

UNIVERSITY OF BAHRAIN
College of Information Technology



The Effectiveness of Deep Learning Methods for Smart Meter Data Analytics

**A Thesis Submitted in Partial Fulfilment of the Requirements for the
Ph.D. Degree in Computing and Information Sciences**

Submitted by
Elham Mohd Eskandarnia
19927274

Supervised by
Dr. Riadh Ksantini
Associate Professor

Dr. Hesham Al-Ammal
Assistant Professor
University of Bahrain

Kingdom of Bahrain
June 2022

UNIVERSITY OF BAHRAIN
College of Information Technology



The Effectiveness of Deep Learning Methods for Smart Meter Data Analytics

A Thesis Submitted in Partial Fulfilment of the Requirements for the **Ph.D.**
Degree in **Computing and Information Sciences**

Submitted by
Elham Mohd Eskandar Eskandarnia
19927274

Supervised by
Dr. Riadh Ksantini
Associate Professor

Dr. Hesham Al-Ammal
Assistance Professor

University of Bahrain

Kingdom of Bahrain
June 2022



Dissertation Deposit Form

Author Name:	Elham Mohamed Eskandarnia		
Email:	19927274@stu.uob.edu.bh	Contact No:	39815815
College:	Information Technology		
Department:	Computer Science		
Thesis Title:	The Effectiveness of Deep Learning Methods for Smart Meter Data Analytics		
Year:	2022		

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:	Elham Eskandarnia	Date:	20-Apr-22
--------------------------	-------------------	-------	-----------

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed:



Date: 20/04/2022

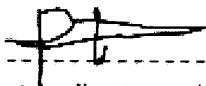
Elham Mohamed Eskandar Eskandarnia (Candidate)

The Thesis defense committee considers the thesis titled:

The Effectiveness of Deep Learning Methods for Smart Meter Data Analytics

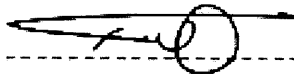
Submitted by **Elham Mohamed Eskandar Eskandarnia** to the College of Information Technology is satisfactory and acceptable for the Master of Science in Cyber Security.

The Defense Committee:



Dr. Riadh Ksantini
(Supervisor)

Department of Computer Science
University of Bahrain



Dr. Hesham Al-Anmal
(Co-Supervisor)

Department of Computer Science
University of Bahrain



Dr. Wael Mohamed Elmedany
(Member)

Department of Computer Science
University of Bahrain



Dr. Mhd Saeed Sharif
(Member)

School of Architecture, Computing and Engineering
University of East London

Abstract

The introduction of smart grid technology came with the promise of better management of energy and power systems, an essential step toward achieving sustainable energy. Smart meter analytics is viewed as a critical building block for sustainable energy since it improves operational efficiency, enhances power grid reliability, and enables better management of energy resources.

This thesis presents a unifying taxonomy to classify the various domains of smart meter analytics and their underlying functions and techniques. This thesis also investigates the deep-learning architectures used for each domain. Current research trends, approaches, and opportunities are introduced and discussed through a the comprehensive updated taxonomy to understand the field of smart meter analytics. This thesis proposes a novel embedded deep clustering base framework for the task of load profiling. The proposed load profiling framework's novelty is twofold. First, using an autoencoder the framework represents data as a hierarchy within the layers of the deep network, allowing for dimensionality reduction and highly nonlinear decision separation between clusters at the autoencoder bottleneck level.

This thesis further investigates the forecasting of big temporal data by Seasonal Autoregressive Integrated Moving Average Model (SARIMAX) and Prophet methods. Hence, accurate and computationally cheap load forecasting techniques are a crucial part of the grid system. Both the deep clustering framework and forecasting methods are then evaluated and tested using real-world smart meter data sets, and the experimental results show that the proposed framework produces significantly better load profiles when compared to classical clustering algorithms as well as previous hybrid frameworks proposed in the literature.

Publications

The following are papers that resulted from the work conducted during the preparation of this thesis.

- **Journal Papers**

1. Eskandarnia, Elham, Hesham M. Al-Ammal, and Riadh Ksantini. An embedded deep-clustering-based load profiling framework. *Sustainable Cities and Society* 78 (2021): 103618. <https://doi.org/10.1016/j.scs.2021.103618>
2. Eskandarnia, Elham, Hesham M. Al-Ammal, and Riadh Ksantini, Mustafa Hammad. Deep Learning Techniques for Smart Meter Data Analytics: A Review. *SN Computer Science* (2022), <https://doi.org/10.1007/s42979-022-01161-6>
3. Eskandarnia, Elham M., and Hesham M. Al-Ammal. A Taxonomy of Smart Meter Analytics: Forecasting, Knowledge Discovery, and Power Management. *International Journal of Computing and Digital Systems* 11.1 (2022). DOI: <https://dx.doi.org/10.12785/ijcds/110170>

- **Conference Papers**

1. Eskandarnia, E. and AlHammad, M. (2020). Predication of future energy consumption using sarimax. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 657–662. <https://digital-library.theiet.org/content/conferences/10.1049/icp.2021.0853>
2. Eskandar, E. M., Alammal, H., and Elmedany, W. (2020). What prophet says about electrical consumption - forecasting techniques for big temporal data. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 543–548 <https://digital-library.theiet.org/content/conferences/10.1049/icp.2021.0863>
3. Iskandarnia, E., Al-Amal, H., and Elmedany, W. (2020). Load forecasting in different scale and horizon-a review. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 464–470. IET. <https://ieeexplore.ieee.org/abstract/document/9545833>

- ORCID <https://orcid.org/0000-0002-3129-2101>

Dedication

This work is dedicated to:

My Family, for their unconditional support throughout this journey. Their everlasting love, dedication, and sacrifice has enabled me to get to this point. Adel, Alaa, and Ayeh without you I could not do it.

Acknowledgments

Throughout the preparation of this dissertation, I received a great deal of assistance and support. First and foremost, I would like to thank my supervisors, Dr. Riadh Ksantini and Dr. Hesham Al-Ammal, for all the help, guidance, support, and great encouragement that they provided during the course of this research. I was able to publish three pieces in highly regarded journals because of your advice and support.

I'd also like to thank Dr. Hesham for his insightful advice during my PhD journey. Your support and advice have assisted me in establishing a solid funding base to conduct the project and to make the proper decision and finish my dissertation effectively. I would also like to thank my PhD instructors, Dr. Mustafa Hammad, Dr. Wael Elmadany, and Dr. Hessa AlJunaid, for their unwavering support and assistance throughout my time at this journey.

