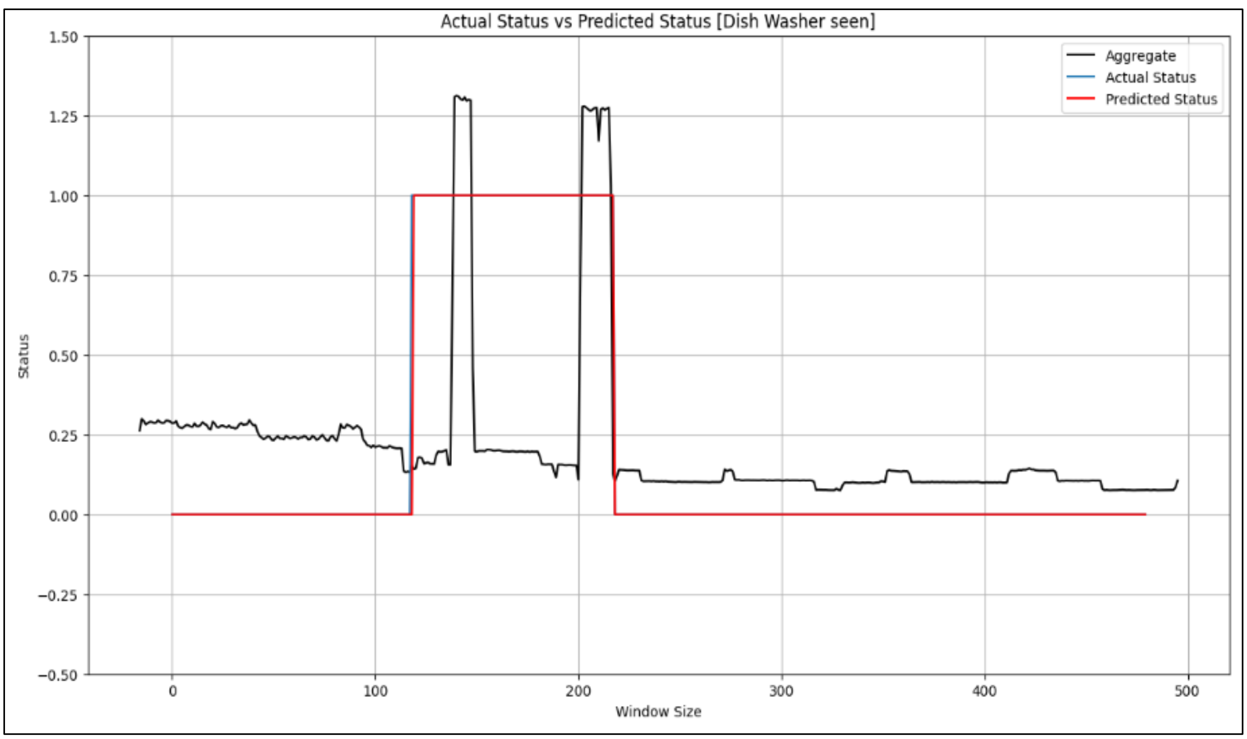

Figure 42: Status Dish Washer Prediction in the seen case for PTPNET.

The PTPNET model performs well in predicting the status of the washing machine and dishwasher, as seen in Figures 41 and 42. The predicted statuses closely match the actual ones, almost overlapping. However, for the fridge, there is a slight difference between predicted and actual statuses in the window shown in Figure 40. This happens because at the start, the aggregate signal includes multiple appliances, making it hard to identify the fridge's pattern. As time goes on, the signal clarifies, showing only the fridge in use. Then, the model predicts more accurately, and the predicted and actual statuses match better.

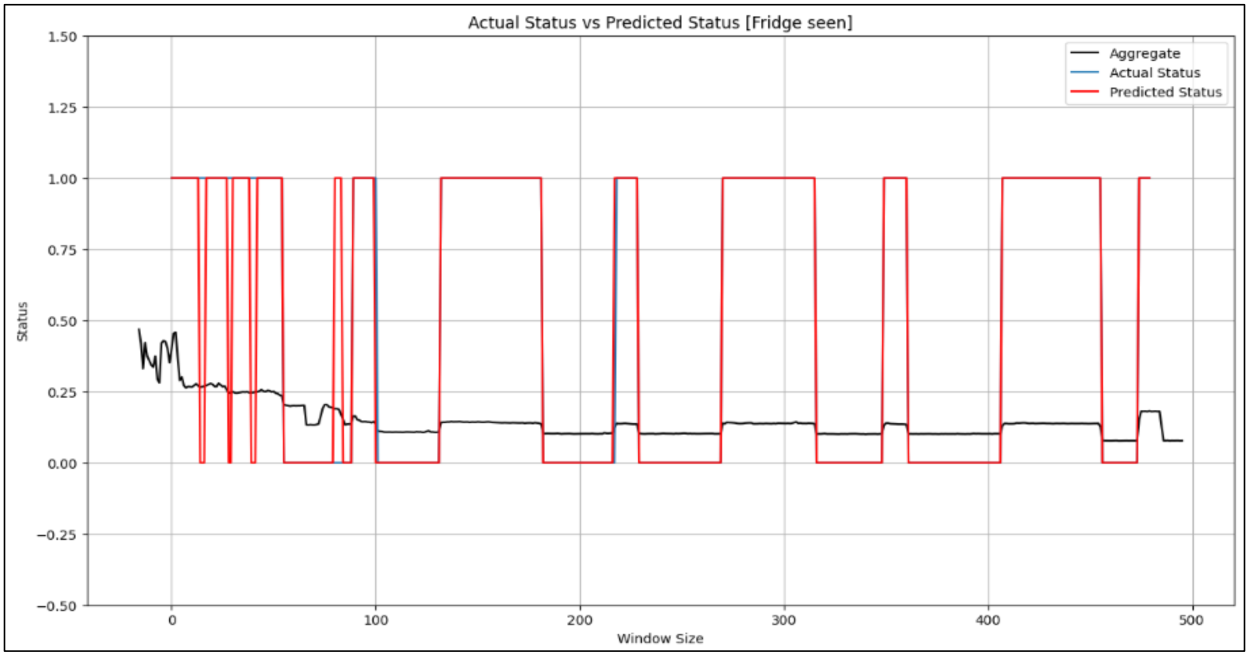

Figure 45: Status Fridge Prediction in the seen case for CONV-BILSTM.

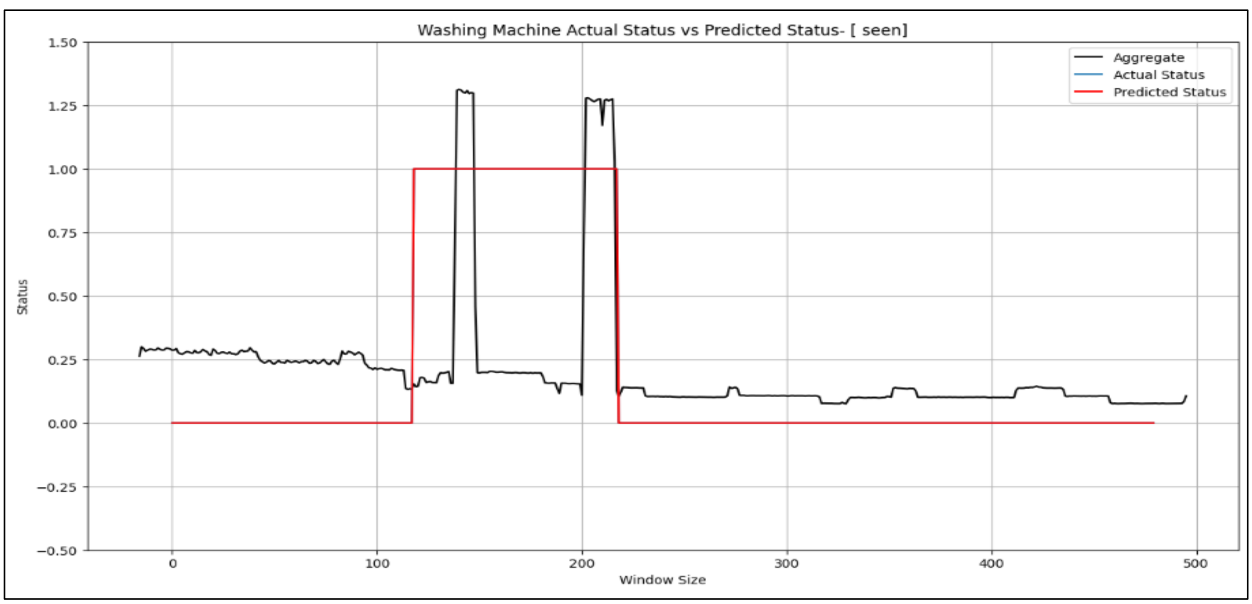

Figure 44: Status Dish Washer Prediction in the seen case for CONV-BILSTM.

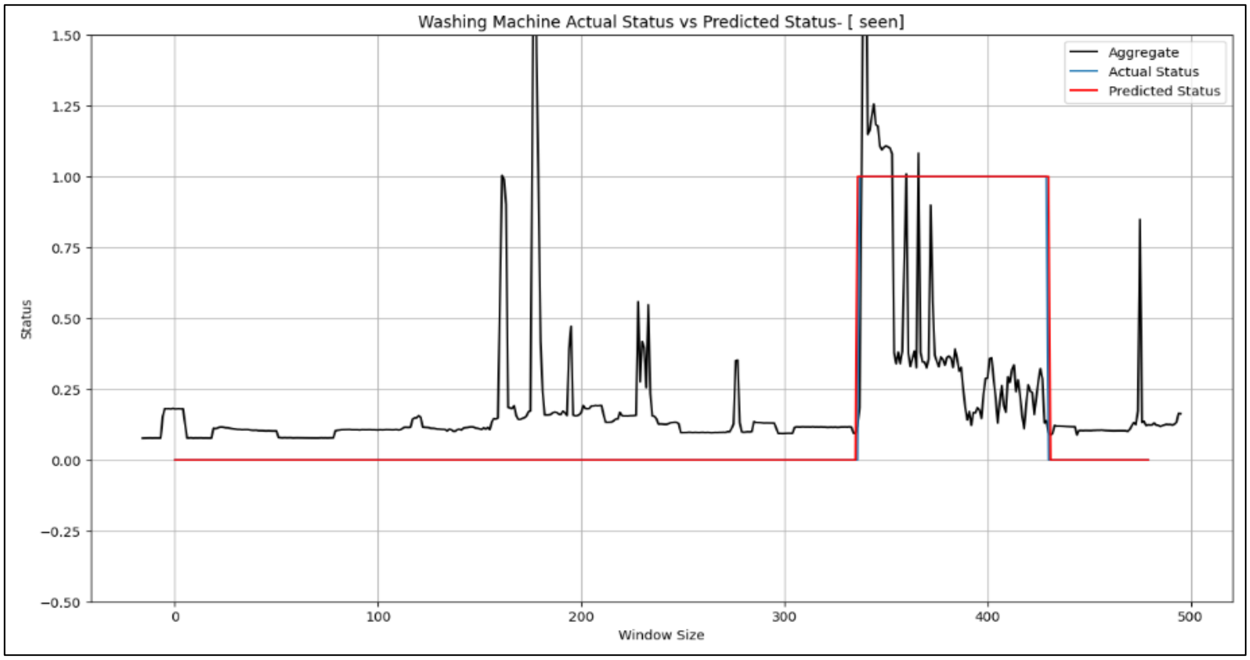

Figure 46: Status washing Machine Prediction in the seen case for CONV-BILSTM.

Similarly, the Conv-BiLSTM model also demonstrates strong performance in the seen case for status prediction. Both PTPNET and Conv-BiLSTM accurately predict the statuses of the appliances, showing close alignment with the actual data.

```
fridge
F1 Score   :0.9707112970711297
Precision  :0.9789029535864979
Recall  :0.9626556016597511
Accuracy  :0.9708333333333333
MCC  :0.9418005057872556

dish_washer
F1 Score  :1.0
Precision  :1.0
Recall  :1.0
Accuracy  :1.0
MCC  :1.0

washing_machine
F1 Score  :0.9893617021276596
Precision  :0.9789473684210527
Recall  :1.0
Accuracy  :0.9958333333333333
MCC  :0.9868577450141074
```

Figure 47: Status Prediction metrics in seen case for CONV-BILSTM.

```
fridge
F1 Score  :0.9565217391304348
Precision  :0.9545454545454546
Recall  :0.9585062240663901
Accuracy  :0.95625
MCC  :0.9125048833338797

dish_washer
F1 Score  :0.9949748743718593
Precision  :1.0
Recall  :0.99
Accuracy  :0.9979166666666667
MCC  :0.9936808213924376

washing_machine
F1 Score  :0.989247311827957
Precision  :0.989247311827957
Recall  :0.989247311827957
Accuracy  :0.9958333333333333
MCC  :0.9866633324997917
```

Figure 48: Status Prediction metrics in seen case for PTPNET.

The results from both the PTPNet and Conv-BiLSTM models demonstrate their effectiveness in determining appliance statuses in seen scenarios.

The F1 Score integrates precision and recall, illustrating how well each model strikes a balance between accurately identifying when appliances are on and minimizing errors. For example, PTPNet achieved an F1 Score of 0.9707 for the fridge, indicating it correctly detects the fridge being on with a precision of 0.9789 and a recall of 0.9627. Precision measures the model's accuracy in positive predictions; a score of 0.9789 means it accurately identifies the fridge as on approximately 97.89% of the time. At the same time, recall measures the model's completeness in capturing all instances of a positive class; PTPNet achieves a recall of 0.9627 for the fridge, meaning it identifies 96.27% of all instances where the fridge is actually on.

