Time-Series Clustering for Improving Predictions on Smart Home Appliances

Master Thesis

Rameez Raja

School of Computer Sciences
Østfold University College
Halden
May 15, 2023

Abstract

Energy consumption forecasting in smart homes has become an important research area due to its potential for optimizing energy consumption and reducing costs. However, accurate forecasting can be challenging due to the time-varying nature of energy consumption patterns, the availability and quality of data from different household appliances, and the challenges associated with identifying the efficient and appropriate clusters for time-series data. In this thesis, we address these challenges by proposing an intelligent data processing approach that utilizes time-series clustering techniques to improve energy and load forecasting for smart home appliances.

In this thesis, we address the challenge of energy consumption forecasting in smart homes utilizing intelligent data processing. We focus on the correlation between the energy consumption of different household appliances, which is an important factor that can be utilized to develop intelligent data processing schemes. The energy consumption data of home appliances is logged as time-series, which can be utilized for several analyses. These time-series act as input to machine learning-based forecasting algorithms. Our research focuses on clustering the time-series of several appliances in a household, and then using the obtained clustering information to improve the performance of a forecasting algorithm where clustered time-series from a set of appliances are fed together as an input.

To perform clustering on time-series data of several household appliances, we used the k-shape algorithm, which extracts the shapes of different time series and provides cluster information by computing their pairwise cross-correlation. We proposed two time-series clustering approaches based on k-shape algorithm: "Static" and "Dynamic" time-series clustering. The Static time-series clustering approach does not consider time-of-the-day, while the Dynamic time-series clustering approach considers this parameter and performs clustering based on time-of-the-day.

Time-series from appliances of a particular cluster are fused together as an input to a forecasting scheme. We forecasted the 24-hr energy consumption of a particular appliance by considering its historical data and its cluster members' data. The experiments for clustering and forecasting are performed on a real dataset of 43 appliances in a household. Improved day-ahead forecasting is observed using the proposed intelligent data processing approach which would potentially result in improved flexibility when implemented in a real environment.

This thesis contributes to the literature by proposing an intelligent data processing approach for energy consumption forecasting in smart homes. Our approach utilizes time-series clustering techniques to improve the accuracy of energy consumption forecasting and load forecasting for smart home appliances. Our results demonstrate that clustering time-series data can be an effective way to identify patterns in energy consumption that can improve forecasting accuracy.

The organization of the thesis is as follows: Chapter 1 provides an introduction to the topic and outlines the key objectives. Chapter 2 includes the literature review, which focuses on energy forecasting and clustering methods. Chapter 3 focuses on time-series clustering techniques for the prediction of smart home appliances. Chapter 4 presents the practical implementation and performance evaluation of the proposed techniques. Finally, Chapter 5 concludes the thesis and provides suggestions for future research.

In conclusion, the proposed intelligent data processing approach has the potential to enhance the accuracy of energy consumption predictions for smart homes. Our study provides a foundation for further research into the application of time-series clustering techniques for energy consumption forecasting in smart homes.

Acknowledgments

I want to convey my sincere gratitude to **Professor Anna-Lena Kjøniksen**. Associate **Professor Thi Thuy Nga Dinh**, **Gunnar Andersson**, and **Susana Garcia Sanfelix** for their essential advice, inspiration, and knowledge while I completed this master's thesis. Their persistent support and astute criticism were crucial in determining the course and caliber of this research. I am very appreciative of their guidance and the chance to benefit from their in-depth industry experience.

I have a lot of gratitude to the personnel, faculty, and administration at **Høgskolen I Østfold** for granting me access to its facilities, resources, and research supplies. It has been impossible to carry out this investigation without their support and collaboration.

I want to express deep appreciation to my family and friends for their constant support, tolerance, and understanding during this trying path. Their unwavering support and confidence in my abilities have served as a constant source of inspiration.

Furthermore, I would like to thank my classmates and colleagues especially **Dr. Surender Redhu** for their insightful comments and lively debates that have improved my study.

Finally, I intend to express my profound gratitude to everyone who helped finish my Master's thesis, whether they were directly involved or not. I've grown academically and achieved a lot thanks to your advice, encouragement, and support.

Glossary of Acronyms

IOT	Internet of Things
HAN	Home Area Network
AI	Artificial Intelligence
HVAC	Heating, Ventilation, and Air Conditioning
ML	Machine Learning
RES	Renewable Energy Sources
STLF	Short-Term Load Forecasting
MSE	Mean Squared Error
HEMS	Home Energy Management System
MG	Micro Grid
AMI	Advanced Meeting Infrastructure
TOU	Time of Use
PV	Photovoltaic
NCAs	Non-Controllable Appliances
ANN	Artificial Neural Network
ACF	Autocorrelation Function
PACF	Partial Autocorrelation Function
AIC	Akaike's Information Criterion
ARIMAX	Autoregressive Integrated Moving Average with Explanatory Variable
ADF	Augmented Dickey-Fuller
RF	Random Forest
SVM	Supervised Machine Learning
OLS	Ordinary Least Square
LSSVM	Least Squares Support Vector Machine
MLR	Multiple Linear Regression
ECMWF	Ensemble Prediction System
NWP	Numerical Weather Predictions
MAED	Model for Analysis of Energy Demand
PSO	Particle Swarm Optimization
LMP	Locational Marginal Pricing
VQ	Vector Quantization
LBG	Linde-Buzo-Gray
LPC	Linear Predictive Coding
UWA	Unsupervised Without Averaging
PCA	Principal Component Analysis
EM	Expectation Maximization
SOM	Soft-Organizing feature Map
KSOM	Kohonen Self-Organizing Feature Maps
SSA	Singular Spectrum Analysis
SAX	Symbolic Aggregate Approximation
PAA	Piecewise Aggregate Approximation
HMM	Hidden Markov Models
ED	Euclidean Distance

DTW	Dynamic Time Wrapping

Contents

Abstract i

Acknowledgments iii

Glossary of Acronyms iv

List of Figures xi

List of Tables xv

1 Introduction 1
- 1.1 AI for Smart Homes . . . 2
- 1.2 Background and Motivation . . . 4
  - 1.2.1 Energy Forecasting . . . 5
    - Load Forecasting . . . 5
  - 1.2.2 Intelligent Data Processing and Energy Prediction . . . 6
- 1.3 Problem Statement and Research Objectives . . . 7
  - 1.3.1 Objectives . . . 7
  - 1.3.2 Potential Challenges . . . 8
  - 1.3.3 Deliverables . . . 9
- 1.4 Organization of Thesis . . . 9

2 Literature Review 11
- 2.1 Review of Energy Forecasting . . . 11
  - 2.1.1 Smart Grid and IoT . . . 11
  - 2.1.2 Forecasting Methods . . . 13
  - 2.1.3 Statistical Methods for Forecasting . . . 15
    - Autoregression (AR) . . . 16
    - Moving Average (MA) . . . 16
    - Autoregressive Moving Average (ARMA) . . . 17
    - Autoregressive Integrated Moving Average (ARIMA) . . . 17
    - Seasonal Autoregressive Integrated Moving Average (SARIMA) . . . 18
  - 2.1.4 Machine Learning (ML) Based Methods . . . 18
    - Linear Regression Method (LR) . . . 20
    - Support Vector Machines (SVM) . . . 21
    - Hierarchical Forecasting . . . 21
    - Ensemble forecasting and Forecast Combination . . . 21
    - Probabilistic Forecasting . . . 22

Artificial Neuron Network (ANN) . . . 22
Energy Forecasting Competitions . . . 23
Energy Forecasting Challenges . . . 23
2.2 Review of Clustering Methods . . . 24
2.2.1 Raw Based Clustering . . . 24
Agglomerative hierarchical . . . 24
Density-Based Spatial Clustering of Applications with Noise (DBSCAN) . . . 25
Partitional Clustering . . . 25
Hierarchical Clustering . . . 26
K-Mean Clustering . . . 27
K-Medoid-Clustering . . . 28
2.2.2 Feature- and Model-Based Methods . . . 28
Feature Based Methods . . . 29
Modified k-Means (MKM) . . . 29
Model Based Methods . . . 29
2.3 Data Analysis Tools . . . 30
2.3.1 Numpy . . . 30
2.3.2 Pandas . . . 31
2.3.3 Matplotlib . . . 31
2.3.4 Seaborn . . . 31
2.3.5 Scikit-learn . . . 31
2.3.6 Ts-Learn . . . 31
2.3.7 Statsmodels . . . 32
2.3.8 Data Science and ML Frameworks . . . 32
2.3.9 XGBoost . . . 32
2.4 Summary . . . 33

3 Time-Series Clustering for Smart Homes Appliances Prediction 35
3.1 Static Time-Series Clustering . . . 35
Time-Series Invariances . . . 35
Shift Invariance . . . 36
Time Series Distance Measure . . . 36
Time-Series Clustering Algorithms . . . 37
3.2 K-shape Clustering . . . 38
3.2.1 Time-Series Shape Similarity . . . 38
3.2.2 Cross-Correlation Measure . . . 39
3.2.3 Shape-Based Distance (SBD) . . . 39
3.2.4 Shape-Based Time-Series Clustering . . . 40
3.3 Dynamic Time-Series Clustering . . . 41
3.3.1 A Potential Solution for Dynamic Clustering . . . 42
3.3.2 Dynamic Time-series Clustering using K-Shape . . . 43
3.4 Summary . . . 44

4 Implementation and Performance Evaluation **45**
4.1 Dataset Details 45
4.1.1 Deployment Scenario 45
4.2 Exploratory Data Analysis(EDA) 46
4.2.1 Data Preprocessing 47
4.2.2 Re-sampling of Dataset 47
4.3 Performance Evaluation of Static Time-Series Clustering 50
4.3.1 Forecasting Analysis using Static Clustering Information 51
4.4 Performance Evaluation of Dynamic Time-Series Clustering 60
4.4.1 Day-Time Clustering Analysis 60
4.4.2 Forecasting Analysis using Day-Time Clustering 63
4.4.3 Night-Time Clustering Analysis 66
4.4.4 Forecasting Analysis using Night-time Clustering 68
4.5 Comparative Analysis of Time-Series Clustering 71

5 Conclusion and Future Scope **75**

6 Publications **77**

Bibliography **79**

List of Figures

1.1 An illustration of a technology-enabled smart home 1
1.2 Load Profile of one Week [24]. 6
1.3 An illustration of the overall framework of the proposed idea in the thesis . 8

2.1 Components of Smart Grid [7]. 12
2.2 Types of Load Forecasting in Power System and Application [24]. 14
2.3 Basic Estimate or Forecasting Model Architecture [53]. 15
2.4 Pseudocode for a typical machine learning implementation, encompassing training and testing phases, as well as a final evaluation stage [80]. 19
2.5 Division of ensemble models for solar and wind energy prediction. [91] . . . 22
2.6 k mean algorithms [137]. 27

3.1 Similarity using: (a) ED (top) and DTW (bottom), (b) Sakoe-Chiba band with the warping path computed under cDTW [111]. 37
3.2 Time-series spread across two clusters, where two series change cluster membership halfway through the time window, are shown in orange [168]. . 43

4.1 Hourly time-series data of appliances (A1-A14) is presented 48
4.2 Hourly time-series data of appliances (A15-A28) is presented 49
4.3 Hourly time-series data of appliances (A29-A43) is presented 50
4.4 Correlation Matrix of 43 Appliances 51
4.5 Correlation Matrix of 43 Appliances in the order of their Clusters 52
4.6 Static Time-series clustering of 43 appliances into 5 clusters. (a) Appliances group of Cluster 1. (b) Appliances group of Cluster 2. (c) Appliances group of Cluster 3. (d) Appliances group of Cluster 4. (e) Appliances group of cluster 5. using k-shape algorithm. The shape of different time series in a cluster shows strong similarity. 53
4.7 The Performance of (a) 10 days Forecasting of Appliance A37 without utilizing clustering. (b) shows a closer look. 54
4.8 Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A37. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 10 days (240 hours). 54
4.9 The Performance of (a) 20 days Forecasting of Appliance A24 without utilizing clustering. (b) shows a closer look. 55

4.10 Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A24. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours). 56
4.11 The Performance of 30 days Forecasting of Appliance A7 without utilizing clustering. 57
4.12 Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A7. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 30 days (720 hours). 57
4.13 The Performance of (a) 35 days Forecasting of Appliance A16 without utilizing clustering. (b) shows a closer look. 57
4.14 Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A16. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 35 days (840 hours). 58
4.15 The Performance of (a) 50 days Forecasting of Appliance A6 without utilizing clustering. (b) shows a closer look. 59
4.16 Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A6. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 50 days (1200 hours). 59
4.17 Correlation Matrix of 43 Appliances before Performing Dynamic (Day-Time) Clustering 61
4.18 Correlation Matrix of 43 Appliances in the order of their Clusters 61
4.19 Dynamic (Day-Time) Time-series clustering of 43 appliances into 5 clusters. Each cluster shows its respective cluster members with similarities in their patterns. 62
4.20 Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A37. (b) a closer look. The forecasting experiments evaluate the performance for the last 10 days (240 hours). 63
4.21 Performance evaluation of (a) proposed Dynamic (Day-Time) clustering for forecasting next- day consumption of Appliance A24. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours). 64
4.22 Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A7. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (720 hours). 64
4.23 Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A16. (b) a closer look.. The forecasting experiments evaluate the performance for last 35 days (840 hours). 65
4.24 Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A6. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (1200 hours). 65

4.25 Correlation Matrix of 43 Appliances before Dynamic (Night-Time) Clustering 66
4.26 Correlation Matrix of 43 Appliances in the order of their Clusters obtained using night-time clustering 66
4.27 Dynamic (Night-Time) Time-series clustering of 43 appliances into 5 clusters. Each cluster shows its respective cluster members with similarities in their patterns. 67
4.28 Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A37. (b) a closer look. The forecasting experiments evaluate the performance for last 10 days (240 hours). 68
4.29 Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A24. (b) a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours). 69
4.30 Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A7. (b) a closer look. The forecasting experiments evaluate the performance for last 30 days (720 hours). 70
4.31 Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A16. (b) a closer look. The forecasting experiments evaluate the performance for last 35 days (840 hours). 70
4.32 Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A6. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (1200 hours). 71
4.33 Comparison of $RMSE$, MSE and SSE (forecasting A37, for 10 days) for Different Clustering Methods 72
4.34 Comparison of $RMSE$, MSE and SSE (forecasting A24, for 20 days) for Different Clustering Methods 73
4.35 Comparison of $RMSE$, MSE and SSE (forecasting A7, for 30 days) for Different Clustering Methods 73
4.36 Comparison of $RMSE$, MSE and SSE (forecasting A16, for 35 days) for Different Clustering Methods 74
4.37 Comparison of $RMSE$, MSE and SSE (forecasting A6, for 50 days) for Different Clustering Methods 74

List of Tables

4.1 Different Circuit Names and their IDs in a House. 46
4.2 Forecasting Performance for Appliance A37 (For last 10 days) using Static Clustering 54
4.3 Forecasting Performance for Appliance A24 (For last 20 days) using Static Clustering 55
4.4 Forecasting Performance for Appliance A7 (For last 30 days) using Static Clustering 56
4.5 Forecasting Performance for Appliance A16 (For last 35 days) using Static Clustering 58
4.6 Forecasting Performance for Appliance A6 (For last 50 days) using Static Clustering 58
4.7 Forecasting Performance for Appliance A37 using Dynamic Clustering (Day-Time) 63
4.8 Forecasting Performance for Appliance A24 using Dynamic Clustering (Day-Time) 64
4.9 Forecasting Performance for Appliance A7 using Dynamic Clustering (Day-Time) 64
4.10 Forecasting Performance for Appliance A16 using Dynamic Clustering (Day-Time) 65
4.11 Forecasting Performance for Appliance A6 using Dynamic Clustering (Day-Time) 65
4.13 Forecasting Performance for Appliance A24 using Dynamic Clustering (Night-Time) 69
4.14 Forecasting Performance for Appliance A7 using Dynamic Clustering (Night-Time) 69
4.12 Forecasting Performance for Appliance A37 using Dynamic Clustering (Night-Time) 69
4.15 Forecasting Performance for Appliance A16 using Dynamic Clustering (Night-Time) 70
4.16 Forecasting Performance for Appliance A6 using Dynamic Clustering (Night-Time) 71
4.18 Comparative Analysis of A24 using Different Time-Series Clustering Methods 72
4.17 Comparative Analysis of A37 using Different Time-Series Clustering Methods 72
4.21 Comparative Analysis of A6 using Different Time-Series Clustering Methods 72
4.19 Comparative Analysis of A7 using Different Time-Series Clustering Methods 73
4.20 Comparative Analysis of A16 using Different Time-Series Clustering Methods 73

Chapter 1

Introduction

There is a growing demand for creative solutions to effectively use the produced energy as the globe struggles to address the issues of climate change and the need for sustainable living. Moreover, there is a rising demand for energy because of the rising population and urbanization. Management of energy usage has thus become essential to contemporary civilization. With technological advancements and computing resources, smart homes are a new trend in the electricity market where smart metering plays a central role in these developments. Generally, smart meters record, process and upload their data to the server [1]. Hence, smart meters keep track of the energy consumption of several household appliances which can also be utilized to develop optimal scheduling algorithms. Moreover, when optimal appliance scheduling is implemented at a major scale, this can result in low electricity costs to the consumer and can also result in improved flexibility in the energy market [2]. Energy efficiency and energy flexibility are important concepts in the context of smart homes, as they can help reduce energy consumption, lower energy bills, and decrease greenhouse gas emissions. Flexibility is important to maintaining or restoring a system's stability since the system can only be balanced by responding flexibly to continually changing variables, such as fluctuating electricity demand at smart homes. However, there are several challenges associated with these developments.

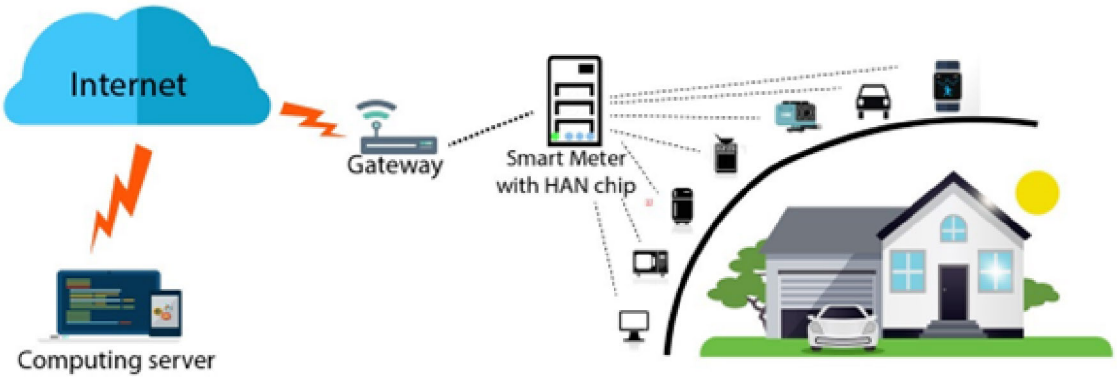


Figure 1.1: An illustration of a technology-enabled smart home

The Internet of Things (IoT), a network of networked computing devices with the capacity to transmit sensor data via internet protocols, has made several applications possible, including smart homes, e-health, smart transportation, etc., in the preceding decade [3]. IoT-enabled smart homes can give computer devices better monitoring and control [4]. By utilizing smart meters, smart plugs, home area networks (HAN), and internet-connected products, these IoT system capabilities can be used to create novel energy flexibility solutions for smart buildings and households. Figure 1.1 shows an example of an IoT-enabled smart home that incorporates these other technologies. The usage of various types of sensors integrated into smart devices in IoT-enabled smart homes and buildings leads to the collection of significant volumes of data that can be analyzed to create data-driven prediction and control algorithms [5] [6].

The data observed from several household appliances generally shows the uncertainty in energy consumption patterns [7]. This random behaviour of users in a household makes it very challenging to forecast short-term and long-term energy consumption. However, several statistical forecasting algorithms have been developed in past. Moreover, with the advancements in the domain of data acquisition and artificial intelligence, several machine learning algorithms have also been developed. The performance of machine learning algorithms is difficult to generalize on different types of dataset and requires an exhaustive feature extraction process.

Data-driven approaches, which offer big data analytics, real-time monitoring, forecasting, and machine learning techniques, have a huge potential to further improve energy flexibility in smart homes [8]. Also, integrating artificial intelligence (AI) with data gleaned from smart homes can be extremely important for the development of energy management systems in the future. Consequently, with precise load forecasting, ideal energy scheduling, prompt predictive maintenance, enhanced demand response, and customized energy management, AI and IoT can aid in boosting energy flexibility [9].

Furthermore, With the development of numerous solutions thanks to technological advancements like wireless sensors, HAN devices, and the Internet of Things (IoT). IoT networks have successfully monitored and managed tasks in a variety of applications, including smart homes and buildings, e-health, etc. In smart homes, various systems and appliances, such as lighting, heating, ventilation, and air conditioning (HVAC), are automated using IoT (Internet of Things) technology [10]. Thus, this technology is used to track the wasteful use of electricity in buildings and aids in the remote management of smart appliances. Smart appliances can be set to automatically turn off when not in use, and smart lighting can turn off when a room is empty. Users of smart homes benefit from better energy consumption and reduced utility costs.

1.1 AI for Smart Homes

By analysing patterns in energy consumption and providing insights to help reduce energy waste, artificial intelligence (AI), the technology that enables machines to learn and think like humans, can be integrated into smart home energy management solutions to make them even more effective. To increase energy efficiency and decrease energy consumption, artificial intelligence (AI) has gained popularity. Smart homes IoT-enabled equipment that includes sensors that collect data on energy consumption and use that data to optimize

energy use are used to achieve this. AI can examine this data to find patterns in energy use and offer suggestions for maximizing energy utilization while taking user comfort into account [11]. When no one is home or it is not necessary to maintain a certain temperature, for example, a smart thermostat can modify the temperature in a home based on occupancy and the outside temperature. To lessen reliance on fossil fuels, smart houses can also be connected to alternative energy sources like solar or wind power. By anticipating when the greatest energy will be produced and then using that energy when it is most needed, artificial intelligence can maximize the usage of renewable energy.

AI can be utilized to reduce energy use in commercial structures like office buildings and shopping malls in addition to smart homes. AI is also essential for the creation of sustainable smart cities when used on a broad scale. These cities employ cutting-edge infrastructure and technologies to raise the standard of living of their citizens while lessening their negative effects on the environment. By collecting data from sensors and other devices and using predictive models to forecast energy demand and optimize energy distribution, AI can assist in managing the energy consumption of large communities. AI can be used, for instance, to assess patterns of energy usage in various city areas and modify the distribution of energy as necessary. This can minimize prices, lessen energy waste, and balance the strain on the energy grid. Copenhagen, which started using AI to improve its heating system, is one example of a smart city application [12]. By analyzing weather patterns and optimizing building heating, the device significantly lowers energy consumption.

At the level of smart homes and cities, the future of AI-enabled energy management solutions is quite promising. The ability of AI to learn and adapt over time is a fundamental benefit in energy management. Machine learning algorithms improve in accuracy as more data is gathered, allowing for more efficient energy management. This is crucial in the context of smart cities because of how much data may be created there. AI can assist in managing and making sense of this data, revealing insights that might guide the development of infrastructure and energy regulations.

The advantages of adopting AI in energy management also present certain difficulties. The challenge of protecting data security and privacy is one of the major issues. Large volumes of data must be gathered and analyzed to employ AI, and some of this data may include sensitive information about users of smart homes. Also, users are less eager to supply service providers with information about their household's energy usage. The combination of an edge computing framework with AI technology, however, can lessen this problem. The central server is not required to receive user data under the edge computing concept. It performs intelligent judgments and processing data at end users. As it is extended to the level of a smart city, a further problem is a requirement for compatibility across various systems and devices. To maintain interoperability and compatibility, it is necessary to create common standards and protocols.

In summary, applying AI to energy management has the potential to revolutionize how we use and handle energy. Researchers and industry professionals are now looking to apply AI to bigger regions such as smart cities considering the continued success of smart homes. In terms of managing sustainable energy, the application of AI in smart cities has the potential to revolutionize the industry. While there are obstacles to be overcome, AI can be a vital component of creating a sustainable future for future generations.

1.2 Background and Motivation

Between 2010 and 2050, the demand for electricity is expected to double. Since providing the electrical power that will be required in the future is challenging, extensive study is being done in this area. Technical developments related to the Internet of Things (IoT) have opened some exciting possibilities for reducing this problem. Setting specific timing of the appliances leads to increased flexibility at the home level[13]. Due to energy consumption uncertainties, this is difficult at the same time and necessitates highly accurate forecasting of appliance-level energy use. Energy prices were already growing as a result of the post-lockdown increase in energy demand, but now that Russia has attacked Ukraine, the energy markets are much more uncertain and unstable, which is having an impact on all EU people. The REPower EU [14] plan can be used to address all of the ways in which the EU has responded to these concerns through various programs and actions. The REPower EU plan, which adds 225 billion euros to the Fit for 55 plan [15] of the European Green Deal, is designed to hasten the transition to a sustainable, equitable, and advantageous energy system for all Europeans. The following are a few pertinent actions encouraged by the plan.

- Improved permitting of RES projects (to accelerate RES deployment)
- Increase the European renewable target for 2030 from 40
- EU-coordination demand reduction plans in case of gas supply disruption
- Investments in an integrated and adapted gas and electricity infrastructure network

In accordance with the aforementioned goals, the EU has made the decision to speed up its energy transition plans in response to the current situation. To do this, it is encouraging greater adoption of RESs along with investments in new infrastructure, but it is also acknowledging the power of demand response and flexibility for a more resilient system.

Furthermore, the development and assessment of machine learning algorithms that allow the analysis of demand attribution and consumption behaviour depend heavily on real-world domestic power demand facts. The main technology for intelligent smart-grid management systems that seek a balance of power supply and demand is thought to be the disaggregation of domestic household electricity use. The development of methods for information retrieval, behaviour analysis, and forecasting is necessary to enable both relevant retrospective insights and prospective recommendations on consumer energy consumption. The flexibility of energy use in smart homes can be greatly increased by scheduling appliances. Consumers may lower their carbon footprint, save money on their energy bills, and contribute to a more sustainable future by utilizing these methods. Consumers used fossil fuel power plants exclusively to produce the electricity they needed in the past, which resulted in greenhouse gas emissions. But, in today's world, conventional electricity networks are unable to satisfy this expanding human need. Engineers and researchers continuously create innovative electricity generation theories for renewable and sustainable energy sources [16]. However, the complexity and dynamics of the power grid have greatly risen as a result of the adoption of renewable energy sources (RES). Researchers have given serious thought to ways to reduce and manage energy use to maximize financial savings and long-term environmental sustainability [17].

1.2.1 Energy Forecasting

The practice of predicting future energy demand or supply is known as energy forecasting. It is dependent on several variables, including population expansion, economic development, weather patterns, and technical improvements. Increasing the accuracy of renewable energy forecasting is essential for power system planning, management, and operations as renewable energy becomes more prevalent in the worldwide electric energy grid. Moreover, Forecasting's primary premise is that the future will in some way resemble the past in terms of patterns or distribution. Accurate forecasting depends on seeing patterns or hidden information in historical data. The growth of energy forecasting is undoubtedly aided by the development of artificial intelligence (AI) and machine learning (ML) approaches. [18] ,[19]. The forecast horizon, or the amount of time into the future for which projections need to be created, is a key idea in energy forecasting. Moreover, using this technology, along with consumer-managed demand response or even automated and remote-controlled appliances that respond to prices, gives the opportunity to improve energy efficiency and minimize peak demand. Energy markets are tasked with performing price and system supply and demand balancing [20]. The effectiveness of a home energy management system is significantly influenced by energy forecasts for appliances in homes. This system can choose the most effective energy allocation strategy and find a healthy balance between energy production and consumption[21].

Load Forecasting

Future load prediction is known as load forecasting, and it is crucial for the energy management system and for more effective planning of the power system. Due to its effect on the economy and the dependable operation of power systems, a significant number of studies on accurate short-term load forecasting (STLF) have been published in recent years. It guarantees the power system's dependable performance, which results in a consumer-supplied power supply that is never interrupted [22]. Accurate load forecasting makes it simple to carry out power system activities like scheduling, maintenance, tariff rate adjustment, and contract review [23].

Consumer load demand varies cyclically throughout the duration of a 24-hour day. It is caused by customers regular, everyday activities, which change depending on the time of day, such as during work hours, school hours, and the night. As a result, the load demand for a power utility changes throughout the day. The load demand for a week from Monday to Sunday, which varies with each hour of the week, is shown in Figure 1.2. It is evident that the pattern of load demand, with its fluctuating peak load demand, repeats throughout the week. The load demand generally declines throughout the night, reaching its lowest level in the morning. But, as the day wears on, the load demand starts to rise as individuals begin to engage in activities. When fewer people are out and about after midnight, the load demand begins to decline once more. Due to various social events that individuals participate in on different days of the week, the load demand also fluctuates on certain days. The load data should be carefully examined, and dynamics should be understood, to create a decent forecast model. To obtain accurate forecasts, the required operations must be carried out on load data based on its behaviour and fluctuation. Furthermore, the load consumption is often higher on working days than on rest days since factories, offices, and other places of employment will restart their output. As a result, there are differences

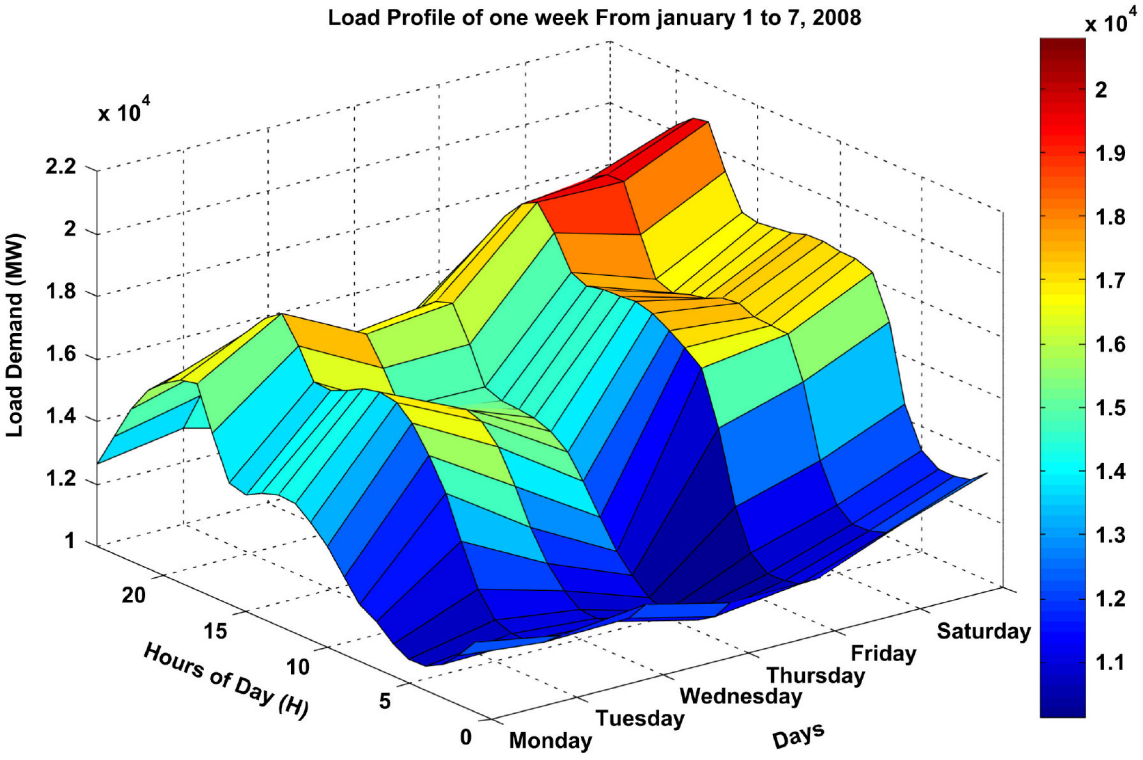


Figure 1.2: Load Profile of one Week [24].

in the load demand patterns of working days and weekends. The load pattern may also be impacted by the regional traditions and customs[24]. Additionally, consumer segmentation enables the utility to comprehend individual and groups of broadband networks delivering specialized services. As not all spectrum allocations are identical, every customer requires a unique set of individualized energy-saving tips at various times of the day. By doing so, this would enhance the efficacy of energy efficiency programs and also positively impact the efficacy of demand response programs that attempt to shift energy usage to avoid peaks or shorten the duration of peaks for broadband networks with high spectrum allotment.