[tps://orcid.org/ 0000-0002-3129-2](https://orcid.org/0000-0002-3129-2)

Contents

Defense committee’s Approval	A
Declaration	B
Abstract	C
Publications	D
Dedication	E
Acknowledgments	F
Table of Contents	G
List of Tables	K
List of Figures	L
List of Abbreviations	N
1 Introduction	2
1.1 Smart Grid and Smart Meters	2
1.2 Research Problem	3
1.3 Research Questions	4
1.4 Research Aim and Objectives	5
1.5 Research Contributions and Significance	6
1.5.1 Local Significance	6
1.5.2 Significance for the Field of Smart Meter Analytics	7
1.6 Research Terminology	8
2 Theoretical Framework and Literature Review	12
2.1 Smart-Meter Analytics	12

2.1.1	Proposed Taxonomy	14
2.1.2	Domain I: Forecasting	16
2.1.3	Domain II: Knowledge Discovery	18
2.1.3.1	Load Profiling	18
2.1.3.2	Load Profiling Application	20
2.1.3.3	Occupant Behaviour	21
2.1.3.4	Load Disaggregation	22
2.1.4	Domain III: Energy Management	24
2.1.4.1	Demand Response	25
2.1.4.2	Home Energy Management Systems(HEMS)	26
2.1.4.3	Security and Privacy	27
2.2	A Review of Deep Learning Techniques for Smart Meter Analytics	28
2.2.1	Deep Learning and Forecasting	30
2.2.2	Deep Learning and Knowledge Discovery	31
2.2.3	Deep Learning and Energy Management	35
2.3	Exploratory Data Analysis (EDA)	36
2.4	Load Forecasting	38
2.4.1	Forecasting Horizon	38
2.4.1.1	Short- term Load Forecasting(STLF)	38
2.4.1.2	Long and Mid-Term Load Forecasting	38
2.4.2	Load Forecasting Scale	40
2.4.2.1	Meter-level Load Forecasting	40
2.4.2.2	Building Load Forecasting	41
2.4.2.3	Integrated Load Forecasting	41
2.4.3	Load Forecasting Techniques	42
2.4.4	Data Prepossessing for Load Forecasting	44
2.4.5	Forecasting Algorithm	45
2.4.6	Evaluation Metric For Forecasting	46

2.5	Time Series Methods for Load Forecasting	47
2.5.1	ARIMA(Box-Jenkins) Background	47
2.5.1.1	Seasonal Autoregressive Integrated Moving Average Model (SARIMA)	49
2.5.2	Prophet Algorithm Background	49
2.6	Chapter Summery	51
3	Methodology	53
3.1	Datasets and Preprocessing	53
3.1.1	Real-World Datasets	53
3.1.1.1	Preprocessing	54
3.2	Deep Embedded Clustering Methodology	57
3.2.1	Enhancing Load Profiling with Deep Clustering	57
3.2.2	Clustering Step	58
3.2.3	The Deep Clustering Algorithms	59
3.3	Methodology of Time Series Forecasting	66
3.3.1	Seasonal ARIMA (SARIMA)	66
3.3.1.1	Proposed Framework	67
3.3.2	Prophet	73
3.3.2.1	Methodology	73
3.3.2.2	Data Prepossessing	73
3.3.2.3	Merge and Split	74
3.3.3	Forecasting Evaluation	74
3.4	Chapter Summery	76
4	Results and Discussion	79
4.1	Results and Discussion for EDA	79
4.1.1	Investigation of Factors Affecting Consumption	83
4.2	Results and Discussion for Embedded Load Profiling	89
4.2.1	Experimental Protocol	89

4.2.2	Results of Embedded Deep Load Profiling	92
4.3	Experimental Result for STLF	96
4.3.1	Experimental Result for SARIMAX	97
4.3.1.1	Evaluation	99
4.3.2	Experimental Result for Prophet	100
4.3.3	Forecasting Evaluation	102
4.4	Chapter Summery	102
5	Conclusions and Future Work	105
5.1	Conclusions	105
5.2	Future Work	109
	Bibliography	109
	Abstract (in Arabic)	

List of Tables

2.1	Articles on Energy Management Functions	25
2.2	HEMS Techniques	26
2.3	Number of deep learning papers per taxonomy task per year	29
2.4	Publications with deep learning techniques per task in knowledge discovery	31
2.5	Neural Network Architectures for Deep Clustering	33
2.6	Deep learning research for tasks of energy management domain	36
2.7	Publications of various load forecasting techniques against scale	46
3.1	Data description	55
3.2	Different Approaches for Dealing with Big-Data	56
3.4	Static Properties of Data Set	68
3.5	ADF Test in daily Data	70
4.1	Silhouette Index (SI) results for classical vs deep clustering methods	94
4.2	Silhouette Index (SI) results for CKSC and the proposed framework for the Irish dataset	97
4.3	The SARIMAX Evaluation	100
4.4	Forecasting for 7 fold	103

List of Figures

2.1	An overview of the smart meter Analysis workflow	13
2.2	Number of General Smart-Meter Analytics Review Papers Per Year	15
2.3	A Taxonomy of Smart Meter Analytic	15
2.4	Number of articles per year based on deep learning architecture	30
2.5	Load Forecasting Approach in scale and horizon	42
2.6	Load forecasting as a data science problem	43
2.7	Semantic view of prophet, forecast in scale	50
3.1	Summary of the deep load profiling framework steps	58
3.2	Overview of the autoencoder architecture	60
3.3	Proposed framework for SARIMA	67
3.4	Decomposition of data set	69
3.5	Auto correlation and PCAF graph	72
3.6	Proposed methodology for Prophet	74
3.7	Holidays data set	75
3.8	Time series Splitting with k=7	76
4.1	Acorn Count	80
4.2	The tariff demonstration	81
4.3	The consumption per month	81
4.4	The consumption per month and Day	82

4.5	The average consumption per month	82
4.6	Consumption per year (2011-2014)	83
4.7	Consumption per month of each year (2011-2014)	84
4.8	London weather temperature	84
4.9	Temperature consumption correlations	85
4.10	OLS Result	86
4.11	Weather factors correlation	86
4.12	Calendar Event Regression	87
4.13	Calendar Analysis	88
4.14	Loss for different learning rates	91
4.15	Elbow methods on Silhouette Index for Hierarchical clustering on sample data.	93
4.16	Loss function and SI value per iteration.	95
4.17	Load profiles of each cluster obtained by DynAE	95
4.18	Load profiles of each cluster obtained by hierarchical clustering	96
4.19	Comparison of Silhouette Index for the Proposed Framework and Sun et al. 2020 on the Irish dataset	97
4.20	Model Coefficients	98
4.22	Predicate vs real DATA	98
4.21	Diagnostic tool with $s=7$ in daily data set	99
4.23	Ensemble Empirical Mode Decomposition result	100
4.24	Seasonality on week and month	101
4.25	Forecasting with Prophet	101

List of Abbreviations

ACF	Auto-Correlation Function
AE	Autoencoders
AMI	Advance Metering Infrastructure
AR	Autoregressive Model
ARIMA	Autoregressive Integrated moving average
ARMA	Autoregressive Moving-average model
CNN	Convolutional Neural Networks
DBN	Deep Belief Networks
DEC	Deep Embedding Clustering
DI	Davies-Bouldin Index
DynAE	Dynamic Auto Encoder.
EDA	Exploratory Data Analysis
GAN	Generative Adversarial Networks
HVAC	Heating, Ventilation, and Air-Conditioning
IDEC	Improved Embedded Clustering
KL	Kullback–Leibler
LTLFs	Long term load forecasting
MAE	Mean Absolute Error

MAPE	Mean Absolute Percentage Error
MTLF	Mid term load forecasting
OLS	Ordinary Least Squares
PACF	Partial Auto-Correlation Function
PCA	Principal Component Analysis
RNN	Recurrent Neural Networks
SARIMA	Seasonal Autoregressive Integrated Moving Average Model
SARIMAX	Seasonal Auto Regressive Integrated Moving Average with eXogenous factors
SEAT	Seasonal Extraction in ARIMA Time Series
STL	Seasonal and Trend Decomposition Using Loss
SI	Silhouette Index/Coefficient
STLF	Short- term Load Forecasting
SVM	Support Vector Machine
TLP	Typical Load Profile
VAE	Variational Auto Encoders

Chapter 1

Introduction

1.1 Smart Grid and Smart Meters

1.2 Research Problem

1.3 Research Questions

1.4 Research Aim and Objectives

1.5 Research Contributions and Significance

1.5 Research Terminology

Chapter 1

Introduction

This introductory chapter will provide an overview of relevant concepts such as the smart grid and smart meters. It will highlight the potential role of smart meter analytics in sustainable energy. Then the research questions will be stated along with the aims and objectives of this thesis.