Accuracy reflects the overall correctness of the model's predictions across all classes. For instance, Conv-BiLSTM achieved an accuracy of 0.9708 for the fridge, indicating that its predictions align with the actual statuses with high consistency. Matthews Correlation Coefficient (MCC) provides a comprehensive measure that considers both true and false predictions. A high MCC value, such as 0.9418 for the fridge using PTPNet, signifies strong alignment between predicted and actual appliance statuses. Similarly, for the dishwasher, Conv-BiLSTM achieved an F1 Score, precision, recall, accuracy, and MCC of 1.0, indicating perfect prediction performance.

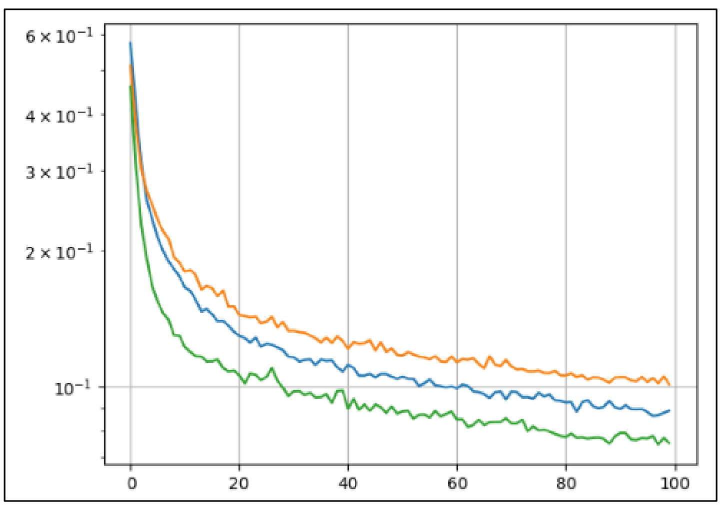

Figure 49: Loss curves of PTPNET in the unseen case- status prediction

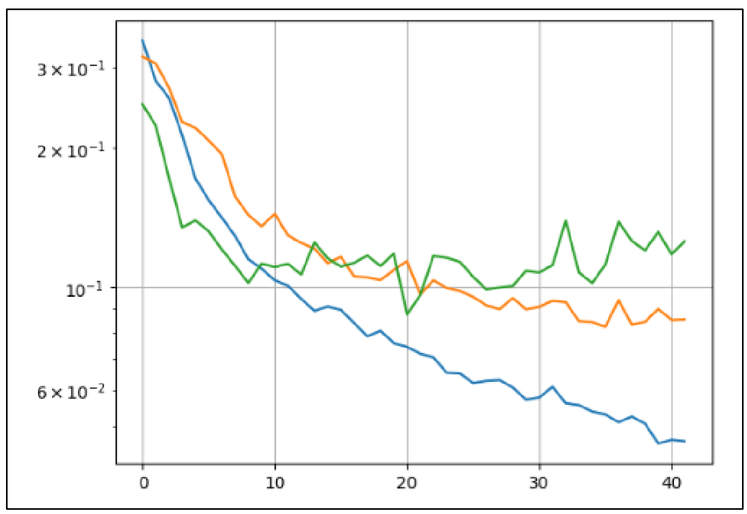

Figure 50: Loss curves of CONV-BILSTM in the unseen case- status prediction

It is evident that the losses for both models increase in the unseen case of status prediction. However, in the PTPNET model, the curves converge more smoothly. In contrast, the Conv-BiLSTM model shows more fluctuations. Although all three curves (training, validation, and testing) are decreasing for the PTPNET model, the validation and testing curves in the Conv-BiLSTM model become constant at certain points, indicating less stable performance.

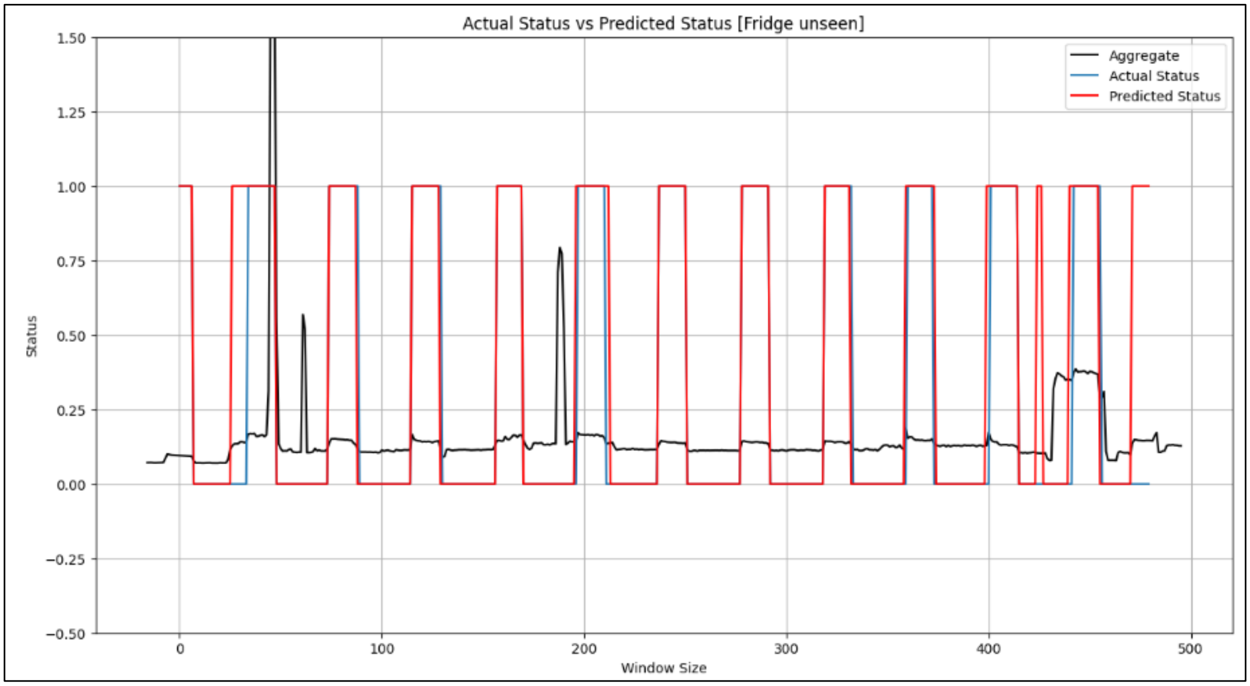

Figure 52: Status Fridge Prediction in the unseen case for PTPNET.

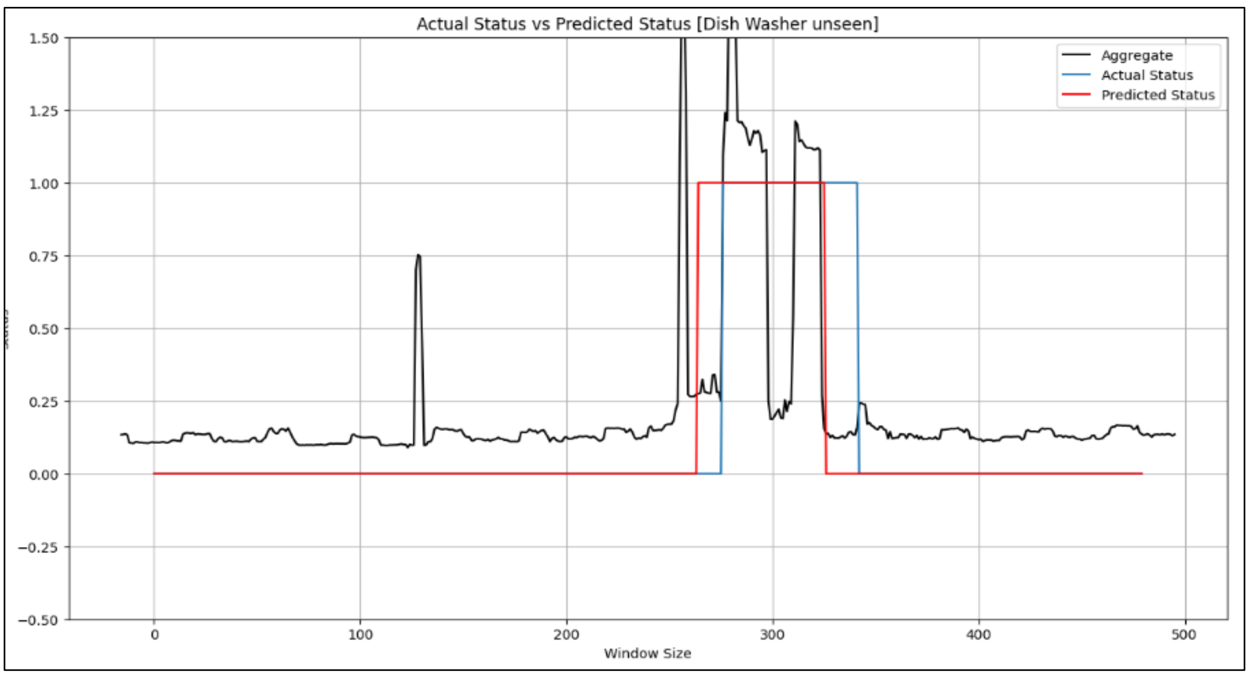

Figure 51: Status Dish Washer Prediction in the unseen case for PTPNET.

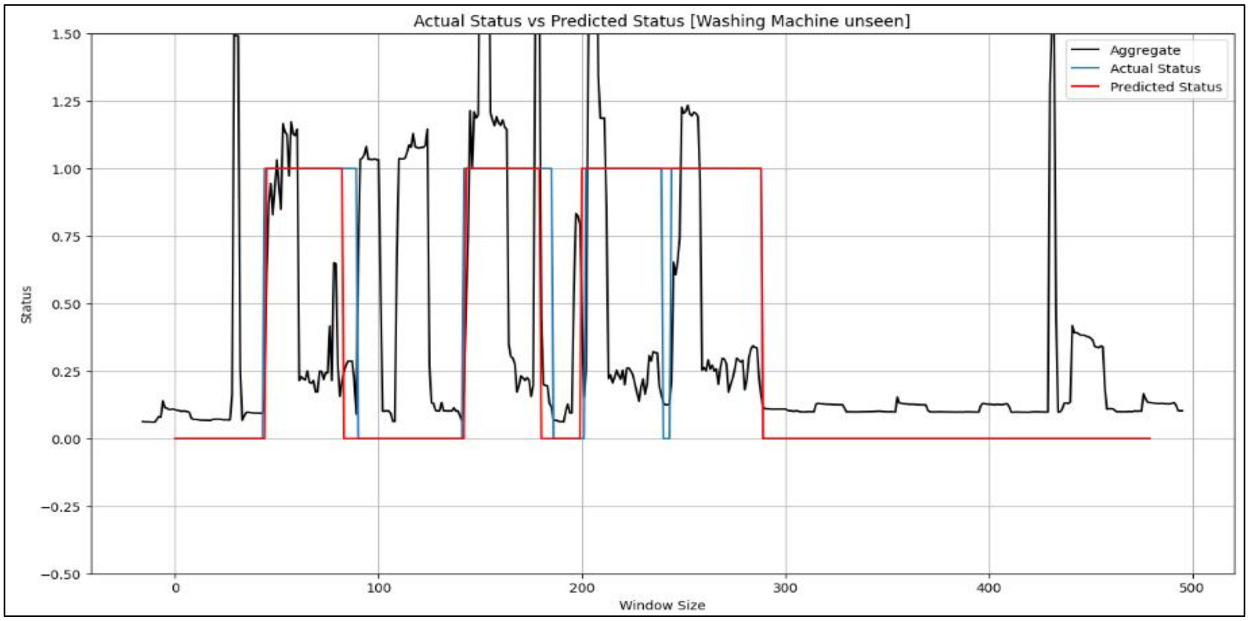

Figure 53: Status Washing Machine Prediction in the unseen case for PTPNET.

Also we can observe that the PTPNET performs less efficiently in the unseen case. This is because in the unseen scenario, the model is trained and validated using data from houses 1 and 5, but then tested on data from house 2, which is a completely different setting. Despite this, the model still accurately predicts whether the appliance is on or off, although with some delay in the prediction at some points.

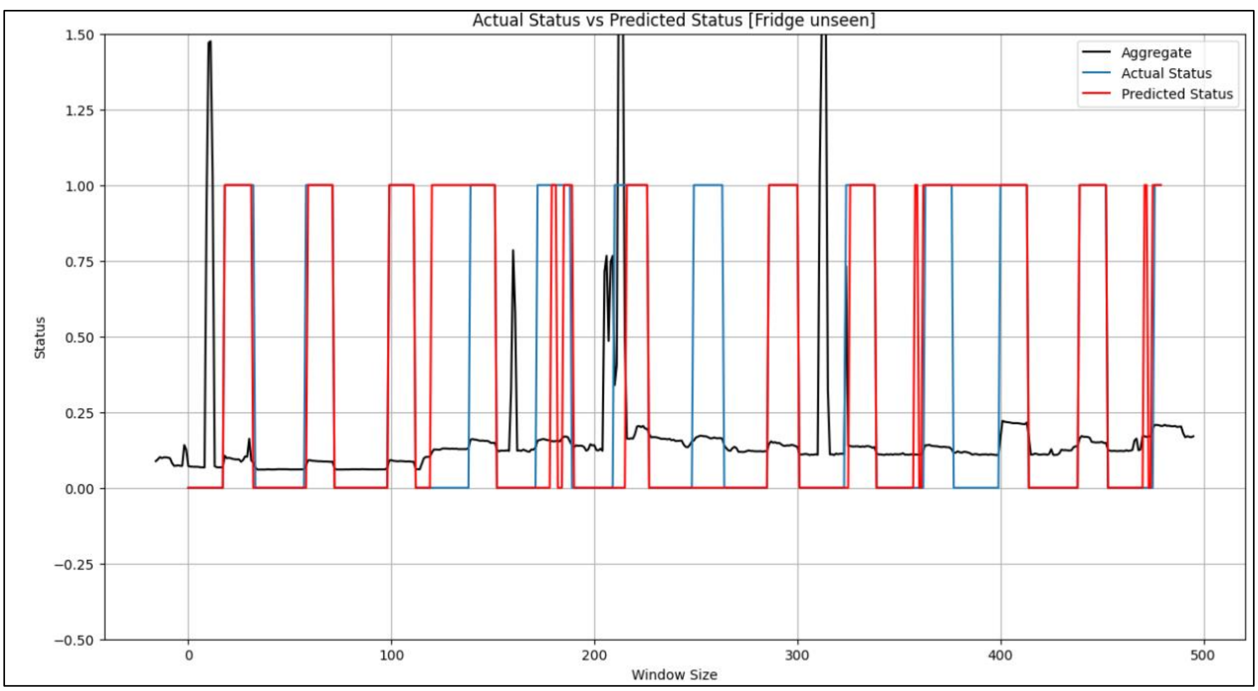

Figure 54: Status Fridge Prediction in the unseen case for CONV-BILSTM.