1.2.2 Intelligent Data Processing and Energy Prediction

Intelligent data processing and energy forecasting are crucial components of smart homes that can assist optimize energy use and cut costs. Intelligent data processing is the process of analyzing data obtained from various sensors and gadgets used in the home using cutting-edge algorithms and machine learning approaches. Data on energy use, temperature, humidity, and occupancy are some examples of this data. Smart home systems can generate forecasts about future energy demand and usage by analyzing this data to find patterns. Smart home systems can improve energy use and cut expenses by automatically altering settings, such temperature and lighting, to match projected energy demand. This is done by precisely estimating energy need.

Energy consumption can be optimized, prices can be cut, and a more sustainable and effective home environment may be produced by smart houses by fusing intelligent data processing and energy forecasting.

1.3 Problem Statement and Research Objectives

Data observed by smart meters from a single or a couple of appliances does not provide a transparent vision of energy consumption patterns in homes. For this, we need to combine the data observed from several different appliances. Thus, intelligent data processing can play a significant role to monitor and control household devices in an optimal manner. Furthermore, the data observed from different appliances can be fused in an intelligent manner such that it can improve flexibility in smart homes. Therefore, intelligent data processing becomes challenging and crucial for smart homes and their flexibility[25]. Moreover, to improve the performance of machine learning methods, some data processing strategies have also been explored. Recent developments in data processing and machine learning for industrial prognosis by focusing on research trends, opportunities and unexplored challenges are presented in[26]. To promote environmental sustainability, the concepts, tools and methods of integrating multi-sensor data combining and machine learning methods are described in[27].

The appliances which possess similarities in their consumption patterns and usage behaviour can be utilized as an addition input to forecasting algorithms to improve their performance. However, identifying the similarity in energy consumption patterns of several home appliances for data processing is very critical and challenging due to time varying nature of consumption patterns.

Based on the above-discussed problem statement, we are identified following objectives of this thesis and the associated challenges.

1.3.1 Objectives

- To develop a more accurate energy consumption forecasting model for smart homes using machine learning techniques: One of the primary objectives of the thesis could be to improve the accuracy of energy consumption forecasting in smart homes. This objective could be achieved by developing a more robust and intelligent data processing algorithm that can help in improving the performance of machine learning-based forecasting models. Intelligent data processing would mainly consider the correlation between different household appliances' energy consumption. The model could be trained on historical energy consumption data and evaluated on unseen data to determine its accuracy.

- Intelligent data processing on appliances' data will be the time-series clustering which would help in improving energy consumption forecasting accuracy in smart homes. The thesis could also focus on exploring the effectiveness of time-series clustering in improving energy consumption forecasting accuracy. This could involve exploring and selecting an appropriate clustering method. The effectiveness of clustering in improving forecasting accuracy could be evaluated by comparing the performance of a clustering-based forecasting model to a baseline model that does not use clustering.

- To explore the potential of incorporating other factors mainly the "time of the day" for energy consumption forecasting: Another possible objective of the thesis could be to investigate the potential of different ways of performing Intelligent data processing (time-series clustering) based on time of the day. This would involve performing time-dependent and time-independent clustering of energy consumption data for improving the forecasting accuracy of appliances. The effectiveness of different ways of time-clustering methods would be evaluated by comparing the performance of the model.

An illustration of the overall framework of the proposed idea in the thesis is presented in Figure 1.3.

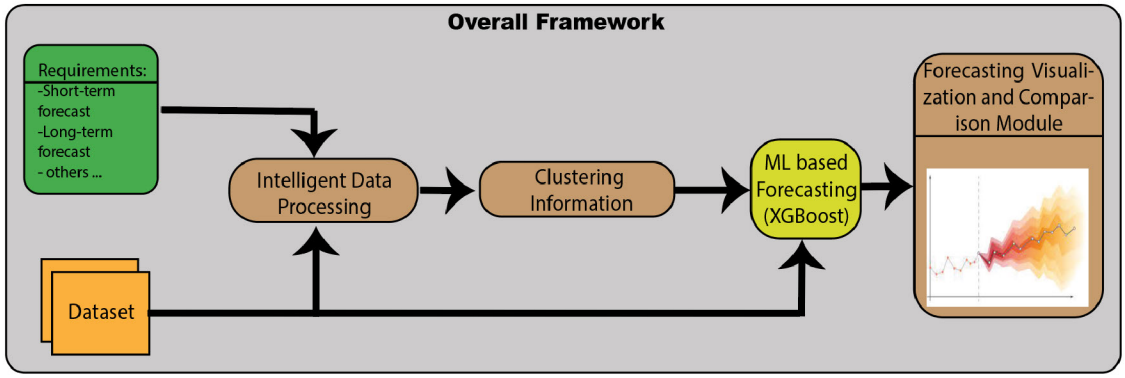


Figure 1.3: An illustration of the overall framework of the proposed idea in the thesis

1.3.2 Potential Challenges

- The time-varying nature of energy consumption patterns in smart homes can make accurate forecasting challenging: Energy consumption patterns in smart homes can vary significantly depending on various factors, including time of day, season, and occupancy. This variability can make accurate forecasting challenging, especially if the forecasting model does not consider the correlation between different household appliances.

- The availability and quality of data from different household appliances may vary, which could affect the accuracy of the forecasting model: The accuracy of the forecasting model could be affected by the availability and quality of data from different household appliances. Some appliances may not be equipped with smart meters, while others may have faulty meters that provide inaccurate data. In such cases, the accuracy of the forecasting model could be compromised.

- Identifying the optimal number of clusters and appropriate clustering method for time-series data can be challenging: Time-series clustering involves identifying patterns in the time-series data and grouping similar time-series together. However, identifying the optimal number of clusters and appropriate clustering methods can be challenging, especially if the data is noisy or contains outliers.

1.3.3 Deliverables

In this thesis, we are addressing the challenge of energy consumption forecasting in smart homes utilising intelligent data processing. The correlation between the energy consumption of different household appliances is an important factor which can be utilized to develop intelligent data processing schemes. Energy consumption data of home appliances is logged as time-series which can be utilized for several analyses. These time-series act as input to machine learning-based forecasting algorithms. This thesis focuses on clustering the time-series of several appliances in a household. More details about time-series clustering are discussed in the next chapters. Next, the obtained clustering information is utilized to improve the performance of a forecasting algorithm where clustered time-series from a set of appliances are fed together as an input. The main contributions of this thesis are summarized as follows.

a. To perform clustering on time-series data of several household appliances, we used the k-shape algorithm, which extracts the shapes of different time series and provides cluster information by computing their pairwise cross-correlation. The motivation behind using k-shape algorithm over other clustering algorithms is discussed in Chapter 2.

b. Time-series from appliances of a particular cluster are fused together as an input to a forecasting scheme. We forecasted the 24-hr energy consumption of a particular appliance by considering its historical data and its cluster members' data.

c. We proposed two time-series clustering approaches based on k-shape algorithm. These two different time-series clustering approaches named "Static" and "Dynamic" time-series clustering. Static time-series clustering does not depend on time-of-the-day while dynamic time-series clustering considers this parameter and performs clustering based on time of the day.

d. The experiments for clustering and forecasting are performed on a real dataset of 43 appliances in a household. An improved day-ahead forecasting is observed using the proposed intelligent data processing approach which can further result in improved flexibility when implemented in real environment.

1.4 Organization of Thesis

Chapter 1: Introduction

Chapter 1 provides an introduction to the topic and outlines the key objectives. It also introduces the concepts of smart homes, energy forecasting, and load forecasting. Also provide a brief background and motivation for the study, highlighting the importance of accurate forecasting for optimizing energy consumption and reducing costs. The chapter concludes by outlining the structure of the thesis, which focuses on using time-series clustering techniques to improve energy and load forecasting for smart home appliances.

Chapter 2: Literature Review

Chapter 2 includes the literature review. The chapter is divided into two main sections: a review of energy forecasting and clustering methods. The section on energy forecasting highlights the most promising techniques for smart home applications, while a review of

clustering methods focuses on statistical methods and machine learning-based methods. The section discusses commonly used statistical methods for clustering. It also covers popular machine learning-based methods. The strengths and weaknesses of each method are discussed, with a particular focus on their suitability for time-series data.

Chapter 3: Time-Series Clustering for Smart Homes Appliances Prediction

Chapter 3 focuses on time-series clustering techniques for the prediction of smart home appliances. The first technique is static time series clustering which does not consider time-of-the-day as a key parameter for implementing the k-shape algorithm. This method can effectively identify clusters of appliances' energy usage patterns in smart homes that remain constant over time. In contrast, the second technique, dynamic time series clustering, is a dynamic approach to clustering time series data that considers time-of-the-day as a key parameter for performing clusters. This technique is ideal for identifying energy usage patterns in smart homes that change over time. Both techniques have the potential to enhance the accuracy of energy usage predictions for smart homes.

Chapter 4: Implementation and Performance Evaluation

Chapter 4 focuses on the practical implementation and performance evaluation of the two time-series clustering techniques proposed in Chapter 3, namely Static Time Series Clustering and Dynamic Time Series Clustering. In Section 4.1, a detailed description of the data set used for the experiments is provided, including the features of the data set and the pre-processing steps applied to prepare the data for clustering. Section 4.2 presents the performance evaluation of the Static Time Series Clustering technique, including a discussion of the results and the accuracy of the energy consumption forecasting. Similarly, Section 4.3 describes the performance evaluation of the Dynamic Time Series Clustering technique, including a comparison of the forecasting results with those of the Static Time Series Clustering technique. Finally, Section 4.4 presents a discussion of the findings, highlighting the strengths and limitations of each technique.

Chapter 5: Conclusion and Future Scope

Chapter 5 presents the conclusion and future scope of the study. The chapter provides a summary of the main findings and acknowledges the challenges and limitations of the study, including potential sources of uncertainty and variability. It also outlines several areas for future research, such as the integration of other data sources and the use of more advanced clustering algorithms to improve prediction accuracy. Finally, the chapter concludes with a brief discussion of the potential implications of the research for smart homes and the broader field of data analytics.

Chapter 6: Publications

This chapter lists the accepted research paper and the ongoing research paper as well.

Chapter 2

Literature Review

For the power and energy sector, forecasting has been crucial. Thousands of papers on forecasting electricity demand, prices, and renewable generation (like wind and solar power) have been written by academics and industry professionals. Numerous influential review articles and original research papers have been among the thousands of energy forecasting and clustering papers that have been published over the past few decades. This section discusses the development of numerous approaches to reviewing various clustering and reviewing energy forecasting that is important to our thesis work.

2.1 Review of Energy Forecasting

"Internet of Things" (IoT) technology makes it possible for two-way communication between the Smart grid and end users, enabling demand response programs that encourage consumers to cut back on their energy use during times of high demand. In order to provide real-time monitoring and control of the energy system, IoT technology serves as a vital link between the physical parts of the smart grid and the digital infrastructure.

For energy forecasting, IoT devices and smart grid technology must be integrated for a number of reasons. First and foremost, precise forecasting and planning of energy resources is critical for preserving grid stability and preventing power outages. Real-time data provided by IoT sensors on numerous aspects, such as weather conditions and building occupancy, plays a crucial part in this process. Second, IoT technology can make it easier to accurately anticipate energy production, which is necessary for the integration of renewable energy sources into the energy system. Furthermore, demand response programs can be implemented with the help of smart grid technology, relieving the grid's load and preventing the need for expensive infrastructure upgrades by motivating users to use less energy during times of high demand.

2.1.1 Smart Grid and IoT

A new idea in smart grids called a micro-grid (MG) increases the efficiency and resilience of power networks by enabling intelligent control of consumer power usage and integrating dispersed generation resources like solar and wind energy [28]. Different components of the smart grid are presented in Figure 2.1. The stability and consistency of micro-grids are guaranteed by a home energy management system (HEMS) [29]. It is sometimes referred to as the method involving domestic users' utilization of home appliances. Because of

the high demand for power in the household sector, HEMS is essential to a smart grid control system. It functions by permitting variances in the demand curve based on each user profile. The variance happens because of a user's participation in the electric power market. Intelligent sensors that are housed at the database-running software are used during the entire operation. The sensors support the user's profile saving at multiple usage points. To enable the power supply, an advanced metering infrastructure (AMI), also known as a smart meter, acts as a connecting connector between the electrical grid and appliances. This load consumption that considers cost and energy is given priority by HEMS [30].

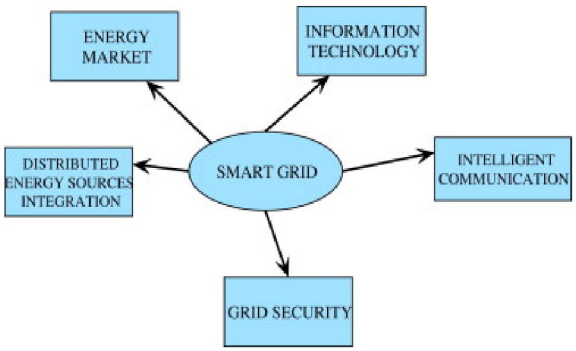


Figure 2.1: Components of Smart Grid [7].

A home habitat typically uses different amounts of energy depending on the time of day, the day of the week, and the season. Time of Use (ToU) pricing was consequently adopted in a variety of scenarios to entice energy consumers to schedule their loads during less expensive off-peak times [31] [32]. Nowadays, residential appliances with value-added smart behaviours can change the execution time to benefit financially from smart metering and varied rates [33]. An Internet of Things-based smart control system for home appliances was suggested in [34]. Despite load-shifting systems and fluctuating tariffs, this approach optimizes energy consumption based on user behaviour. By modifying appliance usage habits to reduce demand during peak hours, new energy management mechanisms save energy costs. Renewable energy production, appliance scheduling, and load balancing are crucial in this context. Experts have sought to replace peak electricity demand with small-scale renewable energy generators, such as photovoltaic (PV) panels, because of the smart grid concept. As a result, it lowers the direct electricity consumption of home appliances.

Furthermore, by flattening the load curve, load shifting further reduces the demand for electricity at peak times. In fact, load balancing is crucial for modern smart homes to optimize their energy use [35]. Simple load-shifting techniques, however, can have negative impacts by introducing additional peak times [36]. Hence, once the peak load is spread out over numerous off-peak hours, a load-shifting method is advantageous [37]–[39]. Using a home load balancing method, the authors of [40] suggested scheduling several users who use the same energy source. The primary goal of [41] which uses game theory to maximize egotistical users on the demand side, was to minimize financial expense. Corresponding [42] loads were moved to more cost-effective times while considering individuals' willingness to change their energy consumption. However, frequent delays in the operation of the

device reduce consumer satisfaction [43]. Hence, load-shifting techniques that offer great financial optimization and little disturbance will be beneficial.

By altering energy, consumption patterns can benefit both consumers and grid operators by reducing overall peak demand, recent smart house layouts have concentrated on increasing customer satisfaction, reducing energy usage, and cutting costs [44]. But creating a smart home architecture that meets all the objectives is a difficult undertaking. Due to frequent disruptions in appliance performance and prolonged completion times, load balancing and power scheduling can have a negative impact on customer comfort. Hence, scheduling procedures should ensure fewer disruptions and the shortest wait times.

Moreover, many household demand models have been presented over the past few decades to help in energy management studies. Long-term and short-term are the two broad categories into which they can be separated. Short-term models are essential for planning the production and procurement of power, dependability and security analysis, economic dispatch, and system maintenance schedules [45]. On the other hand, long-term models are frequently employed to estimate changes in demand under new technological advancements or the introduction of energy use rules, as well as supply capacity augmentation. Various models call for various modeling techniques. A disaggregated top-down strategy is frequently utilized for long-term ones, whereas bottom-up approaches are typically employed for short-term ones [46]. Individual end-users are not considered in the top-down approach, which sees each sector as an energy node. On the other hand, the bottom-up method identifies how much each end-use contributes to total energy consumption.

There has been numerous research addressing different energy parameters. The daily energy cost permitted home temperature ranges, energy use, peak energy use, and consumer comfort were among the factors examined [47], 2020conditional. It has also been investigated how consumption plans with set prices, time-of-use pricing, and real-time pricing affect consumers. Homod et al. introduced the Takagi-Sugeno fuzzy-based technique to meet energy demand in real time. For heating, ventilation, and air conditioning (HVAC) systems that utilize distributed energy resources, non-controllable appliances (NCAs), and battery storage systems, an energy-based operational model was created. Several groups of temperature average data for the entire year were created via clustering, which was employed by output variables. The strategy did not consider the other frequently utilized household loads because it was optimized for HVAC systems [48].

2.1.2 Forecasting Methods

Forecasting energy demand over the short, very short, medium, and long terms requires the application of clustering techniques in the power system. The accuracy of energy demand projections can be increased by using clustering to assist discover patterns and trends in the data. The data can be divided into various groups, each with its own distinct traits and trends, using clustering methods. This may result in more precise and dependable predictions of energy demand, which may then be utilized to improve energy production and distribution for increased efficiency and sustainability of the power system. Moreover, the deregulated economy, power requirement (or electricity load) forecasting is crucial for utility businesses. Applications range from power infrastructure expansion through load shedding, contract appraisal, and electricity production and purchase. For energy prediction, a growing number of numerical strategies have been put forth. Figure 2.5 depicts the various power requirement prediction intervals and their uses for short, medium,

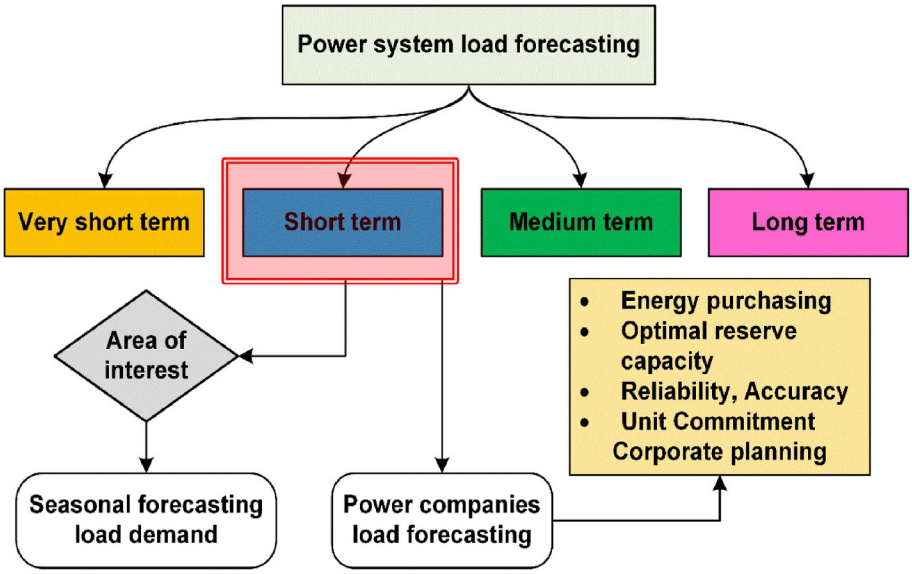


Figure 2.2: Types of Load Forecasting in Power System and Application [24].

and long-term perception of energy, control, planning, and management systems that are dependable and efficient Optimal reserve capacity, accuracy, and reliable, corporate planning for unit commitments, and best reserve capacity handling are the main goals of power companies for load forecasting [24].

About a century has passed since long-term load estimates were first utilized for planning [49]. Throughout the latter half of the 20th century, transmission and distribution planning made extensive use of spatial load forecasting, which provides information on where, when, and how much the demand for energy would increase. Several geographic load forecasting techniques were discussed in the tutorial review by Willis and North Cote-Green at the time, and some of them are still employed in the market today [50]. Short-term load forecasting gradually caught the attention of scholars and practitioners as power firms began to strive for operational excellence.

The forecasting of load, generation, prices, and other factors is a necessity for the energy sector. All sectors of the energy industry are using these projections to plan and run their corporate operations as well as their power systems. Although predicting kWh usage is one way to define energy forecasting, Researchers use a broader definition that refers to forecasting in the energy industry. They concentrate on issues pertaining to power systems, such as electricity demand and costs, as well as the production of wind and solar energy. Furthermore, electricity price forecasting has gained more and more attention from the business community and academia since the 1980s because of deregulation and the expansion of electricity markets. In the 1990s and 2000s, load forecasters experimented with a variety of forecasting methods, with artificial neural networks (ANN) becoming particularly well-liked. An excellent technique-focused analysis pointed out numerous important problems in theory and practice while taking a logical look at the hoopla surrounding ANN for load forecasting in the 1990s [51]. Probabilistic load forecasting has gained popularity in the last ten years.

2.1.3 Statistical Methods for Forecasting

Statistical techniques can be categorized into different groups based on the models they are used to examine and investigate the collection, analysis, explanation, presentation, and association of data [52]. The three main components of forecasting are input variables (past and present data), forecasting estimation methods (analysis of trends), and output variables (future predictions), as shown in Figure 2.3. Forecasting involves making predictions for the future based on the analysis of trends of past and present data. The researched methods can be roughly categorized into stand-alone and hybrid based on the variety of trend analysis methodologies applied. While hybrid methods combine multiple stand-alone techniques, standalone methods only use one technique for trend analysis. The majority of the time, hybridization is done to improve prediction accuracy and rationalize or provide dependable forecast output.

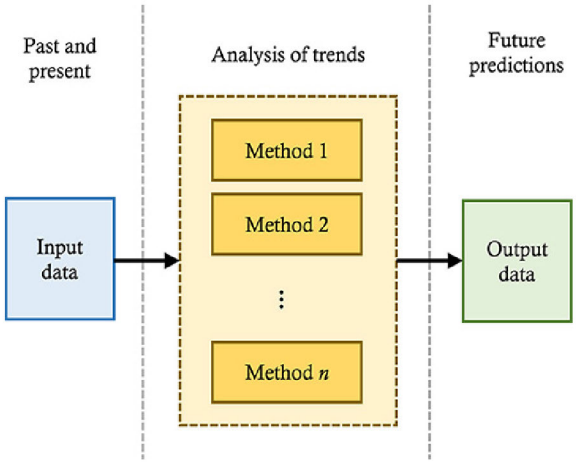


Figure 2.3: Basic Estimate or Forecasting Model Architecture [53].

Both top-down and bottom-up models have been developed to predict energy demand in the context of electric energy use. As an illustration, a few research [54]–[56] use historical aggregate energy data to regress energy consumption as a function of top-level macroeconomic variables including population, GDP, stock index, import, and export. From a representative sample of individual homes, the bottom-up methodologies extrapolate the expected energy consumption of regional levels [46]. The bottom-up methodology employs both statistical and engineering models. To regress the link between end-users and energy consumption, statistical models use a variety of statistical approaches, including regression [57], [58] ARIMA [59], [60] Kalman filtering [61], [62] and neural networks [63]–[65]. The assumption of historical data availability is a flaw in statistical techniques that affects both long-term and short-term models and frequently restricts the use of these models. Engineering models, on the other hand, use data regarding building attributes and end uses to calculate energy usage. Engineering models are the only practical way to properly build energy consumption estimates for a sector without previous energy consumption data. Although the data is harder to collect due to privacy concerns, they are more suited for assessing the effects of scheduling household appliances. The user's comfort hasn't been taken into sufficient consideration in these works because they exclusively concentrate on energy minimization. User comfort can be used to gauge

the possibility of residential DR. By fully utilizing it, homeowners can create a VESS that successfully balances fluctuating peaks and real-time demands.

In this section, an overview of various statistical methods of forecasting is presented.

Autoregression (AR)

According to its name, autoregression is the linear combination of previous values from a time series ("regression"), presuming a relationship between the present and earlier values ("auto"). Lag, also known as order p, refers to the number of prior values employed in the regression. A weighted average of the past values, lag, additional white noise and a constant c are used to estimate the future value:

$$\hat{y}_t = c + \sum_{i=1}^{p} \omega_i y_{t-i} + \epsilon_t \tag{2.1}$$

Here w are the weights of each lag. The Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plot can be used to calculate the number of lags / and order p. If, then we can define order p.

- The ACF is sinusoidal or exponentially decaying
- At lag p, the PACF experiences a sizable rise, but not after

Also, can test several orders P and then choose a model based on a criterion, such as Akaike's Information Criterion (AIC), if the ACF and PACF plots are not helpful. The AIC aids us in locating models that can deliver decent outcomes with the fewest parameters, even though it does not always find the optimum model. Therefore, we essentially test various ordering before selecting the model with the lowest AIC. However, keep in mind that a low AIC does not imply that the model produces accurate forecasts. Time series without a trend or seasonality should be utilized with autoregression, which limits the order of the autoregression. High order (number of lags) suggests the need for extra parameters, such as the addition of a Moving Average (MA).

It's important to remember that the amount of time we can predict a target depends on the sequence we chose. For instance, if we select an order P of 1, we are only able to forecast the subsequent time step. We could anticipate four-time steps into the future if we used an order p of 4, The model feeds back the anticipated values and utilizes them to predict the following time step if we wish to predict more than one time-step [66].

Moving Average (MA)

The historical forecast errors are combined linearly to create the moving average. The strategy thus depends on the relationship between the target value and earlier white noise error terms. It is important to distinguish this method from moving average smoothing:

$$\hat{y}_t = c + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{2.2}$$

In the present scenario, the weights of each lag are represented by , where represent for white noise (i.e., the difference between the forecasted and actual value). When using the Autocorrelation Function (ACF) plot, it is possible to determine the order q of the

moving average, which represents the width of the moving average window. We can determine the order if

- A significant spike in the ACF occurs at lag q, but none follows, and
- The PACF is sinusoidal or exponentially decaying.

The Moving Average (MA) method is usually appropriate for stationary time series [66].

Autoregressive Moving Average (ARMA)

The Moving Average of order q is combined with the Autoregression of order p to create the Autoregressive Moving Average. As a result, the method illustrates how a time series interacts with itself and random noise at earlier time steps:

$$\hat{y}_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{2.3}$$

Here, the moving average is represented by the second summation term whereas the autoregressive component is represented by the first summation term.

The ACF and PACF graphs may not be very helpful if both the p and q components are present, making it difficult to choose the appropriate ordering for them in our ARMA model. They might, however, suggest the sequence and direct us to a reasonable beginning point for our hyperparameter adjustment. Alternatively, we can test several combinations of p and q and then select the orders depending on a predetermined criterion, such as the AIC.

The ARMA method is similar to Autoregression (AR) and Moving Average (MA) in that it only performs well for stationary time series [66].

Autoregressive Integrated Moving Average (ARIMA)

The autoregressive integrated moving average (ARIMA) models are a versatile class of models used for predicting time series data. These models employ a differencing technique to transform a temporal series into a stationary series, assuming that the statistical characteristics of a random variable remain consistent over time. The ARIMA equation for a time series utilizes the delays of the dependent variable and the prediction error as predictors, providing a linear equation framework that serves as a general approach for time series forecasting.

One specific variant of ARIMA models is the ARIMA with eXternal input (ARIMAX) model, proposed by Newsham and Birt (2010) for forecasting energy demand in an office building [67]. In their study, they incorporated occupancy data as an external predictor to improve the accuracy of the model.

Chujai et al. (2013) introduced ARIMA and autoregressive moving average (ARMA) models for analyzing time series data on home electric usage [68]. They found that the ARIMA model was the most effective for determining the optimal forecasting period.

For short-term load forecasting, Mohamed et al. (2011) employed a double seasonal ARIMA model [69]. Their study focused on accurately predicting energy load within shorter time intervals.

As is common knowledge, ARIMA is mostly employed to forecast future values using data from historical time series. Its primary use is for short-term forecasting with at least 38–40 previous data points and a manageable number of outliers. It is advised to look for alternative approaches if you don't have at least 38 data points.

By adding a differencing order d to the ARMA model, we may directly incorporate the differencing rather than performing it in a separate phase, giving us an Autoregressive Integrated Moving Average model. When the order d is 1, the timeseries is only differed once; when the order d is 2, the timeseries is differed twice.

- Plotting our timeseries to compare how the differencing order affects the results.
- utilizing statistical tests like the Augmented Dickey-Fuller (ADF) test
- alternatively examine the ACF and PACF charts
- employ *autoSarima.*

The ARIMA is computationally more expensive than the approaches mentioned above, even though it typically yields superior results. Additionally, we must adjust more hyperparameters [70].

Seasonal Autoregressive Integrated Moving Average (SARIMA)

There are numerous ways that can be used for time series modeling. The seasonal autoregressive integrated moving average (SARIMA) model is one of the most well-liked and often utilized seasonal time series forecasting models. The SARIMA model is widely utilized because of its statistical characteristics and the well-known Box [71] that was used to build the model. Numerous fields have successfully embraced the SARIMA model [72]–[74]. Although the SARIMA model's tremendous success in academic research and industry applications over the past three decades, its presumption of a linear model form causes it to have a significant drawback. Simply said, the time series values are believed to have a linear correlation structure, hence the SARIMA model is unable to detect any nonlinear patterns. The use of linear models to solve difficult problems in the actual world is not always appropriate [75].

Furthermore, we can add a seasonal component to the ARIMA model if our time series contains one. The resulting Seasonal Autoregressive Integrated Moving Average executes an extra Autoregression, Integration, and Moving Average for the seasonal component while also back shifting one of its seasonal components. As a result, the

$$(p, d, q)$$

and $(P, D, Q)m$ components, where $P, D, andQ$ are the parameters of the seasonsal component m, can be used to indicate the SARIMA. Finding the proper parameters for the model takes much longer than for an ARIMA model, even though it provides for a better forecast [70].

2.1.4 Machine Learning (ML) Based Methods

The recent excitement in the field of AI/ML is largely a result of the development of computing technologies. Energy forecasting has adopted several cutting-edge AI/ML

approaches, including deep learning [76], [77] reinforcement learning [78] and transfer learning [79]. The physical properties of the processes involved can be useful for modeling and variable selection in machine learning-based methods for load, wind, solar, and price forecasting, it should be noted. Exogenous data use does not just mean incorporating raw meteorological data into machine learning algorithms. Instead, one ought to delve further and look at the inherent qualities, prominent characteristics, and constraints of these facts.

The standard machine learning (ML) algorithm implementation sequence is depicted in Figure 2.4 The training and testing phases are the two primary stages of an ML algorithm. The ML model is first created during the training phase using a training data-set based on the selected ML classifier models. Artificial neural networks (ANN), support supervised learning machines (SVM), and random forests (RF), the three most well-known ML models, are used. The classifier model's overall performance is guaranteed by the performance validation of the training phase, which is also utilized to prevent the overfitting problem.