1.1 Smart Grid and Smart Meters

The 21st century's economy grows increasingly dependent on energy that utilizes smart power transfer methods. A smart grid is an updated form of an electrical power system, which uses smart meters as one of its main components. This smart infrastructure is a requirement for sustainability, as it is the first building block for sustainable use of energy. Power management, forecasting, planning, and many other functions can only be implemented through the efficient use of this smart meter data.

Smart meters detect the amount of power utilized by consumers on a regular basis and send the data to a data collector over a Local Network (LAN). The data is retrieved by the collector and transmitted to the utility's central collecting sites. The large volume of raw data generated by smart meters needs to be stored, cleaned, and analyzed; in order to

discover commercially usable knowledge and gain insight. This wealth of information can be utilized by individuals, the energy industry, service providers, as well as regulators, and planners. The significant potential of the massive amounts of data collected by smart meters is heavily dependent on the capacity to store, manage, analyze, and extract valuable information from this data. Smart meter analytic refers to the conversion of the massive amounts of data flowing through different layers to actionable knowledge.

1.2 Research Problem

Smart meters generate an unprecedented data in volume, speed and complexity; some of the characteristics associated with Big Data. The data generated by smart meters are also temporal in nature making them even more difficult to handle as the time horizon and the temporal frequency are issues which need to be resolved when modelling these data. Thus, the issues related to smart meter data analytic are the issues associated with big data analytic coupled with that of the time series data. Data must be prepossessed; cleansed, scrubbed of anomalies, and free of bias. In addition, it must be structured appropriately to be effectively analysed. One of most important task of smart meter analysis is Load profiling which group the consumers based on their consumption pattern.

Hence, Most published works in load profiling used partition clustering which are parametric in nature and are thus subject to the researcher's choice (Panapakidis and Christoforidis, 2017). At the same time, partition clustering does not adequately model the volatile nature of the data that is seen in smart meter readings. On the other hand, traditional clustering algorithms such as hierarchical clustering are computationally expensive for temporal large data sets such as smart meter data (Wang et al., 2015).

One suggested solution for this limitation is to apply dimensionality

reduction prior to clustering. Although the appropriate dimensionality reduction methods will speed up the clustering process and reduce the execution time and hence provide adequate clustering quality, there are very few efforts in the comparison of different feature extraction methods in smart meter analysis. These all justify the importance of investigation and imply the value of attempting new techniques such as deep learning for the load profiling task.

Thus, the research problem is to firstly identifies different Knowledge extractables from smart meter data. Secondly efficiently group the consumer base of their consumption patterns, since this is a primary task in many domains of smart meter analytics. Efficiency in this research is measured in terms of more meaningful clustering results which is assessed with evaluation metrics along with visualization. For more details on the analytics framework that will be considered in this research, please refer to section 2.1 below. To plan the electricity providers need to forecast needed consumption in a different zone and scale so, the third research problem is to find integrated consumption needed for the day ahead.

1.3 Research Questions

Although the some aspects of smart meter analysis such as prediction of feature needed electricity was traditionally in the domain of electrical engineering, but smart meter reading creates a lot of opportunities for data scientists to explore and extract knowledge. Smart meter produce multi dimension big volume data base, which is represent as matrix $m \times n \times D$. Where m is number of consumer and D is number of days and n is granularity of data which can be varied from 24 (every 1 hour) to 86400 (every one second).

Consumption data has very stochastic nature and can claim from

minimum to maximum amount in less than a minute. Data collected from smart meter transfer to collector by radio frequency and then through the power line carrier to distributor which can save the consumption data in the central data base. This infrastructure raised issues about privacy and in batch or stream processing. To extract knowledge from data smart meter Analysis is implied. Smart meter analysis is a challenging computational problem due to nature of data.

In addition to that, The trend in smart meter analytic is to process consumption data along with other information like economic data base, metro logical data and etc. This raise another aspect of big data to problem Multivariate data.

This thesis will formulate smart meter analysis as computational problem. In this contrast the research questions are :

- Q1. How to use vast amounts of smart meter data to promote and improve the electricity grid's efficiency and sustainability.
- Q2. What are computational methods to segment consumer base of multivariate big data produced by smart meter efficiently.
- Q3. How to forecast the consumption needed on the day ahead.

1.4 Research Aim and Objectives

This thesis aims at providing efficient computational methods for clustering large amounts of temporal data sets by implementing new techniques such as deep learning and time series modeling.

By considering the research questions the objectives of this thesis are formulated as follows:

- Provide a taxonomy for smart meter analysis.
- Perform Exploratory Data Analysis (EDA) to determine the nature and pattern of data in different time zone.
- Apply deep embedded clustering algorithms for electrical load profiling.
- Investigate the effectiveness of deep embedded clustering algorithms for electrical load profiling by comparing it to traditional clustering methods.
- Apply time series techniques for load forecasting.

1.5 Research Contributions and Significance

In many countries the smart grid is replacing traditional electricity's grid. This transformation is because of important role that smart grid plays on sustainability of energy, Smart meter as part of a new grid system recently introduced in many countries. In Bahrain, Electricity and Water Authority (EWA) started installing smart meters in most households. The following section discusses the importance of smart meter analysis in the local context.

1.5.1 Local Significance

Over the last decade Bahrain's peak electricity demand has increased rapidly due to the following circumstances:

1. Highly subsidized electricity prices that are lower than the benchmark price.
2. Rapidly expanding population due to high fertility rates and rising expatriate population.

3. Increase in national income due to increased worldwide demand for gas.
4. High demand for cooling and air-conditioning during the hot summer months.
5. End of the quest for a higher level of living.

Bahrain is renowned as one of the world's most energy-intensive countries (e.g. According to Energy Administration Information (Eia) the Kingdom has an average electrical power per capita=16,724 kwh per year). Nonetheless, little is known about its national or consumer power use patterns and customer base profiles.

Classifying the consumers based on their electricity consumption, will help in better understanding the electricity consumption pattern in the kingdom. Understanding domestic power consumption patterns provides for a better reflection of future energy system planning and allows for a variety of related assessments, such as the need for grid expansion or load forecast for electricity markets. This understanding will also be useful at the national level to provide ways for reducing energy usage per-capita. This all shows the importance of smart meter analysis in Bahrain.

As a second problem, Bahrain intends to eliminate energy subsidies and aims to adopt time-of-use or dynamic pricing mechanisms to recoup the cost of power delivery. At the start of 2018, Bahrain removed the subsidies for the expatriate population, this research endeavours to find the relation between tariff and consumers patterns by comparing the typical expatriates' load curve (without subsidies) and that of the Bahrainis (with subsidies).