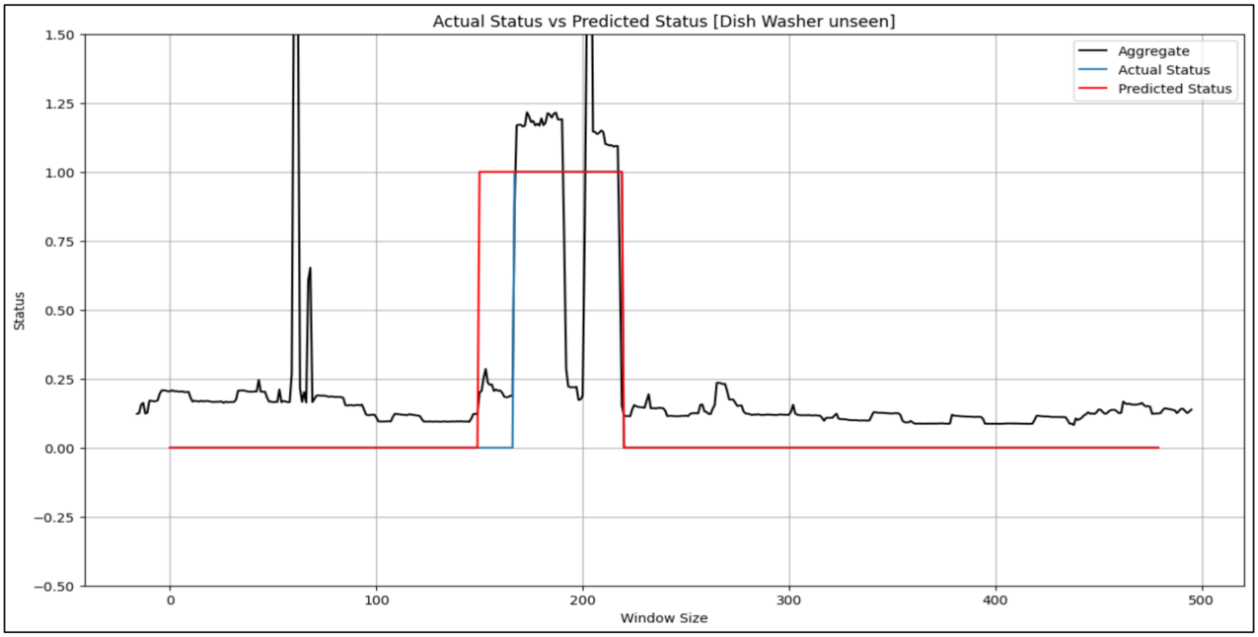

Figure 56: Status Dish Washer Prediction in the unseen case for CONV-BILSTM.

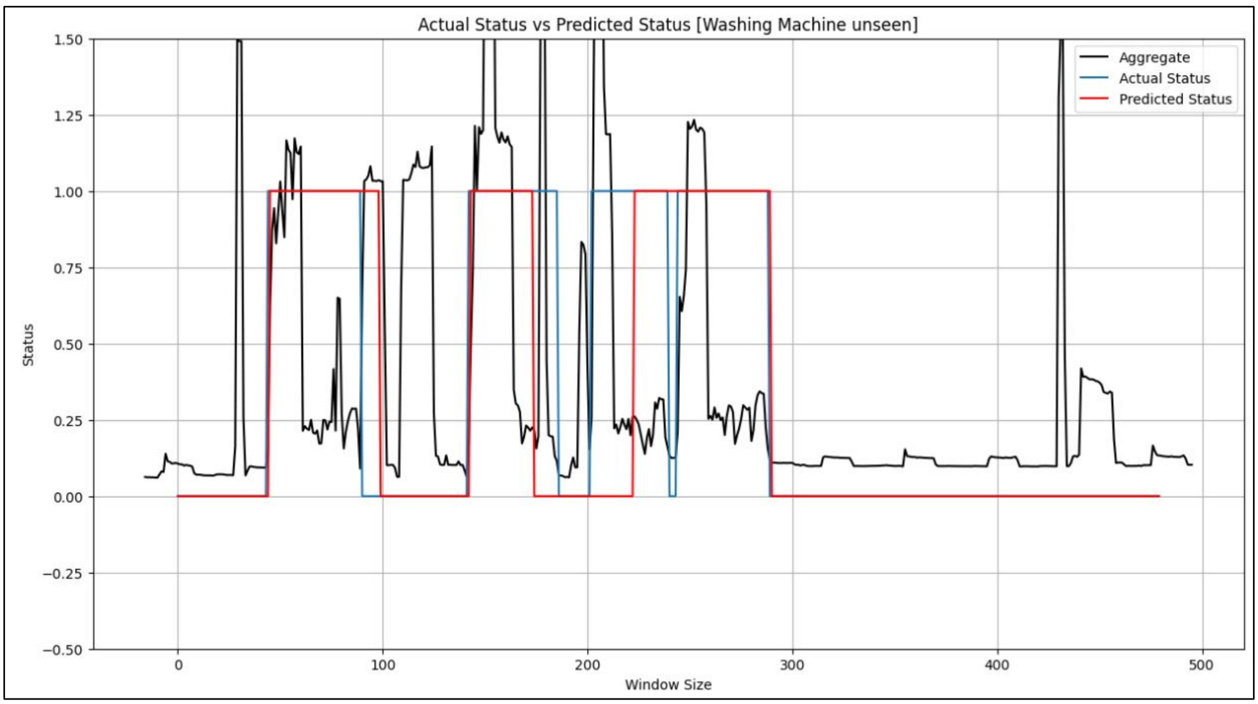

Figure 55: Status Washing Machine Prediction in the unseen case for CONV-BILSTM.

Unfortunately, the CONV-BILSTM model is not always able to accurately predict whether the appliance is on or off. In contrast, the PTPNET model demonstrates more accurate predictions of the appliance's status. This difference in performance is clear from the metrics shown in Figures 56 and 57.

```
fridge
F1 Score  :0.7754010695187166
Precision  :0.7474226804123711
Recall  :0.8055555555555556
Accuracy  :0.825
MCC  :0.6335754305211765

dish_washer
F1 Score  :0.8617886178861789
Precision  :0.7571428571428571
Recall  :1.0
Accuracy  :0.9645833333333333
MCC  :0.8526423613214625

washing_machine
F1 Score  :0.8492307692307692
Precision  :0.9078947368421053
Recall  :0.7976878612716763
Accuracy  :0.8979166666666667
MCC  :0.7762481853292692
```

Figure 58: Status Prediction metrics in the unseen case for CONV-BILSTM.

```
fridge
F1 Score  :0.9048991354466858
Precision  :0.8440860215053764
Recall  :0.9751552795031055
Accuracy  :0.93125
MCC  :0.8569418223967017

dish_washer
F1 Score  :0.7812499999999999
Precision  :0.8064516129032258
Recall  :0.7575757575757576
Accuracy  :0.9416666666666667
MCC  :0.7481204184498087

washing_machine
F1 Score  :0.9376854599406528
Precision  :0.9634146341463414
Recall  :0.9132947976878613
Accuracy  :0.95625
MCC  :0.9047804938235995
```

Figure 57: Status Prediction metrics in the unseen case for PTPNET.

In the unseen case predictions, PTPNet demonstrates a mixed performance compared to Conv-BiLSTM across the evaluated appliances. PTPNet achieves higher scores for some appliances but not all. For example, PTPNet achieves a higher F1 Score for the fridge (0.9049) compared to Conv-BiLSTM (0.7754), indicating better performance. However, for the dishwasher, Conv-BiLSTM achieves a higher F1 Score (0.8618) than PTPNet (0.7812). Similarly, for the washing machine, PTPNet has a higher F1 Score (0.9377) compared to Conv-BiLSTM (0.8492). This indicates that PTPNet generally performs better in predicting the status of the fridge and washing machine, while Conv-BiLSTM is more effective for the dishwasher. Overall, both models show meaningful predictive capability, with PTPNet generally performing better for most appliances.

REGRESSION TASK (POWER PREDICTION):

We conducted a similar comparison as in the status prediction task, but our focus shifted to predicting the power consumption of specific appliances such as the fridge, dishwasher, and washing machine. This task involved regression analysis, where we used evaluation metrics like MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error), and R squared to assess the models' performance. Additionally, we calculated and compared the total aggregate energy consumption, comparing the actual total energy consumed by each appliance with the predicted energy consumption as estimated by each model.

Moreover, we compared the models from an efficiency standpoint, evaluating their running time and prediction speed. This comprehensive evaluation was conducted for both seen and unseen data scenarios, allowing us to compare the overall performance of the PTPNet and Conv-BiLSTM models across multiple dimensions.

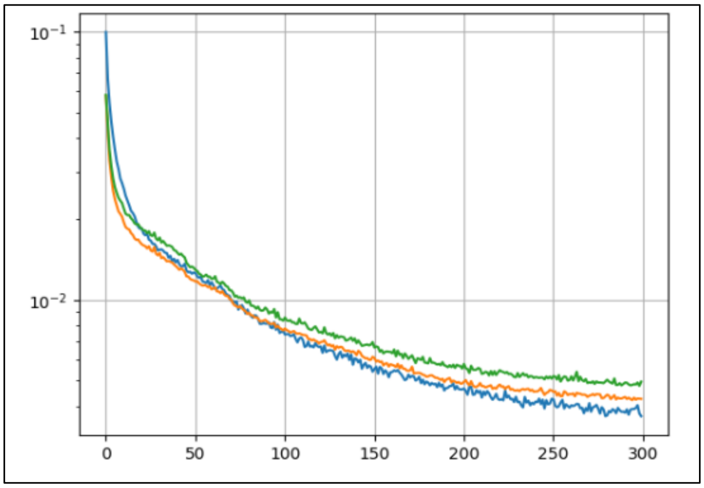

Figure 60: Loss curves of PTPNET in the seen case – power prediction.

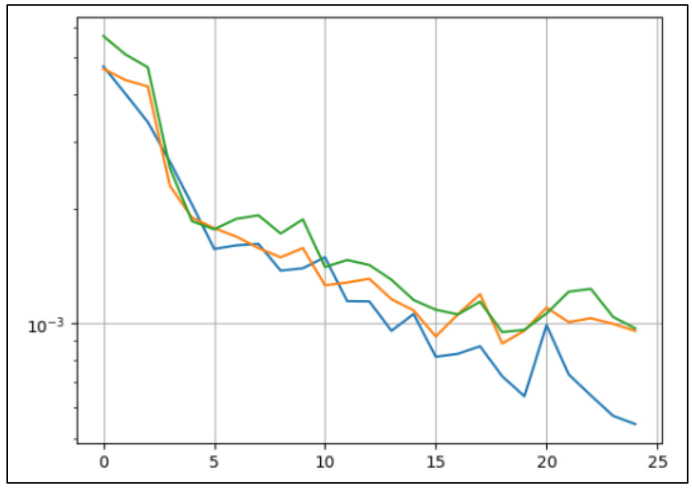

Figure 59: Loss curves of CONV-BILSTM in the seen case – power prediction

In Figures 58 and 59, tracking seen-case power prediction, PTPNet's loss curves are notably smoother and synchronized compared to Conv-BiLSTM, which shows more fluctuations and less stable convergence. PTPNet concludes training at 300 epochs based on optimal hyperparameter tuning as mentioned in previous sections, while Conv-BiLSTM stops around 25 epochs using early stopping to prevent overfitting.

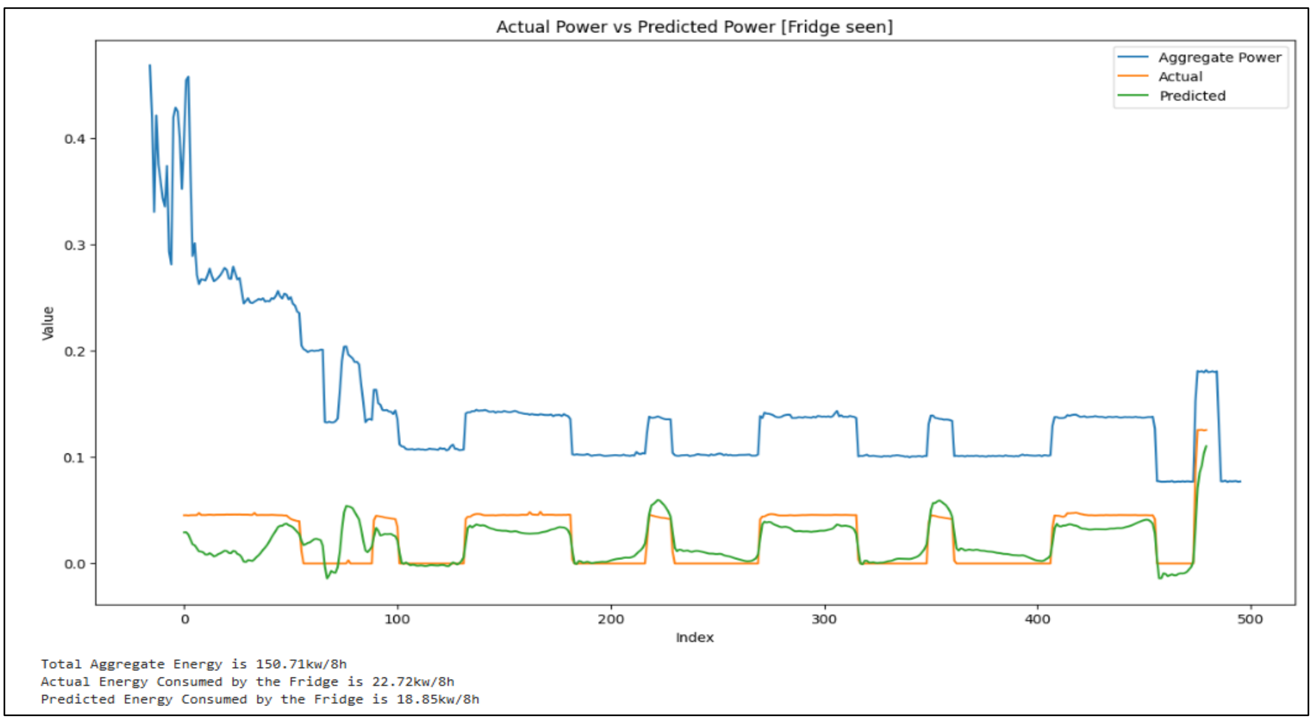

Figure 61: Power Fridge Prediction in the seen case for CONV-BILSTM.