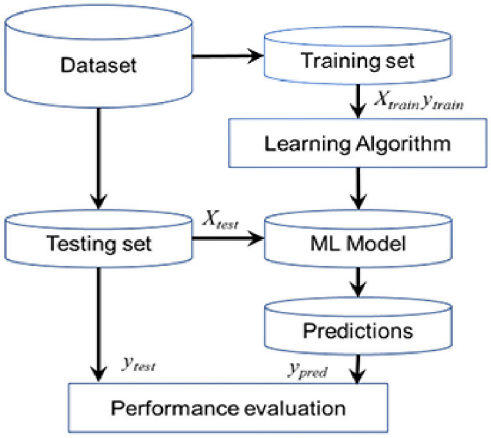


Figure 2.4: Pseudocode for a typical machine learning implementation, encompassing training and testing phases, as well as a final evaluation stage [80].

Compute classification accuracy and F-score Compute classification confusion matrix The trained classifier is then used in the testing stage to verify the trained model using the testing dataset as input. This testing dataset is made up of the additional partitioned data from the initial dataset and shares all the same properties as the training dataset. 70 percent and 30 percent, respectively, of the original dataset are divided for training and testing. Both phases are evaluated using performance measures. Any overfitting issues can be identified by comparing the results of both the training and testing phases. It happens when training performance is significantly better than testing performance.

According to the pseudocode of [81], Algorithm 2.1.4 displays the implemented program ML classifier. Cross-validation was carried out k times. 10-time cross-validation was applied in this investigation. Another ten times were spent repeating this cycle. Random data rearranging and the total average performance were used. This check assisted in preventing any overfitting problems. This was accomplished by presenting the difference between the training and testing results to the random forest's loss function. The root-mean-square error (RMSE) function in Equation (1) is used to evaluate the conventional technique, which is a regression-type problem. The proposed classifier uses

Algorithm 1 Machine Learning Implementation

Input: Dataset D, number of repeats n, number of folds k
Output: Classification accuracy, F-score, and confusion matrix
Preprocessing: Shuffle-split D into training, validation, and testing sets, and import necessary libraries
for $i = 1$ to n **do**
  Shuffle-split D into training, validation, and testing sets
  for $j = 1$ to k **do**
    Train an ML algorithm on the training set
    Evaluate the performance of the trained model on the validation set
  end for
  Test the trained model on the testing set
  Evaluate the performance of the tested model
end for
Compute the average classification accuracy and F-score over n repeats
Compute the confusion matrix over n repeats
Output: Classification accuracy, F-score, and confusion matrix [80].

the following metrics: F-score, classification accuracy, and confusion matrix. Equation (2)'s F-score accuracy metrics consider the importance of the ML model's precision and recall performance. While recall gauges sensitivity, precision gauges its positive predictive value. The accuracy of classification in (3) was also measured. This measure represents the proportion of correctly categorized levels over all levels taken.

$$\begin{aligned} \text{RMSE} &= \sqrt{\sum_{i=1}^{n} \frac{\left(w^T x(i) - y(i)\right)^2}{n}} \\ \text{F score} &= \frac{2 * \text{ Precision } * \text{ Recall}}{\text{Precision * Recall}} \\ \text{Accuracy} &= \frac{\text{No. of correctly classified energy levels}}{\text{Total number of classified energy levels}} \end{aligned} \tag{2.4}$$

The classification performance of the model may also be seen using a confusion matrix with a normalized range of 0 to 1. A platform that offers Python libraries and assistance is called scikit learn in [82]. These three classifiers—ANN, SVM, and RF—were implemented using ML models.

Linear Regression Method (LR)

Linear regression (LR) was employed to model the relationship between a scalar dependent variable and one or more independent variables. In this study, the multivariate linear regression (MLR) technique, a variant of LR, was utilized. MLR models have been commonly applied in the prediction of energy loads for buildings due to their simplicity. For this research, the linear time series regression model was employed with ordinary least squares (OLS), which requires defining the data's characteristics and the error term. Various important asymptotic and finite sample results were reported and compared with the statistical properties of time series regression. LR models have been widely used to forecast energy consumption in buildings due to their ease of use. Catalina et al. (2008)

developed regression models to predict monthly heating demand in single-family residential buildings in temperate climates, offering architects and design engineers a valuable tool to explore energy-efficient solutions during the early stages of their projects [83]. To assess the energy performance of buildings in their initial design phases, Hygh et al. (2012) proposed a novel modeling technique [84] based on multivariate regression. The main advantage of the LR method lies in its simplicity, as it requires no hyperparameter tuning. However, it is important to note that the LR approach is limited to solving linear problems and cannot handle nonlinear ones [85].

Support Vector Machines (SVM)

Many machine learning applications, including pattern recognition, object classification, and time series prediction, particularly the forecasting of energy usage, use support vector machines. SVM was used by Dong et al. (2005) to forecast the energy use of buildings in a tropical area [86]. To create and test models, they collected average monthly utility bills. Their forecasts exhibited percentage errors of less than 4 percent and coefficients of variance of less than 3 percent. For estimating short-term load, Bozic et al. (2010) suggested a Least Squares Support Vector Machine (LSSVM) [87]. To forecast the hourly and daily load, they used a week's worth of hourly data. For daily projections, the findings showed mean absolute percentage error rates ranging from 0.93 percent to 3.04 percent.

Hierarchical Forecasting

Due to temporal or geographical groupings, time series with aggregation constraints are frequently encountered in energy forecasting. As an illustration, the total load at a distribution feeder should be equal to the load at the corresponding transmission, with fewer losses, which are normally a tiny percentage. Hierarchical forecasting, which combines base estimates produced independently at several levels of a hierarchy, is crucial in these cases. Comparing hierarchical forecasting to traditional forecasting, there are two clear benefits. First off, a hierarchy's ultimate forecasts are cohesive. To put it another way, the total of lower-level projections is like or equal to the total of the corresponding higher-level forecast. Second, compared to base estimates, reconciled forecasts are frequently, if not always, more accurate. Hyndman's research team has largely contributed to recent advancements in hierarchical forecasting [88]. Numerous publications have addressed computation issues that prevent the widespread use of hierarchical forecasting [89],[90].

Ensemble forecasting and Forecast Combination

As shown in Figure 2.5, ensemble models for wind and solar prediction are employed in [91] two classes, each of which has two additional sub-classes. The competitive ensemble model (CEM) makes predictions by using a variety of factors and models. The final projections are calculated from the weighted average of the network's various estimates. The CEM divides the prediction process into several sections, using a variety of classifications at different stages to produce the final accurate and trustworthy forecasts.

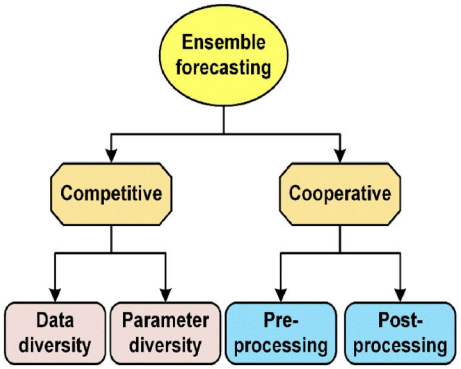


Figure 2.5: Division of ensemble models for solar and wind energy prediction. [91]

One of the most effective forecasting techniques is the combination of forecasts. In 1969[92] the benefits of forecast combination were formally articulated. Afterward, many empirical studies were released to demonstrate both the beneficial and detrimental consequences of integrating forecasts [93].

Probabilistic Forecasting

Artificial Neuron Network (ANN)

Artificial neural network (ANN) is a machine learning approach that models the human brain and consists of several artificial neurons. Their ability to learn by example makes them very flexible and powerful. Artificial neural networks (ANN) have been the focus of ML-based models for building energy systems, both in terms of energy demand prediction [94] and energy system design optimization [95]. The ANNs are a subset of AI that act as superb modeling tools for analysis, just like its rivals, to ascertain the non-linear network function assessment, pattern detection, data sorting, simulation, grouping, and optimization. Black-box optimization techniques are a type of setup used to eliminate non-linear tendencies. The hidden layer, input layer, biases and weights, output layer, summation node, and activation function are the main components of the design. It is split into two phases: learning and generalization (recalling) (training). Biases and weights are used in the network training to create the desired output by reducing the network function's error. The networks in the model networks were trained using repetitions, which are the final cycle of each dataset, and were used to improve the systems using machine learning. The three types of modelling learning techniques are unsupervised, evolutionary, and reinforcement supervised. For a specific application, these supervised models add the variance between the desired output and the actual network output outcomes. Modelling methods based on artificial neurons, or a network of connected units or nodes, are known as artificial neural networks. ANNs models have been widely utilized to forecast residential and commercial energy usage [96]. The efficiency of ANNs in forecasting was examined by Adya and Collopy in 1998 [97]. Using ANNs, Ekici and Aksoy (2009) suggested a model to forecast building energy needs and their correlation with orientation, insulation thickness, and transparency ratio [98]. According to the outcomes of their simulation, ANN offers satisfactory results with a variance of 3.43percent and an accuracy prediction rate of 94.8–98.5 percent. To anticipate net electricity usage in Turkey, Hamzacebi (2007) created

models [99]. The outcomes showed that the ANN technique outperforms the Model for Analysis of Energy Demand (MAED) technique in terms of performance.

Hybrid Models In hybrid models, optimization strategies are used with machine learning approaches. They are more effective than single models because they frequently combine the benefits and make up for the drawbacks of the many methodologies used, increasing the predicting accuracy. Hybrid models can be built with one or more phases that correspond to various objectives for solving problems.

Energy Forecasting Competitions

One of the earliest notable energy forecasting competitions took place in the early 1990s, specifically focused on day-ahead load forecasting and hosted by Puget Sound Power and Light Company. This competition involved ten participants who employed various models, including neural network models, state space models, and multiple regression models. Out of the 14 competing models, a multiple regression model emerged as the top performer [100]. Subsequently, one of the contestants further refined their algorithms, leading to the development of a commercial load forecasting system [101], [102].

In 2001, the EUNITE network organized another competition, challenging participants to predict daily load for a month. The winning submission prominently featured the use of support vector machine (SVM), marking its initial successful application in load forecasting [18]. Notably, the primary author of the publication detailing this competition is also the creator of various SVM libraries, including the well-known MATLAB library LIBSVM.

Hong and his collaborators organized a series of Global Energy Forecasting Competitions, also known as GEFCom2012, GEFCom2014, and GEFCom2017 [103], [104], [105] with the goal of fostering reproducible research and recognizing effective methods. The IEEE Power and Energy Society provided financial support for the competitions. More and more energy organizations have recently begun to hold forecasting competitions for a variety of reasons, including choosing software providers and hiring intern students. While some of these challenges were designed to mirror the production environment's forecasting process, others weren't. The results of these contests, including the information and strategies used to win, have not been widely reported in the academic literature. We should also be aware that some tournaments may not be able to identify winners if they are not set up thoroughly. Competitions for forecasting include several restrictions, as explained in [106].

Energy Forecasting Challenges

Net-load forecasting was a specific example given for the fusion of energy forecasting issues back when the forecast was made in 2015. There weren't many studies on net-load forecasting and behind-the-meter solar estimating at the time. Five years later, the literature contains several credible investigations on this subject [107], [108] Deep coupling exists between load and locational marginal pricing (LMP). A study on how to create probabilistic LMP projections while taking load uncertainties into account may be found in [109]. In the upcoming years, experts anticipate that the variety of energy forecasting difficulties will only increase.

2.2 Review of Clustering Methods

In the literature, numerous clustering schemes have been suggested for time series analysis. These schemes employ various distance measurements either directly on the raw time series data (raw-based methods) or by transforming the sequences into feature vectors to be used with traditional algorithms (feature- and model-based methods) [110], [111]. Feature- and model-based techniques often rely on domain-specific considerations, requiring adjustments to the features or models for different application domains. Among the raw-based methods, the top three clustering approaches are agglomerative hierarchical, spectral, and partitional clustering techniques.

2.2.1 Raw Based Clustering

This method of clustering includes techniques that use unprocessed data in the time or frequency domain. Although the length (or number of time points) of the two-time series under comparison is typically the same, this is not always the case.

Komelj and Batagelj [112] improved the relocation clustering method initially designed for static data to accommodate multivariate time-changing data. Their approach involved two key steps. Firstly, they devised a specialized model using the notion of compound interest to calculate time-dependent linear weights. Secondly, they introduced a general model employing a cross-sectional approach, which incorporated the time dimension to measure dissimilarity between trajectories, as necessitated by the procedure. However, it is important to note that the proposed cross-sectional method only considers time series of equal lengths and overlooks any correlations that may emerge among the variables over time. The clustering with the smallest generalized Ward criterion function, out of all those that could exist, is the one that forms the desired number of clusters. With the aim of forming a definite number of combat states, Liao et al [113] used a variety of clustering techniques, such as K-means, fuzzy c-means, and genetic clustering, to multivariate battle simulation time series data of unequal length. The simple linear interpolation approach was used to uniformly sample the unbalanced original time series data. Data on daily power use were agglomeratively hierarchically clustered by van Wijk and van Selow [114] using the root mean square distance. With the use of calendar-based visualization, the distribution of the clusters throughout the course of the week and the year was investigated as well.

Agglomerative hierarchical

A hierarchical structure is not imposed by partitional clustering algorithms, which simultaneously discover all clusters as a partition of the data. Algorithms for hierarchical clustering identify nested groupings. These are examples of hierarchical clustering algorithms:

1. Agglomerative mode. This is a bottom-up way of clustering. I begin with a single data point as its own cluster and merge the most comparable pairs of clusters one at a time until we reach a final cluster that contains all the data points.

2. Divisive mode - This is a top-down clustering technique in which all the data points are first grouped into a single cluster, which is then recursively subdivided into smaller clusters.

Agglomerative hierarchical clustering methods frequently perform poorly because they are unable to adapt after making a merger decision. The same is true for techniques used in divisive hierarchical clustering. Time series of different lengths can be grouped together using hierarchical clustering. If an appropriate distance measure, such as dynamic temporal warping, is used to determine the distance/similarity, it is also applicable to series of uneven length [115].

Hierarchical clustering groups data objects (e.g., time series) into clusters to form a tree-like structure. There are two types: agglomerative and divisive. Agglomerative clustering, the more popular method, starts with each object in its own cluster and progressively merges them until all objects are in one cluster or specific termination conditions are met. The single linkage algorithm merges clusters based on the smallest distance, while Ward's minimal variance procedure combines clusters with the least impact on the sum-of-squares variance. It tests all possible mergers, selecting the one with the lowest value. [110]

The hierarchical clustering method CHAMELEON [116] is another one. Only when there is a high level of interconnection and closeness (proximity) between two clusters relative to the internal inter-connectivity of the clusters and the closeness of the items inside the clusters are the clusters combined.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

DBSCAN is a clustering algorithm, It simply needs one input parameter and helps the user choose a suitable value for it. It unearths clusters of any shape. Last but not least, DBSCAN is effective even for big spatial databases. When using density-based algorithms, like DBSCAN [117] the fundamental concept is to keep forming a cluster as long as the density (number of objects or data points) in the "neighborhood" is higher than a predetermined limit. OPTICS [118] computes an augmented cluster ordering for automatic and interactive cluster analysis rather than constructing a clustering explicitly. The ordering overcomes the challenge of choosing parameter values by containing information that is equal to density-based clustering and collected from a wide variety of parameter choices.

Partitional Clustering

Using a set of k non-overlapping groups, partition clustering methods divide the n objects. The number of clusters we wish to split is indicated by the input parameter K. By minimizing the objective function, partitioned clustering algorithms divide the data into k clusters in an iterative manner. When using the partitioning procedure, finding two points referred to as seed points to establish two clusters is necessary. The distance between a point and both seed points must be determined in order to allocate it to the closest seed point in order to determine the nearest seed point to all the points. The selection of the seed point is crucial in this process. We could arrive at the incorrect solution if we choose the wrong seed points. The best way to selecting good seed points is to select just two of the available options rather than more. The benefit of this is that null clusters won't exist. When selecting seed points, it's important to keep in mind that they should also be suitably far enough from one another in order for the right clusters to form. In this category, algorithms like k-mean, k-modes, PAM, CLARA, CLARANS, fuzzy-C-means, DBSCAN, etc. are researched. K-means is an algorithm that divides your items into K number of groups depending on qualities or traits. The number K is a positive integer. By

reducing the sum of squares of distances between the data and the relevant cluster centroid, the data are grouped [119]. The k-means objective function is provided as follows:

$$E = \sum ||X_i - m_i||^2 \tag{2.5}$$

E is the sum of square errors for all objects in the data, Xi is a point in a cluster C, and mi is the mean of cluster ki in the expression above. K-means seeks to reduce the total squared error across all K clusters. According to the technique, initial group centroids should be placed at k places in space that reflect the items that need to be clustered. The next step is to assign each object to the cluster center that is closest to it. then determine the mean of each cluster to get new centroids. Until there is no change in centroids, repeat these steps. Even though K-means was first presented more than 50 years ago, it is still one of the most popular clustering methods [120].

As with k means, the fuzzy-C means (FCM) algorithm also uses centroid-based clustering, but it requires that the number of clusters, k, be predetermined. Instead of allocating each object to a distinct cluster, each object is given a membership number between 0 and 1 to indicate whether it belongs to that cluster. Data mining, pattern recognition, classification, and picture segmentation are applications of FCM. PAM, CLARA, and CLARANS are some other partitioning algorithms that make use of the idea of selecting the seed points from the available points [121].

Hierarchical Clustering

Using agglomerative or divisive methods, hierarchical clustering [122] creates a hierarchy of clusters for cluster analysis. The bottom-up, agglomerative method treats each item as a cluster before progressively merging the clusters. In contrast, the divisive method starts with a single cluster that contains all objects and then breaks that cluster to find clusters that only contain one object (top-down). Due to their inability to alter the clusters after a cluster has been split in two using the divisive approach or after merging using the agglomerative method, hierarchical algorithms are generally considered to be of low quality. To address this problem, hierarchical clustering methods are frequently paired with another algorithm to form a hybrid clustering solution. The performance of hierarchical clustering is also improved by several extended efforts, such as Chameleon [116], CURE [123], and BIRCH [124], where the merging approach is improved or created clusters are improved.

A pair-wise distance matrix of time-series is used to create a layered hierarchy of related groups, which is analogous to hierarchical clustering of time-series [125]. Hierarchical clustering is a method that can be utilized for time-series clustering to a large extent because it has excellent visualization capabilities [126], [114]. For instance, Oates, Schmill, and Cohen [127] build clusters of an autonomous agent's experiences via agglomerative clustering. With a dataset that includes 150 trials of actual Pioneer data in a range of scenarios, they use Dynamic Time Warping (DTW) as a dissimilarity metric. Hirano and Tsumoto apply average linkage agglomerative clustering, a form of hierarchical technique for time-series clustering, in a different study [128]. Due to its strength in visualization, hierarchical is also frequently used in research to assess dimensionality reduction or distance metric. For instance, the authors of a study [129] provided a Symbolic Aggregate Approximation (SAX) representation and utilized hierarchical clustering to assess their

results. They demonstrate that hierarchical clustering produces results like Euclidean distance when utilizing SAX.

Another well-known and notable characteristic of this technique is that unlike for the most part algorithms, hierarchical clustering does not require the number of clusters as an initial parameter. It is also a time-series clustering strength because it is typically challenging to specify the number of clusters in real-world problems. Furthermore, despite a variety of techniques, hierarchical clustering can group time series of different lengths. If an appropriate elastic distance measure, such as Dynamic Time Warping (DTW) [130], [131] or Longest Common Subsequence (LCSS) [132], [133] is employed to compute the dissimilarity/similarity of time-series, it is possible to cluster unequal time-series using this technique. The ability of this algorithm to tolerate unequal time-series is a result of the fact that prototypes are not required during its procedure. However, due to its quadratic computational complexity and limited scalability, hierarchical clustering is essentially unable to deal with huge time-series [134]. As a result, it is constrained to tiny data-sets.

K-Mean Clustering

One of the most well-known, widely used, and straightforward clustering methods is the k-means algorithm [135], [136] which is frequently used to address clustering issues. The given data set is categorized in this technique using a user-defined number of clusters, k. To define k centroids, one for each cluster, is the main notion. This is how the objective function J is presented.

$$\text{Minimize } J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 \tag{2.6}$$

Where is a $\left\| x_i^{(j)} - c_j \right\|^2$ of the distance between a data point and $x_i^{(j)}$· the cluster centre cj The k-means algorithm's flow diagram is shown in Figure 2.6

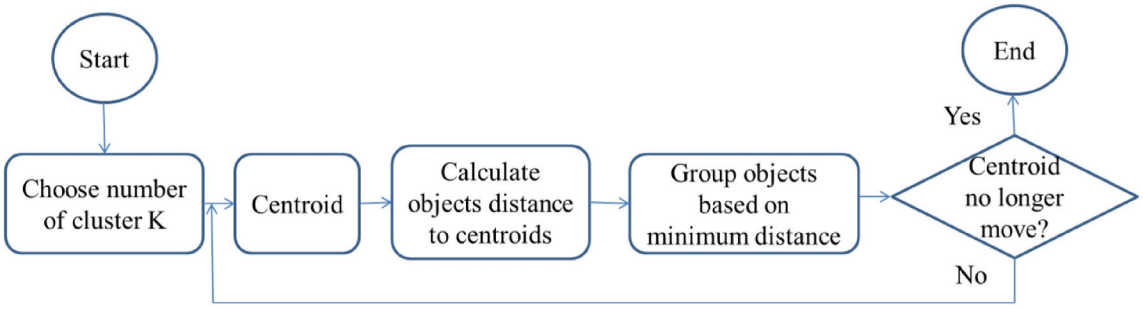


Figure 2.6: k mean algorithms [137].

The Linde-Buzo-Gray (LBG) algorithm, a k-means-like algorithm, was proposed for vector quantization (VQ) [138] for signal reduction. Prototype vectors are referred to as code words in this context, and together they make up a code book. VQ seeks to minimize information loss while representing the data with fewer pieces. Although one of the most widely used clustering techniques, k-means clustering has some limitations.

These limitations include: The initial divisions and the number of clusters, k, cannot be determined efficiently or universally, thus k-means is susceptible to noise and outliers. An object is nevertheless driven into a cluster even though it is far from the cluster centroid, (However, despite its effectiveness, many labelled data points are needed to produce the ideal cluster centroid values.) changing the shape of the clusters [139].

Algorithm 2 *K-means Algorithm*

1: **Initialization:** Choose k distinct points randomly as the initial centroids.
2: Assign each object to the closest centroid.
3: Recalculate the positions of the centroids based on the objects assigned to them.
4: Repeat Steps 2 and 3 until the centroids no longer move. This results in a grouping of the objects, allowing calculation of the minimized metric [137].

K-Medoid-Clustering

A traditional method of clustering that divides a set of n objects into k different groups is known as K-medoid. This k: the necessary number of clusters must be supplied by the user. Based on the idea of minimizing the total difference between each item and its matching reference point, this method operates. The algorithm selects k objects at random from dataset D to serve as initial medoids, or representative objects. A medoid is the point in the supplied data set that is the most centrally placed and has an average dissimilarity to all the other points in the cluster that is minimum. The new medoid is then chosen for each medoid following each assignment of a data object to a certain cluster [140]. Moreover, the fundamental drawback of the k-Means approach is that it is susceptible to outliers because an object with an unusually big value has the potential to skew the distribution of data. The most centrally situated object in a cluster, known as a medoid, can be used as a reference point instead of the mean value of the objects in the cluster. So long as the dissimilarities between each object and its matching reference point are kept to a minimum, the partitioning approach can still be applied. The k-Medoids approach is founded on this idea. In order to find k clusters in a set of n objects, k-Medoids clustering algorithms first choose a representative object (the medoids) at random for each cluster. The remaining items are grouped together with the medoid that they are most related to. Instead of using the mean value of the items in each cluster, the k-Medoids technique employs representative objects as reference points. The number of clusters to be divided across a set of n items, represented by the input parameter k, is provided to the algorithm [140].

2.2.2 Feature- and Model-Based Methods

Feature- and model-based techniques are frequently domain-specific, requiring their adaptation for applications across a range of domains. Machine learning uses feature-based and model-based techniques to draw out relevant information from data and provide predictions or classifications. When employing feature-based techniques, relevant features are chosen from the input data. These features can be extracted or pre-defined using methods like reduction of dimensionality or feature engineering. In model-based techniques, a suitable machine learning algorithm is chosen, and the input data are trained on it to create a predictive model that can make precise predictions on new data.

Feature Based Methods

When clustering based on raw data, especially for data taken at high sample rates, working in a high-dimensional space is necessary. Working directly with excessively noisy raw data is not something that is also advisable. To deal with these issues, several feature-based clustering techniques have been put forth. Despite the fact that the majority of feature extraction techniques are general in nature, the characteristics that are often extracted depend on the application. In other words, a set of qualities that are effective for one application could not be applicable to another. Some studies even go so far as to add another feature selection stage to the feature extraction process to further cut down on the amount of feature dimensions.

Modified k-Means (MKM)

Wilpon and Rabiner [141] modified the standard k-means clustering algorithm for the recognition of isolated words with the goals of developing an automatic clustering algorithm that could be implemented by any user with a minimal understanding of clustering procedures and to provide the template sets as accurate as those created by other clustering algorithms. The adjustments deal with issues like how to find cluster centers, how to divide clusters to make more clusters, and how to make the final cluster representations. Each pattern used to replicate a single spoken word has an inherent time (for example, it lasts for a certain number of frames), and each frame contains a vector of coefficients for linear predictive coding (LPC). Based on the Itakura distance for gauging the separation between two frames, a symmetric distance measure was developed to gauge the separation between two spoken word patterns. At that time, it was demonstrated that the suggested modified k-means (MKM) clustering technique performed better than the renowned unsupervised without averaging (UWA) clustering approach.

By employing two hierarchical clustering techniques, the Ward's minimal variance algorithm and the single linkage algorithm, to normalized spectra (normalized by the amplitude of the highest peak), Shaw and King [142] indirectly grouped time series. The original time series were used to create the spectra, with the means set to zero. When the filtered spectra from principal component analysis (PCA) were grouped, it was discovered that the 14 most important eigenvectors may produce equivalent outcomes. It made advantage of the Euclidean distance.

Model Based Methods

Model-based approaches aim to best match the data to the presumptive model for each of the clusters. Model-based methods can be categorized into two main categories: statistical approach and neural network approach. Furthermore, AutoClass [143] which employs Bayesian statistical analysis to determine the number of clusters, serves as an illustration of a statistical technique. Competitive learning, including ART [144] and self-organizing feature maps [145] are two well-known neural network approach to clustering techniques.

In model-based clustering, optimization tries to match the provided data to a certain mathematical model. It is predicated on the idea that various underlying probability distributions combine to produce data. It contains:

An effective iterative refining approach is the EM (Expectation Maximization) algorithm. It is a development of k-means. Each item is given a weight (probabilistic

distribution) before being assigned to a cluster, and the new means are computed using weighted measures.

COBWEB: Fisher created it in 1987. It is a well-liked and straightforward approach to progressive conceptual learning. It produces a classification tree-like hierarchical grouping. Each node offers a probabilistic description of the topic to which it refers. It automatically modifies the partition's class count. The user is not required to supply this input parameter. CLASSIT: It is a COBWEB extension for incremental continuous data clustering. It also experiences the same issues as COBWEB [146] Auto Class: It was created in 1996 by Cheeseman and Stutz. To determine how many clusters there are, Bayesian statistical analysis is used. It enjoys huge acclaim in business.

SOM (Soft-Organizing feature Map) Kohonen Self-Organizing Feature Maps (KSOMs), also known as SOMs or topological ordered maps. It converts every point from a high-dimensional source space into a 2 to 3-dimensional destination space while preserving as much of the topology (i.e., distance and proximity relationships) as possible. The cluster centers typically lie in a low-dimensional manifold in the feature space, like k-means. The clustering process in this case involves many units vying for the present object. The winning unit has a weight vector that is most similar to the current object. By having their weights changed, the winner and its neighbours gain knowledge. It is thought that SOMs resemble the processing that can take place in the brain. It can be used to visualize high-dimensional data in a two- or three-dimensional space [115].

Furthermore, Model-based clustering techniques use some mathematical models to improve and assess the appropriateness of the supplied data. Model-based clustering algorithms discover feature information for each cluster, where each cluster represents a concept or class, in a manner similar to conventional clustering.

2.3 Data Analysis Tools

To extract valuable insights and knowledge, one must first examine and understand the raw data. Data must be organized, cleaned, transformed, and visualized using statistical and computational methods in order to find patterns, trends, and linkages that can guide decision-making. The purpose of data analysis is to make sense of massive amounts of data, find patterns and connections, and derive inferences from the data.

There are many tools for data analysis, particularly in Python, a well-liked programming language with a number of potent tools and modules. Several of the frequently used Python data analysis tools which listed below.

2.3.1 Numpy

A powerful library for numerical computation that offers capabilities for working with arrays, matrices, and other types of data structures required in data analysis. It is far more effective than conventional Python code because it was created in C and operates on whole arrays and matrices. NumPy is widely used in the back-ends of the other libraries on this website. A thorough user's manual for the library, as well as tutorials and examples, are available in the official NumPy documentation. A vibrant community for NumPy also offers assistance and learning tools [147].

2.3.2 Pandas

Data cleaning, filtering, grouping, and merging utilities are provided by a library for data manipulation and analysis while working with tabular data. It features robust plotting capabilities utilizing Matplotlib as the default plotting backend and employs effective libraries like Numpy as its compute and data representation backend. Detailed explanations and examples of how to use the library for data manipulation and analysis can be found in the Pandas documentation, which is another great source [148].

2.3.3 Matplotlib

A library for data visualization that offers resources for making line charts, scatter plots, bar charts, and histograms, among other types of plots. It can run interactively in applications like Jupyter notebooks, although it is ideally suited for producing static plots and figures of publication quality [149].

2.3.4 Seaborn

A high-level interface is provided by the Python data visualization library Seaborn, which is based on Matplotlib and allows for the creation of intricate and aesthetically beautiful statistical charts. It features pre-built color schemes and themes, supports popular statistical plot types, connects with Pandas for simple data processing, and offers tools for displaying intricate correlations between numerous variables. Overall, Seaborn is an effective tool for sharing data ideas with others and conducting exploratory data analysis [150].

2.3.5 Scikit-learn

An open-source Python machine learning package called Scikit-learn offers a variety of supervised and unsupervised learning algorithms, tools for data preprocessing and feature engineering, and tools for model selection and evaluation. Scikit-learn is built on top of NumPy, SciPy, and Matplotlib and is intended to work in tandem with other Python scientific computing tools. Scikit-learn is a well-liked option for data scientists and machine learning professionals working on classification, regression, and clustering tasks because of its intuitive API and comprehensive documentation [151].

2.3.6 Ts-Learn

A set of tools for time series analysis and classification are provided by the Python package TSLearn. Preprocessing, feature extraction, model creation, and visualization are just a few of the functions it offers. It comes with built-in time series data preprocessing tools like scaling, imputation, and normalization. Other feature extraction techniques include Singular Spectrum Analysis (SSA), Symbolic Aggregate Approximation (SAX), and Piecewise Aggregate Approximation (PAA). The package also includes model-building methods including K-Nearest Neighbors, Random Forests, Support Vector Machines (SVM), and Hidden Markov Models (HMM) [152].