1.5.2 Significance for the Field of Smart Meter Analytics

One of the most critical tasks in smart grid analytics and sustainable energy practice is load profiling. The performance of load profiling is usually

an input to many other smart meter analysis tasks, such as load analysis and demand response. Load profiling is also paramount to improving the performance of other sustainable energy management tasks such as Short Term Load Forecasting (STLF) and Home Energy Management Systems (HEMS). All of these reasons emphasize the need for better load profiling algorithms.

This thesis proposed an enhanced load profiling framework that utilizes deep learning for the clustering step. While deep learning has recently been applied for other realms of smart meter analysis, such as load forecasting, load disaggregation, and dynamic pricing; it has rarely been used for improving the performance of load profiling. This thesis covered this gap in the literature by formulating load profiling as a clustering problem and solving it with various deep learning clustering algorithms.

1.6 Research Terminology

1. **Advanced Metering Infrastructure:** AMI communicates the consumption of electricity of households internally in the form of kilowatt-hour (kwh) with utility. The information sends to and from the smart meter, the heart of AMI. This allows two-way communication of information between consumers and suppliers. This includes information about the quality of electricity provided by the supplier and temporal consumption measurement of consumers. Smart meter features are listed as (Mohassel et al., 2014):

- Captures distributed generation and real-time or near real-time energy usage.
- Capability of remotely reading.
- Facilities for remotely control and cut off supply

- Possibility to link to gas and water supply
 - Capacity for capture events.
2. **The Smart Grid** is a modern electric power grid infrastructure for enhanced efficiency and reliability through automated control, high-power converters, modern communications infrastructure, sensing, and metering technologies, and modern energy management techniques based on the optimization of demand, energy, and network availability (Gungor et al., 2011).
 3. **Load Profiling** is employed to determine a customer's energy demand trend over a given time frame, which is considered as the typical *load curve* of the group.
 4. **Feature Drift** which takes place when different objective functions compete inappropriately or unfairly against each other (Mrabah et al., 2020a).
 5. **Feature Randomness** is defined as the deviation of the gradient of the real supervised objective function compared with the pseudo-supervised gradient (Goodfellow et al., 2014).
 6. **Ordinary Least Squares (OLS)**: For estimating the unknown parameters in a linear regression model, ordinary least squares (OLS) is a sort of linear least-squares approach. By minimizing the sum of the squares of the differences between the observed dependent variable (values of the variable being observed) in the given dataset and those predicted by the linear function of the independent variable, OLS chooses the parameters of a linear function of a set of explanatory variables.

7. **Min-Max Normalization** is a popular strategy for scaling data that are used to train a deep neural network.
8. **Hyper-parameters** are model parameters that have their values pre-programmed. They're all numbers that are established before the training process and have an impact on the model's behavior.
9. **Grid Search** is the most basic hyper-parameter tuning approach. The hyper-parameter tuning is crucial because a model with variable hyper-parameters is truly a separate model, and its performance may suffer. The hyper-parameters' domain is organized into a discrete grid. Then, using cross-validation, a combination of values from this grid is tried, and various performance indicators are calculated. The ideal combination of values for the hyper-parameters is the point on the grid that maximizes the average value in cross-validation.

chapter summery

This chapter provides an introduction to the aims, objectives, and significance of this research. In the next chapter literature review of the smart meter will be provided and a survey for each task will be disused.

Chapter 2

Theoretical Framework and Literature Review

2.1 Smart-Meter Analytics

2.2 A Review of Deep Learning Techniques for Smart Meter Analytics

2.3 Exploratory Data Analysis (EDA)

2.4 Load Forecasting

2.5 Time Series Methods for Load Forecasting

2.6 Chapter Summery

Chapter 2

Theoretical Framework and Literature Review

This chapter will provide a theoretical framework and literature review. First smart meter analysis will be defined, then a taxonomy will be proposed. Each task of the proposed domain will be explained and formulated from a data science perspective, Then different algorithms for solving the task will be introduced. In the next section, a survey of deep networks will be provided. The two last sections of this chapter will discuss exploratory data analysis and load forecasting in detail.

2.1 Smart-Meter Analytics

The significant potential of the massive amounts of data collected by smart meters is heavily dependent on the capacity to store, manage, analyze, and extract valuable information from this data, this fact is reflected in published review paper as demonstrated in Figure 2.2. Smart meter analytic refers to the conversion of the massive amounts of data flowing through different layers to actionable knowledge. The data science workflow outlined in

Figure 2.1 shows the main steps in smart meter data analytics.

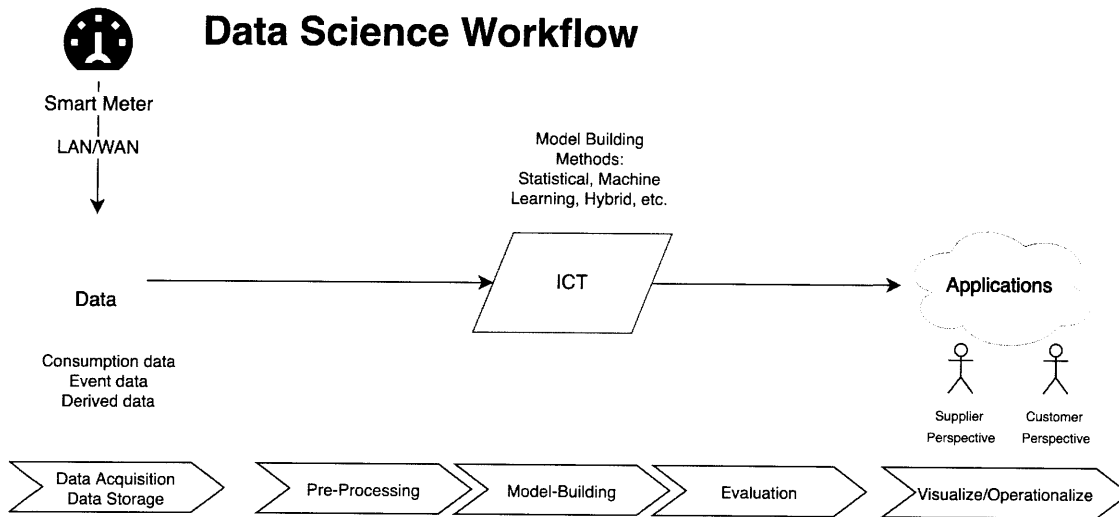


Figure 2.1: An overview of the smart meter Analysis workflow

1. *Data Acquisition*: Smart meters generate data and communicate it to storage areas through LANs and WANs. Other related data might also be stored (e.g. weather data, national holidays, etc.)
2. *Pre-Processing and Exploration*: The data is pre-processed through data wrangling and munging techniques. This includes dealing with missing data and detecting plus filtering anomalies. The data must be transformed to a format that can be processed by the machine learning model.
3. *Model-Building*: Suitable models are selected and trained on the pre-processed data.
4. *Evaluation*: The model is evaluated based on some ground truth that is established by the data scientist.
5. *Visualize/Operationalize*: The final step is performed once good performance is established by the model-building step. It includes

visualizing and presenting the results to the stakeholders, and operationalizing the process for future data.

Looking through the past literature, there is a need for a comprehensive survey that covers all aspect of smart meter data analytics, None of the previous review papers covers all aspects of smart meter analytic. This thesis covered this gap by providing a broad overview of the current research spectrum while identifying future challenges for smart meter data analytics within a detailed taxonomy.