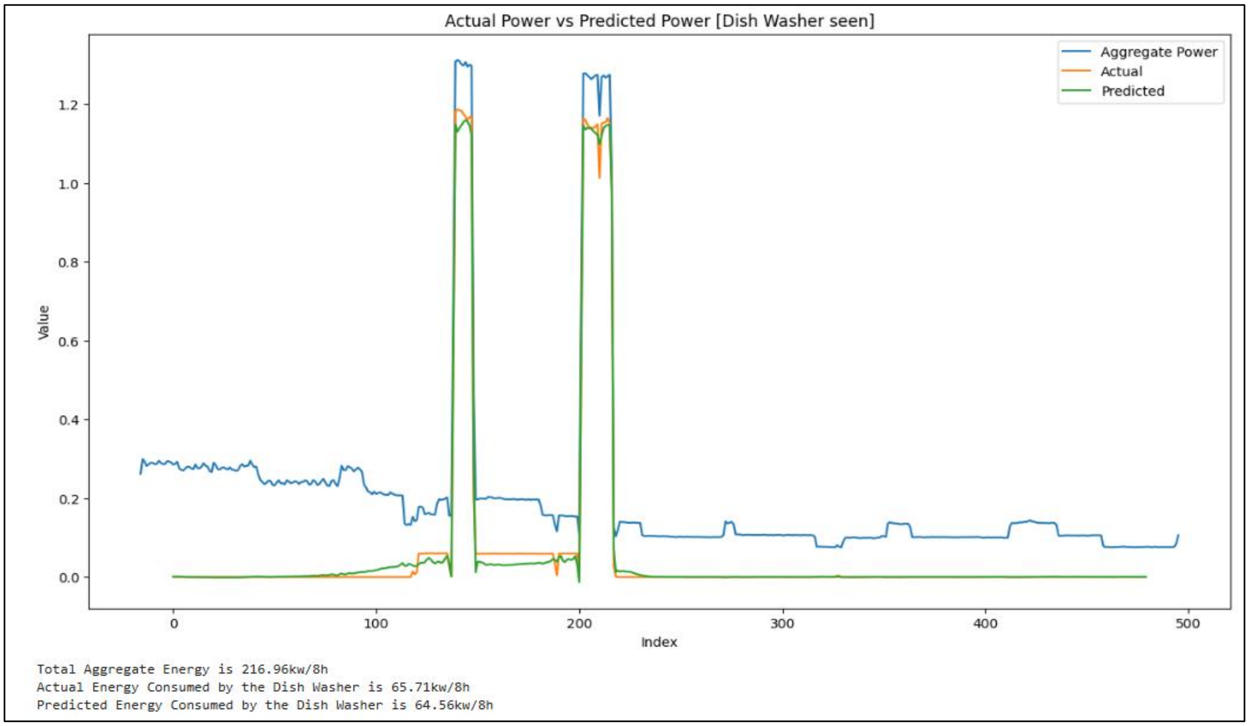

Figure 62: Power Dish Washer Prediction in the seen case for CONV-BILSTM.

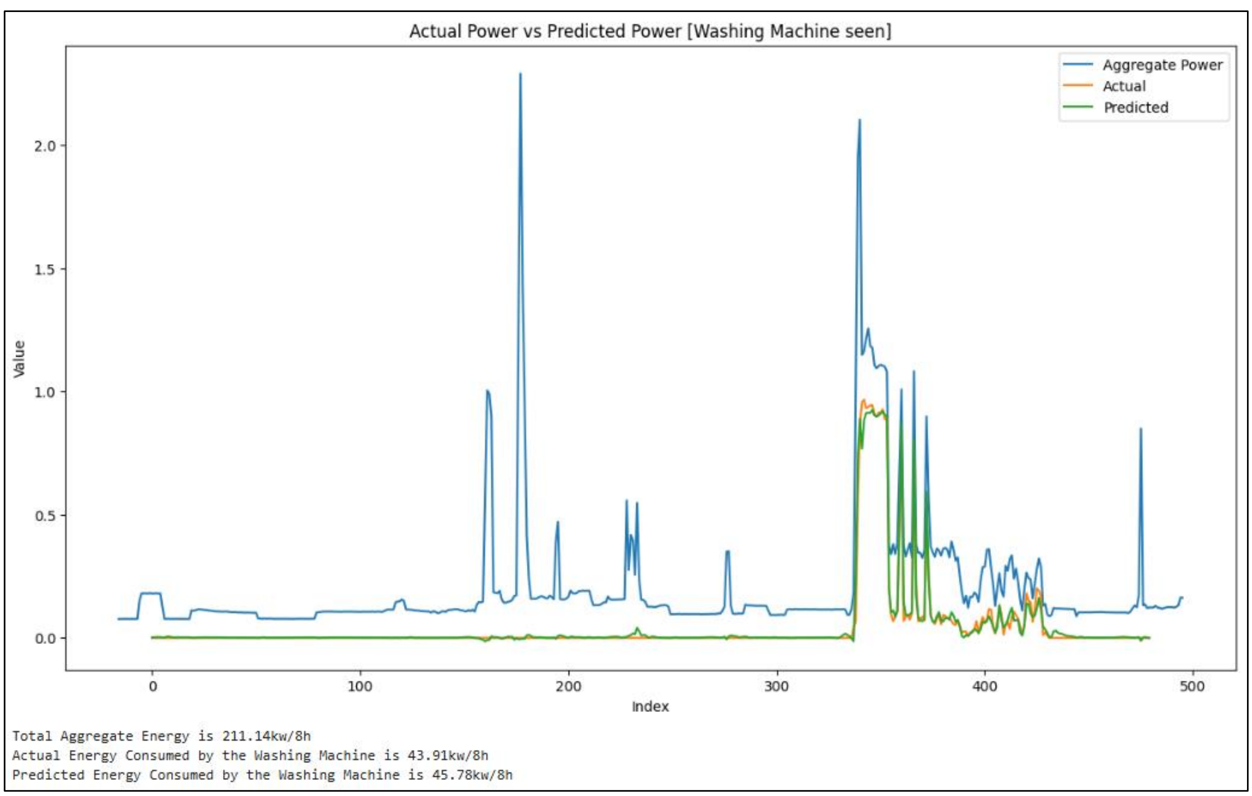

Figure 63: Power Washing Machine Prediction in the seen case for CONV-BILSTM.

Conv-BiLSTM performs better in predicting the power consumption of the dishwasher and washing machine compared to the fridge. The actual energy consumed by both appliances closely matches the predicted values. While the predicted energy consumption for the fridge is 18.85 kW/8h, close to the actual consumption of 22.72 kW/8h, it still shows reasonable accuracy.

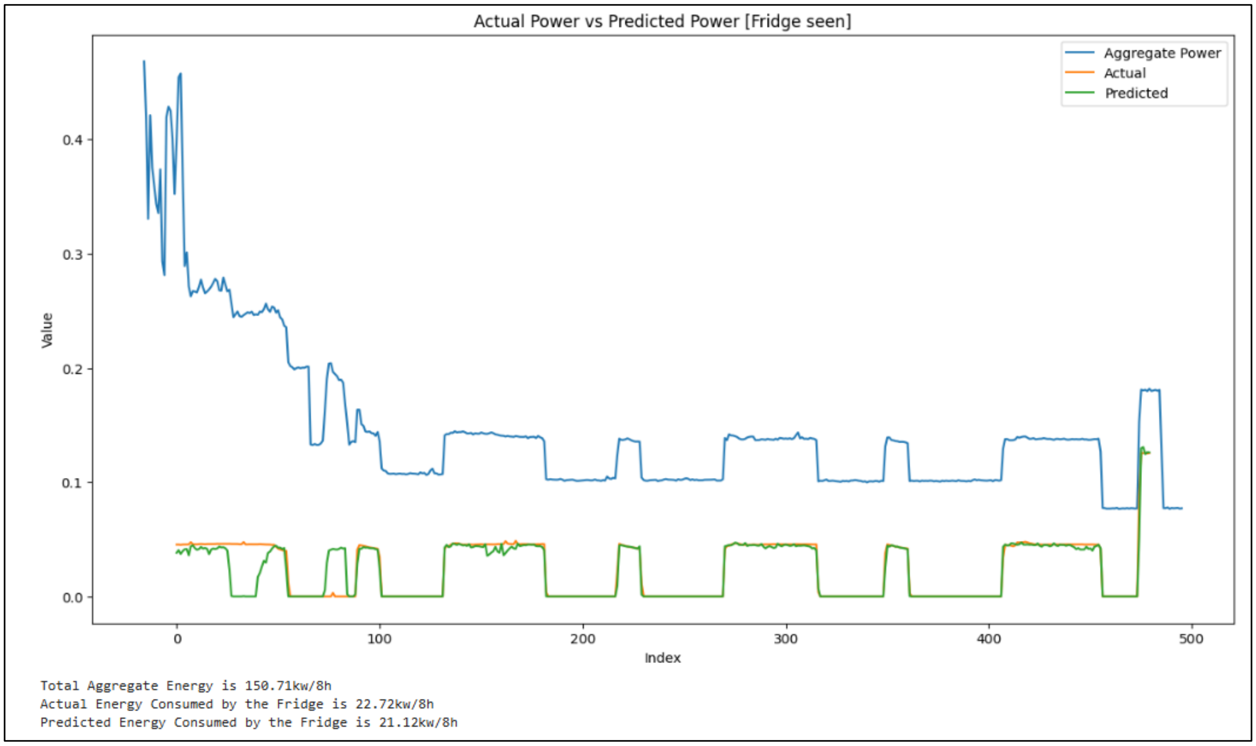

Figure 66: Power Fridge Prediction in the seen case for PTPNET.

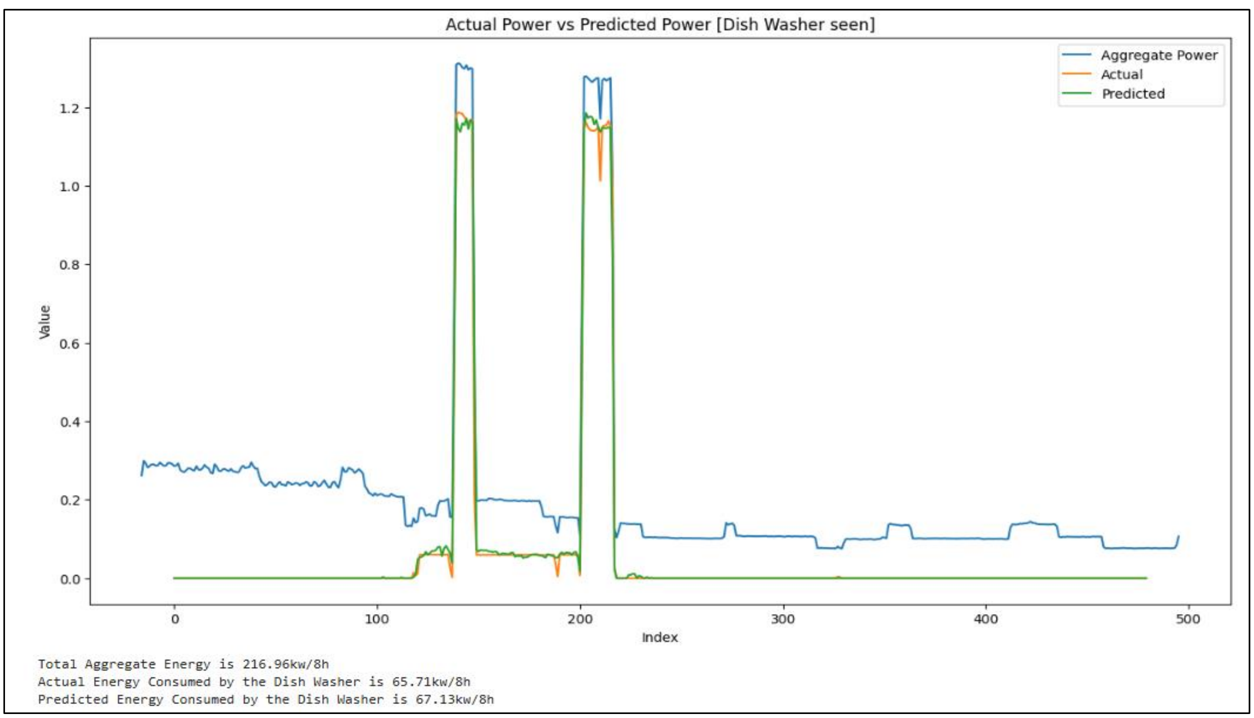

Figure 65: Power Dish Washer Prediction in the seen case for PTPNET.