2.3.7 Statsmodels

A variety of tools are available for analyzing data, estimating statistical models, and running statistical tests in the Statsmodels Python package for statistical modeling and econometric analysis. In addition to a variety of statistical models like linear regression, generalized linear models, time series analysis, and survival analysis, Statsmodels also contains tools for hypothesis testing, statistical inference, and model selection. It is built on top of NumPy and Pandas. Statsmodels is a useful tool for data analysis and modeling due to its interoperability with other well-liked scientific computing libraries in Python and its extensive statistical capabilities [153].

2.3.8 Data Science and ML Frameworks

Frameworks for data science and machine learning (ML) offer a complete set of libraries and modules for developing, testing and deploying data-driven models. Popular choices for these frameworks include TensorFlow, PyTorch, Scikit-learn, and Keras, among others. They make it possible for data scientists and machine learning (ML) specialists to work with huge datasets, test out different model architectures, and carry out effective model training and deployment. These frameworks frequently include a selection of pre-built models as well as tools for feature engineering and data preprocessing. In addition, they frequently integrate with other well-known Python libraries like NumPy and Pandas. Overall, these frameworks are crucial for creating dependable and scalable data-driven solutions for a range of applications and industries [154].

2.3.9 XGBoost

Machine learning models can be trained quickly and scaled up with the help of the distributed gradient boosting library known as XGBoost. It is an ensemble learning technique that combines the results of several ineffective models to yield a more accurate result. The machine learning algorithm known as XGBoost, which stands for "Extreme Gradient Boosting," has grown to be one of the most well-liked and widely used due to its capacity for handling large datasets and its ability to deliver cutting-edge results in a variety of machine learning tasks, including classification and regression. One of XGBoost's important strengths is its effective handling of missing values, which enables it to handle actual data with missing values without the need for a lot of pre-processing. Additionally, XGBoost includes built-in parallel processing capabilities, enabling quick model training on big datasets [155].

Due to its capability to manage enormous quantities of time-series data with high-dimensional features, which are frequently present in energy systems, XGBoost is a widely used technique for energy forecasting. Based on historical data and additional pertinent variables like weather, time of day, and seasonality, this algorithm can be used to forecast energy demand, production, and consumption trends. Overall, XGBoost has demonstrated that it can greatly improve the precision and effectiveness of energy management and planning in a variety of applications. In our thesis work, we utilized XGBoost, a powerful machine learning algorithm, for the purpose of energy forecasting for various appliances. XGBoost is known for its exceptional performance in regression. By implementing XGBoost, we were able to generate highly accurate energy usage predictions

for the appliances. These findings have significant implications for promoting energy efficiency and sustainability.

2.4 Summary

In view of their capacity to predict energy consumption patterns in smart homes with accuracy, and machine learning techniques for energy forecasting have drawn more and more interest in recent years. In a literature review, a review of these techniques and their usefulness in energy forecasting has been provided. For energy forecasting, clustering techniques have also been investigated, notably time-series clustering, which can increase the predictability of appliances. In general, the potential for increasing energy efficiency in smart homes is very high when using machine learning and clustering techniques for energy forecasting. Overall, the information presents a thorough analysis of the numerous approaches utilized in energy forecasting and clustering, highlighting the advantages and disadvantages of each strategy.

Chapter 3

Time-Series Clustering for Smart Homes Appliances Prediction

Since our time series data contains time samples, we must cluster them using certain time series data-appropriate techniques. Specialized time series data clustering methods are used to look for patterns and commonalities in the data. Static time series clustering is a technique for putting time series data into groups depending on how similar their patterns or traits are and the clusters are independent of any parameter like time-of-the-day. Utilizing algorithms, the clustering process finds groups or clusters of time series that display comparable behaviours. The clusters can then be studied to acquire insights into the underlying patterns or trends in the data. The choice of clustering method will rely on the specific features of the data. Static time series clustering can be used to discover subgroups of data that need various kinds of analysis or action in a variety of industries, we utilize it for the predictions of the smart home appliance. This can also be utilized in future for the scheduling of appliances.

3.1 Static Time-Series Clustering

This section presents the time-series k-shape clustering algorithm and its incorporation in Static Time-Series Clustering in measures other than Euclidean Distance (ED). In [111] a novel time-series clustering scheme named k-shape is developed considering the shortcoming of existing raw-based schemes. The k-shape time-series clustering algorithm creates homogeneous and well-separated clusters. Therefore, we incorporate this algorithm in this proposed static time-series clustering of different appliances in smart homes. More details about the k-shape algorithm and its incorporation in the proposed intelligent data processing scheme are provided in the next section after the discussion of basic definitions which are required to understand time series data. These are presented as follows.

Time-Series Invariances

Sequences are frequently warped in some way depending on the domain, and to compare sequences effectively, distance metrics must meet several invariances. The common time-series distortions and their invariances are discussed in [156] for a more thorough evaluation.

Shift Invariance

Two time-series sequences can be treated as comparable if their phases are different (global alignment) or if some parts of the sequences are aligned but not others (local alignment). Heartbeats, for instance, may not be in phase depending on when the measurements are taken (global alignment), and different people's handwriting may require local alignment based on the size of the letters and the spacing between sentences.

Time Series Distance Measure

The two cutting-edge methods for comparing time series first z -normalize the sequences before using a distance metric to assess how similar they are and perhaps capture more invariances. The basic ED [157] is the distance metric that is most frequently employed. ED compares two time series $\vec{x} = (x_1, \ldots, x_m)$ and $\vec{y} = (y_1, \ldots, y_m)$ of length m as follows:

$$ED(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \tag{3.1}$$

DTW, also known as Dynamic Time Warping, is a widely used distance measurement method [131]. It can be seen as an extension of Euclidean Distance (ED) that allows for local (non-linear) alignment. In DTW, a matrix M of size m-by-m is created, where each element represents the ED between two points from vectors $\vec{x}$ and $\vec{y}$. A warping path, denoted as $W = w_1, w_2, \ldots, w_k$ with k $\geq$ m, is a contiguous set of matrix elements that defines a mapping between $\vec{x}$ and $\vec{y}$. Various constraints are applied to determine this mapping [158]:

$$DTW(\vec{x}, \vec{y}) = \min \sqrt{\sum_{i=1}^{k} w_i} \tag{3.2}$$

Dynamic programming can be used to compute this path on matrix M for the evaluation of the following recurrence:

$$\gamma(i, j) = ED(i, j) + min\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \tag{3.3}$$

Constraining the warping path to only travel over a portion of matrix M's cells is a popular procedure. The warping window refers to the band's width and the subset matrix's shape, respectively. The Sakoe-Chiba band is the most popular band for constrained Dynamic Time Warping (cDTW) [131].

Figure 3.1 (b) shows the computation of the warping path (dark cells) for cDTW confined by the Sakoe-Chiba band with width 5 cells (light cells). Figure 3.1 (a) displays the difference in alignments of two sequences supplied by ED and DTW distance measures.

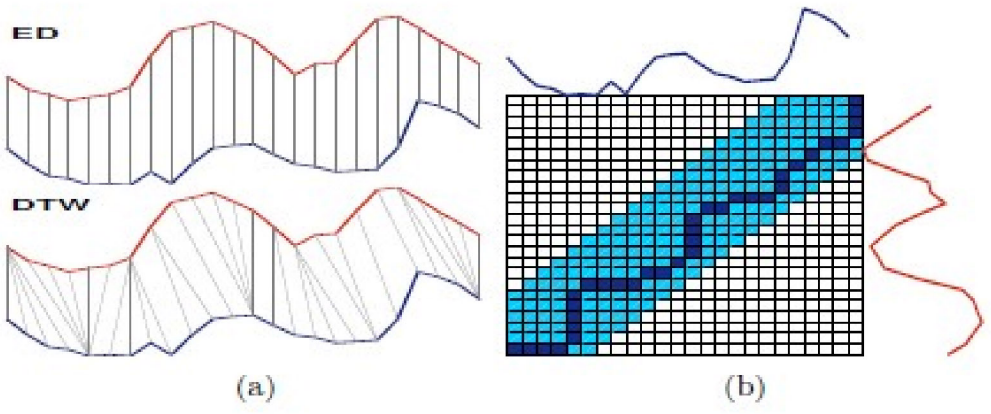


Figure 3.1: Similarity using: (a) ED (top) and DTW (bottom), (b) Sakoe-Chiba band with the warping path computed under cDTW [111].

Wang et al. [159] has conducted a thorough evaluation of 9 distance metrics and their variations. They discovered that ED is the most effective measure with a respectable level of accuracy and that, in contrast to other measures, DTW and cDTW work remarkably well. DTW greatly saves computing time and performs marginally better than DTW. To increase the speed of cDTW even further, some optimizations have been suggested [160]. A few time-series clustering techniques that make use of these distance measurements are discussed in the next section.

Time-Series Clustering Algorithms

Various techniques have been developed in the literature to cluster time series. Typically, a new algorithm modifies existing algorithms in two ways. Firstly, it may change the default distance measures to those that are better suited for comparing time series, known as raw-based methods. Secondly, it may transform the sequences into data that can be directly used in traditional algorithms, known as feature- and model-based methods [110]. Raw-based methods can readily benefit from the extensive literature on distance measurements 3.1. This literature has shown that certain measures, such as DTW, provide invariances that are generic and suitable for almost every domain [161]. On the other hand, feature- and model-based techniques are typically specific to particular domains, requiring modifications of features or models for applications in different domains. In this study, we opt for a raw-based methodology due to the limitations of feature- and model-based techniques.

Agglomerative hierarchical, spectral, and partitional clustering are the three main raw-based techniques [156]. The single, average, and complete linkage versions are the "linkage" criteria for hierarchical clustering that are most frequently used [122]. The success of spectral clustering [162] in comparison to other data types [163] has lately caused it to gain attention [156]. K-means [136] and k-medoids [122] are the most illustrative examples of partitional algorithms. We classify partitional methods as shape-based strategies when they employ distance measures that provide invariances to scaling, translating, and shifting.

K-medoids is typically preferred among these methods [110] because, unlike K-means, it computes the dissimilarity matrix of all data sequences and uses real sequences as cluster centroids. K-means, on the other hand, requires the computation of artificial sequences as centroids, which makes it difficult to easily adapt distance measures other than ED. Only the k-means class of algorithms, though, can scale linearly with the size of the datasets out of all these techniques. DTW [164] and a distance metric that allows for pairwise scaling and shifting of time-series sequences [165] have both recently been added to the list of k-means' compatible modifications. Both of these changes rely on fresh approaches to computing cluster centroids, which we are going to explore next.

3.2 K-shape Clustering

Our goal is to utilize a time-series clustering method that is accurate, scalable, and invariant to scaling and shifting. In this scenario, k-Shape, a clustering approach that can maintain the forms of time series and is based on (i) a distance measure and (ii) a centroid computation technique is utilised herein. The distance measurement, which is based on the cross-correlation measure (Section 3.2.1), is covered first. It determines the centroids of time-series clusters based on this distance measure (Section. 3.2.4). k-Shape clustering technique creates homogeneous and well-separated clusters via an iterative refinement process that scales linearly in the number of sequences (Section. 3.2.4).

3.2.1 Time-Series Shape Similarity

As already stated, distance measurements that adequately account for amplitude and phase distortions are necessary for capturing shape-based similarity. Unfortunately, the most effective distance measures that provide invariances to these distortions, like DTW, discussed in 3.1 are computationally expensive. In order to get around this efficiency restriction, a normalized cross-correlation metric is used.

Cross-correlation is a commonly utilized similarity metric in signal and image processing for time-lagged signals. However, recent comprehensive evaluations of state-of-the-art distance measures for comparing time-series have largely overlooked cross-correlation. Cross-correlation involves comparing individual points between signals, and it was not considered in previous experimental assessments of various distance measures. Several studies have examined different distance measures, such as 9 measures and their variations in [161] [159], as well as 48 measures in [166], but they did not include cross-correlation. The difficulty in finding suitable normalizations for both the data and the cross-correlation measure contributes to its limited use, as different areas and applications have diverse requirements. Additionally, slow implementations of cross-correlation can give the false impression of it being as slow as Dynamic Time Warping (DTW). These challenges associated with cross-correlation have hindered its widespread adoption as a time-series distance metric. In the following sections, we will address these issues and propose an efficient and domain-independent normalization approach that enables the development of a shape-based distance measure for quick and accurate time series comparison.

3.2.2 Cross-Correlation Measure

A statistical tool for assessing the similarity of two sequences is cross-correlation $\vec{x} = (x_1, \ldots, x_m)$ and $\vec{y} = (y_1, \ldots, y_m)$ even when they are not perfectly positioned. Cross-correlation maintains y as constant and slides x across y to compute their inner product for each shift s of x in order to achieve shift-invariance. In k-shape algorithm, a normalized version of the cross-correlation measure is utilised as a distance measure to compare and cluster the time-series. In cross-correlation process, the time-series $\vec{x}$ is shifted over time to check the correlation with other time-series $\vec{y}$. In this way, it computes all the possibilities of similarity between two time-series at different time-lags. A shift of sequence $\vec{x}$ is denoted as follows.

$$\vec{x}_{(s)} = \begin{cases} (\overbrace{0, \ldots, 0}^{|s|}, x_1, x_2, \ldots, x_{m-s}), & s \geq 0 \\ (x_{1-s}, \ldots, x_{m-1}, x_m, \underbrace{0, \ldots, 0}_{|s|}), & s < 0 \end{cases} \tag{3.4}$$

With all the possible shifts$\vec{x}_{(s)}$ are considered, with $s \in [-m, m]$, cross-correlation sequence $CC_w(\vec{x}, \vec{y}) = (c_1, \ldots, c_w)$, is computed with length $2m - 1$ which is given by.

$$CC_w(\vec{x}, \vec{y}) = R_{w-m}(\vec{x}, \vec{y}), \quad w \in \{1, 2, \ldots, 2m - 1\} \tag{3.5}$$

where $R_{w-m}(\vec{x}, \vec{y})$ is computed, in turn, as:

$$R_k(\vec{x}, \vec{y}) = \begin{cases} \sum_{l=1}^{m-k} x_{l+k} \cdot y_l, & k \geq 0 \\ R_{-k}(\vec{y}, \vec{x}), & k < 0 \end{cases} \tag{3.6}$$

The point w at which $CC_w(\vec{x}, \vec{y}$ is maximized is what we are trying to determine. The best shift of $\vec{x}$ with respect to $\vec{y}$ is then $\vec{x}_(s))$, where $s = w - m$, based on this value of w. $CC_w(\vec{x}, \vec{y}$ may need to be normalized differently depending on the domain or the application. Time series may also need to be normalized to eliminate intrinsic distortions in addition to the cross-correlation normalization.

3.2.3 Shape-Based Distance (SBD)

Based on the previous discussion, we use the coefficient normalization, which produces values between -1 and 1, regardless of the data normalization, to create a shape-based distance metric. The cross-correlation sequence is normalized by dividing it by the geometric mean of the autocorrelations of the separate sequences. The objective is to obtain a position w where $CC(\vec{x}, \vec{y}$ is maximum. The cross-correlation sequence is normalized by dividing it by the geometric mean of the autocorrelations of the separate sequences.

$$SBD(\vec{x}, \vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x}) \cdot R_0(\vec{y}, \vec{y})}} \right) \tag{3.7}$$

The Shape Extraction Algorithm 3is a technique for grouping time series data according to how similar their shapes are. Each time series in a dataset is first aligned to a reference sequence, after which the Shape-Based Distance metric is used to calculate their pairwise distances. Finally, spectral clustering is used to group the time series that have

similar forms into clusters. Before applying a matrix transformation to this dot product matrix to produce a matrix with the eigenvectors corresponding to the greatest eigenvalue, the technique first creates a matrix of dot products between the aligned time series. A centroid vector, which represents the average shape of all the time series in a cluster, is the last result of the process. This approach can be used for a variety of purposes, including predicting energy usage in smart homes, where clustering can increase forecasting models' precision.

Algorithm 3 *Shape Extraction Algorithm:* $C = SE(X, R)$

1: **Input:** X is an $n \times m$ matrix with z-normalized time series. R is a $1 \times m$ vector with the reference sequence against which time series of X are aligned.
2: **Initialize:** $X' \leftarrow [\ ]$
3: **for** $i = 1, 2, ..., n$ **do**
4: $\quad [dist, x'] \leftarrow SBD(R, X(i))$ using Eq. 3.7
5: $\quad X' \leftarrow [X'; x']$
6: **end for**
7: **Compute:** $S \leftarrow X'^T.X'$
8: **Compute:** $Q \leftarrow I - \frac{1}{m}.O$
9: **Compute:** $M \leftarrow Q^T.S.Q$
10: **Compute:** $C \leftarrow Eig(M, 1)$
11: **Output:** C is a $1 \times m$ vector with the centroid of clusters. [167]

3.2.4 Shape-Based Time-Series Clustering

The novel time-series clustering technique, k-Shape, is now described. To effectively create time series clusters, k-Shape uses the shape extraction technique from Algorithm 3 and the SBD distance measure from Section 3.2.1.

A partitional clustering technique called k-Shape is based on an iterative refinement process similar to that of k-means. K-Shape minimizes the sum of squared distances through this iterative process and is able to (i) construct homogeneous and well-separated clusters and (ii) scale linearly with the number of time series. Under the constraints of scaling, translation, and shift invariances, our algorithm computes centroids effectively while efficiently comparing sequences. The only scalable technique that considerably outperforms k-means is k-Shape, which is a nontrivial instantiation of k-means. In contrast to earlier attempts in the literature, [164], [165] k-Shape's distance measure and centroid computation method make this the case.

Two steps are carried out by k-Shape in each iteration: Each time series is compared to all computed centroids in the assignment step, and then each time series is assigned to the cluster with the closest centroid; this updates the cluster memberships; and in the refinement step, the cluster centroids are updated to reflect the changes in cluster memberships from the assignment step. Until there is no change in cluster membership or the allotted number of iterations has been reached, the algorithm repeats these two steps. K-Shape uses the distance metric from (Section. 3.2.1) for the assignment stage and the centroid computation method from (Section. 3.2.4) for the refining step.

The time series data and the number of clusters we wish to construct are the two inputs that k-Shape expects. The algorithm first groups the time series into clusters at random. The shape extraction approach is then used to calculate each cluster centroid (see Section.

3.2.4). After computing the centroids, SBD distance metric is used to further narrow the memberships of the clusters. This process is repeated until the algorithm converges or hits the maximum number of iterations, which is typically a low number, like 100. The algorithm's output includes the centroids for each cluster as well as the assignment of sequences to clusters.

The k-Shape algorithm utilizes two inputs: a time series set X and the desired number of clusters, denoted as k. Initially, the time series in X are randomly clustered. The algorithm computes the centroid of each cluster using Algorithm 3 (lines 5-10). This process continues until the algorithm converges or reaches a maximum number of iterations, typically set to a low value such as 100.

Algorithm 4 *K-Shape Algorithm:* $[ID_{Cluster}, C] = K - shape(X, k)$

1: **Input:** X is an $n \times m$ matrix containing n time series of length m that are initially z-normalized. k is the required number of clusters.
2: $iter \leftarrow 0$
3: $ID'_{Cluster} \leftarrow [\ ]$
4: **while** $ID_{Cluster} \neq ID'_{Cluster}$ and $iter < 100$ **do**
5: $\quad ID'_{Cluster} \leftarrow ID_{Cluster}$
6: $\quad //Refinementstep$
7: $\quad$ **for** $j \leftarrow 1$ to K **do**
8: $\quad\quad X' \leftarrow [\ ]$
9: $\quad\quad$ **for** $i \leftarrow 1$ to n **do**
10: $\quad\quad\quad$ **if** $ID_{Cluster}(i) = j$ **then**
11: $\quad\quad\quad\quad X' \leftarrow [X'; X(i)]$
12: $\quad\quad C(j) \leftarrow SE(X', C(j))$ using Algorithm 1.
13: $\quad //Assignment - step$
14: $\quad$ **for** $i \leftarrow 1$ to n **do**
15: $\quad\quad dist_{min} \leftarrow \infty$
16: $\quad\quad$ **for** $j \leftarrow 1$ to K **do**
17: $\quad\quad\quad [dist, x'] \leftarrow SBD(C(j), X(i))$
18: $\quad\quad\quad$ **if** $dist < dist_{min}$ **then**
19: $\quad\quad\quad\quad dist_{min} \leftarrow dist$
20: $\quad\quad\quad\quad ID_{Cluster}(i) \leftarrow j$
21: $\quad iter \leftarrow iter + 1$
22: **Output:** $ID_{Cluster}$ is an n-by-1 vector of clusters) [167].

The algorithm's output includes the centroids for each cluster and the assignment of sequences to clusters. Algorithm 4 provides a comprehensive outline of the k-shape clustering process. It takes as input all the time-series data and the desired number of clusters, and it returns the number of clustered time-series of appliances that are used in the forecasting process. The next section presents the evaluation of clustering performance and its application in forecasting schemes.

3.3 Dynamic Time-Series Clustering

The use of dynamic clustering techniques in this thesis work is driven by the need to investigate their potential in improving smart homes' analysis and prediction of energy

consumption patterns. This method aims to identify patterns and trends that may not be apparent using static clustering methods. Dynamic clustering is significant because it provides an opportunity to develop more accurate energy forecasting models, identify opportunities for energy savings, and ultimately contribute to the development of more sustainable smart homes. Furthermore, this thesis work offers valuable insights into how clustering techniques can be used to analyze time-dependent data on energy consumption patterns in smart homes, providing practical solutions that can benefit both homeowners and the environment.

3.3.1 A Potential Solution for Dynamic Clustering

In this thesis work, to shift the cluster members dynamic clustering is employed by segmenting a dataset of 43 smart home appliances into day and night periods and applying separate clustering algorithms for each period. Given the variations in energy consumption patterns between day and night, this approach provides a more comprehensive understanding of the data and offers greater potential for optimizing energy usage in smart homes. The day period was defined from 6:00 in the morning until 21:00, while the night period was from 21:00 to 5:59. By running separate clustering algorithms for the day and night periods, this thesis work aims to gain deeper insights into the dynamics of energy consumption during different times of the day. Additionally, the methodology employed in this thesis work is also applied to forecasting appliance consumption patterns to identify any impacts on accuracy. The use of separate algorithms for day and night periods is expected to improve the accuracy of appliance consumption forecasting in this thesis work.

The majority of clustering methods have traditionally focused on static data. But as the frequency of data collection has increased, more focus has been placed on grouping time-varying observations. Despite the fact that there have been many advancements in this topic over the past few decades, Liao. [110] and Aghabozorgi et al. [169] provide a useful overview of the most significant research.

In dynamic time series clustering, cluster membership is subject to change over time, allowing for the identification of evolving patterns and trends in the data. As energy consumption patterns can change over time, dynamic clustering enables the identification of new clusters that reflect these changes in the data. This approach is particularly relevant in the context of smart homes, where energy consumption patterns may vary according to daily use, seasonal factors, and user behaviour. Dynamic clustering can provide a more accurate depiction of energy consumption patterns and provide feasible options for smart home energy use optimization by considering these aspects.

Illustration of a time series stretched across two clusters in which two of the time series, shown by the orange colors, switch across clusters midway through the time window. As shown in the Figure 3.2. The presented cluster analysis, indicated by the orange and black clusters, remained consistent for the first 35 hours. However, beyond this period, the consumption patterns of the orange cluster began to align more closely with those of the cluster situated below it. This finding demonstrates the dynamic nature of cluster membership, as changes in consumption patterns over time can result in shifts in cluster membership. Such insights highlight the importance of employing dynamic clustering methods in the analysis of time-dependent data, particularly in the context of smart homes where energy consumption patterns are subject to evolving user behavior and external factors.

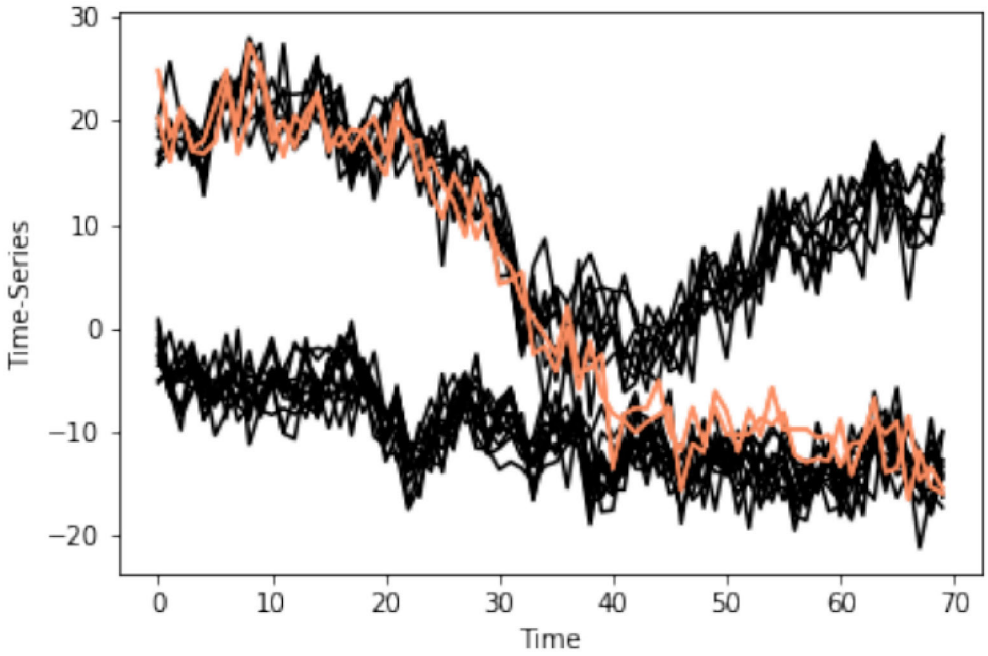


Figure 3.2: Time-series spread across two clusters, where two series change cluster membership halfway through the time window, are shown in orange [168].

Clearly, one could contend that this behavior represents a third, new cluster. To capture these changes, however, one would need a large number of clusters with very few members in each of them, which in a way goes against the fundamental goal of clustering as more observational units begin exhibiting this behaviour at various time-points.

3.3.2 Dynamic Time-series Clustering using K-Shape

The "Dynamic Time-series Clustering Algorithm" is a process that clusters time-series data of appliances based on the time interval for dynamic clusters and the required number of clusters. The process of dynamic time-series clustering is enumerated in Algorithm 5.

The Algorithm 5 follows the below steps:

1. The time-series data of appliances X is separated based on the time interval for dynamic clusters $\mathcal{T}$, resulting in two datasets $X^{\mathcal{T}}$ and $X^{\mathcal{T}'}$.

2. Initialize an empty cluster ID list $ID^{\mathcal{T}}_{Cluster}$ and $ID^{\mathcal{T}'}_{Cluster}$ for each time interval dataset, and concatenate them into a single list $ID_{Cluster}$.

3. For each time interval in $\mathcal{T}$, perform K-Shape clustering on the corresponding dataset using Algorithm 4 to obtain cluster labels IDX and centroids C.

4. Assign the cluster labels obtained in step 3 to the corresponding time interval in $ID_{Cluster}$.

Algorithm 5 Dynamic Time-series Clustering Algorithm

Input: Time-series data of appliances X, Time Interval for Dynamic Clusters $\mathcal{T}$, k is the required number of clusters.

1: **Separating Time-series data based on Interval, $\mathcal{T}$:**
$X = \begin{bmatrix} X^{\mathcal{T}} & X^{\mathcal{T}'} \end{bmatrix}$
2: **Initialize**:
$ID^{\mathcal{T}}_{Cluster} = [\ ],\ ID^{\mathcal{T}'}_{Cluster} = [\ ]$
3: $ID_{Cluster} = [ID^{\mathcal{T}}_{Cluster}\ ID^{\mathcal{T}'}_{Cluster}]$
4: **for** $j \leftarrow 1$ to $Length(\mathcal{T})$ **do**
5: $[IDX, C] = K - Shape(X(j), k)$ using Algorithm 4.
6: $ID_{Cluster}(j) = IDX$
7: $IDX = [\ ]$
8: **end for**
9: **return** Clusters for different time intervals, $ID_{Cluster} = [ID^{\mathcal{T}}_{Cluster}\ ID^{\mathcal{T}'}_{Cluster}]$

5. Return the clustering information for different time intervals, $ID_{Cluster}$, which contains the cluster labels for each time interval dataset.

The algorithm uses the K-Shape clustering algorithm to perform the clustering, which is a method for clustering time-series data based on shape similarity described earlier.

3.4 Summary

This chapter presented the k-shape time-series clustering algorithm and its incorporation in static time-series clustering. The k-shape algorithm is a novel time-series clustering scheme that addresses the shortcomings of existing raw-based schemes by creating homogeneous and well-separated clusters. The algorithm is accurate, scalable, and invariant to scaling and shifting.

To investigate the potential of dynamic clustering techniques in improving smart home analysis and prediction of energy consumption patterns, this thesis work employs a segmentation approach of clustering algorithms for each day and night periods. By running separate clustering algorithms for different time intervals (for example, the day and night periods), the thesis aims to gain deeper insights into the dynamics of energy consumption during different times of the day and provide practical solutions that can benefit both homeowners and the environment. Additionally, the methodology is applied to forecasting appliance consumption patterns to identify any impacts on accuracy. The use of algorithms for datasets on different periods is expected to improve the accuracy of appliance consumption forecasting which is evaluated and discussed in the next chapter.

Chapter 4

Implementation and Performance Evaluation

This section outlines the performance and evaluation of the proposed intelligent data processing approach within the framework of forecasting appliance consumption and presents a thorough discussion of the resulting outcomes across various dimensions.

4.1 Dataset Details

For the performance evaluation, we considered a public dataset [170]. This dataset considers a house with 43 smart plugs installed on several circuits. Along with different varieties of appliances, this dataset has a record of generation from solar panels installed at the house and the total consumption and generation records. Thus, this dataset contains all the possible values from a household. Different circuit names and their IDs are summarized in Table. 4.1. The house is a two-story building with 1700 square feet of area. The dataset is recorded in 2016, and it has hourly data samples of energy consumption (in KW)from all the smart plugs installed on appliances and circuits.

The dataset utilized for this study was compiled from January 1 at 00:00:00 to December 31 at 23:59:00 in 2016. The dataset consists of 499,635 instances, each of which represents a distinct data point, and 44 attributes that describe various aspects of each instance. At regular intervals of 0 days and 01:00:00, the data was gathered. The data was cleaned, normalized, and divided into training and testing sets before to analysis in order to assure its accuracy and dependability. In this thesis work, the variables that were utilized to train and evaluate machine learning models were carefully chosen based on their applicability to the research topics.

4.1.1 Deployment Scenario

The 1700 square feet, two-story Home is occupied by three people full-time. Eight rooms in all, including the basement, make up the house. The living room, bedroom, kitchen, and bathroom are located on the main level, and the second story houses two bedrooms and a bathroom. The residence is devoid of central air conditioning (A/C). Three window A/C units are used by the residents in the summer: one in the living room and one in each of the bedrooms upstairs. The heating system in the house burns natural gas. Other significant equipment includes a heat recovery ventilation (HRV) system, an electric dryer

Appliance ID	**Appliance Name**	**Appliance ID**	**Appliance Name**
A1	Generation	A2	House Panel
A3	Guest House Kitchen	A4	Basement
A5	Fresh Air Ventilation	A6	Studies
A7	Master Bedroom	A8	Dining Room Receptacles
A9	Guest House Bathroom	A10	Guest House Bedroom
A11	Workshop Receptacle Bath Heater	A12	Second Floor Bathroom
A13	Guest House Kitchen	A14	Second Floor Bathroom
A15	Ground Source Heat Pumps	A16	Photovoltaics
A17	Well Pump	A18	Range
A19	Panel Receptacles	A20	Washing Machine
A21	Refrigerator	A22	Microwave
A23	Dryer	A24	Net House Power Usage
A25	Garage Receptacles	A26	Garage Receptacles
A27	Shed Receptacles	A28	Shed Lights
A29	Panel Receptacles	A30	Garage PV
A31	Guest House Living Room	A32	Heat Circulator Pumps
A33	Domestic Hot Water Reserve	A34	Kitchen Island
A35	Radiant Heat Reserve Tank	A36	Basement Receptacles
A37	Kitchen Lighting	A38	Guest House Dining Room Receptacles
A39	Porch	A40	Veranda Lighting
A41	Hall Lighting	A42	Outside Lighting
A43	Lights		

Table 4.1: Different Circuit Names and their IDs in a House.

and washing machine, a dishwasher, a refrigerator, and a freezer. The house includes 35 wall switches, most of which control the lighting in the rooms and closets. Other switches also operate each bathroom's garbage disposal and exhaust fan. There are 26 separate circuits in the electrical panel.