2.1.1 Proposed Taxonomy

The proposed taxonomy identifies the main domains within the field, as well as the various functions that are conducted within each domain. It includes a survey of the various algorithms and techniques that were proposed by the researchers to achieve these functions. This classification will enumerate the current research trends and provide a better understanding of the available opportunities for researchers. In addition, very few review articles explored novel approaches such as deep learning for smart meter analysis (Yin et al., 2020; Ibrahim et al., 2020). Overall, the review papers can be divided into two categories: reviews about special domains of smart meter analytics (Amasyali and El-Gohary, 2018; Khuntia et al., 2016b; Sun et al., 2010; Kakran and Chanana, 2018), and reviews about specific techniques used within these domains (Deb et al., 2017; Wei et al., 2018; Zor et al., 2017).

Recently, a taxonomy for smart meter analysis was proposed but it did not include all tasks in knowledge discovery or management domain (Wang et al., 2018b).

To remedy the lack of a comprehensive taxonomy of the field of smart meter analytics, an analysis of around 200 papers from the literature resulted

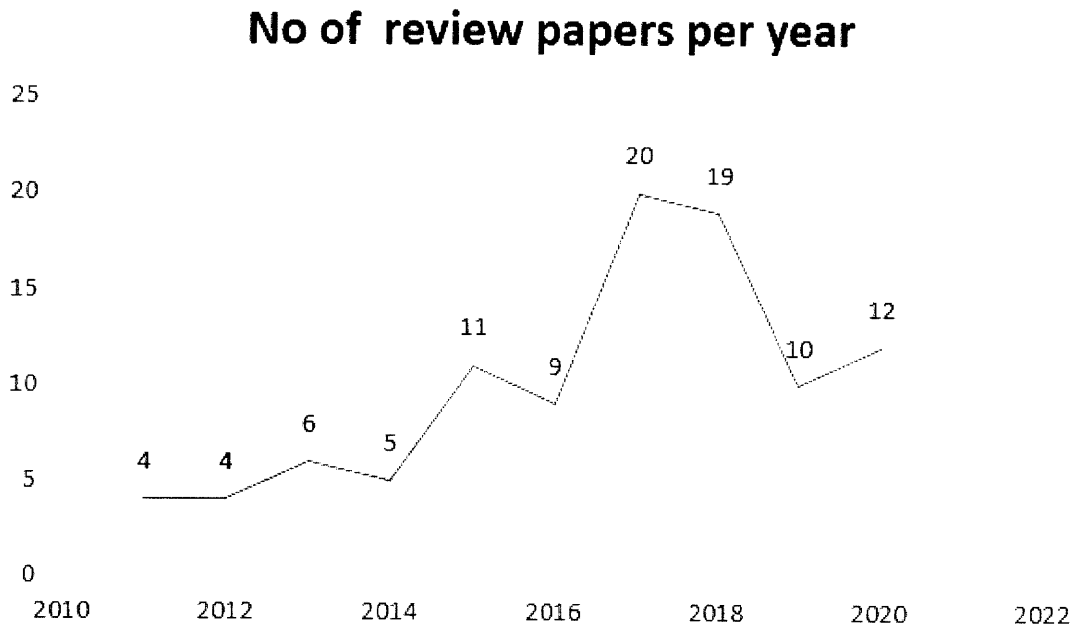


Figure 2.2: Number of General Smart-Meter Analytics Review Papers Per Year

in a clear *taxonomy* of the domains and tasks within the field. The proposed taxonomy is presented in Figure 2.3 and it includes three main domains: Load forecasting, Knowledge Discovery, and Energy management.

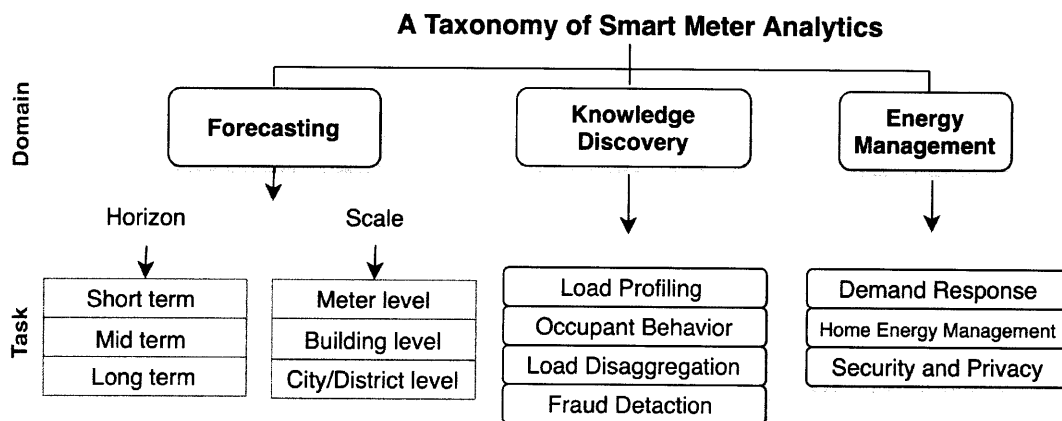


Figure 2.3: A Taxonomy of Smart Meter Analytic

2.1.2 Domain I: Forecasting

Load Forecasting refers to the prediction of future electricity demand in different horizons and scales. *Knowledge discovery* refers to new information extracted from raw databases of smart meter data by applying techniques such as machine learning, pattern recognition, classification, clustering, statistical analysis, etc. While *Energy Management* refers to the processing of smart grid data in order to deploy functions that are related to the management of grid resources, such as: improving energy efficiency, reducing the cost of energy production, response to demand fluctuation, the introduction of dynamic pricing, etc.

In the next sections, we discuss each of the three domains with the different tasks contained in each of them, and summarize related research from the literature.

In today's competitive energy market, more precise projections on various scales, ranging from a single smart meter (the end-user) to an entire grid system, are increasingly required. An extended review of load forecasting techniques was represented in (Iskandarnia et al., 2020). As a result, load forecasting is regarded as a crucial component of the electric industry's planning process. Furthermore, load forecasting is critical for customers to optimize their use of electrical energy. It can be performed on different scale (the size of the unit that the load forecast is carried out at) and on a certain horizon (time period of forecasting); which can be divided into (Khuntia et al., 2016a):

- *STLF* Short term load forecasting : a minute or a hour or a day.
- *MTLF* Mid term load forecasting : 1 weeks to a year.
- *LTLF* Long term load forecasting : 1 to 50 years.

The output of forecasting algorithms in the STLF zone is used to assess power system security and to schedule generation. The major goal of MTLF

is the scheduling of maintenance and the efficient functioning of power systems. While the LTLF is an important tool for planning and forecasting future power network circumstances and future planning.