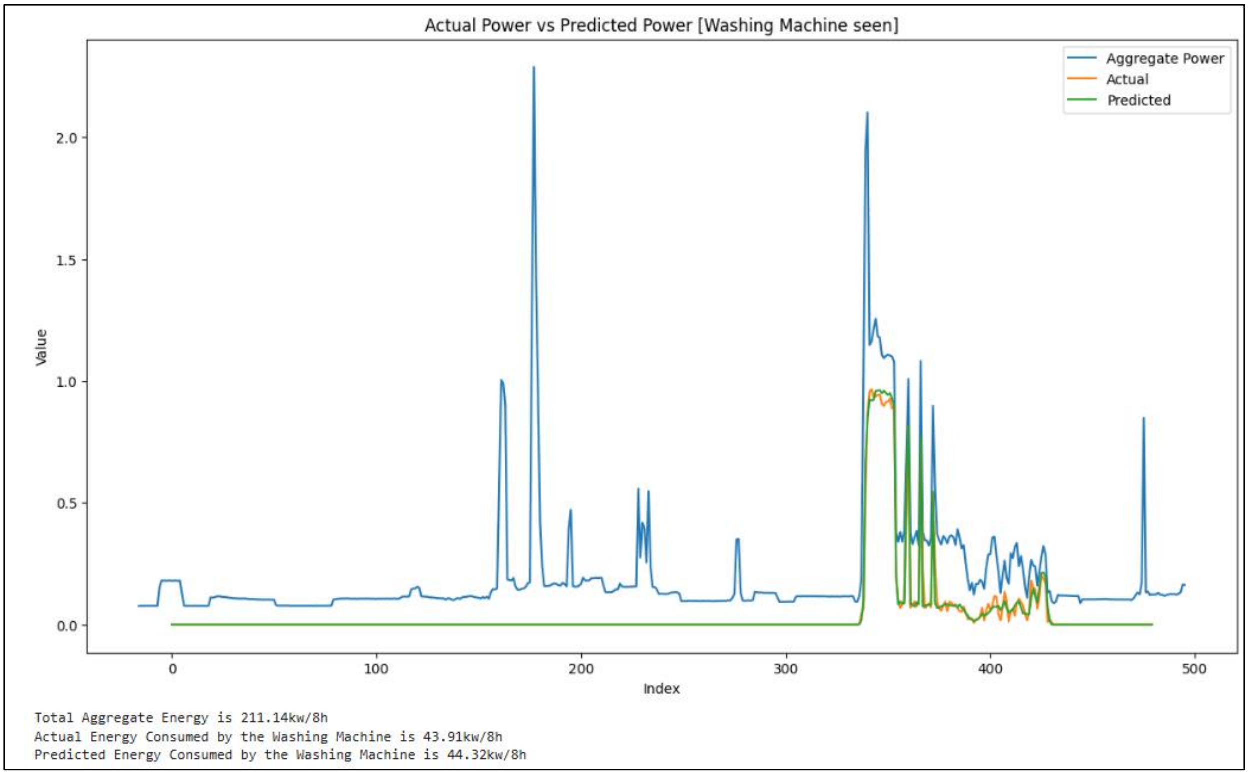

Figure 67: Power Washing Machine Prediction in the seen case for PTPNET.

PTPNet clearly outperforms Conv-BiLSTM in predicting power consumption, showing smoother and more confident signals across all appliances. Moreover, the difference between the actual and predicted energy for the fridge is smaller compared to Conv-BiLSTM.

```
fridge
MAE  :0.012747319415211678
MSE  :0.0002751215943135321
RMSE  :0.01658678986132145
R Squared  :0.5536022186279297

dish_washer
MAE  :0.007094299886375666
MSE  :0.00018994430138263851
RMSE  :0.01378202810883522
R Squared  :0.9969937014393508

washing_machine
MAE  :0.007470957934856415
MSE  :0.00038617735845036805
RMSE  :0.01965139620006084
R Squared  :0.9863424133509398
```

Figure 69: Power Prediction metrics in the seen case for Conv-BiLSTM.

```
fridge
MAE  :0.003587055951356888
MSE  :0.00010567851859377697
RMSE  :0.010280005633831024
R Squared  :0.8285316228866577

dish_washer
MAE  :0.003578596282750368
MSE  :0.00035551696782931685
RMSE  :0.018855158239603043
R Squared  :0.9943731389939785

washing_machine
MAE  :0.004715790040791035
MSE  :0.00027398887323215604
RMSE  :0.016552608460187912
R Squared  :0.9903100822120905
```

Figure 68: Power Prediction metrics in the seen case for PTPNET.

The power prediction results for the seen case show that both PTPNet and Conv-BiLSTM models work well, but PTPNet is better overall.

Mean Absolute Error (MAE) measures the average error in predictions. PTPNet has a lower MAE for the fridge (0.0036) compared to Conv-BiLSTM (0.0127), showing it's more precise. Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) measure the average squared errors and their square root. PTPNet has lower MSE (0.000106) and RMSE (0.0103) for the fridge than Conv-BiLSTM (0.000275 and 0.0166), indicating better accuracy. R-squared (R^2) indicates how much variance in the data is explained by the model. PTPNet's higher R^2 of 0.8285 means it explains more variance than Conv-BiLSTM's 0.5536.

For the dishwasher and washing machine, both models perform well, but PTPNet is slightly better. PTPNet's MAE for the dishwasher is 0.0036 and for the washing machine is 0.0047, compared to Conv-BiLSTM's 0.0071 and 0.0075. Lower RMSE values for PTPNet mean more accurate predictions. High R^2 values for both models (0.997 for Conv-BiLSTM and 0.994 for PTPNet for the dishwasher; 0.986 for Conv-BiLSTM and 0.990 for PTPNet for the washing machine) show they both explain the variance well. Overall, PTPNet's predictions are smoother and more precise, making it the better model for seen data scenarios.

When comparing the training times of the PTPNet and Conv-BiLSTM models on the seen dataset, PTPNet took 5054.93 seconds to complete 300 epochs, while Conv-BiLSTM took 4288.94 seconds for around 25 epochs. This similarity in training times despite the large difference in epochs indicates that PTPNet is more efficient. If we extrapolate the Conv-BiLSTM time to 300 epochs, it would be significantly slower, approximately 10 times slower than PTPNet. Thus, PTPNet not only performs better but also achieves this efficiency with faster training per epoch.

Also, PTPNet took just 1.73 seconds to predict power in the seen case, whereas Conv-BiLSTM took 14.83 seconds. This further demonstrates that PTPNet is faster and more efficient in both training and prediction, making it a better choice for appliance status and power consumption prediction.

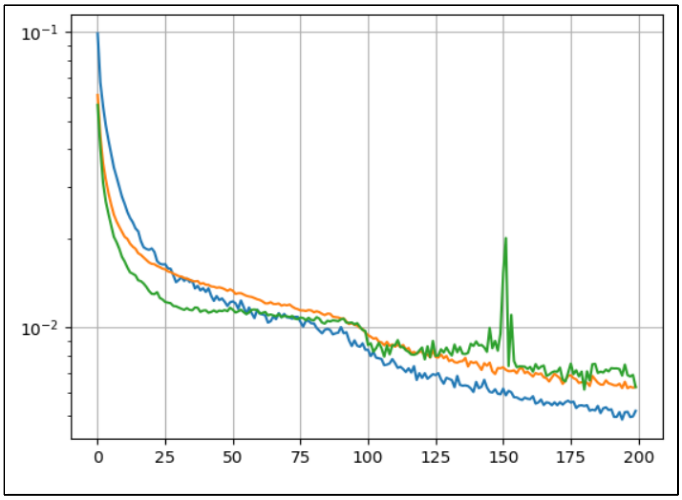

Figure 71: Loss curves of PTPNET in the unseen case – power prediction.

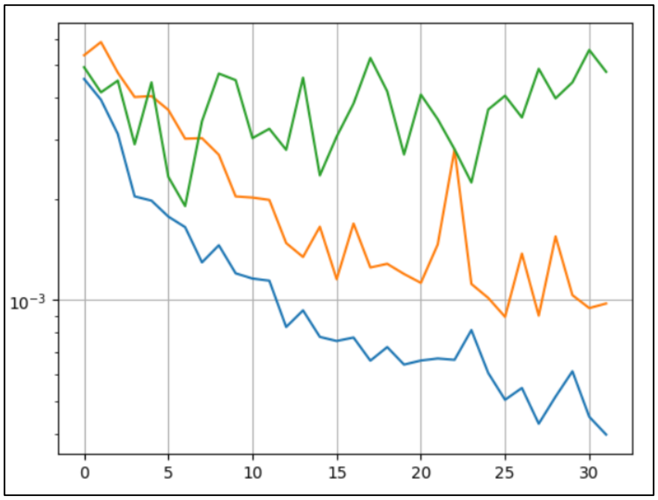

Figure 70: Loss curves of CONV-BILSTM in the unseen case – power prediction.

Figures 69 and 70 show the power prediction curves in the unseen case, similar to the loss curves of status prediction. The PTPNet model's curves converge smoothly with minor fluctuations, while the Conv-BiLSTM model has more fluctuations. PTPNet's training, validation, and testing curves all decrease steadily, but Conv-BiLSTM's validation and testing curves become constant at times, showing less stable performance.

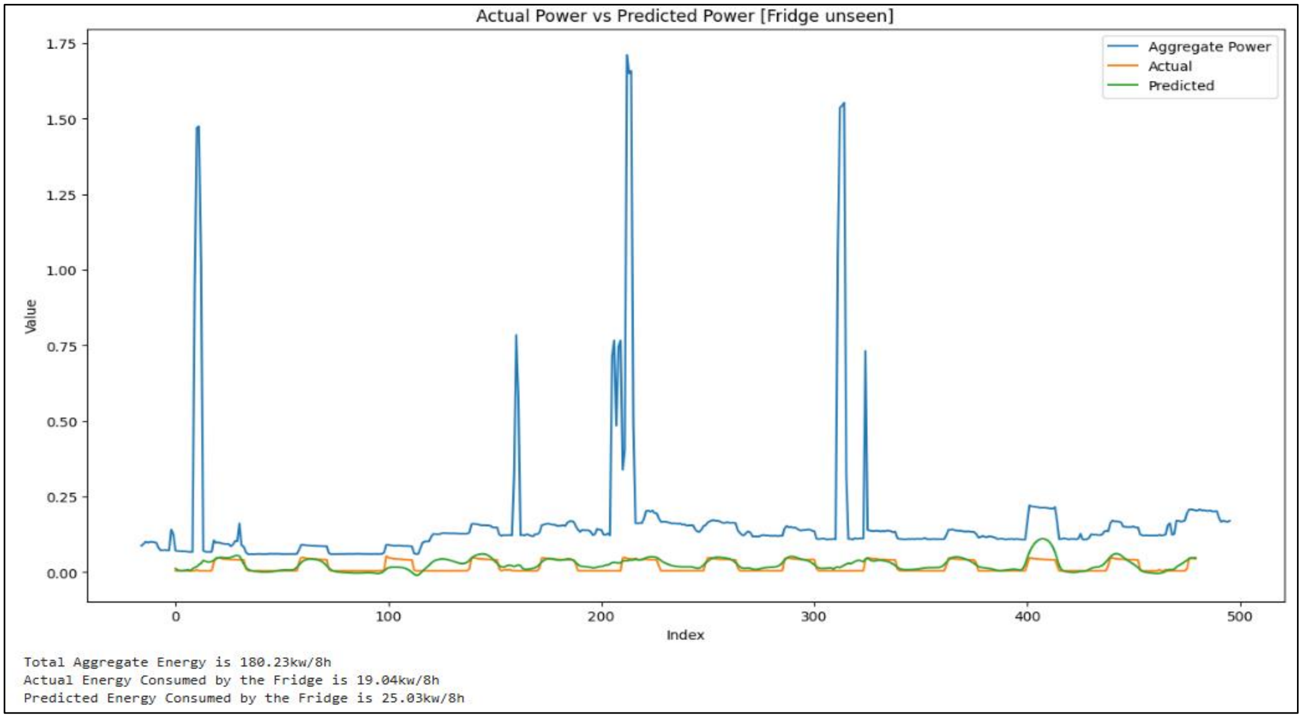

Figure 72: Power Fridge Prediction in the unseen case for CONV-BILSTM.