Power data is gathered from the entire home and each circuit every second using sensors installed in the main panel. Installed devices broadcast on-off dim events for the switches through the powerline to a gateway server in place of 30 of the home's 35 wall switches. The remaining five switches could not be replaced for a variety of reasons, including the absence of neutral wires in the switch boxes for three basement switches, the garbage disposal's power exceeding the capacity of the programmable switches, and the lack of an exact replacement for one kitchen switch. Because the switches in the basement are on dedicated circuits, the garbage disposal is on a circuit with only the dishwasher (which has a very different power profile), and the kitchen switch is on a circuit for kitchen lights that has just one other instrumented load, the power usage can be determined from the uninstrumented switches using the circuit data. Our data collection is further aided by the electrical wiring of the house. Each circuit is assigned to one of three things: outlets (which are monitored at plug meters), lighting (which is monitored at wall switches), or specific major appliances (which are monitored at the main panel). Having the lighting on separate rate circuits makes it straightforward to correlate lighting events with power usage using the circuit data because our wall switches report on off-dim events rather than raw power.

4.2 Exploratory Data Analysis(EDA)

Exploratory Data Analysis (EDA) of 43 appliances' energy consumption trends in a home is crucial for understanding energy use patterns and guiding energy forecasting.

For accurate energy forecasting, it is essential to carefully examine historical data on energy usage and take into account external variables. To make these techniques more effective, data processing and analysis tools like numpy, scipy, and Data Science and ML Frameworks are frequently used in conjunction with them. Energy forecasting can optimize energy use, advance energy efficiency, and ultimately support sustainable energy practices by making use of such tools and methodologies. EDA can assist spot anomalies and outliers, detect relationships between appliance usage and environmental conditions, and optimizing energy use. For managing energy systems, advancing energy efficiency, and achieving sustainable energy use, EDA and energy forecasting are crucial tools.

4.2.1 Data Preprocessing

Effective data preprocessing is crucial for ensuring the accuracy and reliability of analysis and modelling results, especially when working with real-world data that may contain noise, redundancies, and discontinuities. The initial step typically involves noise reduction to minimize the impact of random variations that are not related to the underlying phenomenon being studied.

Outliers are another critical factor to consider when training machine learning algorithms, as incorrect or anomalous values can have a detrimental impact on the trained model's performance. Such outliers are often the result of measurement errors or system anomalies and should generally be removed from the data before proceeding to further analysis or modelling. However, in the datset used, we could not find outliers.

Missing values can also pose a significant challenge in data preprocessing. It is important to identify and address missing data, which can be either continuous or discontinuous. For instance, if the data is missing continuously in large chunks, it can be filled with data from the same time period in another year. If there is a missing segment, nearby values can be used to fill in the gap. There were some missing values which were filled using appropriate methods.

Overall, a rigorous data preprocessing approach is essential for generating reliable and accurate analysis and modelling results. We caefully addressed noise reduction, outliers, and missing data in the EDA process.

4.2.2 Re-sampling of Dataset

The dataset contains the values at every second, we resampled the data into hourly data which is a common demand of several industrial players in the market. Total of 8784 data samples of hourly data from 43 appliances is prepared.

An hourly time-series data of 43 appliances is presented in Figure 4.1, 4.2 and 4.3. It must be noted from these figures that there is good variations in patterns of data between several appliances.

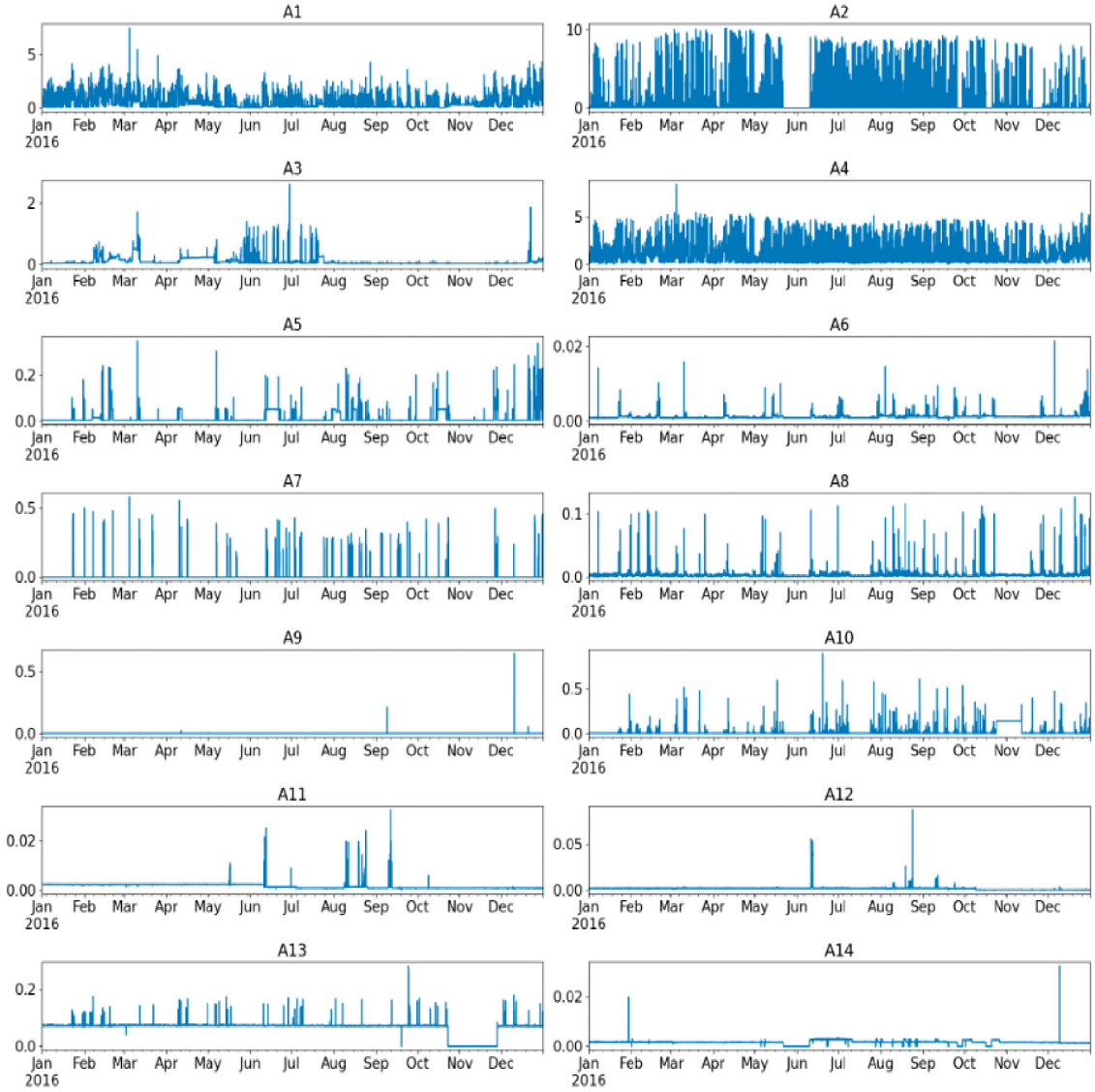


Figure 4.1: Hourly time-series data of appliances (A1-A14) is presented

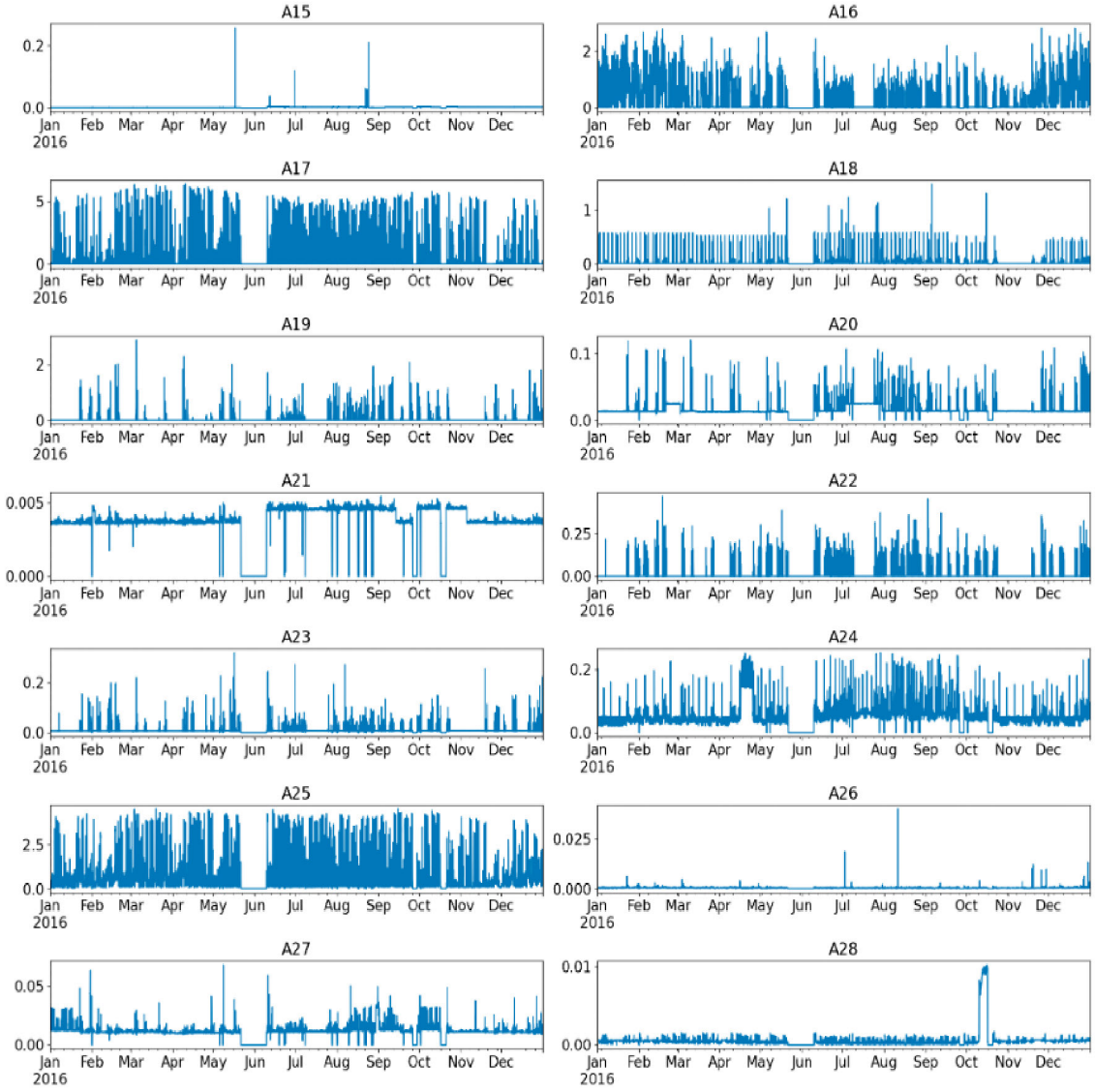


Figure 4.2: Hourly time-series data of appliances (A15-A28) is presented

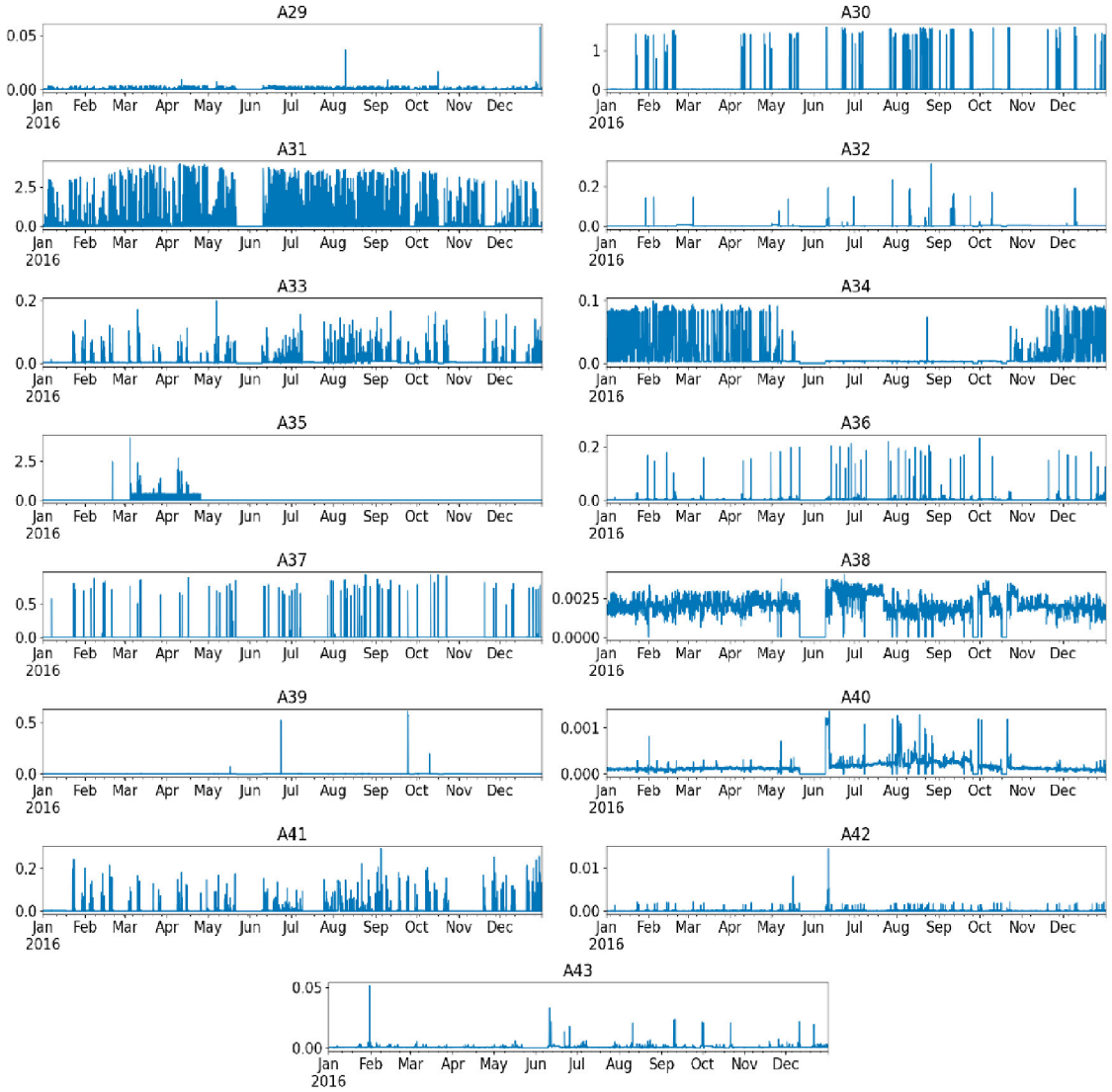


Figure 4.3: Hourly time-series data of appliances (A29-A43) is presented

The correlation matrix of all the appliances is also shown in the figure. 4.4.

4.3 Performance Evaluation of Static Time-Series Clustering

We performed clustering Algorithm 4 on the dataset described in the Section 4.1, and for this, we choose 5 clusters as a thumb rule (clusters are less than equal to the square root of the total number of time series).The obtained result of correlation matrix before performing the clustering algorithm is show in Figure 4.4 and result obtained after the clustering process are shown in Figure. 4.5. It shows 5 different clusters and their

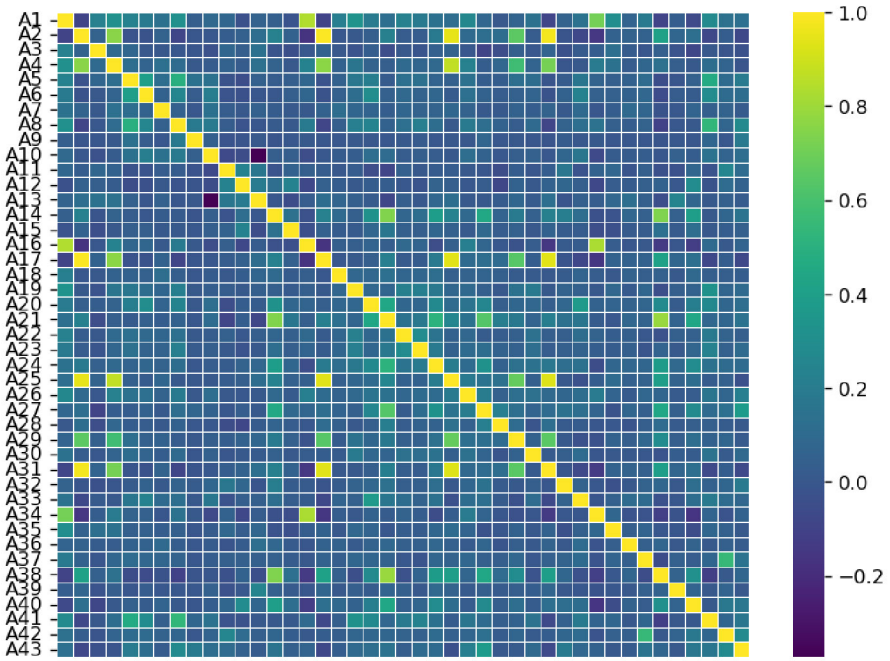


Figure 4.4: Correlation Matrix of 43 Appliances

associated appliance as cluster members. It is important to note that the clustered time-series within a specific cluster demonstrate comparable patterns. Moreover, the yellow color in the correlation matrix represents the high correlation between the appliances.

The relationship between various appliances within each cluster was also analyzed in addition to Static clustering of the time-series data using an Algorithm 4 with a group of 5 clusters. Figure. 4.6 displays the outcomes of this analysis. The k-shape clustered time series display a strong correlation coefficient, which suggests that the appliances within each cluster have similar usage patterns. This is demonstrated in the image. This is a desirable result because it shows that the clustering algorithm was effective in assembling appliances with comparable usage patterns. This implies that the clustering algorithm was successful in grouping similar appliances based on their energy consumption patterns.

4.3.1 Forecasting Analysis using Static Clustering Information

This section discusses the use of static time-series clustering for forecasting, which involves applying a clustering Algorithm 4 to a dataset and obtaining 5 clusters. The study revealed that time-series shapes within a given cluster exhibit similar patterns, and a strong correlation was observed between the various appliances in a cluster, particularly for time series with a k-shape cluster. Leveraging these commonalities within clusters provides an efficient and practical approach to forecasting the future values of appliances.

To demonstrate this, we performed energy consumption forecasting on various appliances for different days, selecting one appliance from each of the 5 clusters for analysis. In the following section, the impact of forecasting appliances will be discussed in detail.

After obtaining the clustering information obtained from k-shape algorithm, we performed forecasting of the energy consumption of a device with ID - A37 which is a

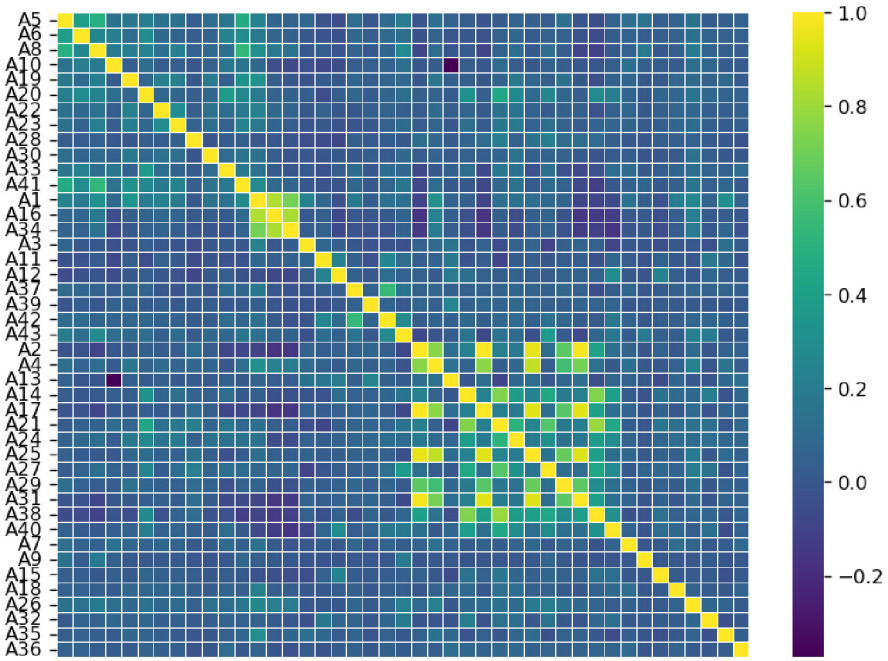


Figure 4.5: Correlation Matrix of 43 Appliances in the order of their Clusters

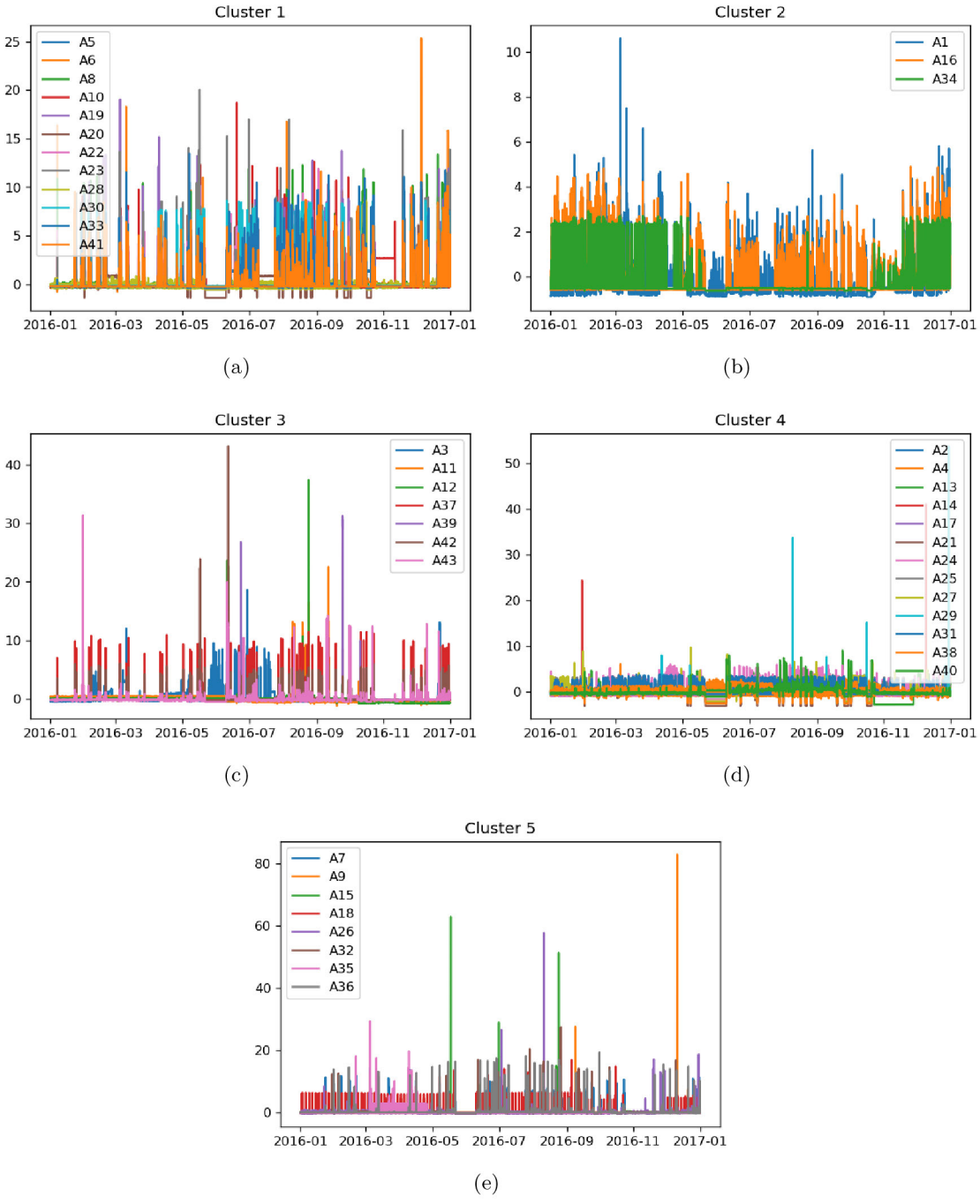
member of cluster-3. For forecasting we considered the extreme gradient boosting scheme [155]. In the experiment settings, we trained the forecasting model with yearly data and tested it on the last 10 days of the year. For computing the next-day energy consumption of A37, we fed an extreme gradient boosting scheme with historical data of A37, its time-lagged values and the time-series of other cluster members of cluster-3 which are A3, A11, A12, A39, A42, A43. Further, to evaluate the impact of clustering, we also fed the extreme gradient boosting scheme with only historical data of A37 and its time-lagged values, not the time series from other appliances. The performance of the extreme gradient boosting scheme without using the clustering information is shown in Figure 4.7, which shows the result before performing Clustering. The performance of the forecasting method utilizing clustering information is also presented in Figure 4.8 in terms of Root Mean Squared Error (RMSE) 4.1, Mean Squared Error (MSE) 4.2 and Sum of Squared Error (SSE) 4.3 which is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\text{ Predicted }_i - \text{ Actual }_i)^2}{N}} the \tag{4.1}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\text{Predicted}_i - \text{Actual}_i)^2 \tag{4.2}$$

$$SSE = \sum_{i=1}^{N}(\text{Predicted}_i - \text{Actual}_i)^2 \tag{4.3}$$

The values of these different metrics are given in Table 4.2. It must be noted that all these metrics are improved very significantly. This improved forecasting would result in better flexibility models at the smart homes level.



(a) (b) (c) (d) (e)

Figure 4.6: Static Time-series clustering of 43 appliances into 5 clusters. (a) Appliances group of Cluster 1. (b) Appliances group of Cluster 2. (c) Appliances group of Cluster 3. (d) Appliances group of Cluster 4. (e) Appliances group of cluster 5. using k-shape algorithm. The shape of different time series in a cluster shows strong similarity.

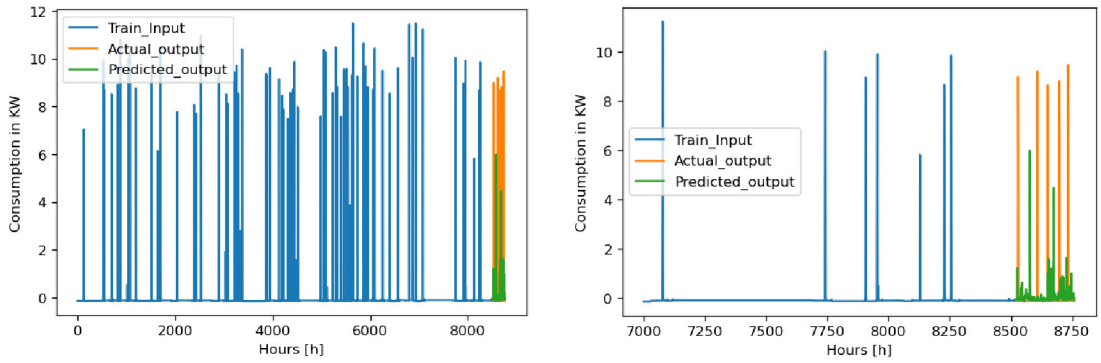


Figure 4.7: The Performance of (a) 10 days Forecasting of Appliance A37 without utilizing clustering. (b) shows a closer look.

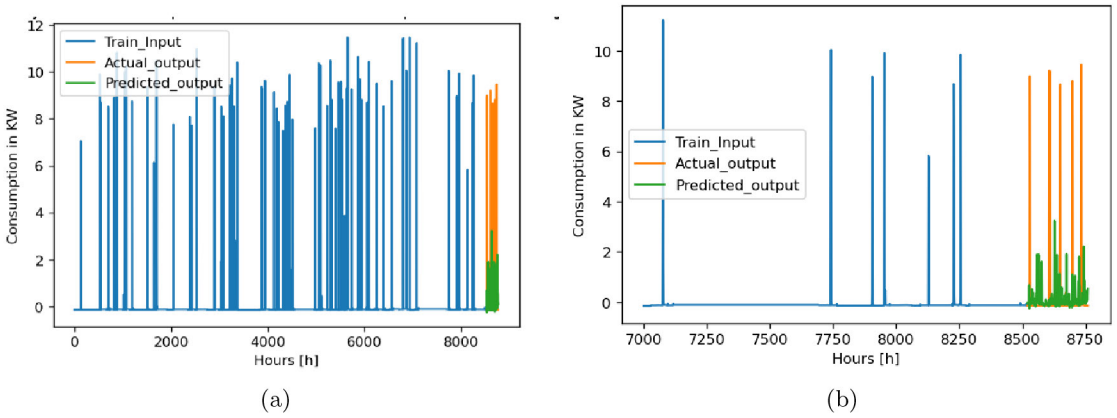
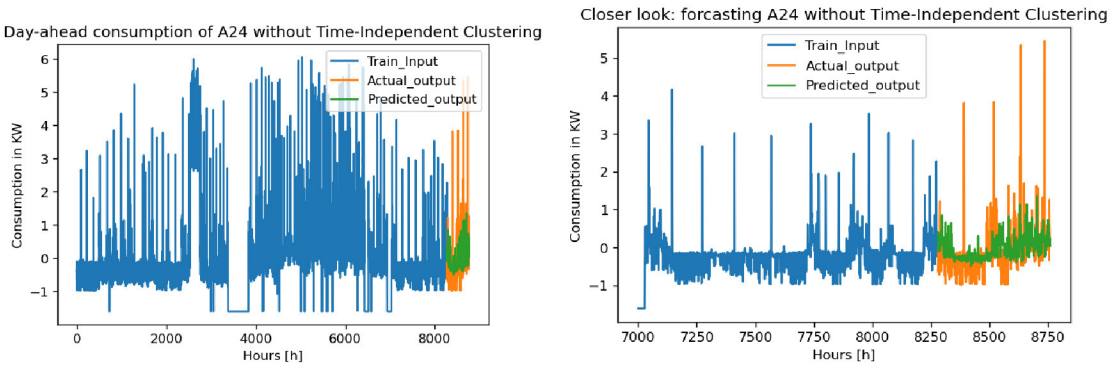


Figure 4.8: Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A37. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 10 days (240 hours).

Table 4.2: Forecasting Performance for Appliance A37 (For last 10 days) using Static Clustering

Metric	Without Clustering	With Static Clustering
$RMSE$	1.662	1.646
MSE	2.765	2.711
SSE	663.70	650.83



(a) (b)

Figure 4.9: The Performance of (a) 20 days Forecasting of Appliance A24 without utilizing clustering. (b) shows a closer look.