The forecast's scale can be divided into the following levels:

- *Meter-level load forecasting* (Kong et al., 2017): This type of forecasting refers to domestic load, user-level load, and dwelling load; and is considered as a relatively new area. Because of the volatile nature of domestic power load profiles, forecasting at the meter-level is very challenging. To the best of our knowledge only few studies were published for short-term load forecasting at the meter level that use data granularity provided by smart meters (Kuster et al., 2011; Yildiz et al., 2011; Dedinec et al., 2016; Marino et al., 2016).
- *Commercial building level forecasting*: There are two approaches for building load forecasting in the literature: (i) physical models (also known as white-box models) rely on thermodynamic rules for detailed energy modeling which is out of scope for this paper, and (ii) data-driven energy consumption prediction modeling that learns from historical data for prediction (Amasyali and El-Gohary, 2018). The grey box approach was proposed by (Wei et al., 2018) for data driven and engineering methods.
- *Integrated load forecasting (district, region, and country level)*: Three different approaches to aggregated load forecasting: (i) forecasting meter level load and then adding it to find the aggregated load, (ii) adding demand to forecast aggregate load, and (iii) clustering customers to perform forecasting for each cluster. (Ma et al., 2017)

Load Forecasting Techniques

Considering the data science approach, the framework for forecasting will include the use of exploratory analysis and statistical methods to understand trends within the given data sets; and applying smoothing techniques in order to remove noise prior to applying the algorithms. Load forecasting is done based on this approach discussed in details (Iskandarnia et al., 2020). In general, this framework includes: Pre-Processing and Data Wrangling, forecasting Algorithms, and evaluation of forecasting techniques. Statistical techniques along with machine learning techniques are widely used in load forecasting (Eskandarnia and AlHammad, 2020; Eskandar et al., 2020). A wide literature review for load forecasting will be provided in section 2.4 since integrated STLF is one of the objectives of this thesis.

2.1.3 Domain II: Knowledge Discovery

The extraction of new information from data is the second area of smart meter data analytics. Techniques used in this area range from traditional data mining to deep learning and meta-heuristics. The following subsections survey different functions within the area of knowledge discovery.

2.1.3.1 Load Profiling

A *load profile* is a graph/curve that represents the variation of electrical usage over time. Power companies traditionally create aggregate load profiles and create *Typical Load Profiles* (TLP) and group their consumers and model each group by a *Standard Load Profile* (SLP). By employing the opportunities provided by smart meters, customers can now be classified by analyzing their load curves. Load profiling helps in grouping the consumers based on similarity in their consumption behavior, and subsequently, produces the typical load profile for each group.

Traditionally, load profiling was implemented using engineering or statistical methods (McLoughlin et al., 2013). However, in the smart grid era load profiling problem can be formulated as an unsupervised machine learning task, and hence, it is completely data-driven. Load profiling is mostly performed via clustering-based methods (Yang et al., 2013). Cluster-driven load profiling attempts to group similar consumers into a single cluster based on similarities in consumption patterns. Based on the available literature, load profiling can be defined as clustering high dimensional data with no labels, and it is usually performed in three main stages:

- *Data preparation:* the dataset is preprocessed and dimension reduction is performed to deal with the problem of high dimensionality and compress the data. This stage seeks to map raw data into a new feature space where the resulting data can be separated by existing classifiers more easily. Traditional techniques to data transformation include linear and non-linear transformations such as Principal Component Analysis (PCA), Kernel, and spectral methods. feature reduction techniques implemented for load profiling task is explained in (Wang et al., 2015).
- *Load curve clustering:* K-means, Fuzzy Clustering (FCM), Hierarchical Clustering Process, and Self Organize Maps (SOM) are some of the traditional and most widely used clustering algorithms in load profiling. The advantages and disadvantages of each approach were presented in (Yang et al., 2013).
- *Evaluation:* Smart meter readings are unlabeled and no structural knowledge about the data set is available. As a result, calculating the optimum number of clusters and evaluating the accuracy of clustering outcomes is a challenging task. The most popular method for

determining the optimum number of clusters is to run the clustering algorithm several times while attempting various numbers of clusters, and then choose the number of clusters that produces the best results in a predefined evaluation criterion function. Clustering validity is usually evaluated with visualization, goodness of fit, as well as scalar measurements functions (Aghabozorgi et al., 2015). The clustering quality also can be assessed with a clustering evaluation index such as the Silhouette Coefficient (SI), and Davies-Bouldin Index.

2.1.3.2 Load Profiling Application

Load profiling is considered as a key tool for smart meter analytics as it produces input for the other functions of the knowledge discovery domain. This highlights the significance of load profiling and justifies the second research question (i.e. Q2) of this thesis.

Load profiling applications are classified as follows:

- *Energy management task:* Load profiling can be used as input to electricity management applications such as demand response and security monitoring. Kwac et al. (2014) applied load profiling to create a dictionary for consumer patterns. The TLP of each consumer was matched by representative load shapes in the dictionary, to target consumers for demand response. Load profiling can also be used for detecting unusual usage and initiating warnings for such usage (Ozawa et al., 2016). It also can be used for other functions such as tariff setting, where consumers are first divided into different categories based on their individual information by a clustering algorithm; then, a pricing model is presented as a nonlinear programming problem from the perspective of a load service company, with the goal of minimizing the overall operating cost.

- *Cluster-based aggregate load forecasting:* Load profiling can improve the accuracy of forecasting if done beforehand. When load forecasting is performed on an aggregate scale, consumers with common load trends are identified, and forecasting can be extended to each group (Al-Wakeel and Wu, 2016; Li et al., 2016).
- *Other tasks in Knowledge discovery:* Load Profiling improves the accuracy and efficiency of other tasks in this domain. For example, it feeds information to outlier detection algorithms and to improve fraud detection tasks (Villar-Rodriguez et al., 2017). It may also be used prior to the occupant and load analysis tasks. Consumer segmentation is usually accomplished using a combination of clustering and classification approaches. Following the grouping of users based on the similarities of their actions by load profiling, supervised methods (classification) can be used to extract further information for decision-making purposes (Vercamer et al., 2015).

Data Driven Load Profiling Techniques

Load profiling attempts to group similar consumers into a single cluster based on similarities in consumption patterns. Based on the available literature, load profiling can be defined as clustering high dimensional data with no labels, and it is usually performed in three main stages which will be discussed in section 2.4.1.

2.1.3.3 Occupant Behaviour

Occupant behaviour refers to the discovery of elements that impact the load curve and, as a result, the consumer's lifestyle. To do so, smart meter readings are combined with other related data sets, such as building characteristics, weather statistics, calendar information, and etc. From the

data science perspective this is a clustering and feature extraction task.

Data Driven Occupant Behaviour Techniques

In the literature, various techniques were used for occupant behavior identification but all of them have two main steps: discovering frequent itemsets and generating association rules based on the frequent itemsets. For instance, Hidden Markov Models (HMMs) was used by (Muratori and Rizzoni, 2016) and Association rule mining (ARM) was implemented by (Mahmud and Sant, 2017) for this task.

Another techniques for doing this task is motif discovery. The goal of motif discovery is to find frequently recurring subsequences in a time series. A univariate time series is first segmented into distinct subsequences depending on user-specified window size. Certain distance measures are employed to assess the correlations between different pairs of subsequences. Motifs are discovered as sub-sequences with a significant degree of similarity.

2.1.3.4 Load Disaggregation

Load disaggregation is accomplished by examining the load of a household to identify the various equipment used by the occupant. Giving end-users direct and real-time input on the appliance's usage can lead to energy savings. Furthermore, load disaggregation can be used to further describe the lifestyle of the consumers (Hosseini et al., 2017).

According to (Esa et al., 2016), there are two main approaches to load disaggregation :

- **Intrusive Appliance Load Monitoring (IALM):** By using sensors that are attached to every appliance in the house, IALM can produce the operating condition precisely.