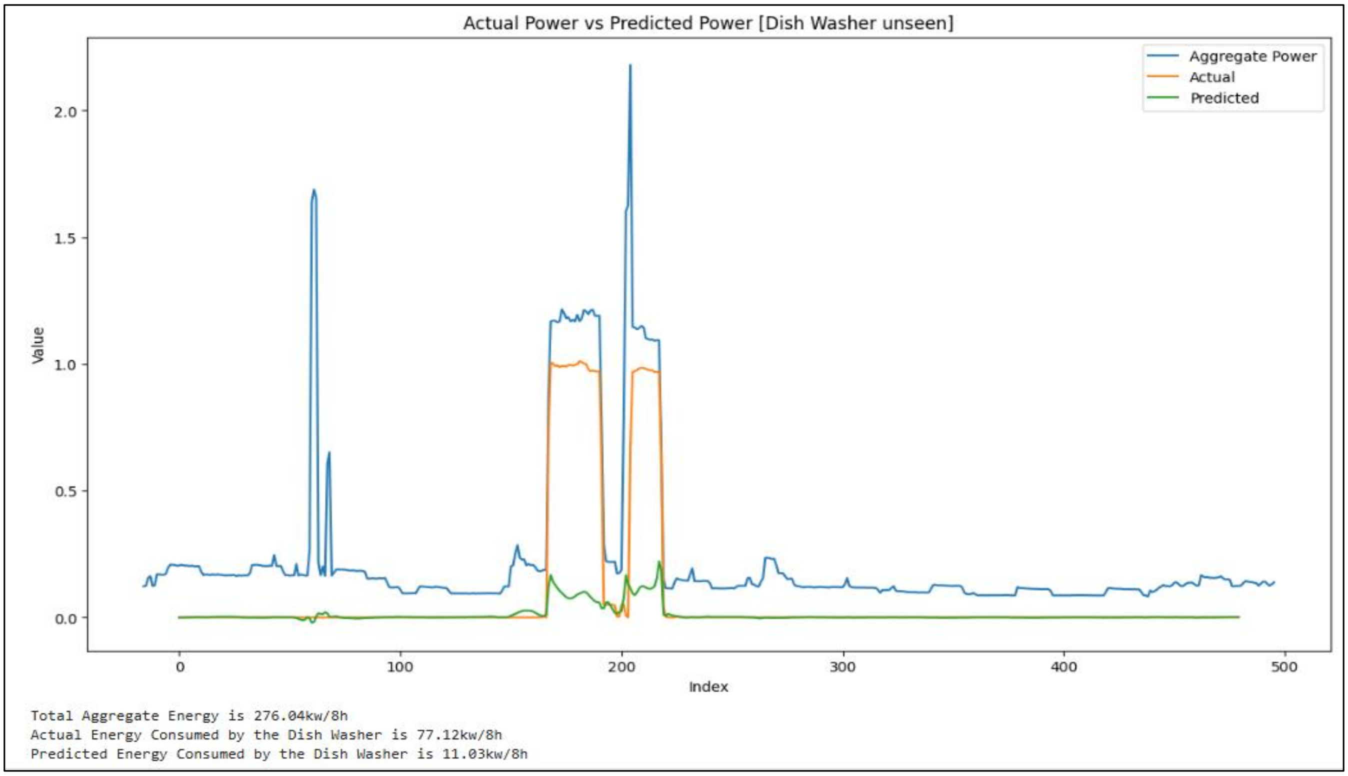

Figure 73: Power Dish Washer Prediction in the unseen case for CONV-BILSTM.

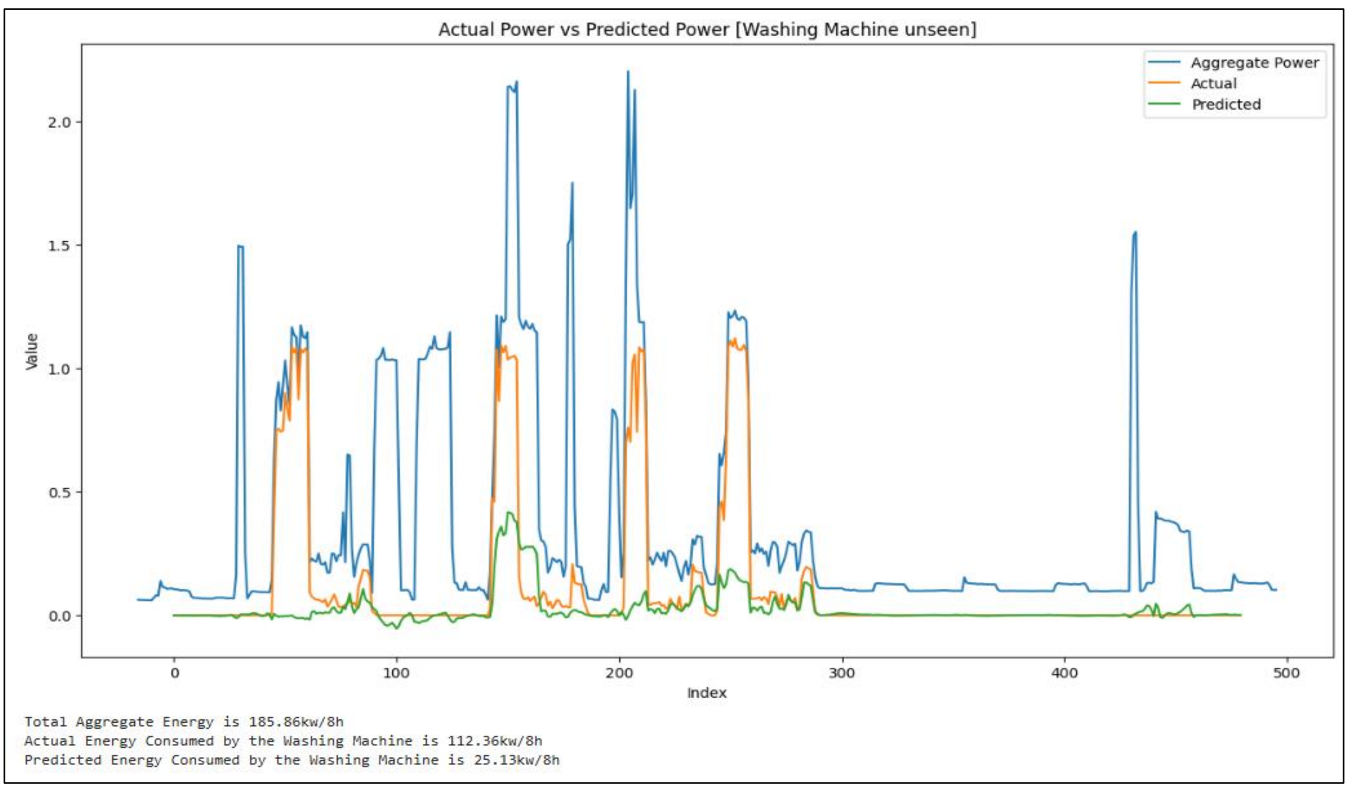

Figure 74: Power Washing Machine Prediction in the unseen case for CONV-BILSTM.

Conv-BiLSTM doesn't perform well on unseen data when predicting power usage. There is a clear difference between its predictions and the actual energy consumed by appliances. For example, it predicts the dishwasher will use 11.03 kW/8h, but it actually uses 77.03 kW/8h, showing it is not accurate.

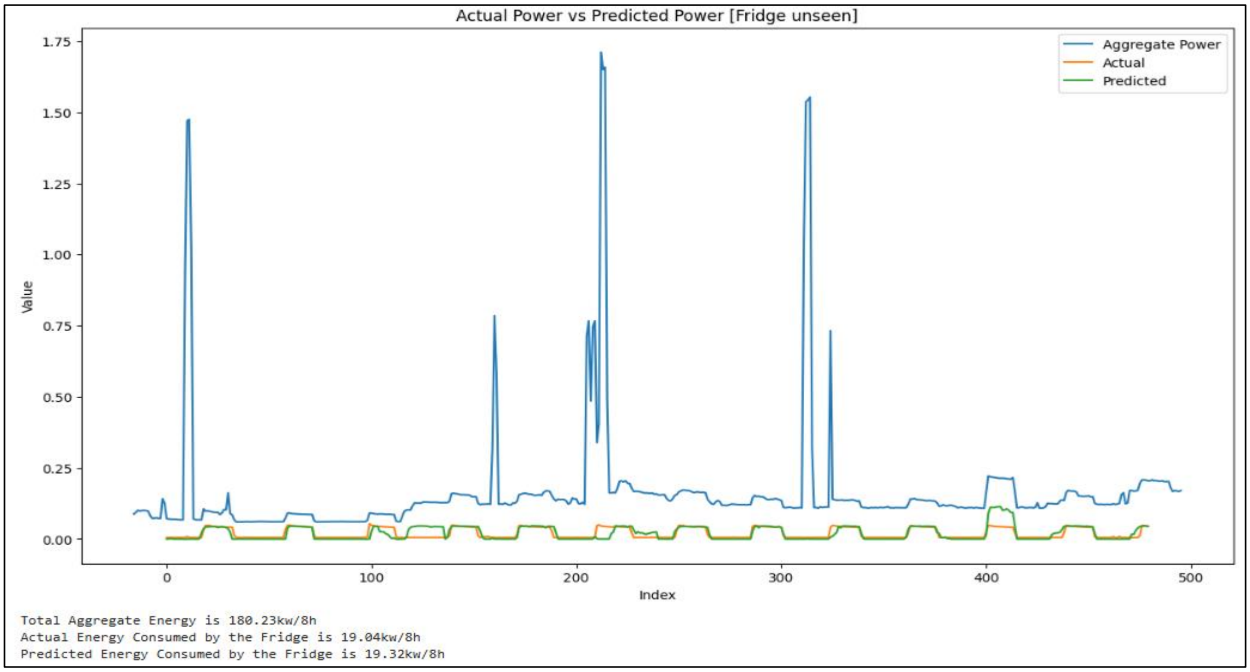

Figure 75: Power Fridge Prediction in the unseen case for PTPNET.

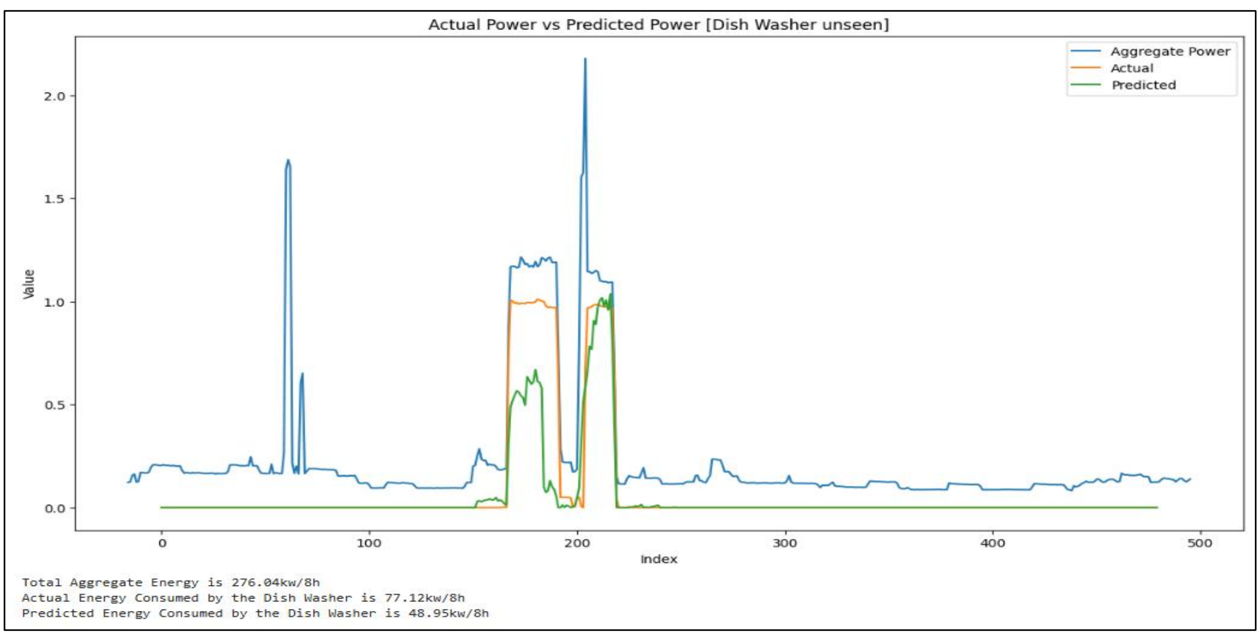

Figure 76: Power Washer Machine Prediction in the unseen case for PTPNET.

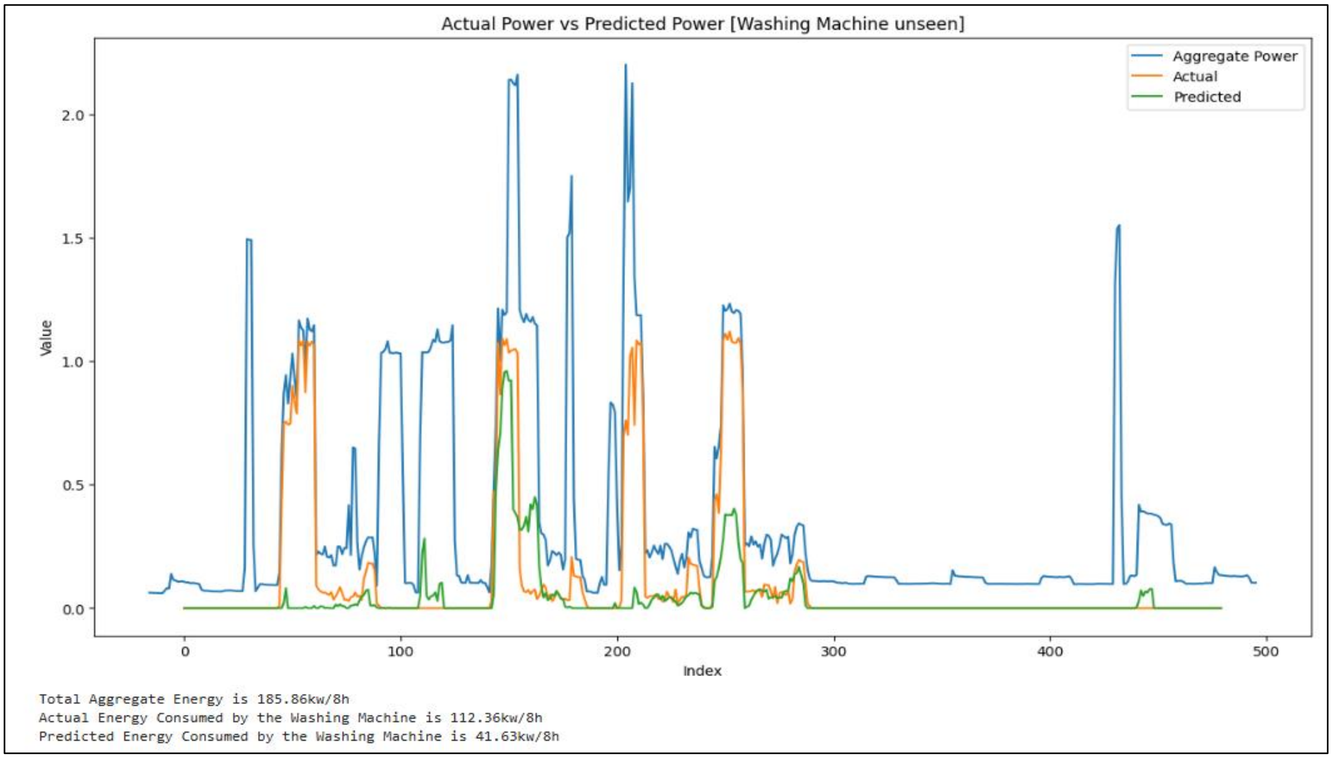

Figure 77: Power Washing Machine Prediction in the unseen case for PTPNET.