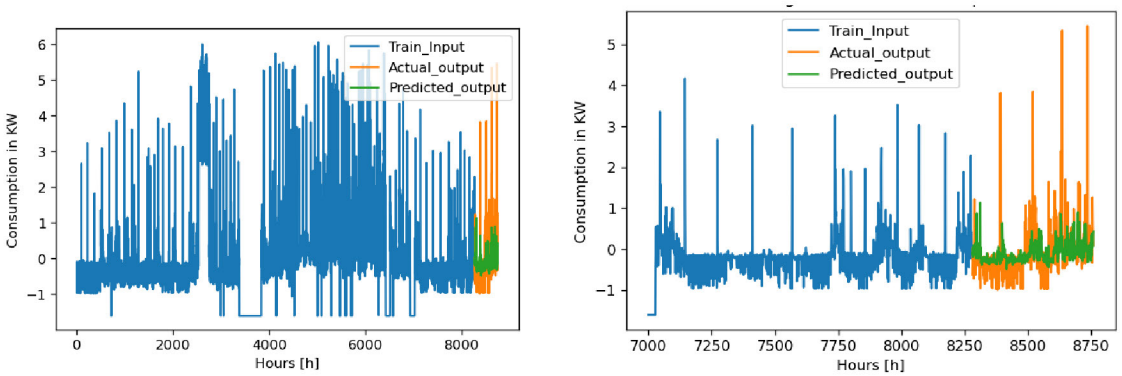
Table 4.3: Forecasting Performance for Appliance A24 (For last 20 days) using Static Clustering

Metric	Without Clustering	With Static Clustering
$RMSE$	0.6368	0.6251
MSE	0.4055	0.3907
SSE	194.663	187.568

To compute the next 20-day energy consumption of A24, we used an extreme gradient boosting (XGBoost) algorithm with historical data of A24, its time-lagged values, and the time-series of other cluster members from Cluster 4 (A2, A4, A13, A14, A17, A21, A25, A27, A29, A31, A38, and A40). Additionally, we evaluated the impact of clustering by running the XGBoost algorithm with only the historical data of A24 and its time-lagged values, without the time-series data from other appliances. Figure 4.9 shows the performance of the XGBoost algorithm without using clustering information, while the performance of the clustering-based forecasting method is presented in Figure 4.10 using Root Mean Squared Error (RMSE) 4.1, Mean Squared Error (MSE) 4.2, and Sum of Squared Error (SSE) 4.3 as evaluation metrics.

Table 4.3 presents the values of the different evaluation metrics for the forecasting methods. It is important to note that all of these metrics show significant improvement. The enhanced forecasting accuracy can lead to the development of more flexible models at the smart home level.

To predict the next-day energy consumption of A7, for the next 30 days, we utilized an extreme gradient boosting (XGBoost) algorithm with historical data of A7, its time-lagged values, and the time-series of other appliances in Cluster 5 (A9, A15, A18, A26, A32, A35 and A36. Furthermore, we evaluated the impact of clustering by running the XGBoost algorithm only with the historical data of A7 and its time-lagged values, without incorporating time-series data from other appliances. Figure 4.11 depicts the performance of the XGBoost algorithm without clustering information, while Figure 4.12 presents the performance of the clustering-based forecasting method, using Root Mean Squared Error



(a) (b)

Figure 4.10: Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A24. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours).

Table 4.4: Forecasting Performance for Appliance A7 (For last 30 days) using Static Clustering

Metric	Without Clustering	With Static Clustering
$RMSE$	1.3287	1.3154
MSE	1.7654	1.7305
SSE	1694.837	1245.980

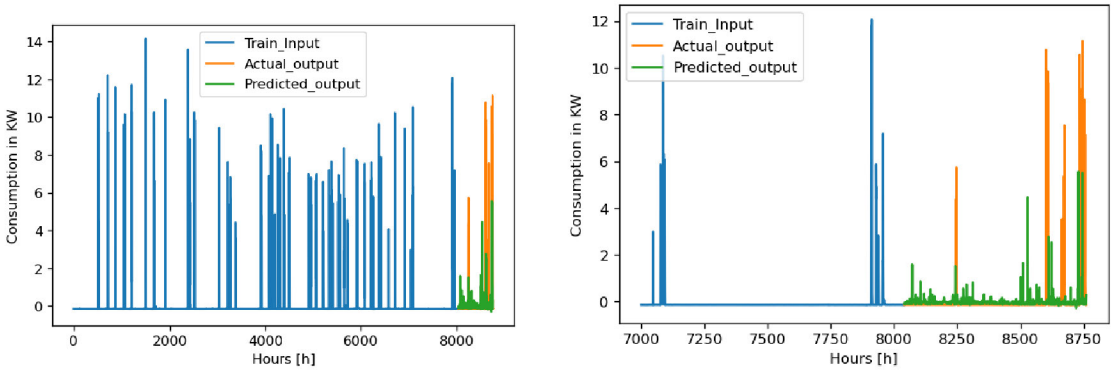
(RMSE) 4.1, Mean Squared Error (MSE) 4.2, and Sum of Squared Error (SSE) 4.3 as evaluation metrics.

"Table 4.4 summarizes the performance of the forecasting methods with different evaluation metrics. It is noteworthy that all of the evaluation metrics show a significant improvement. This improved forecasting accuracy can lead to the development of more flexible models at the smart home level."

For forecasting the energy consumption of appliance A16 for the next 35 days, we applied an extreme gradient boosting (XGBoost) algorithm. We utilized historical data of A16, its time-lagged values, and the time-series of other appliances belonging to Cluster 2, which include A1 and A34. To evaluate the impact of clustering, we also ran the XGBoost algorithm using only the historical data of A16 and its time-lagged values, without considering the time-series data from other appliances. The performance of the XGBoost algorithm without using clustering information is illustrated in Figure 4.13, while the performance of the clustering-based forecasting method is presented in Figure 4.14, using Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Sum of Squared Error (SSE) as evaluation metrics. The improved accuracy of energy consumption forecasting can facilitate the development of more flexible models at the smart home level.

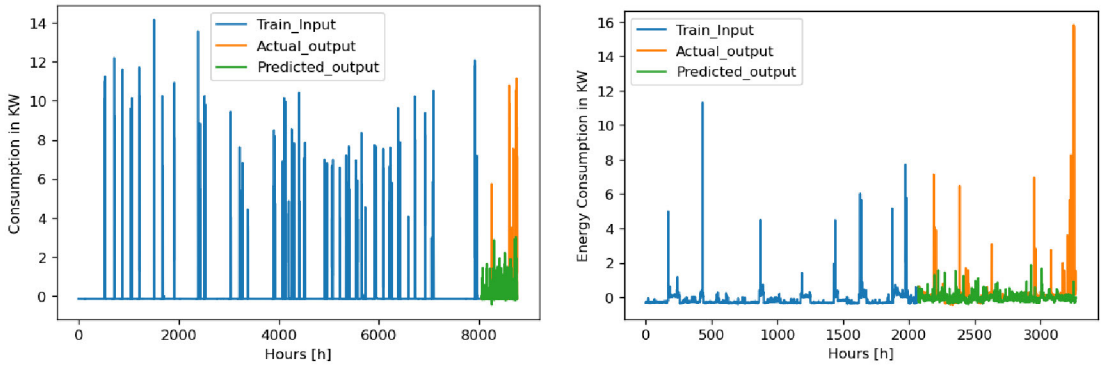
Table 4.6 presents a summary of the forecasting methods' performance of Appliance A16 using various evaluation metrics.

To forecast the energy consumption of appliance A6 for the next 50 days, we employed an extreme gradient boosting (XGBoost) algorithm that utilized both the historical data



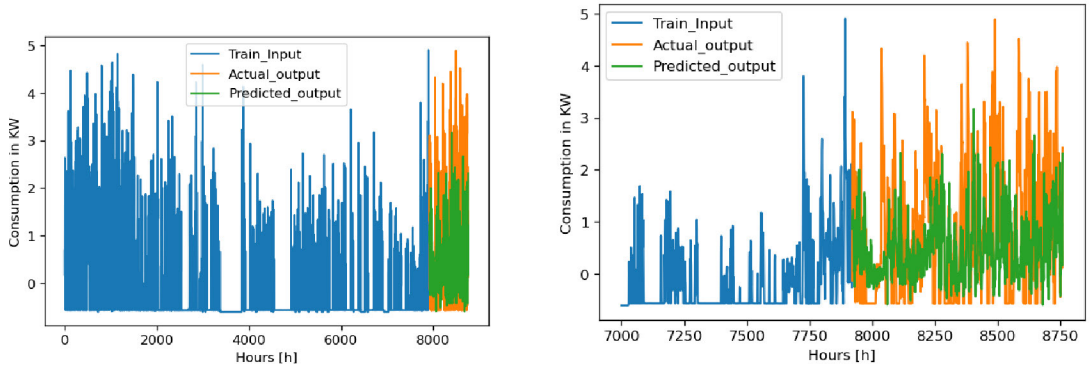
(a) (b)

Figure 4.11: The Performance of 30 days Forecasting of Appliance A7 without utilizing clustering.



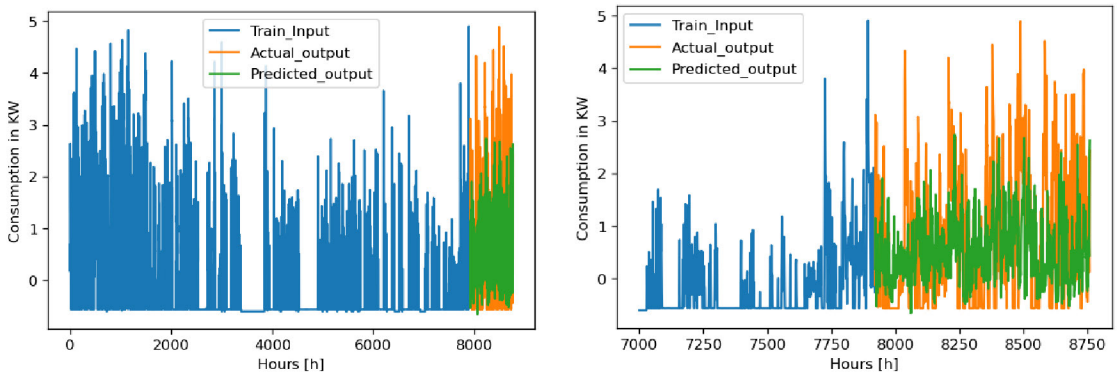
(a) (b)

Figure 4.12: Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A7. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 30 days (720 hours).



(a) (b)

Figure 4.13: The Performance of (a) 35 days Forecasting of Appliance A16 without utilizing clustering. (b) shows a closer look.



(a) (b)

Figure 4.14: Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A16. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 35 days (840 hours).

Table 4.5: Forecasting Performance for Appliance A16 (For last 35 days) using Static Clustering

Metric	Without Clustering	With Static Clustering
$RMSE$	1.2765	1.2688
MSE	1.6296	1.6100
SSE	1368.914	1352.426

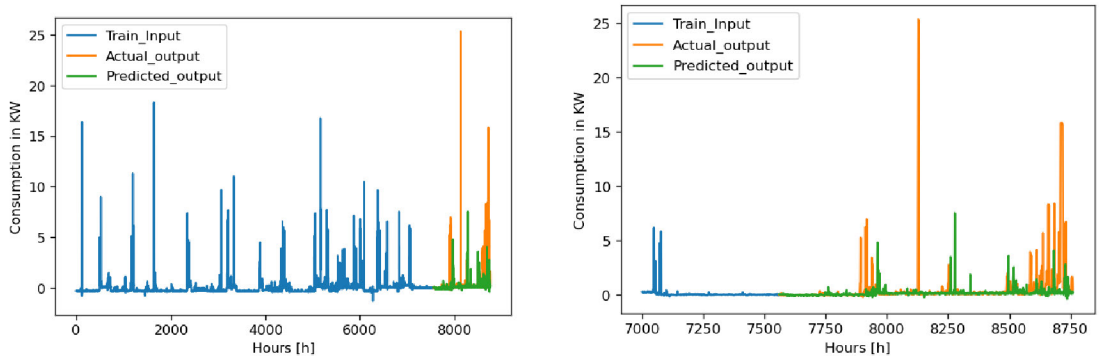
of A6 and its time-lagged values, as well as the time-series of other appliances in Cluster 1, which include A5, A8, A10, A19, A20, A22, A23, A28, A30, A33, and A41. To assess the impact of clustering on the forecasting performance, we also ran the XGBoost algorithm using solely the historical data of A6 and its time-lagged values, without considering the time-series data of other appliances. The forecasting performance of the XGBoost algorithm without using clustering information is illustrated in Figure 4.15, while the forecasting performance of the clustering-based method is presented in Figure 4.16, with Root Mean Squared Error (RMSE), Mean Squared Error (MSE), and Sum of Squared Error (SSE) used as evaluation metrics

Table 4.6 summarizes the performance of different forecasting methods for Appliance A16 based on various evaluation metrics.

Thus, time-dependent (dynamic), and online clustering of time-series data would be interesting and crucial to explore. Also, the accuracy can be improved further by eliminating outliers in time series.

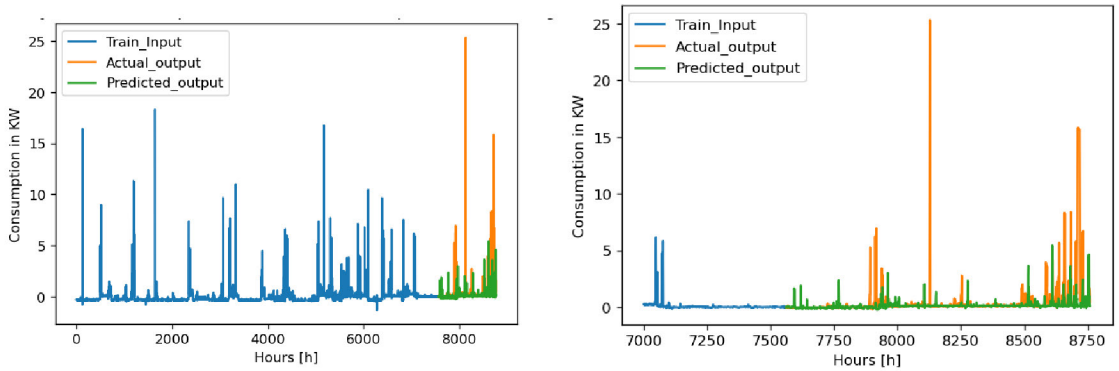
Table 4.6: Forecasting Performance for Appliance A6 (For last 50 days) using Static Clustering

Metric	Without Clustering	With Static Clustering
$RMSE$	1.9386	1.9216
MSE	3.7583	3.6926
SSE	4510.033	4431.143



(a) (b)

Figure 4.15: The Performance of (a) 50 days Forecasting of Appliance A6 without utilizing clustering. (b) shows a closer look.



(a) (b)

Figure 4.16: Performance evaluation of (a) proposed Static clustering for forecasting next-day consumption of Appliance A6. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 50 days (1200 hours).

4.4 Performance Evaluation of Dynamic Time-Series Clustering

In this study, we employed Algorithm 5 to cluster the dataset described in Section 4.1. To enable dynamic clustering, we segmented a dataset of 43 smart home appliances into day and night periods and applied separate clustering algorithms for each period. To achieve this, we defined the day period as the interval between 6:00 in the morning until 21:00, while the night period spanned from 21:00 to 5:59.

As highlighted in an earlier chapter, we partitioned the entire dataset based on these time intervals, resulting in two sub-datasets: one for the day-time period and another for the night-time period. Subsequently, we utilized Algorithm 5 to perform clustering on both datasets, resulting in 5 clusters for each dataset.

This approach enabled us to capture the dynamic changes in appliance usage patterns during the day and night periods separately, providing more accurate and granular insights into energy consumption behaviour. By clustering each sub-dataset independently, we were able to identify patterns and similarities unique to each period, improving the overall clustering accuracy and reducing any potential noise and redundancies that may arise from clustering the entire dataset as a single entity.

Overall, our dynamic clustering approach has demonstrated its effectiveness in capturing the time-dependent patterns in energy consumption behaviour, offering new avenues for energy-efficient smart home management.

4.4.1 Day-Time Clustering Analysis

The two figures show correlation matrices for a dataset consisting of 43 smart home appliances, before and after performing dynamic (day-time) clustering.

Figure 4.17 displays the correlation matrix of the 43 appliances before performing clustering. This matrix shows the pairwise correlations between each appliance in the dataset, with correlations ranging from 0 to 1. The diagonal of the matrix contains the correlations of each appliance with itself, which is always 1. As shown in the figure, the matrix appears to be quite complex, with multiple appliances exhibiting strong correlations with each other.

Figure 4.18 shows the correlation matrix of the 43 appliances after dynamic clustering has been applied. In this figure, the appliances have been reordered based on their assigned clusters, resulting in a clearer and more organized correlation matrix. This clustering has resulted in the grouping of similar appliances, which are shown as clusters along the diagonal. The figure also displays the correlation values between appliances, but now they are easier to interpret, as we can see the relationships between the clusters and the appliances within them.

By grouping similar appliances together, we can gain a more intuitive understanding of how they relate to each other and how their energy consumption patterns change over time thus utilizing it for improving the forecasting accuracy.

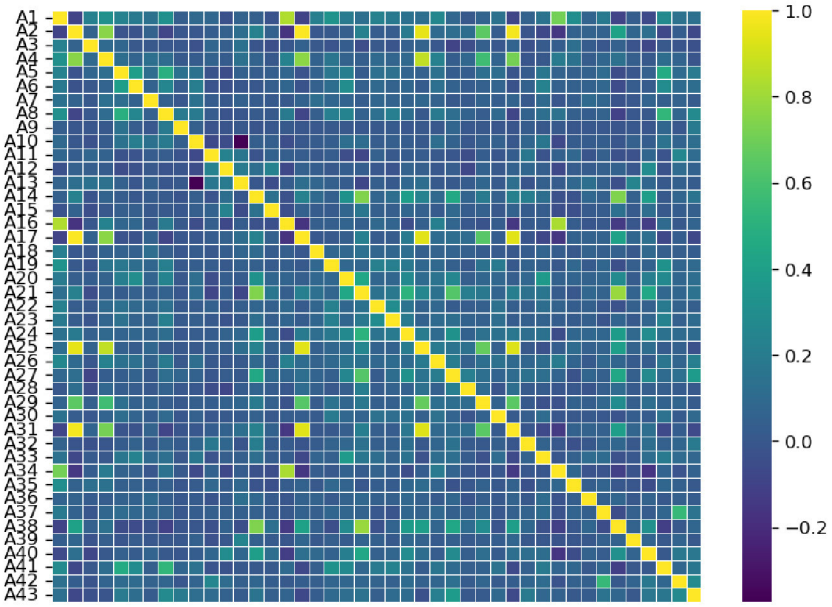


Figure 4.17: Correlation Matrix of 43 Appliances before Performing Dynamic (Day-Time) Clustering

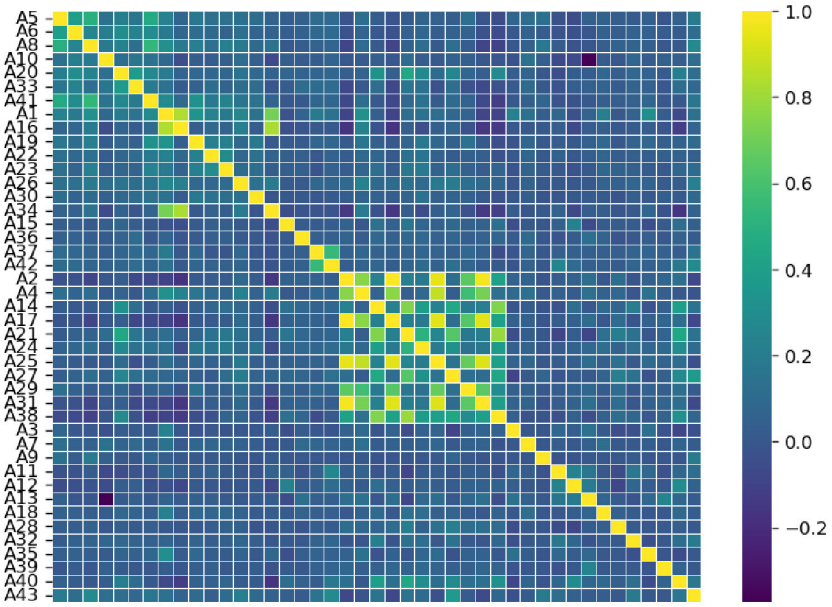


Figure 4.18: Correlation Matrix of 43 Appliances in the order of their Clusters

Figure 4.19 shows the results of dynamic time-series clustering of 43 appliances into 5 clusters for the day-time period. Each subfigure corresponds to one cluster and shows the appliances that belong to that cluster. The clustering is done using the k-shape algorithm, which clusters the time-series data based on their shape similarity. The appliances within each cluster exhibit strong similarities in their time-series shapes, which is evident from

the plots of each appliance's energy consumption over time. These clusters can be useful for forecasting the future energy consumption of these appliances since appliances within the same cluster are likely to exhibit similar energy consumption patterns.

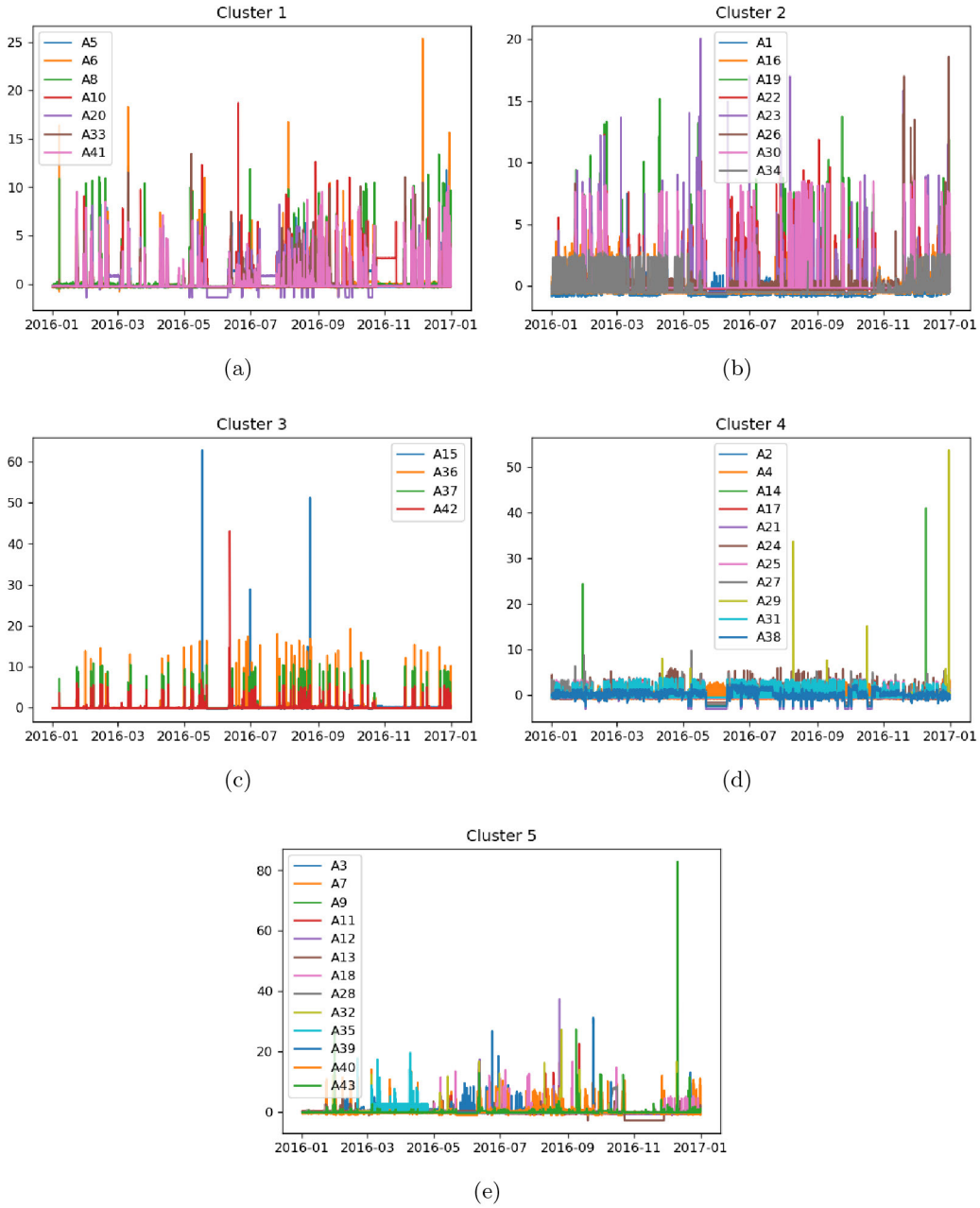


Figure 4.19: Dynamic (Day-Time) Time-series clustering of 43 appliances into 5 clusters. Each cluster shows its respective cluster members with similarities in their patterns.

4.4.2 Forecasting Analysis using Day-Time Clustering

Figure 4.20 shows the performance evaluation of Dynamic (Day-time) clustering for forecasting the next-day consumption of Appliance A37. The cluster member of A37 in this scenario is A15, A36, and A42. And, we performed forecasting using XGBoost. The figure presents two subfigures: (a) shows the comparison between the actual and predicted values of the appliance consumption for the last 10 days (240 hours), while (b) presents a closer look at the same comparison for the last two days.

Table 4.7 presents the forecasting performance metrics for Appliance A37 using Dynamic Clustering (Day-Time). The table shows the comparison of metrics. The results suggest that dynamic clustering for day-time data provides slightly worse performance compared to the case without clustering, mainly because the day-time consumption generally has more randomness in patterns. However, the difference is not significant.

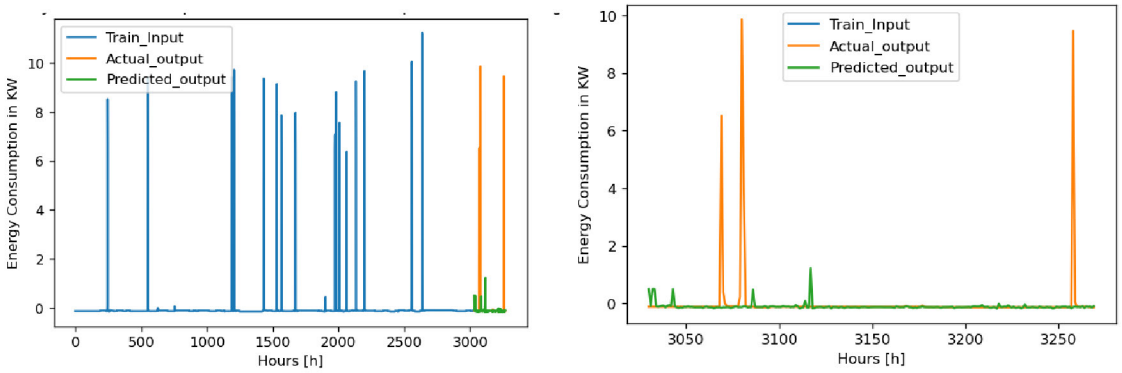
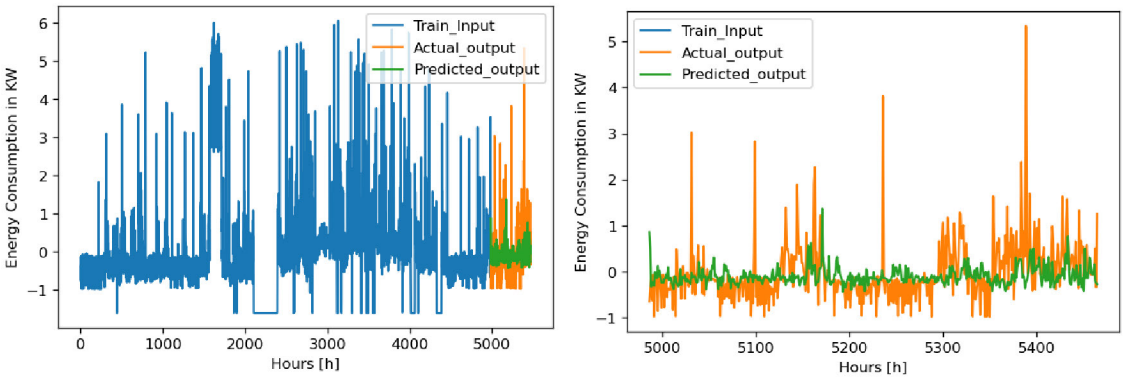


Figure 4.20: Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A37. (b) a closer look. The forecasting experiments evaluate the performance for the last 10 days (240 hours).

Table 4.7: Forecasting Performance for Appliance A37 using Dynamic Clustering (Day-Time)

Metric	Without Day-time Clustering	With Day-Time Clustering
$RMSE$	1.426	1.5374
MSE	2.034	2.3638
SSE	488.37	567.313

Figure 4.21 and Table 4.8 show the results of Appliance A24. Figure 4.22 and Table 4.9 show the results of Appliance A7. Figure 4.23 and Table 4.10 show the results of Appliance A16. Figure 4.24 and Table 4.11 show the results of Appliance A6. The cluster member information of these appliances is provided in Figure 4.19.

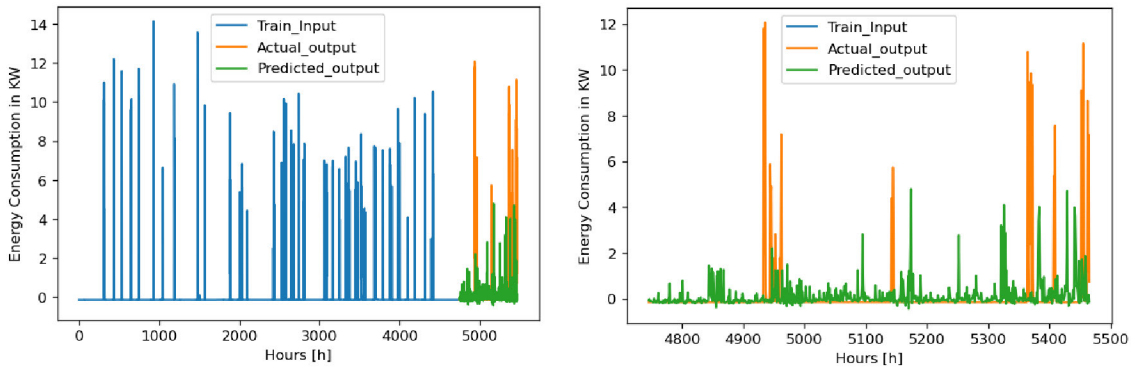


(a) (b)

Figure 4.21: Performance evaluation of (a) proposed Dynamic (Day-Time) clustering for forecasting next- day consumption of Appliance A24. (b) Shows a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours).