- **Non-Intrusive Appliance Load Monitoring (NILM):** This technology involves the practice of disaggregating the household's total electrical load measured into distinct appliance signals (Basu et al., 2015).

Data Driven for Load Disaggregation

The NILM method consists of two steps: extraction and identification. The extraction phase transforms the smart meter readings into a collection of characteristics that are utilized for load recognition. Following categorization, each class will be allocated with an electrical signature. The detected events are connected with a loads signature database in order to determine the load state (on or off) and track the appliance usage over time. The load is identified by using machine learning techniques (Azaza and Wallin, 2017; Esa et al., 2016).

Fraud Detection

Energy fraud detection is a crucial component of the smart grid. Energy fraudsters can manipulate smart meter data in two ways: they consume more energy while reporting less, or they report more energy for certain meters to obtain additional advantages from the utility. Physical tampering, such as illegal tapping, bypassing meters, firmware modification, and cyber attacks against the AMI infrastructure are used to manipulate smart meter reading. Non-technical loss (NTL) in power systems is mostly the result of electricity theft, billing problems, and unpaid bills.

Data Driven Fraud Detection

Machine learning and data mining have been widely utilized by researchers for comprehensive intelligent data analysis in order to recognize typical patterns of behavior so that deviations may be recognized as anomalies.

Fraud detection consists of the following stages (Messinis and Hatziargyriou, 2018):

- *Pre-possessing*: Raw data goes through the pre-processing stage to extract features for the NILM algorithm. Since feature selection has a direct effect on algorithm output, feature selection algorithms can be used for defining an optimal set of features (Vercamer et al., 2015).
- *Machine learning*: Two main categories of machine learning are used in: supervised methods (for labeled data) including SVM, Artificial Neural Networks (ANN), Optimum Path Forrest (OPF) and the Naïve Bayes Classifier; and Unsupervised methods (for unlabeled data) which include self-organizing maps and regression methods.
- *Evaluation*: Depending on the type of algorithm used, different metrics are used for assessing the performance of detection methods. Besides metrics that evaluate the performance of machine learning algorithms, other metrics are also defined to quantify the effects of the detection methods.

From the data science perspective this is a clustering and feature extraction task.

2.1.4 Domain III: Energy Management

Smart grid technology provides innovative power approaches such as renewable energy sources, energy storage systems, car charged batteries, and so on. Although the use of these new approaches provides several benefits, it also poses certain problems for utilities. Energy management strategies are critical to overcoming these problems.

Different tasks within the domain of energy management are discussed briefly in the following sections. Table 2.1 provides a summary that can

Table 2.1: Articles on Energy Management Functions

Function		References
Demand Response	Price-Based	Ambreen et al. (2017)
	Incentive-Based	Mahmud and Sant (2017) Shafie-khah et al. (2017)
Home Energy Management Systems		Muratori and Rizzoni (2016) Tasdighi et al. (4 01)
Dynamic Pricing		Assaf et al. (4 06)
		Shuai et al. (2017)
Microgrid		Nafi et al. (2016)
		Shuai et al. (2017)

be used for further study of the topic of energy management, providing a classified set of references addressing the subject.

2.1.4.1 Demand Response

This task's purpose is to reduce the utility's overall load through lowering consumer energy consumption. One approach to do this is for consumers to alter their electricity consumption in response to the price set by suppliers. Consumers are more motivated to limit energy consumption when prices are dynamic, especially during peak hours.

The ultimate purpose of dynamic pricing is to reduce peak to average

energy demand, by using various price strategies,

2.1.4.2 Home Energy Management Systems(HEMS)

HEMS is one of the most important parts of smart cities since it can regulate household loads, schedule appliances, and create the ideal load. From the programmer's point of view, HEMS is a scheduling problem that aims to produce optimal production and consumption schedules by considering multiple objectives such as reducing energy costs, environmental concerns, and consumer comfort (Rodriguez Fernandez et al., 6 01).

Home Energy Management Techniques

There are many algorithms in the literature for solving multi-objective optimization problems in general; among them, the two most popular are game theory and optimization. Table 2.2 gives a summary of work conducted by these methods.

Table 2.2: HEMS Techniques

Technique	References
Game Theory	(Assaf et al., 4 06)
	(Yu and Hong, 2017)
Optimization	(Mahmud and Sant, 2017)
	(Shafie-khah et al., 2017)
	(Tariq et al., 2017)

2.1.4.3 Security and Privacy

Smart grid devices save and analyze vast quantities of data in cloud-based storage units. This connection between smart grid IOT devices and cloud storage providers is subject to several security risks. A weak security policy allows intruders to seize control of power generation networks, steal customer information, or inflict more harm. Furthermore, the smart grid offers two-way real-time communication, it gathers sub-hourly power consumption from smart meters and sends alerts to smart meters if required. Since the communication between the smart meter and provider is mostly through open networks, they introduce many vulnerable points for network-based threats. Fine grained smart meter data may be used to infer sociodemographic information. Consumption data collected and communicated by smart meters may disclose sensitive information about households such as habits, lifestyle choices, and even religious views.

These all emphasize the growing interest in privacy-enhancing technology in the smart meter analysis since providing data protection along with smart meter analysis is crucial. In literature, cryptographic strategies for ensuring privacy-preserving smart meter data management under the trusted operator model, as well as privacy-preserving solutions for data processing under the non-trusted operator model, were investigated (Asghar et al., 2017).

Differential Privacy (DP) methods were recently used to secure personal information. Differential privacy is a technique that allows researchers and database analysts to extract important information from databases containing people's personal information without revealing personal information about individuals. For example, in (Soykan et al., 2019) DP property added to forecasting implementation through benefiting from Google's TensorFlow Privacy library (Soykan et al., 2019).

2.2 A Review of Deep Learning Techniques for Smart Meter Analytics

Many of the tasks in smart meter analytic literature follow the data science methods and rely on machine learning techniques. Deep learning models outperform shallow machine learning models and traditional data analysis techniques in many applications, this justifies why utilizing deep learning has been gaining momentum in the past few years for data science tasks.

The literature review showed there is a need for advance techniques such as deep learning for smart meter analysis. This is because Today high-performance computing and many cloud base platforms made it possible to broadly use the deep net to extract more actionable knowledge and pattern from massive data collected by the smart grid. Also, deep learning has the ability to extract important features for analysis tasks. This encouraged academics to look at deep learning architectures that are ideal for smart meter analytic.

A denotative research analysis was performed to determine the deep net architecture used in each task, while all the domains of the taxonomy (see Figure 2.3) were considered during data collection. The papers were then subjected to a comparative analysis to decide the paper that underpins each tasks (Table 2.3).

Following the taxonomy presented in Figure 2.3, a bibliometric analysis of around 100 articles is presented in Table 2.3 which shows the number of papers published annually in each domain. As Table 2.3 shows, the trend of utilizing deep learning has been gaining momentum in the past few years, even though it has receded in 2021. Deep learning techniques were mainly used for the load forecasting domain and other specific tasks (i.e. load disaggregation and fraud detection) as demonstrated in Table 2.3.