PTPNet attempts to capture the higher power values for the dishwasher and washing machine better than Conv-BiLSTM, although not perfectly accurate. Additionally, PTPNet shows a lower discrepancy between the actual and predicted energy for the fridge compared to Conv-BiLSTM, indicating its overall better performance.

```
fridge
MAE   :0.011578688398003578
MSE   :0.00026720258756540716
RMSE   :0.01634633168578148
R Squared   :0.20619404315948486

dish_washer
MAE   :0.07359108328819275
MSE   :0.060938797891139984
RMSE   :0.24685785174369812
R Squared   :0.12969970703125

washing_machine
MAE   :0.10277135670185089
MSE   :0.07412384450435638
RMSE   :0.2722569406032562
R Squared   :0.10540097951889038
```

Figure 78: Power Prediction metrics in the unseen case for Conv-BiLSTM.

```
fridge
MAE   :0.00978526659309864
MSE   :0.00027516798581928015
RMSE   :0.01658818870782852
R Squared   :0.1825304627418518

dish_washer
MAE   :0.036120783537626266
MSE   :0.02009369060397148
RMSE   :0.1417522132396698
R Squared   :0.7130310237407684

washing_machine
MAE   :0.09319838136434555
MSE   :0.06351472437381744
RMSE   :0.25202128291130066
R Squared   :0.23344218730926514
```

Figure 79: Power Prediction metrics in the unseen case for PTPNET.

The power prediction results for the unseen case indicate that both PTPNet and Conv-BiLSTM models perform, though not significantly. The shorter duration of recorded data in the test home may have contributed to this, making training and testing more challenging. Despite this limitation, PTPNet generally outperforms Conv-BiLSTM with lower errors across all metrics: Mean Absolute Error (MAE) is 0.0098 for the fridge, 0.0361 for the dishwasher, and 0.0932 for the washing machine, compared to Conv-BiLSTM's higher MAE values of 0.0116, 0.0736, and 0.1028, respectively. PTPNet also shows slightly better accuracy with lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values for most appliances. Moreover, PTPNet's higher R-squared (R^2) values indicate it explains more variance in the data, making it more reliable for predicting power consumption in these conditions.

In the unseen data scenario, PTPNet trained for 200 epochs in approximately 3521.32 seconds, whereas Conv-BiLSTM required about 4759.07 seconds for around 32 epochs. This suggests that PTPNet is more efficient in terms of training time despite running more epochs, likely due to its faster convergence. Also, PTPNet took only 3.54 seconds to make predictions on unseen data, whereas Conv-BiLSTM required approximately 30.39 seconds "almost ten times longer". This significant difference in prediction time further highlights PTPNet's efficiency in real-time applications. Although both models show room for improvement in predicting power consumption in unseen data, PTPNet demonstrates better performance with lower training times and faster predictions compared to Conv-BiLSTM.

5.8 Comparative Analysis of Appliance Status Prediction Models: Our PTPNet, Reference PTPNet, and Conv-BiLSTM

SEEN CASE:

Table 1: Comparison of Status Prediction Metrics for Seen Case between our PTPNet, reference PTPNet, and Conv-BiLSTM.

Fridge					
	Precision	**Recall**	**Accuracy**	**F1 Score**	**MCC**
Refrence PTPNet	0.88	0.86	0.88	0.87	0.76
Our PTPNet	0.954	0.958	0.956	0.956	0.912
Conv-BiLSTM	0.978	0.962	0.970	0.970	0.941
Dishwasher					
PTPNet - Paper	0.94	0.92	1.00	0.93	0.93
Our PTPNet	1.0	0.99	0.997	0.994	0.993
Conv-BiLSTM	1.0	1.0	1.0	1.0	1.0
Washing Machine					
Refrence PTPNet	0.98	0.98	1.00	0.98	0.98
Our PTPNet	0.989	0.989	0.995	0.989	0.986
Conv-BiLSTM	0.978	1.0	0.995	0.989	0.986

UNSEEN CASE:

Table 2: Comparison of Status Prediction Metrics for unseen Case between our PTPNet, reference PTPNet, and Conv-BiLSTM.

Fridge					
	Precision	**Recall**	**Accuracy**	**F1 Score**	**MCC**
Refrence PTPNet	0.89	0.85	0.91	0.87	0.80
Our PTPNet	0.844	0.975	0.931	0.904	0.856
Conv-BiLSTM	0.747	0.805	0.825	0.775	0.633
Dishwasher					
Refrence PTPNet	0.79	0.84	0.99	0.81	0.81
Our Implementaion	0.806	0.757	0.941	0.781	0.748
Conv-BiLSTM	0.757	1.0	0.964	0.861	0.852
Washing Machine					
Refrence PTPNet	0.86	0.87	1.00	0.86	0.86
Our Implementaion	0.963	0.913	0.956	0.937	0.904
Conv-BiLSTM	0.907	0.797	0.897	0.849	0.776

In both seen and unseen cases, we compare three models: our implementation of PTPNet, PTPNet from the reference paper, and Conv-BiLSTM.

Seen Case:

For the fridge, our PTPNet and Conv-BiLSTM both outperform the reference paper's PTPNet. Conv-BiLSTM slightly edges out with a higher F1 Score (0.970) and MCC (0.941) compared to our PTPNet (F1 Score 0.956, MCC 0.912). The dishwasher shows perfect performance by Conv-BiLSTM (F1 Score 1.0) and very high performance by our PTPNet (F1 Score 0.994), both better than the reference paper's PTPNet. For the washing machine, our PTPNet and Conv-BiLSTM perform similarly, both achieving an F1 Score of 0.989, which is slightly higher than the reference paper's PTPNet (F1 Score 0.98).

Unseen Case:

For the fridge in the unseen case, our PTPNet (F1 Score 0.904) performs better than both the reference paper's PTPNet (F1 Score 0.87) and Conv-BiLSTM (F1 Score 0.775). For the dishwasher, Conv-BiLSTM (F1 Score 0.861) performs better than both our PTPNet (F1 Score 0.781) and the reference paper's PTPNet (F1 Score 0.81). For the washing machine, our PTPNet (F1 Score 0.937) again outperforms both the reference paper's PTPNet (F1 Score 0.86) and Conv-BiLSTM (F1 Score 0.849).

In summary, while Conv-BiLSTM excels in the seen case, especially for the dishwasher, our PTPNet demonstrates greater performance in the unseen case, particularly for the fridge and washing machine, indicating its robustness in handling new data.

It is important to note that we cannot compare power prediction performance with the reference paper's PTPNet, as it does not predict power but rather multiplies predicted status with a constant, as detailed in previous sections.

Chapter 6: Conclusion, Challenges and Future Works

In conclusion, our goal was to reduce power consumption via load disaggregation, enabling users to monitor and manage their energy usage more effectively by identifying how much power each appliance consumes. Our work was divided into two tasks: the classification task and the regression task. The classification task focused on predicting the status of the target appliances (on/off), while the regression task aimed to predict the power consumption values over time for these appliances. By focusing on three main appliances: the fridge, washing machine, and dishwasher, we utilized two primary models, PTP-Net and Conv-BiLSTM, trained and tested on the UK-DALE dataset in both seen and unseen cases. Our comparative analysis involved evaluating their performance in predicting the status of target appliances and their power consumption, assessing the efficiency of the models, and discussing and comparing the results of the evaluation metrics with the reference paper. Our models demonstrated strong performance in predicting both appliance status and power consumption, outperforming the reference paper in several instances. This highlights the effectiveness of our approach in accurately disaggregating appliance-level activation status and power consumption.

Our study faced several challenges. Firstly, the data used for training and testing poses challenges due to its imbalance and the differences between houses. House 1's appliances are significantly larger than those in other houses, and the recorded power duration is much longer. A more balanced and varied dataset would likely yield better results, especially in predicting unseen scenarios. This limitation also affected the number of appliances included in our study, as we focused on those consistently monitored across all houses. Also, tasks and models of this nature require high computational resources, such as GPUs, and considerable runtime, making platforms like Kaggle or Colab insufficient, especially considering they offer GPUs for limited durations.

Looking ahead, our future work aims to enhance both models to achieve more accurate predictions. This is crucial because accurate data is essential for empowering consumers to adopt better energy-saving habits. We plan to develop a user-friendly application where users can input data such as their daily departure and return times, this will help in optimizing appliance usage, ensuring that appliances are not used unnecessarily when household members are absent. The application will suggest optimal times for using appliances, such as postponing washing machine use until nighttime during summer to avoid peak power consumption hours. Additionally, we will present prediction results in a user-friendly format, providing insights into each appliance's power usage to facilitate informed decision-making based on this data.

Bibliography

[1] J. D. Kelly, "Disaggregation of domestic smart meter energy data," *Imperial College London,* 2017.

[2] X. Zhou, J. Feng and Y. Li, "Non-intrusive load decomposition based on CNN–LSTM hybrid deep learning model," *Energy Reports,* vol. 7, no. 2352-4847, pp. 5762-5771, 2021.

[3] C. C. YANG, C. S. SOH and V. V. YAP, "A systematic approach in load disaggregation utilizing a multi-stage classification algorithm for consumer electrical appliances classification," *Frontiers Energy,* vol. 13, no. 2, pp. 386-398, 2019.

[4] L. Tokam and S. Ouro-djobo, "Non-Intrusive Load Monitoring (NILM), Interests and Applications," *Journal of the Nigerian Society of Physical Sciences,* 2023.

[5] Y.-B. Chiu and C.-C. Lee, "Effects of financial development on energy consumption: The role of country risks," *Energy Economics,* vol. 90, no. 0140-9883, 2020.

[6] M. Taha, . O. Sharqieh and M. A. D. Al Arda , "Recommended National Sustainable Urban and Energy Savings Actions for Palestine," 2015.

[7] J. Chontanawat, "Relationship between energy consumption, CO2 emission and economic growth in ASEAN: Cointegration and causality model," *Energy Reports,* vol. 6, no. 2352-4847, pp. 660-665, 2020.

[8] "PALESTINE RENEWABLE ENERGY PROJECT (PREP)," AECOM International Development, 2017.

[9] "The environmental benefits of energy efficiency," *energysage,* 2023.

[10] "UK-DALE," *paperswithcode.*

[11] Y. Kwak, J. Hwang and T. Lee, "Load Disaggregation via Pattern Recognition: A Feasibility Study of a Novel Method in Residential Building," *energies,* 2018.

[12] R. Ramadan, Q. Huang, O. Bamisile, A. Zalhaf, K. Mahmoud, M. Lehtonen and M. Darwish, "Towards Energy-Efficient Smart Homes via Precise Non-Intrusive Load Disaggregation Based on Hybrid ANN-PSO," *Energy Science & Engineering,* 2023.

[13] A. Moradzadeh, O. Sadeghian, K. Pourhossein, B. Mohammadi-Ivatloo and A. Anvari-Moghaddam, "Improving Residential Load Disaggregation for Sustainable Development of Energy via Principal Component Analysis," *Sustainability,* 2020.

[14] M. Zhuang, M. Shahidehpour and Z. Li, "An Overview of Non-Intrusive Load Monitoring: Approaches, Business Applications, and Challenges," pp. 4291-4299, 2018.

[15] L. Norford and S. Leeb, "Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms," *Energy and Buildings,* vol. 24, pp. 51-64, 1995.

[16] I. Rahman, M. Kuzlu and S. Rahman, "Power disaggregation of combined HVAC loads using supervised machine learning algorithms," *Energy and Buildings,* vol. 172, no. 0378-7788, pp. 57-66, 2018.

[17] F. Kalinke, P. Bielski, S. Singh, E. Fouché and K. Böhm, "An Evaluation of NILM Approaches on Industrial Energy-Consumption Data," *The Twelfth ACM International Conference on Future Energy,* 2021.

[18] Y. Zhang, G. Yang and S. Ma, "Non-intrusive Load Monitoring based on Convolutional Neural Network with Differential Input," *Procedia CIRP,* vol. 83, no. 2212-8271, pp. 670-674, 2019.

[19] C. Li, Z. Ding, D. Zhao, J. Yi and G. Zhang, "Building Energy Consumption Prediction: An Extreme Deep Learning Approach," *Energies,* vol. 10, no. 1996-1073, p. 1525, 2017.

[20] T.-Y. Kim and S.-B. Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy,* vol. 182, no. 0360-5442, pp. 72-81, 2019.

[21] L. Massidda, M. Marrocu and S. Manca, "Non-Intrusive Load Disaggregation by Convolutional Neural Network and Multilabel Classification," *Applied Sciences,* vol. 10, 2020.

[22] M. Irani Azad, R. Rajabi and A. Estebsari, "Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a Transformer Based Attention Mechanism and Temporal Pooling," *MDPI,* vol. 13, 2024.

[23] J. Kelly and W. Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," 2015.

[24] O. Krystalakos, C. Nalmpantis and D. Vrakas, "Sliding Window Approach for Online Energy Disaggregation Using Artificial Neural Networks," *ACM Digital Library,* JULY 2018.

[25] A. Langevin, M.-A. Carbonneau, M. Cheriet and G. Gagnon, "Energy Disaggregation using Variational Autoencoders," 2021.

[26] D. Garcia, "How do smart meters communicate?," emnify, 22 9 2021. [Online]. Available: https://www.emnify.com/blog/how-smart-meters-communicate#authorbio.

[27] K. Oliynyk, "6 Advantages of Smart Meters for Utilities," Webbylab, 23 March 2023. [Online]. Available: https://webbylab.com/blog/6-benefits-of-smart-meters-for-utilities/.