Table 4.8: Forecasting Performance for Appliance A24 using Dynamic Clustering (Day-Time)

Metric	Without Day-time Clustering	With Day-Time Clustering
$RMSE$	0.6956	0.6427
MSE	0.4839	0.4131
SSE	232.275	198.297

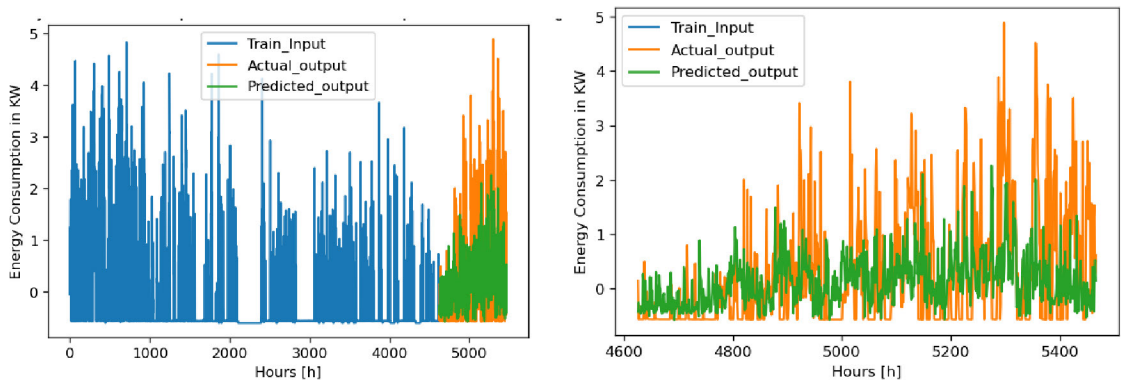


(a) (b)

Figure 4.22: Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A7. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (720 hours).

Table 4.9: Forecasting Performance for Appliance A7 using Dynamic Clustering (Day-Time)

Metric	Without Day-time Clustering	With Day-Time Clustering
$RMSE$	1.5047	1.5400
MSE	2.2642	2.3716
SSE	1630.253	1707.574

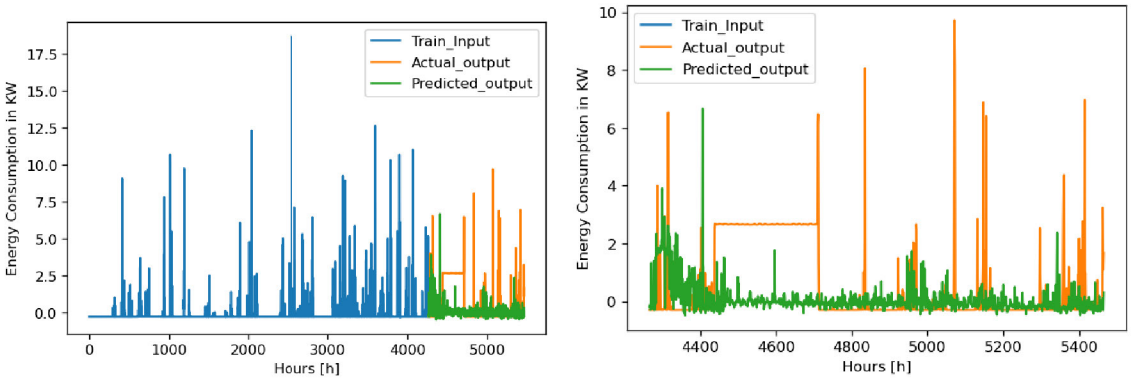


(a) (b)

Figure 4.23: Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A16. (b) a closer look.. The forecasting experiments evaluate the performance for last 35 days (840 hours).

Table 4.10: Forecasting Performance for Appliance A16 using Dynamic Clustering (Day-Time)

Metric	Without Day-time Clustering	With Day-Time Clustering
$RMSE$	1.0746	1.0606
MSE	1.1549	1.1249
SSE	970.164	944.926



(a) (b)

Figure 4.24: Performance evaluation of (a) Dynamic (Day-time) clustering for forecasting next-day consumption of Appliance A6. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (1200 hours).

Table 4.11: Forecasting Performance for Appliance A6 using Dynamic Clustering (Day-Time)

Metric	Without Day-time Clustering	With Day-Time Clustering
$RMSE$	1.4694	1.5839
MSE	2.1594	2.5088
SSE	2591.314	3010.615

4.4.3 Night-Time Clustering Analysis

Figure 4.25 shows the correlation matrix of 43 appliances before performing dynamic (night-time) clustering. In this figure, the appliances are not ordered based on their correlation with each other.

Figure 4.26 presents a correlation matrix of appliances but in the order of their clusters obtained using night-time clustering. The clustering has organized the matrix in a way that appliances that are highly correlated with each other are grouped together, resulting in a pattern of distinct clusters.

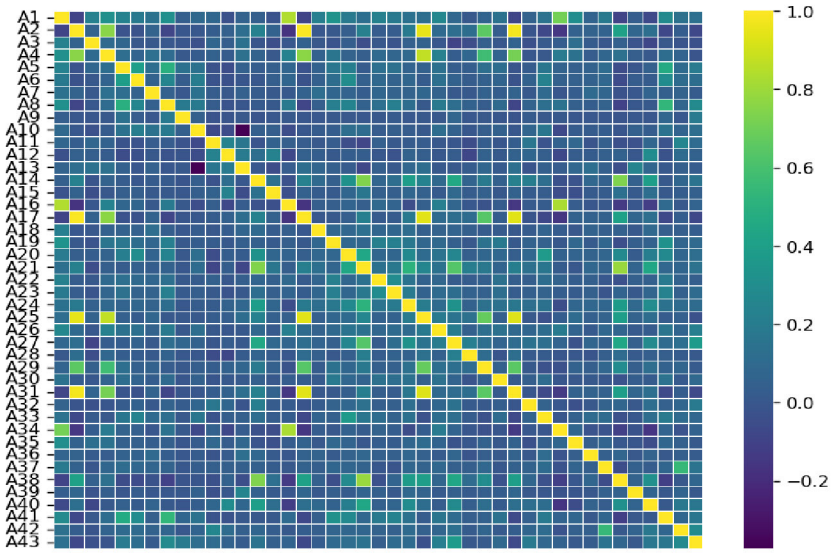


Figure 4.25: Correlation Matrix of 43 Appliances before Dynamic (Night-Time) Clustering

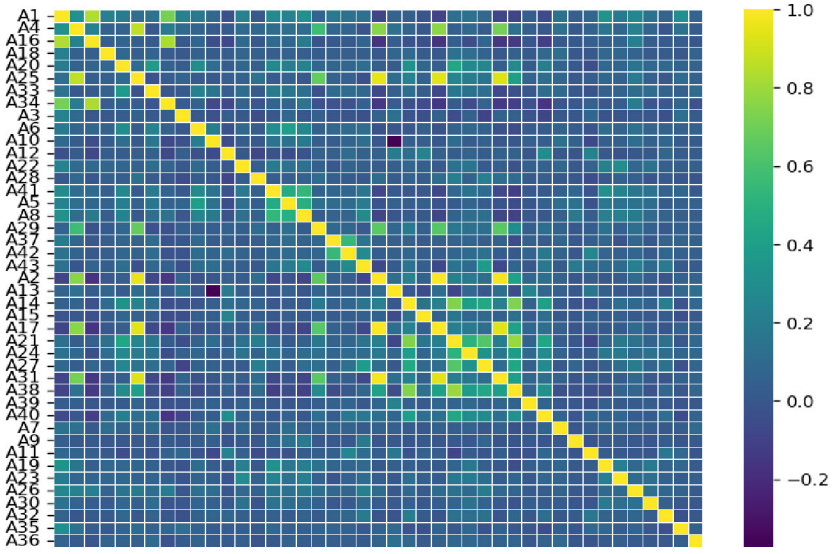


Figure 4.26: Correlation Matrix of 43 Appliances in the order of their Clusters obtained using night-time clustering

These figures show the results of a dynamic (night-time) time-series clustering of 43 appliances into 5 clusters using the Algorithm 9. Figure 4.27 consists of five subfigures, each showing the appliances' group of a specific cluster. The x-axis represents time, and the y-axis represents the power consumption of each appliance. Each line represents the power consumption time-series of an individual appliance. The shape of different time-series shows strong similarity within each cluster. These figures provide insights into the usage patterns and behaviours of different appliances and can be useful for various applications.

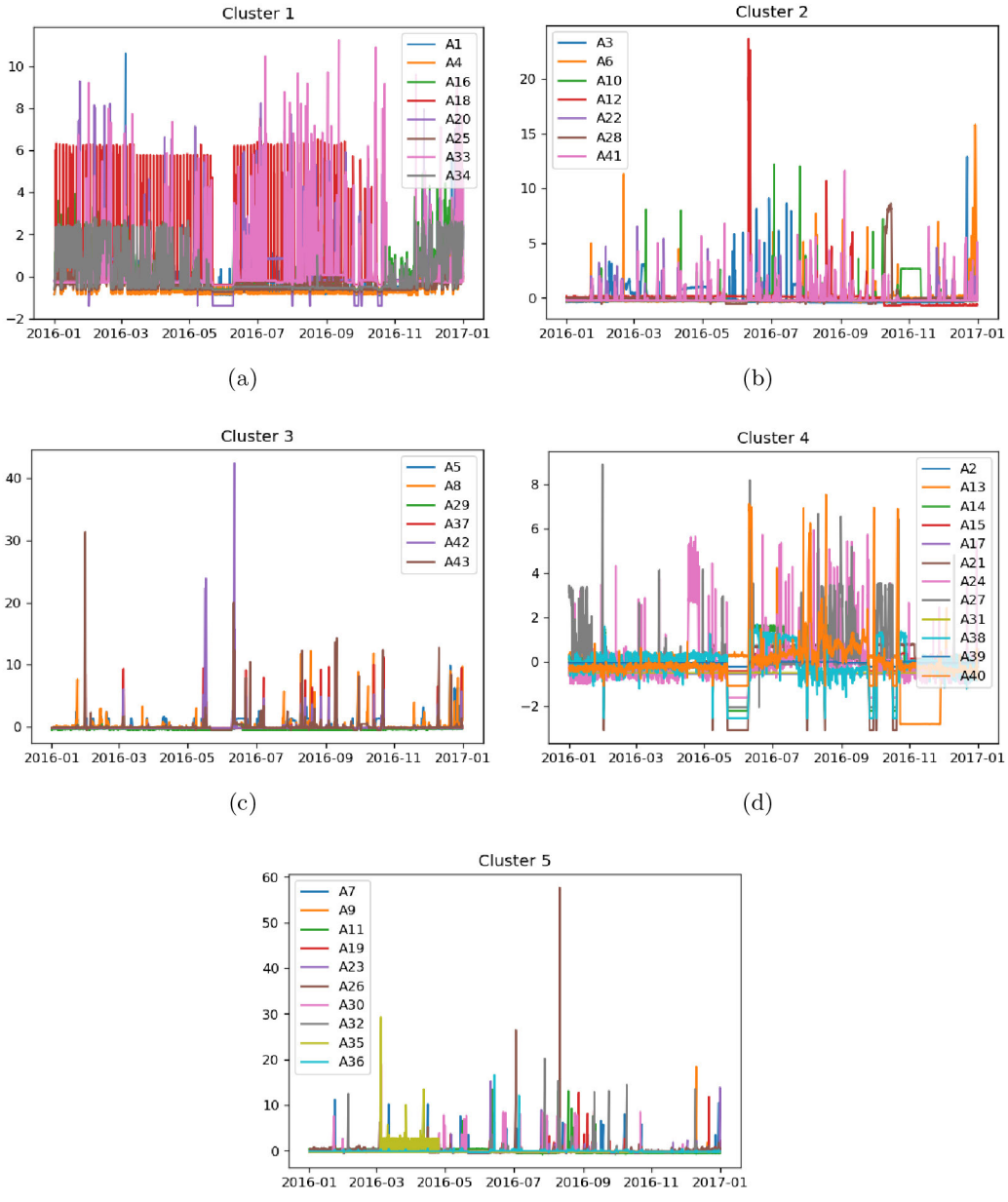


Figure 4.27: Dynamic (Night-Time) Time-series clustering of 43 appliances into 5 clusters. Each cluster shows its respective cluster members with similarities in their patterns.

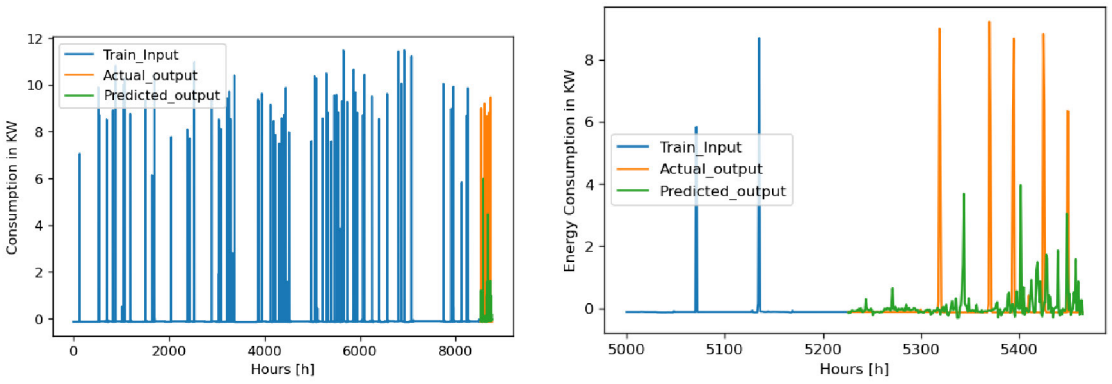
4.4.4 Forecasting Analysis using Night-time Clustering

This section presents the forecasting of different appliances using the clustering information presented in the previous section.

The figure and table show the performance evaluation of dynamic (night-time) clustering for forecasting the next-day 10 days consumption of Appliance A37. we employed an extreme gradient boosting (XGBoost) algorithm that utilized both the historical data of A37 and its time-lagged values, as well as the time-series of other appliances in Cluster 3,in night time energy forecasting which include A5, A8, A29, A42, and A43. To assess the impact of clustering on the night-time forecasting performance, we also ran the XGBoost algorithm using extensively the historical data of A37 and its time-lagged values, without considering the time-series data of other appliances. By using Root Mean Squared Error (RMSE) 4.1, Mean Squared Error (MSE) 4.2, and Sum of Squared Error (SSE) 4.3 as evaluation metrics as mentioned in the earlier sections.

Figure 4.28(a) shows the comparison of the actual consumption of A37 with the predicted consumption using night-time clustering. The orange line represents the actual consumption, while the green line represents the predicted consumption using night-time clustering. Figure 4.28(b) shows a closer look at the performance of the predicted consumption using night-time clustering.

Table 4.12 summarizes the performance metrics for forecasting the next-day consumption of Appliance A37 with and without night-time clustering. The performance metrics evaluated are root mean square error ($RMSE$), mean square error (MSE), and sum of squared error (SSE). As can be seen, the performance of the forecasting model with night-time clustering outperforms the model without night-time clustering in all three metrics, indicating that dynamic clustering can help improve the accuracy of the forecasting model.



(a) (b)

Figure 4.28: Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A37. (b) a closer look. The forecasting experiments evaluate the performance for last 10 days (240 hours).

Table 4.13: Forecasting Performance for Appliance A24 using Dynamic Clustering (Night-Time)

Metric	Without Night-time Clustering	With Night-time Clustering
RMSE	0.5490	0.5225
MSE	0.3014	0.2730
SSE	144.689	131.042

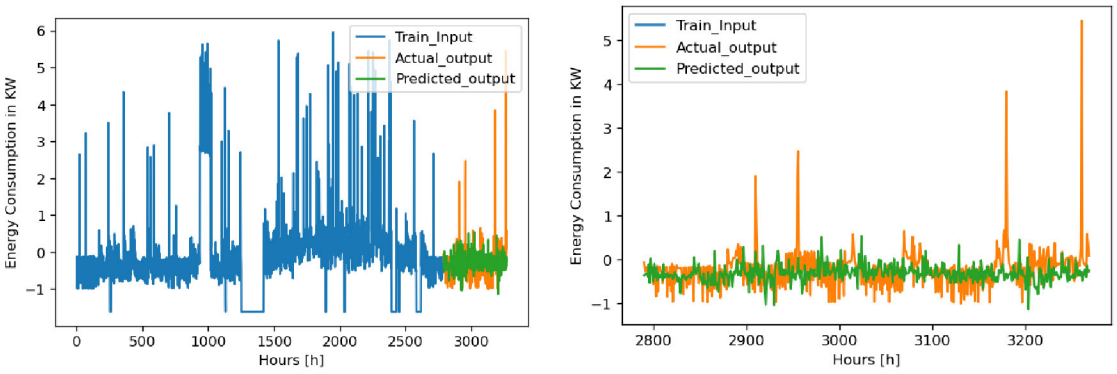
Table 4.14: Forecasting Performance for Appliance A7 using Dynamic Clustering (Night-Time)

Metric	Without Night-time Clustering	With Night-time Clustering
RMSE	0.6848	0.6766
MSE	0.4690	0.4578
SSE	337.697	329.6221

Table 4.12: Forecasting Performance for Appliance A37 using Dynamic Clustering (Night-Time)

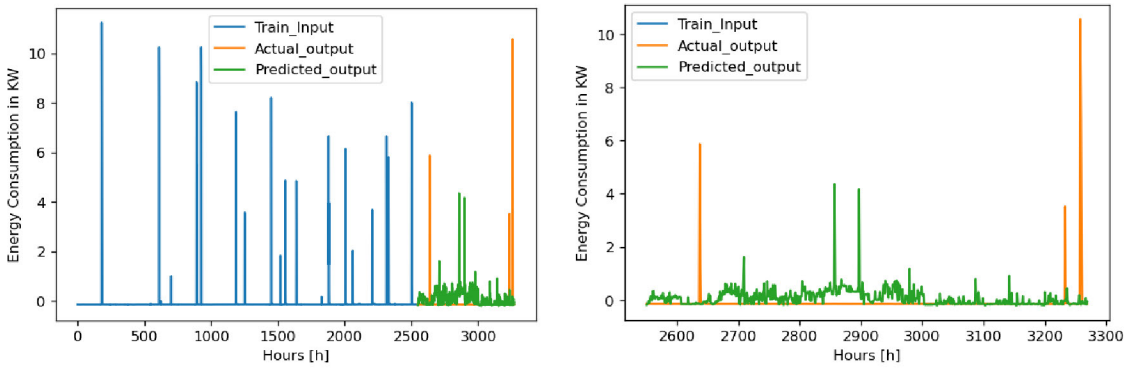
Metric	Without Night-time Clustering	With Night-Time Clustering
RMSE	1.317	1.0361
MSE	1.735	1.0735
SSE	416.625	257.6555

Figure 4.29 and Table 4.13 show the results of Appliance A24. Figure 4.30 and Table 4.14 show the results of Appliance A7. Figure 4.31 and Table 4.15 show the results of Appliance A16. Figure 4.32 and Table 4.16 show the results of Appliance A6. The cluster member information of these appliances is provided in Figure 4.27.



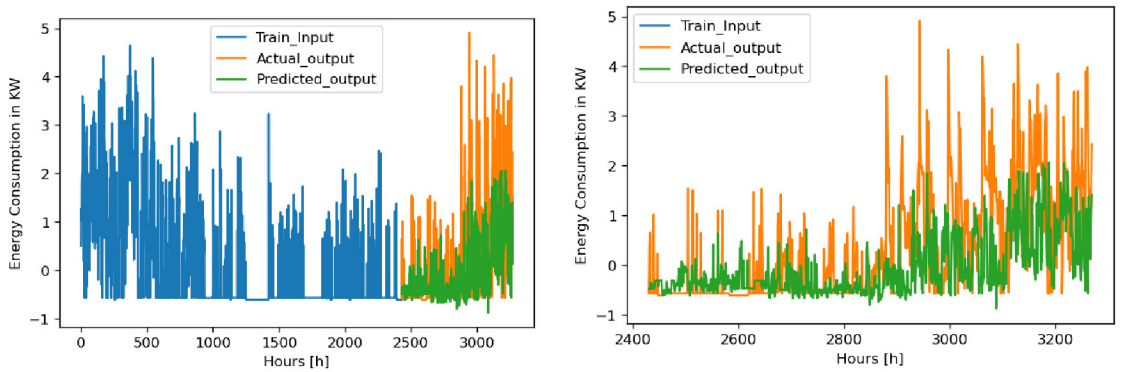
(a) (b)

Figure 4.29: Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A24. (b) a closer look. The forecasting experiments evaluate the performance for the last 20 days (480 hours).



(a) (b)

Figure 4.30: Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A7. (b) a closer look. The forecasting experiments evaluate the performance for last 30 days (720 hours).

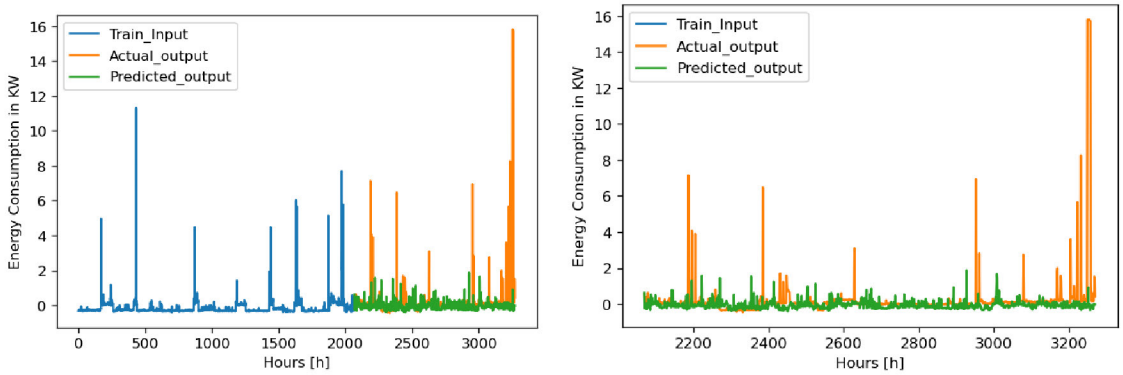


(a) (b)

Figure 4.31: Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A16. (b) a closer look. The forecasting experiments evaluate the performance for last 35 days (840 hours).

Table 4.15: Forecasting Performance for Appliance A16 using Dynamic Clustering (Night-Time)

Metric	Without Night-time Clustering	With Night-time Clustering
$RMSE$	0.9998	1.0968
MSE	0.9996	1.2030
SSE	839.671	1010.573



(a) (b)

Figure 4.32: Performance evaluation of (a) Dynamic (Night-time) clustering for forecasting next-day consumption of Appliance A6. (b) a closer look. The forecasting experiments evaluate the performance for last 50 days (1200 hours).

Table 4.16: Forecasting Performance for Appliance A6 using Dynamic Clustering (Night-Time)

Metric	Without Night-time Clustering	With Night-time Clustering
$RMSE$	1.5038	1.5022
MSE	2.2616	2.2568
SSE	2713.970	2708.273

4.5 Comparative Analysis of Time-Series Clustering

Table 4.17 and Figure 4.33 present the comparison of the forecasting performance of Appliance A37 for 10-day forecasting of A37 under different clustering methods using three different metrics, namely Root Mean Squared Error ($RMSE$), Mean Squared Error (MSE), and Sum of Squared Errors (SSE).

Both static and dynamic clustering methods result in lower RMSE, MSE, and SSE than without clustering. Dynamic clustering performs better than static clustering, with the lowest values for RMSE, MSE, and SSE. The improvements in accuracy between the different methods are relatively small but still important for some applications. Overall, these results suggest that dynamic clustering is the most effective method for improving the accuracy of the A37 forecasting, based on the metrics analyzed in this table. Additionally, it is worth noting that the improvements in accuracy between the different methods are relatively small, particularly in terms of RMSE and MSE. However, even small improvements in accuracy can be important for some applications, and the results here suggest that dynamic clustering is a promising approach for improving time-series forecasting.

Table 4.18: Comparative Analysis of A24 using Different Time-Series Clustering Methods

Metric	Without Clustering	With Static Clustering	With Dynamic Clustering		
			Day	Night	Average
RMSE	0.6368	0.6251	0.6427	0.5225	0.5826
MSE	0.4055	0.3907	0.4131	0.2730	0.3430
SSE	194.663	187.568	198.297	131.042	164.66

Table 4.17: Comparative Analysis of A37 using Different Time-Series Clustering Methods

Metric	Without Clustering	With Static Clustering	With Dynamic Clustering		
			Day	Night	Average
RMSE	1.6629	1.6467	1.5374	1.0361	1.2867
MSE	2.7654	2.7117	2.3638	1.07356	1.7186
SSE	663.705	650.830	567.313	257.655	412.484

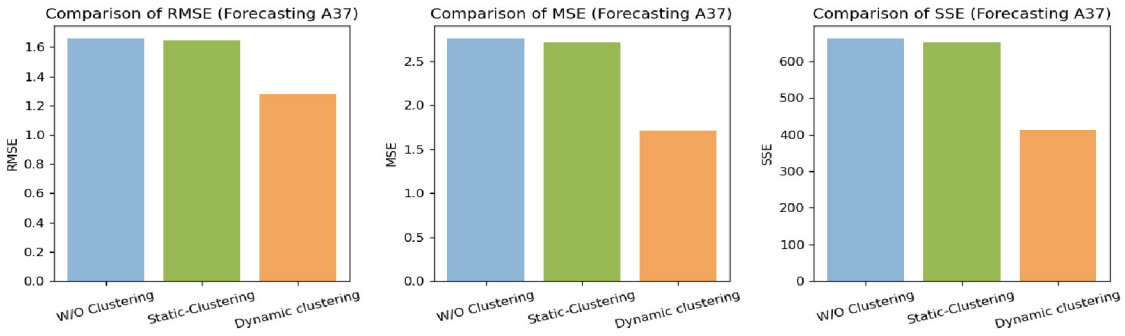


Figure 4.33: Comparison of $RMSE$, MSE and SSE (forecasting A37, for 10 days) for Different Clustering Methods

The RMSE value for forecasting Appliance A24 for 20 days without clustering is 0.6368, while the values for A24 with static clustering and dynamic clustering are 0.6251 and 0.5826, respectively. This indicates that both static and dynamic clustering methods perform better than those without clustering, with dynamic clustering producing the lowest RMSE value. Similar trends are observed for MSE and SSE.

Overall, these results suggest that dynamic clustering is the most effective method for improving the accuracy of the appliance A24 forecasting model, based on the metrics analyzed in Table 4.18. These results are also presented in Figure 4.34.

The comparison of other appliances A7 (for 30 days), A16 (for 35 days) and A6 (for 50 days) are presented in Table 4.19, 4.20 and 4.21 respectively. And, we observed performance trends similar to the earlier appliances. These results are also presented in Figure 4.35, 4.36 and 4.37 respectively.

Table 4.21: Comparative Analysis of A6 using Different Time-Series Clustering Methods

Metric	Without Clustering	With Static Clustering	With Dynamic Clustering		
			Day	Night	Average
RMSE	1.9386	1.9216	1.5839	1.5022	1.5430
MSE	3.7583	3.6926	2.5088	2.2568	2.3828
SSE	4510.033	4431.143	3010.615	2708.273	2,859.444

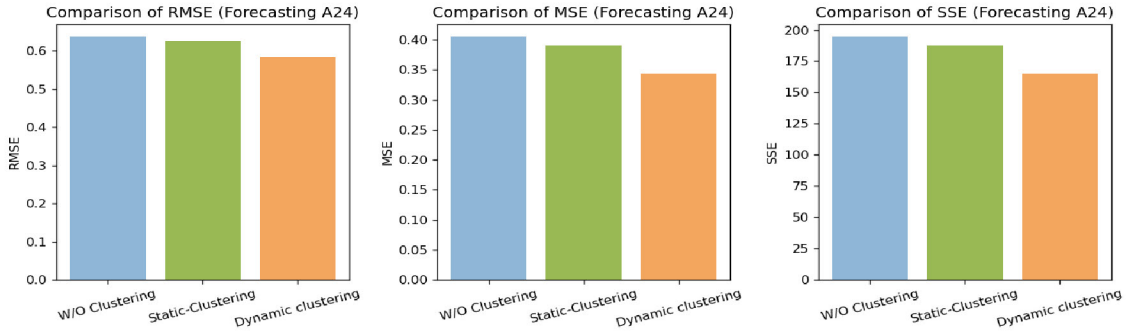


Figure 4.34: Comparison of $RMSE$, MSE and SSE (forecasting A24, for 20 days) for Different Clustering Methods

Table 4.19: Comparative Analysis of A7 using Different Time-Series Clustering Methods

Metric	**Without Clustering**	**With Static Clustering**	**With Dynamic Clustering**		
			Day	**Night**	**Average**
RMSE	1.3287	1.3154	1.5400	0.676615	1.1083
MSE	1.7654	1.7305	2.3716	0.4578	1.4147
SSE	1694.837	1245.980	1707.574	329.622	1,018.598

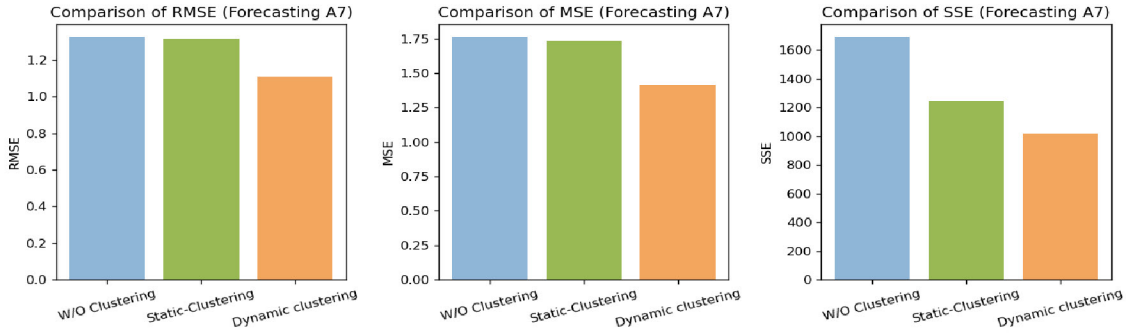


Figure 4.35: Comparison of $RMSE$, MSE and SSE (forecasting A7, for 30 days) for Different Clustering Methods

Table 4.20: Comparative Analysis of A16 using Different Time-Series Clustering Methods

Metric	**Without Clustering**	**With Static Clustering**	**With Dynamic Clustering**		
			Day	**Night**	**Average**
RMSE	1.2765	1.2688	1.0606	1.0968	1.0787
MSE	1.6296	1.6100	1.1249	1.2030	1.1663
SSE	1368.9148	1352.426	944.926	1010.573	847.073

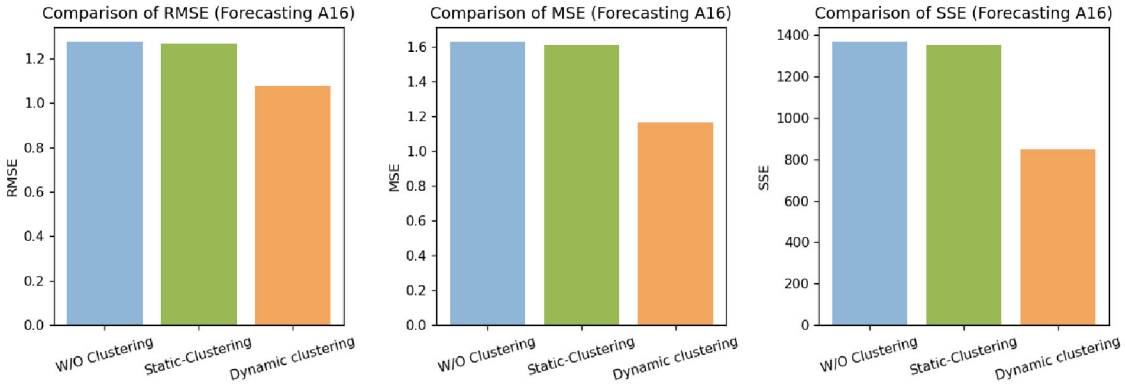


Figure 4.36: Comparison of $RMSE$, MSE and SSE (forecasting A16, for 35 days) for Different Clustering Methods

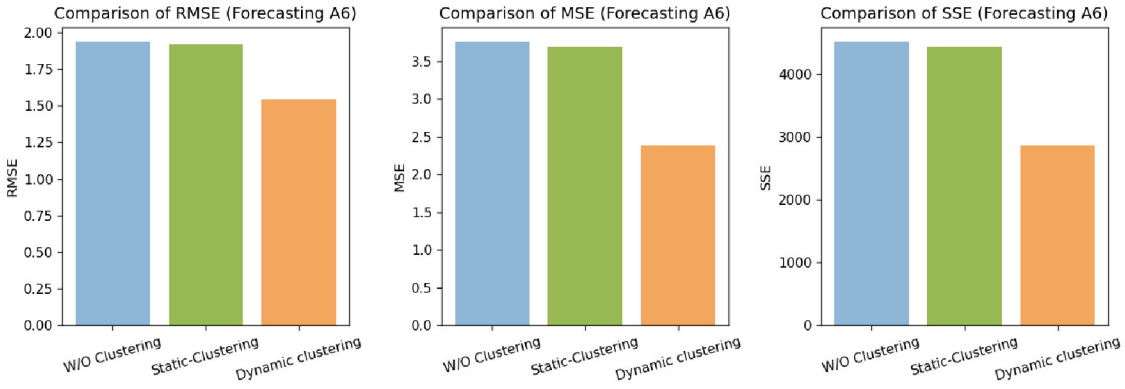


Figure 4.37: Comparison of $RMSE$, MSE and SSE (forecasting A6, for 50 days) for Different Clustering Methods

Chapter 5

Conclusion and Future Scope

This thesis presents a comprehensive investigation into the use of clustering techniques for improving energy consumption forecasting in smart homes. The research demonstrates that incorporating the connections between various appliances through clustering can enhance the accuracy of energy consumption forecasts compared to conventional models. The proposed static and dynamic clustering algorithms are particularly effective in this regard.