[28] "Everything You Need To Know About Smart Electric Meters," Genus energizining lives. [Online].

[29] "The Benefits of Smart Meters," California Public Utilities Commission, [Online]. Available: https://www.cpuc.ca.gov/industries-and-topics/electrical-energy/infrastructure/the-benefits-of-smart-meters.

[30] "What are smart meters and how can they save you money?," Whats Watt, [Online]. Available: https://whatswatt.com.au/what-are-smart-meters-how-can-they-save-money/.

[31] "HOW TO USE A SMART METER," smart meter, [Online]. Available: https://www.smartenergygb.org/smart-meter-benefits/benefits-for-you/how-to-use-a-smart-meter-to-save-money#:~:text=How%20to%20use%20a%20smart%20meter%20to%20save%20money,allowing%20you%20to%20make%20savings.

[32] "DO SMART METERS HELP THE ENVIRONMENT?," Smart Energy Consumer Collaborative, [Online]. Available: https://whatissmartenergy.org/faqs/do-smart-meters-help-the-environment.

[33] "Advanced Metering Infrastructure (AMI) and Customer Systems," Office of Electricity Delivery and Energy Reliability, 2016.

[34] M. I. Azad, R. Rajabi and A. Estebsari, "Non-Intrusive Load Monitoring (NILM) using Deep Neural Networks : AReview," *Cornell University,* 2023.

[35] A. Hemmatifar and M. Mogadali, "Household Energy Disaggregation basedon Difference Hidden Markov Model," 2016.

[36] Y. Zhang, G. Yang and S. Ma, "Non-intrusive Load Monitoring based on Convolutional Neural Network with Differential Input," *Procedia CIRP,* vol. 83, no. 2212-8271, pp. 670-674, 2019.

[37] A. Sankara, "Energy disaggregation in NIALM using hidden Markov models," *Library and Learning Resources,* 2015.

[38] J. Kim, T.-T.-H. Le and T.-T.-H. Le, "Nonintrusive Load Monitoring Based on Advanced Deep Learning and Novel Signature," *Computational Intelligence and Neuroscience,* pp. 1-22, 2017.

[39] Y. Zhang, W. Qian, Y. Ye, Y. Li, Y. Tang, Y. Long and M. Duan, "novel non-intrusive load monitoring method based on ResNet-seq2seq networks for energy disaggregation of distributed energy resources integrated with residential houses," *Applied Energy,* vol. 349, no. 0306-2619, 2023.

[40] N. Fumo and M.A. Rafe Biswas, "Regression analysis for prediction of residential energy consumption," *Renewable and Sustainable Energy Reviews,* vol. 47, pp. 332-343, 2015.

[41] Y. Chen, Peng Xu, Yiyi Chu, Weilin Li, Yuntao Wu, Lizhou Ni, Yi Bao and Kun Wang, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings," *Applied Energy,,* vol. 195, no. 0306-2619, pp. 659-670, 2017.

[42] AndreyBogomolov, BrunoLepri, RobertoLarcher, FabrizioAntonelli, FabioPianesi and AlexPentland, "Energy consumption prediction using people dynamics derived from cellular network data," *EPJ Data Science a SpringerOpen Journal,* 2016.

[43] C. A. Ronao and Sung-Bae Cho, "Anomalous query access detection in RBAC-administered databases with random forest and PCA," *Information Sciences,* vol. 369, no. 0020-0255, pp. 238-250, 2016.

[44] "A Simple Introduction to Sequence to Sequence Models," *Analytics Vidhya,* 2020.

[45] "Spatiotemporal," *HEAVY.AI.*

[46] P. Grieve, "Deep learning vs. machine learning: what's the difference?," Zendesk, 22 September 2023. [Online]. Available: https://www.zendesk.co.uk/blog/machine-learning-and-deep-learning/.

[47] A. Pai, "Analyzing Types of Neural Networks in Deep Learning," Analytics Vidhya, 13 Jul 2023 . [Online]. Available: https://www.analyticsvidhya.com/blog/2020/02/cnn-vs-rnn-vs-mlp-analyzing-3-types-of-neural-networks-in-deep-learning/#:~:text=The%20three%20main%20types%20of,unsupervised%20learning%2C%20and%20reinforcement%20learning..

[48] "What are convolutional neural networks?," IBM, [Online]. Available: https://www.ibm.com/topics/convolutional-neural-networks.

[49] S. Saxena, "What is LSTM? Introduction to Long Short-Term Memory," analytics vidhya, 4 Jan 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/.

[50] Y. Verma, "Bidirectional LSTM (Long-Short Term Memory) with Python Codes," *analyticsindiamag,* 2024.

[51] "Nonintrusive Load Monitoring (NILM) Using a Deep Learning Model with a Transformer-Based Attention Mechanism and Temporal Pooling," *Electronics,* vol. 13, no. 2079-9292, 2024.

[52] "Accuracy vs. precision vs. recall in machine learning: what's the difference?," *Evidently AI*.

[53] "Matthews Correlation Coefficient(MCC) one of the best metric when 2 classes are imbalanced," *Medium,* 2023.

[54] "Top 10 Machine Learning Evaluation Metrics For Regression - Implemented In R," *Appsilon.*

[55] "F1 Score in Machine Learning," *geeksforgeeks,* 2023.

[56] "Top Performance Metrics in Machine Learning: A Comprehensive Guide," *v7*, 2023.

[57] P. Pushp, "Beyond Accuracy: Understanding and Measuring Efficiency in Machine Learning Models," LinkedIn, 21 September 2023. [Online]. Available: https://www.linkedin.com/pulse/7-beyond-accuracy-understanding-measuring-efficiency-machine-pushp/.

[58] K. Yasar, "Pytorch".

[59] "Difference between Apparent Power and Real Power," *tutorialspoint.*

[60] "Why reactive power and power factorbcorrection are so important," 2021.

[61] M. A. D. A. Arda, . O. S. and M. T. , "Recommended National Sustainable Urban and Energy Savings Actions for Palestine," Human Dynamics Consortium , June, 2015.

[62] A. Abdel-Majeed and S. Tenbohlen, Design and Building of a Cheap Smart Meter, 2013.

[63] N. Mogles, N. Walker, A. Ramallo-González, J. Lee, S. Natarajan, J. Padget, E. Gabe-Thomas, T. Lovett, G. Ren, S. Hyniewska, E. O'Neill, R. Hourizi and D. Coley, "How smart do smart meters need to be?," *Building and Environment,* vol. 125, no. 0360-1323, pp. 439-450, 2017.

[64] M. Abdalzaher, M. Fouda and M. Ibrahem, "Data Privacy Preservation and Security in Smart Metering Systems," *Energies,* vol. 15, no. 1996-1073, 2022.

[65] M. Weiss, A. Helfenstein, F. Mattern and T. Staake, "Leveraging smart meter data to recognize home appliances," *IEEE International Conference on Pervasive Computing and Communications,* 2012.

[66] O. Munoz, A. Ruelas, P. Rosales, A. Acuña, A. Suastegui and F. Lara, "Design and Development of an IoT Smart Meter with Load Control for Home Energy Management Systems," *Sensors,* vol. 22, 2022.

[67] B. Hemus, "What are Smart Meters and How Do They Work?," eminfy, 30 6 2023. [Online]. Available: https://www.emnify.com/iot-glossary/what-is-a-smart-meter.

[68] "Statistical Modeling," complexica, [Online]. Available: https://www.complexica.com/narrow-ai-glossary/statistical-modeling.

[69] V. Kanade , "What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022," spiceworks, 3 April 2023. [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/.

[70] "What is Regression Analysis and Why Should I Use It?," alchemer, 8 June 2021. [Online]. Available: https://www.alchemer.com/resources/blog/regression-analysis/.

[71] N. Fumo and R. Biswas, "Regression analysis for prediction of residential energy consumption," *Renewable and Sustainable Energy Reviews,* vol. 47, no. 1364-0321, pp. 332-343, 2015.

[72] A. R. Rout, "ML – Advantages and Disadvantages of Linear Regression," GeeksforGeeks, 30 March 2023. [Online]. Available: https://www.geeksforgeeks.org/ml-advantages-and-disadvantages-of-linear-regression/.

[73] C. Nalmpantis and D. Vrakas, "Machine learning approaches for non-intrusive load monitoring: from qualitative to quantitative comparation," *Artificial Intelligence Review,* vol. 52, 2019.

[74] K. L. Sung and N. C. Yang, "Non-Intrusive Load Classification and Recognition Using Soft-Voting Ensemble Learning Algorithm With Decision Tree, K-Nearest Neighbor Algorithm and Multilayer Perceptron," *IEEE Access,* 2023.

[75] "Decision Tree," GeeksForGeeks, 20 August 2023. [Online]. Available: https://www.geeksforgeeks.org/decision-tree/.

[76] R. Stephen, L. Stankovic and V. Stankovic, "Explainability-Informed Feature Selection and Performance Prediction for Nonintrusive Load Monitoring," *Sensors,* 2023.

[77] A. Saini, "Guide on Support Vector Machine (SVM) Algorithm," analytics vidhya, 23 Jan 2024 . [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/.

[78] A. Raj, "Everything About Support Vector Classification — Above and Beyond," Medium, 30 Mar 2022. [Online]. Available: https://towardsdatascience.com/everything-about-svm-classification-above-and-beyond-cc665bfd993e#:~:text=Disadvantages%20of%20SVM%20Classifier%3A&text=SVM%20algorithm%20is%20not%20suitable,samples%2C%20the%20SVM%20will%20underperform..

[79] "What is the k-nearest neighbors algorithm?," IBM, [Online]. Available: https://www.ibm.com/topics/knn#:~:text=the%20next%20step-,K%2DNearest%20Neighbors%20Algorithm,of%20an%20individual%20data%20point..

[80] M. M. R. Khan, A. B. Siddique and S. Sakib, "Non-Intrusive Electrical Appliances Monitoring and Classification using K-Nearest Neighbors," *International Conference on Innovation in Engineering and Technology (ICIET),* pp. 1-5, 2019.

[81] "Advantages and disadvantages of KNN," my educator, [Online]. Available: https://app.myeducator.com/reader/web/1421a/11/q07a0/.

[82] I. Dabbura, "K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks," Medium, 17 Sep 2018. [Online]. Available: https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a.

[83] "K means Clustering – Introduction," GeeksForGeeks, 21 Dec 2023. [Online]. Available: https://www.geeksforgeeks.org/k-means-clustering-introduction/.

[84] S. Lu, X. Feng, G. Lin, J. Wang and Q. Xu, "Non-Intrusive Load Monitoring and Controllability Evaluation of Electric Vehicle Charging Stations Based on K-Means Clustering Optimization Deep Learning," *World Electric Vehicle Journal,* vol. 13, 2022.

[85] "Introduction to Recurrent Neural Network," GeeksForGeeks, 4 Dec 2023. [Online]. Available: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.

[86] N. V. Linh and P. Arboleya, "Deep Learning Application to Non-Intrusive Load Monitoring," *2019 IEEE Milan PowerTech,* pp. 1-5, 2019.

[87] D. Kalita, "A Brief Overview of Recurrent Neural Networks (RNN)," analytics vidhya, 7 Feb 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/#:~:text=Disadvantages%20of%20RNNs%3A,than%20other%20neural%20network%20architectures..

[88] A. Tongta and K. Chooruang, "Long Short-Term Memory (LSTM) Neural Networks Applied to Energy Disaggregation," *International Electrical Engineering Congress (iEECON),* 2020.

[89] N. Barla, "Dimensionality Reduction for Machine Learning," neptune ai, 8 Aug 2023. [Online]. Available: https://neptune.ai/blog/dimensionality-reduction#:~:text=Dimensionality%20reduction%20is%20the%20task,variables%20are%20also%20called%20features..

[90] Z. Jaadi, "A Step-by-Step Explanation of Principal Component Analysis (PCA)," builtin, 29 March 2023. [Online]. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[91] Z. Wu, C. Wang, H. Zhang, W. Peng and W. Liu, "A time-efficient factorial hidden Semi-Markov model for non-intrusive load monitoring," *Electric Power Systems Research,* vol. 199, no. 0378-7796, 2021.

[92] G. Saranathan and D. Coleman, "What is the relationship between accuracy and efficiency in ANN research?," LinkedIn, [Online]. Available: https://www.linkedin.com/advice/1/what-relationship-between-accuracy-efficiency-a0kge.

[93] K. Amber, M.W. Aslam and S.K. Hussain, "Electricity consumption forecasting models for administration buildings of the UK higher education sector," *Energy and Buildings,* vol. 90, no. 0378-7788, pp. 127-136, 2015.

[94] Y. Zhang, Guanghua Yang and Shaodan Ma, "Non-intrusive Load Monitoring based on Convolutional Neural Network with Differential Input," vol. 83, no. 2212-8271, pp. 670-674, 2019.

[95] M. K. S. R. Imran Rahman∗, "Power disaggregation of combined HVAC loads using supervised," *Energy & Buildings,* vol. 172, pp. 57-66, 2018.

[96] "What is random forest?," *IBM.*

[97] C. Li, Zixiang Ding, Dongbin Zhao, Jianqiang Yi and Guiqing Zhang, "Building Energy Consumption Prediction: An Extreme Deep Learning Approach," *energies,* 2017.

[98] P. Everton Gomede, "Unveiling the Power of Stacked Autoencoders," *Medium,* 2023.

[99] T.-Y. Kim and Sung-Bae Cho, "Predicting residential energy consumption using CNN-LSTM neural networks," *Energy,* vol. 182, no. 0360-5442, pp. 72-81, 2019.

[100] J. Kelly and William Knottenbelt, "Neural NILM: Deep Neural Networks Applied to Energy Disaggregation," *Research Gate,* 2015.

[101] C. Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard and Charles Sutton, "Sequence-to-Point Learning With Neural Networks for Non-Intrusive Load Monitoring," *Proceedings of the AAAI Conference on Artificial Intelligenc,* 2018.