To enhance the precision of energy consumption forecasting, the thesis introduces a dynamic time series clustering technique. The k-shape algorithm is employed for static clustering, which partitions a dataset of 43 smart home appliances into five clusters. Subsequently, using static clustering, the forecasting model predicts the energy consumption of five appliances from each cluster. The dataset is further segmented into day and night periods for dynamic clustering purposes. By applying dynamic day and night time clustering, the model forecasts the energy consumption of the same five appliances for durations of 10, 20, 30, 35, and 50 days, and compares the results with static time series clustering. The findings clearly indicate that the proposed algorithm significantly enhances the accuracy of energy consumption forecasting in smart homes. Moreover, the thesis explores the efficacy of time series clustering and investigates the potential of incorporating time of day and night time periods in energy forecasting.

Moving forward, there are several avenues for future research in this domain. One area of focus could be the exploration of machine learning approaches to further improve the accuracy of energy consumption forecasts. Additionally, incorporating additional factors such as weather information and occupancy statistics may enhance forecasting precision. Overall, this research provides a solid foundation for future studies on leveraging machine learning and clustering techniques to enhance energy consumption forecasting in smart homes.

Chapter 6

Publications

Peer-Reviewed Conference

S. Redhu, **R. Raja**, B. Bremdal, "COGNITIVE DATA FUSION FOR IMPROVING FLEXIBILITY IN SMART HOMES", Accepted to appear in CIRED 2023, Rome.

In Progress

R. Raja, D. Nga, "Dynamic Time-Series Clustering for Improving Appliances' Forecasting in Smart Homes", (work in progress).

Bibliography

[1] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: An overview," in *2013 IEEE Green Technologies Conference (GreenTech)*, IEEE, 2013, pp. 57–64.

[2] Y. Parag and G. Butbul, "Flexiwatts and seamless technology: Public perceptions of demand flexibility through smart home technology," *Energy Research & Social Science*, vol. 39, pp. 177–191, 2018.

[3] A. Khanna and S. Kaur, "Internet of things (iot), applications and challenges: A comprehensive review," *Wireless Personal Communications*, vol. 114, pp. 1687–1762, 2020.

[4] C. Paul, A. Ganesh, and C. Sunitha, "An overview of iot based smart homes," in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, IEEE, 2018, pp. 43–46.

[5] L. Salman, S. Salman, S. Jahangirian, *et al.*, "Energy efficient iot-based smart home," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, IEEE, 2016, pp. 526–529.

[6] S. Aheleroff, X. Xu, Y. Lu, *et al.*, "Iot-enabled smart appliances under industry 4.0: A case study," *Advanced engineering informatics*, vol. 43, p. 101 043, 2020.

[7] N. Arghira, L. Hawarah, S. Ploix, and M. Jacomino, "Prediction of appliances energy use in smart homes," *Energy*, vol. 48, no. 1, pp. 128–134, 2012.

[8] A. Kathirgamanathan, M. De Rosa, E. Mangina, and D. P. Finn, "Data-driven predictive control for unlocking building energy flexibility: A review," *Renewable and Sustainable Energy Reviews*, vol. 135, p. 110 120, 2021.

[9] I. Antonopoulos, V. Robu, B. Couraud, *et al.*, "Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review," *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109 899, 2020.

[10] M. R. Alam, M. B. I. Reaz, and M. A. M. Ali, "A review of smart homes—past, present, and future," *IEEE transactions on systems, man, and cybernetics, part C (applications and reviews)*, vol. 42, no. 6, pp. 1190–1203, 2012.

[11] L. Prieto González, A. Fensel, J. M. Gómez Berbís, A. Popa, and A. de Amescua Seco, "A survey on energy efficiency in smart homes and smart grids," *Energies*, vol. 14, no. 21, p. 7273, 2021.

[12] H. G. Bergsteinsson, T. S. Nielsen, J. K. Møller, S. B. Amer, D. F. Dominković, and H. Madsen, "Use of smart meters as feedback for district heating temperature control," *Energy Reports*, vol. 7, pp. 213–221, 2021.

[13] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[14] C. Kuzemko, M. Blondeel, C. Dupont, and M. C. Brisbois, "Russia's war on ukraine, european energy policy responses & implications for sustainable transformations," *Energy Research & Social Science*, vol. 93, p. 102 842, 2022.

[15] B. Č. Erceg, A. Vasilj, and A. Perković, "Fit for 55–does it fit all? air and rail transport after covid–19 pandemic," *EU and comparative law issues and challenges series (ECLIC)*, vol. 6, pp. 66–101, 2022.

[16] M. Ramezani, D. Bahmanyar, and N. Razmjooy, "A new optimal energy management strategy based on improved multi-objective antlion optimization algorithm: Applications in smart home," *SN Applied Sciences*, vol. 2, pp. 1–17, 2020.

[17] R. Zafar, A. Mahmood, S. Razzaq, W. Ali, U. Naeem, and K. Shehzad, "Prosumer based energy management and sharing in smart grid," *Renewable and Sustainable Energy Reviews*, vol. 82, pp. 1675–1684, 2018.

[18] B.-J. Chen, M.-W. Chang, *et al.*, "Load forecasting using support vector machines: A study on eunite competition 2001," *IEEE transactions on power systems*, vol. 19, no. 4, pp. 1821–1830, 2004.

[19] G. Kariniotakis, G. Stavrakakis, and E. Nogaret, "Wind power forecasting using advanced neural networks models," *IEEE transactions on Energy conversion*, vol. 11, no. 4, pp. 762–767, 1996.

[20] A. Fensel, D. K. Tomic, and A. Koller, "Contributing to appliances' energy efficiency with internet of things, smart data and user engagement," *Future Generation Computer Systems*, vol. 76, pp. 329–338, 2017.

[21] D. L. Ha, S. Ploix, E. Zamai, and M. Jacomino, "Realtimes dynamic optimization for demand-side load management," *International Journal of Management Science and Engineering Management*, vol. 3, no. 4, pp. 243–252, 2008.

[22] K. Methaprayoon, W. Lee, P. Didsayabutra, J. Liao, and R. Ross, "Neural network-based short term load forecasting for unit commitment scheduling," in *IEEE Technical Conference on Industrial and Commercial Power Systems, 2003.*, IEEE, 2003, pp. 138–143.

[23] S. Ruzic, A. Vuckovic, and N. Nikolic, "Weather sensitive method for short term load forecasting in electric power utility of serbia," *IEEE Transactions on Power Systems*, vol. 18, no. 4, pp. 1581–1586, 2003.

[24] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1352–1372, 2015.

[25] D. De Silva, D. Alahakoon, and X. Yu, "A data fusion technique for smart home energy management and analysis," in *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*, IEEE, 2016, pp. 4594–4600.

[26] A. Diez-Olivan, J. Del Ser, D. Galar, and B. Sierra, "Data fusion and machine learning for industrial prognosis: Trends and perspectives towards industry 4.0," *Information Fusion*, vol. 50, pp. 92–111, 2019.

[27] N.-B. Chang and K. Bai, *Multisensor data fusion and machine learning for environmental remote sensing*. CRC Press, 2018.

[28] F. S. Al-Ismail, "Dc microgrid planning, operation, and control: A comprehensive review," *IEEE Access*, vol. 9, pp. 36 154–36 172, 2021.

[29] F. Moazeni, J. Khazaei, and A. Asrari, "Step towards energy-water smart microgrids; buildings thermal energy and water demand management embedded in economic dispatch," *IEEE Transactions on Smart Grid*, vol. 12, no. 5, pp. 3680–3691, 2021.

[30] B. Mahapatra and A. Nayyar, "Home energy management system (hems): Concept, architecture, infrastructure, challenges and energy management schemes," *Energy Systems*, vol. 13, no. 3, pp. 643–669, 2022.

[31] J. Abushnaf, A. Rassau, and W. Górnisiewicz, "Impact on electricity use of introducing time-of-use pricing to a multi-user home energy management system," *International Transactions on Electrical Energy Systems*, vol. 26, no. 5, pp. 993–1005, 2016.

[32] A. Muller and R. Bourdais, "Dynamic pricing for local energy management: Towards a better integration of local production," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6749–6754, 2017.

[33] F. Qayyum, M. Naeem, A. Khwaja, and A. Anpalagan, "Appliance scheduling optimization in smart home networks comprising of smart appliances and a photovoltaic panel," in *2015 IEEE electrical power and energy conference (EPEC)*, IEEE, 2015, pp. 457–462.

[34] M. Khan, B. N. Silva, and K. Han, "A web of things-based emerging sensor network architecture for smart control systems," *Sensors*, vol. 17, no. 2, p. 332, 2017.

[35] M. Khan, B. N. Silva, and K. Han, "Internet of things based energy aware smart home control system," *Ieee Access*, vol. 4, pp. 7556–7566, 2016.

[36] Q. Xu, C. Zhang, C. Wen, and P. Wang, "A novel composite nonlinear controller for stabilization of constant power load in dc microgrid," *IEEE Transactions on Smart Grid*, vol. 10, no. 1, pp. 752–761, 2017.

[37] M. Muratori and G. Rizzoni, "Residential demand response: Dynamic energy management and time-varying electricity pricing," *IEEE Transactions on Power systems*, vol. 31, no. 2, pp. 1108–1117, 2015.

[38] A. Mnatsakanyan and S. W. Kennedy, "A novel demand response model with an application for a virtual power plant," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 230–237, 2014.

[39] M. Muratori, B.-A. Schuelke-Leech, and G. Rizzoni, "Role of residential demand response in modern electricity markets," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 546–553, 2014.

[40] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE transactions on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.

[41] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Demand-side management via distributed energy generation and storage optimization," *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 866–876, 2012.

[42] C. Joe-Wong, S. Sen, S. Ha, and M. Chiang, "Optimized day-ahead pricing for smart grids with device-specific scheduling flexibility," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1075–1085, 2012.

[43] B. Ramanathan and V. Vittal, "A framework for evaluation of advanced direct load control with minimum disruption," *IEEE Transactions on Power Systems*, vol. 23, no. 4, pp. 1681–1688, 2008.

[44] C. Chen, J. Wang, Y. Heo, and S. Kishore, "Mpc-based appliance scheduling for residential building energy management controller," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1401–1410, 2013.

[45] M. Karimi, H. Karami, M. Gholami, H. Khatibzadehazad, and N. Moslemi, "Priority index considering temperature and date proximity for selection of similar days in knowledge-based short term load forecasting method," *Energy*, vol. 144, pp. 928–940, 2018.

[46] F. Farzan, M. A. Jafari, J. Gong, F. Farzan, and A. Stryker, "A multi-scale adaptive model of residential energy demand," *Applied Energy*, vol. 150, pp. 258–273, 2015.

[47] K. Song, Y. Jang, M. Park, H.-S. Lee, and J. Ahn, "Energy efficiency of end-user groups for personalized hvac control in multi-zone buildings," *Energy*, vol. 206, p. 118 116, 2020.

[48] R. Z. Homod, H. Togun, H. J. Abd, and K. S. Sahari, "A novel hybrid modelling structure fabricated by using takagi-sugeno fuzzy to forecast hvac systems energy demand in real-time for basra city," *Sustainable Cities and Society*, vol. 56, p. 102 091, 2020.

[49] T. Hong *et al.*, "Energy forecasting: Past, present, and future," *Foresight: The International Journal of Applied Forecasting*, no. 32, pp. 43–48, 2014.

[50] H. L. Willis and J. E. Northcote-Green, "Spatial electric load forecasting: A tutorial review," *Proceedings of the IEEE*, vol. 71, no. 2, pp. 232–253, 1983.

[51] H. S. Hippert, C. E. Pedreira, and R. C. Souza, "Neural networks for short-term load forecasting: A review and evaluation," *IEEE Transactions on power systems*, vol. 16, no. 1, pp. 44–55, 2001.

[52] Y. Dodge and D. Cox, *The Oxford dictionary of statistical terms*. Oxford University Press, USA, 2003.

[53] K. B. Debnath and M. Mourshed, "Forecasting methods in energy planning models," *Renewable and Sustainable Energy Reviews*, vol. 88, pp. 297–325, 2018.

[54] S. H. A. Kaboli, J. Selvaraj, and N. Rahim, "Long-term electric energy consumption forecasting via artificial cooperative search algorithm," *Energy*, vol. 115, pp. 857–871, 2016.

[55] S. H. A. Kaboli, A. Fallahpour, N. Kazemi, J. Selvaraj, and N. Rahim, "An expression-driven approach for long-term electric power consumption forecasting," *American Journal of Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 16–28, 2016.

[56] S. H. A. Kaboli, A. Fallahpour, J. Selvaraj, and N. Rahim, "Long-term electrical energy consumption formulating and forecasting via optimized gene expression programming," *Energy*, vol. 126, pp. 144–164, 2017.

[57] J. Wu, J. Wang, H. Lu, Y. Dong, and X. Lu, "Short term load forecasting technique based on the seasonal exponential adjustment method and the regression model," *Energy Conversion and Management*, vol. 70, pp. 1–9, 2013.

[58] G.-F. Fan, L.-L. Peng, and W.-C. Hong, "Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model," *Applied energy*, vol. 224, pp. 13–33, 2018.

[59] J. W. Taylor and P. E. McSharry, "Short-term load forecasting methods: An evaluation based on european data," *IEEE Transactions on Power Systems*, vol. 22, no. 4, pp. 2213–2219, 2007.

[60] K. G. Boroojeni, M. H. Amini, S. Bahrami, S. Iyengar, A. I. Sarwat, and O. Karabasoglu, "A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon," *Electric Power Systems Research*, vol. 142, pp. 58–73, 2017.

[61] H. Takeda, Y. Tamura, and S. Sato, "Using the ensemble kalman filter for electricity load forecasting and analysis," *Energy*, vol. 104, pp. 184–198, 2016.

[62] H. Al-Hamadi and S. Soliman, "Short-term electric load forecasting based on kalman filtering algorithm with moving window weather and load model," *Electric power systems research*, vol. 68, no. 1, pp. 47–59, 2004.

[63] M. Aydinalp, V. I. Ugursal, and A. S. Fung, "Modeling of the appliance, lighting, and space-cooling energy consumptions in the residential sector using neural networks," *Applied energy*, vol. 71, no. 2, pp. 87–110, 2002.

[64] P. Singh and P. Dwivedi, "Integration of new evolutionary approach with artificial neural network for solving short term load forecast problem," *Applied energy*, vol. 217, pp. 537–549, 2018.

[65] M. Ghofrani, M. Ghayekhloo, A. Arabali, and A. Ghayekhloo, "A hybrid short-term load forecasting with a new input selection framework," *Energy*, vol. 81, pp. 777–786, 2015.

[66] J. Dancker, *Towards data science*. [Online]. Available: `https://towardsdatascience.com/a-brief-introduction-to-time-series-forecasting-using-statistical-methods-d4ec849658c3`, (Last Accessed: 18.04.2023).

[67] G. R. Newsham and B. J. Birt, "Building-level occupancy data to improve arima-based electricity use forecasts," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*, 2010, pp. 13–18.

[68] P. Chujai, N. Kerdprasop, and K. Kerdprasop, "Time series analysis of household electric consumption with arima and arma models," in *Proceedings of the international multiconference of engineers and computer scientists*, IAENG Hong Kong, vol. 1, 2013, pp. 295–300.

[69] N. Mohamed, M. H. Ahmad, Z. Ismail, and S. Suhartono, "Short term load forecasting using double seasonal arima model," in *Proceedings of the regional conference on statistical sciences*, vol. 10, 2010, pp. 57–73.

[70] J. Dancker, *Towards data science*. [Online]. Available: `https://towardsdatascience.com/a-brief-introduction-to-time-series-forecasting-using-statistical-methods-d4ec849658c3`, (Last Accessed: 22.04.2023).

[71] G. E. Box and G. C. Tiao, "Comparison of forecast and actuality," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 25, no. 3, pp. 195–200, 1976.

[72] C. Goh and R. Law, "Modeling and forecasting tourism demand for arrivals with stochastic nonstationary seasonality and intervention," *Tourism management*, vol. 23, no. 5, pp. 499–510, 2002.

[73] Y. Li, E. Campbell, D. Haswell, R. Sneeuwjagt, and W. Venables, "Statistical forecasting of soil dryness index in the southwest of western australia," *Forest Ecology and Management*, vol. 183, no. 1-3, pp. 147–157, 2003.

[74] J. Navarro-Esbrı, E. Diamadopoulos, and D. Ginestar, "Time series analysis and forecasting techniques for municipal solid waste management," *Resources, conservation and Recycling*, vol. 35, no. 3, pp. 201–214, 2002.

[75] G. P. Zhang, "Time series forecasting using a hybrid arima and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.

[76] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—a novel pooling deep rnn," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2017.

[77] L. Wang, Z. Zhang, and J. Chen, "Short-term electricity price forecasting with stacked denoising autoencoders," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2673–2681, 2016.

[78] C. Feng, M. Sun, and J. Zhang, "Reinforced deterministic and probabilistic load forecasting via Q-learning dynamic model selection," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1377–1386, 2019.

[79] L. Cai, J. Gu, and Z. Jin, "Two-layer transfer-learning-based architecture for short-term load forecasting," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1722–1732, 2019.

[80] Y.-T. Chen, E. Piedad Jr, and C.-C. Kuo, "Energy consumption load forecasting using a level-based random forest classifier," *Symmetry*, vol. 11, no. 8, p. 956, 2019.

[81] H. Chang, C. Kuo, Y. Chen, W. Wu, and E. J. Piedad, "Energy consumption level prediction based on classification approach with machine learning technique," in *4th World Congress on New Technologies*, 2018, pp. 1–8.

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[83] T. Catalina, J. Virgone, and E. Blanco, "Development and validation of regression models to predict monthly heating demand for residential buildings," *Energy and buildings*, vol. 40, no. 10, pp. 1825–1832, 2008.

[84] J. S. Hygh, J. F. DeCarolis, D. B. Hill, and S. R. Ranjithan, "Multivariate regression as an energy assessment tool in early building design," *Building and environment*, vol. 57, pp. 165–175, 2012.

[85] Z. Wang and R. S. Srinivasan, "A review of artificial intelligence based building energy prediction with a focus on ensemble prediction models," in *2015 Winter simulation conference (WSC)*, IEEE, 2015, pp. 3438–3448.

[86] B. Dong, C. Cao, and S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy and Buildings*, vol. 37, no. 5, pp. 545–553, 2005.

[87] M. Bozic, M. Stojanovic, and Z. Stajic, "Short-term electric load forecasting using least square support vector machines," *Facta Universitatis, Series: Automatic Control and Robotics*, vol. 9, no. 1, pp. 141–150, 2010.

[88] G. Athanasopoulos, R. A. Ahmed, and R. J. Hyndman, "Hierarchical forecasts for australian domestic tourism," *International Journal of Forecasting*, vol. 25, no. 1, pp. 146–166, 2009.

[89] R. J. Hyndman, R. A. Ahmed, G. Athanasopoulos, and H. L. Shang, "Optimal combination forecasts for hierarchical time series," *Computational statistics & data analysis*, vol. 55, no. 9, pp. 2579–2589, 2011.

[90] S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman, "Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 804–819, 2019.

[91] Y. Ren, P. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting—a state-of-the-art review," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 82–91, 2015.

[92] J. M. Bates and C. W. Granger, "The combination of forecasts," *Journal of the operational research society*, vol. 20, no. 4, pp. 451–468, 1969.

[93] M. S. Roulston and L. A. Smith, "Combining dynamical and statistical ensembles," *Tellus A: Dynamic Meteorology and Oceanography*, vol. 55, no. 1, pp. 16–30, 2003.

[94] A. Krumins, *Gearbox fault detection, based on machine learning of multiple sensors*, 2021.

[95] C. Voyant, G. Notton, S. Kalogirou, *et al.*, "Machine learning methods for solar radiation forecasting: A review," *Renewable energy*, vol. 105, pp. 569–582, 2017.

[96] E. D'Andrea and B. Lazzerini, "Rosario sld," *Neural network-based forecasting of energy consumption due to electric lighting in office buildings. Sustainable Internet and ICT for Sustainability (SustainIT) Pisa, Italy: IEEE*, 2012.

[97] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? a review and evaluation," *Journal of forecasting*, vol. 17, no. 5-6, pp. 481–495, 1998.

[98] B. B. Ekici and U. T. Aksoy, "Prediction of building energy consumption by using artificial neural networks," *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, 2009.

[99] C. Hamzaçebi, "Forecasting of turkey's net electricity energy consumption on sectoral bases," *Energy policy*, vol. 35, no. 3, pp. 2009–2016, 2007.

[100] R. Ramanathan, R. Engle, C. W. Granger, F. Vahid-Araghi, and C. Brace, "Short-run forecasts of electricity loads and peaks," *International journal of forecasting*, vol. 13, no. 2, pp. 161–174, 1997.

[101] A. Khotanzad, R. Afkhami-Rohani, T.-L. Lu, A. Abaye, M. Davis, and D. J. Maratukulam, "Annstlf-a neural-network-based electric load forecasting system," *IEEE Transactions on Neural networks*, vol. 8, no. 4, pp. 835–846, 1997.

[102] A. Khotanzad, R. Afkhami-Rohani, and D. Maratukulam, "Annstlf-artificial neural network short-term load forecaster-generation three," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1413–1422, 1998.

[103] T. Hong, J. Xie, and J. Black, "Global energy forecasting competition 2017: Hierarchical probabilistic load forecasting," *International Journal of Forecasting*, vol. 35, no. 4, pp. 1389–1399, 2019.

[104] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, *Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond*, 2016.

[105] T. Hong, P. Pinson, and S. Fan, *Global energy forecasting competition 2012*, 2014.

[106] J. W. Messner, P. Pinson, J. Browell, M. B. Bjerregård, and I. Schicker, "Evaluation of wind power forecasts—an up-to-date view," *Wind Energy*, vol. 23, no. 6, pp. 1461–1481, 2020.

[107] H. Shaker, H. Zareipour, and D. Wood, "Estimating power generation of invisible solar sites using publicly available data," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2456–2465, 2016.

[108] M. Sun, T. Zhang, Y. Wang, G. Strbac, and C. Kang, "Using bayesian deep learning to capture uncertainty for residential net load forecasting," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 188–201, 2019.

[109] R. Bo and F. Li, "Probabilistic lmp forecasting considering load uncertainty," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1279–1289, 2009.

[110] T. W. Liao, "Clustering of time series data—a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.

[111] J. Paparrizos and L. Gravano, "K-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, 2015, pp. 1855–1870.

[112] K. Košmelj and V. Batagelj, "Cross-sectional approach for clustering time varying data," *Journal of Classification*, vol. 7, no. 1, pp. 99–109, 1990.

[113] T. Liao, B. Bolt, J. Forester, *et al.*, "Understanding and projecting the battle state," in *23rd Army Science Conference, Orlando, FL*, vol. 25, 2002.

[114] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis' 99)*, IEEE, 1999, pp. 4–9.

[115] A. Nagpal, A. Jatain, and D. Gaur, "Review based on data clustering algorithms," in *2013 IEEE conference on information & communication technologies*, IEEE, 2013, pp. 298–303.

[116] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *computer*, vol. 32, no. 8, pp. 68–75, 1999.

[117] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise.," in *kdd*, vol. 96, 1996, pp. 226–231.

[118] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure," *ACM Sigmod record*, vol. 28, no. 2, pp. 49–60, 1999.

[119] A. Ahmad and L. Dey, "A k-mean clustering algorithm for mixed numeric and categorical data," *Data & Knowledge Engineering*, vol. 63, no. 2, pp. 503–527, 2007.

[120] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[121] R. T. Ng and J. Han, "E cient and e ective clustering methods for spatial data mining," in *Proceedings of VLDB*, 1994, pp. 144–155.

[122] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons, 2009.

[123] S. Guha, R. Rastogi, and K. Shim, "Cure: An efficient clustering algorithm for large databases," *ACM Sigmod record*, vol. 27, no. 2, pp. 73–84, 1998.

[124] T. Zhang, R. Ramakrishnan, and M. Livny, "Birch: An efficient data clustering method for very large databases," *ACM sigmod record*, vol. 25, no. 2, pp. 103–114, 1996.

[125] M. V. J. L. E. Keogh and D. Gunopulos, "A wavelet-based anytime algorithm for k-means clustering of time series,"

[126] E. J. Keogh and M. J. Pazzani, "An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback.," in *Kdd*, vol. 98, 1998, pp. 239–243.

[127] T. Oates, M. D. Schmill, and P. R. Cohen, "A method for clustering the experiences of a mobile robot that accords with human judgments," in *AAAI/IAAI*, 2000, pp. 846–851.

[128] S. Hirano and S. Tsumoto, "Empirical comparison of clustering methods for long time-series databases," in *Active Mining: Second International Workshop, AM 2003, Maebashi, Japan, October 28, 2003. Revised Selected Papers*, Springer, 2005, pp. 268–286.

[129] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, 2003, pp. 2–11.

[130] H. Sakoe, "Dynamic-programming approach to continuous speech recognition," in *1971 proc. the international congress of acoustics, budapest*, 1971.

[131] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[132] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proceedings 18th international conference on data engineering*, IEEE, 2002, pp. 673–684.

[133] A. Banerjee and J. Ghosh, "Clickstream clustering using weighted longest common subsequences," in *Proceedings of the web mining workshop at the 1st SIAM conference on data mining*, Citeseer, vol. 143, 2001, p. 144.

[134] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data mining and knowledge Discovery*, vol. 13, pp. 335–364, 2006.

[135] D. Lam and D. C. Wunsch, "Clustering," *Academic Press Library in Signal Processing*, vol. 1, pp. 1115–1149, 2014.

[136] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, University of California Los Angeles LA USA, 1967, pp. 281–297.

[137] A. Saxena, M. Prasad, A. Gupta, *et al.*, "A review of clustering techniques and developments," *Neurocomputing*, vol. 267, pp. 664–681, 2017.

[138] A. Gersho, R. M. Gray, A. Gersho, and R. M. Gray, "Finite—state vector quantization," *Vector Quantization and Signal Compression*, pp. 519–553, 1992.

[139] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on neural networks*, vol. 16, no. 3, pp. 645–678, 2005.

[140] T. Velmurugan and T. Santhanam, "Computational complexity between k-means and k-medoids clustering algorithms for normal and uniform distributions of data points," *Journal of computer science*, vol. 6, no. 3, p. 363, 2010.

[141] J. Wilpon and L. Rabiner, "A modified k-means clustering algorithm for use in isolated work recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 3, pp. 587–594, 1985.

[142] C. Shaw and G. King, "Using cluster analysis to classify time series," *Physica D: Nonlinear Phenomena*, vol. 58, no. 1-4, pp. 288–298, 1992.

[143] P. C. Cheeseman, J. C. Stutz, *et al.*, "Bayesian classification (autoclass): Theory and results.," *Advances in knowledge discovery and data mining*, vol. 180, pp. 153–180, 1996.

[144] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer vision, graphics, and image processing*, vol. 37, no. 1, pp. 54–115, 1987.

[145] T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, vol. 78, no. 9, pp. 1464–1480, 1990.

[146] D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine learning*, vol. 2, pp. 139–172, 1987.

[147] N. D. Copyright 2008-2022, *Towards data science.* [Online]. Available: `https://numpy.org/doc/stable/`, (Last Accessed: 03.05.2023).

[148] ©. 2. pandas via numfocos., *Towards data science.* [Online]. Available: `https://pandas.pydata.org/docs/`, (Last Accessed: 03.05.2023).

[149] M. D. John Hunter, *Towards data science.* [Online]. Available: `https://matplotlib.org/stable/index.`, (Last Accessed: 05.05.2023).

[150] ©. 2. pandas via numfocos., *Michael waskom.* [Online]. Available: `https://seaborn.pydata.org/`, (Last Accessed: 05.05.2023).

[151] L. Buitinck, G. Louppe, M. Blondel, *et al.*, "API design for machine learning software: Experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[152] J.-P. Vandijck, R. Tavenard, F. Divo, *et al.*, *Tslearn: A python library for time series classification*, `https://tslearn.readthedocs.io/en/stable/`, Accessed: May 2, 2023, 2021.

[153] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with python," in *9th Python in Science Conference*, 2010.

[154] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd. O'Reilly Media, Inc., 2019, ISBN: 1492032646.

[155] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[156] G. E. Batista, E. J. Keogh, O. M. Tataw, and V. M. De Souza, "Cid: An efficient complexity-invariant distance for time series," *Data Mining and Knowledge Discovery*, vol. 28, pp. 634–669, 2014.

[157] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM Sigmod Record*, vol. 23, no. 2, pp. 419–429, 1994.

[158] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, pp. 358–386, 2005.

[159] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, pp. 275–309, 2013.

[160] T. Rakthanmanon, B. Campana, A. Mueen, *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 262–270.

[161] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[162] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.

[163] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta, "A survey of kernel and spectral methods for clustering," *Pattern recognition*, vol. 41, no. 1, pp. 176–190, 2008.

[164] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[165] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proceedings of the fourth ACM international conference on Web search and data mining*, 2011, pp. 177–186.

[166] R. Giusti and G. E. Batista, "An empirical comparison of dissimilarity measures for time series classification," in *2013 Brazilian Conference on Intelligent Systems*, IEEE, 2013, pp. 82–88.

[167] J. Paparrizos and L. Gravano, "Fast and accurate time-series clustering," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 2, pp. 1–49, 2017.

[168] V. S. Sartório and T. C. Fonseca, "Dynamic clustering of time series data," *arXiv preprint arXiv:2002.01890*, 2020.

[169] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information systems*, vol. 53, pp. 16–38, 2015.

[170] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, J. Albrecht, *et al.*, "Smart*: An open data set and tools for enabling research in sustainable homes," *SustKDD, August*, vol. 111, no. 112, p. 108, 2012.