On the other hand, Figure 2.4 presents the number of articles published

Table 2.3: Number of deep learning papers per taxonomy task per year

Domain	Task	2016	2017	2018	2019	2020	2021	Total
Load Forecasting	Meter Level	1	2	3	6	13	1	25
	Building Level	0	1	1	2	4	0	8
Knowledge Discovery	Load profile	0	0	0	0	0	0	0
	Occupant Behavior	1	1	1	4	3	0	10
	Load Disaggregation	0	5	3	8	8	1	24
	Fraud Detection	1	0	2	7	6	0	16
Energy Management	HEMS	0	1	1	4	3	0	9
	Demand response	0	1	0	1	1	1	4
	Security and privacy	0	1	0	7	3	0	11

per year according to the deep learning technique used to tackle the analytics problem. It demonstrates the growth in researchers' interest in deep learning as an approach for solving smart grid tasks. As shown, the number of papers kept growing, and reached a peak in 2020 as new deep learning methods were being introduced. This also demonstrates the promising results obtained by implementing deep neural network in the field of smart meter analytics.

It is clear from the analysis that represented in Table 2.3 and Figure 2.4 research articles started in 2016 by using general deep neural network architectures, and then starting from 2018 specific architectures such as Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) gained more popularity. This is due to the time-series nature of the smart meter data, as these two deep network architectures work well for learning localized temporal data. Also, Long Short-Term Memory gained a lot of momentum since 2017, due to its effectiveness in learning localized temporal patterns in time-series data. There was noticeable drop in deep learning articles in 2021, which may be due to the need for more open data and use of novel techniques. In the following subsection deep learning for each domain will be discussed.

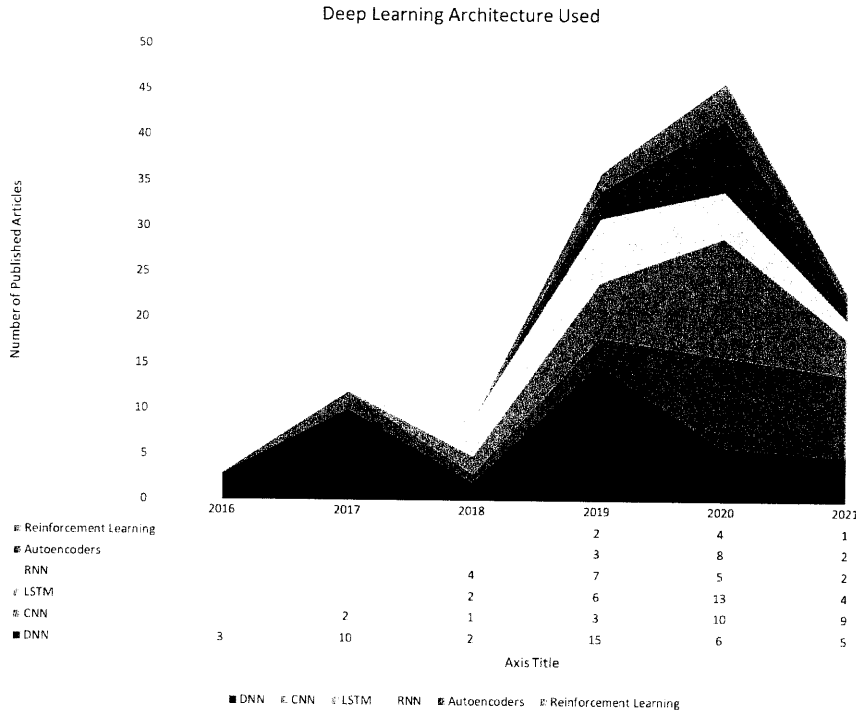


Figure 2.4: Number of articles per year based on deep learning architecture

2.2.1 Deep Learning and Forecasting

Among Artificial Intelligence methods, deep neural networks are considered as one of the most promising methods for short-term load forecasting (STLF) due to two critical characteristics of deep learning: the ability to learn extremely non-linear interactions and the ability to learn mutual ambiguity in data sets. This fact is also reflected in Table 2.3, which shows that several papers used deep learning for STLF at the meter level.

The deployment of deep networks at other scales and horizons (other than STLF) was not addressed in the literature. Deep learning networks architecture used for STLF include: recurrent neural networks (RNN) (Yuan et al., 2020) and Long Short-Term Memory (LSTM) (Yang et al., 2019a), Convolutional Neural Networks (Fahiman et al., 2017), and Bayesian deep learning (Yang et al., 2019b; Sun et al., 2019). An online RNN model was proposed in (Fekri et al., 2021) and an online LSTM model was proposed

by (Razavi et al., 2020). In addition to these, recently a hybrid network was also designed for STLF (i.e. LSTM network with CNN) (Madhure et al., 2020; Khan et al., 2020) and an RNN with LSTM (Bouktif et al., 2020).

2.2.2 Deep Learning and Knowledge Discovery

The extraction of new information from data is the second area of smart meter data analytic. Deep learning techniques used in this area is represented in Table 2.4.

Table 2.4: Publications with deep learning techniques per task in knowledge discovery

	Load Profiling	Load Analysis	Load Dis-aggregation	Fraud Detection
Hybrid Architecture			Chen et al. (2020) Paresh et al. (2019) (CNN-Auto Encoder)	He and Chai (2016) (CNN-RNN)
Auto Encoder		Ullah et al. (2020), Zhang et al. (2019)		Gong et al. (2020)
RNN				Kelly and Knottenbelt (2015) Nabil et al. (2019)
CNN		Wang et al. (2018a)	Garcia and Macabebe (2017)	Li and Wang (2020), (Hassani et al., 2019)
Restricted Boltzmann			Paresh et al. (2019), Singh and Majumdar (2017)	
Multi-Layer Perceptron (MLP)		Devlin and Hayes (2019a)	Devlin and Hayes (2019b)	
State-of-art networks	Farkas and Lacko (2017) Singhal et al. (2018)			

Load Profiling

Traditional load profiling methods typically suffer from the curse of dimensionality and thus perform poorly on high-dimensional smart meter readings. As a result, feature reduction and feature transformation approaches have usually been used before clustering algorithms.

Furthermore, on large-scale smart meters datasets, these approaches typically suffer from large computational complexity. This all leads to the use of more advantageous techniques in the literature such as hybrid machine learning and deep learning.

As seen in Table 2.3, the use of deep learning for load profiling is a relatively recent research field. This is due to the fact that deep learning has historically been employed for supervised-learning tasks, and the use of deep learning for clustering is a novel and challenging research area in a variety of applications. While deep learning has recently been applied for other realms of smart meter analysis, such as load forecasting, load disaggregation and dynamic pricing, it has rarely been used for improving the performance of load profiling. While traditional clustering-based load profiling algorithms require a complex feature reduction step to reduce the dimensionality of smart meter readings, deep learning learning non-linear mappings that enable data to be transformed into more *clustering-friendly representations* without the need for the dimensionality reduction or feature selection steps.

As a result, in traditional methods the consistency of the obtained load profile usually suffers because of the high complexity of the data provided by the feature reduction process. Deep clustering provides more efficient load profiles by eliminating the pre-processing step and at the same time provides better and more robust clusters for the various profiles. This thesis fills this gap in the literature by formulating load profiling as a clustering problem and solving it with various deep learning clustering algorithms (Eskandarnia et al., 2022b).

Deep Learning Based Clustering

As mentioned above, traditional clustering methods suffer from computational complexity and the curse of dimensionality. Dimensionality

Table 2.5: Neural Network Architectures for Deep Clustering

Network type	Description of the architecture	Used by
Feedforward Fully-Connected Neural Network	each neuron in each layer is connected to every neuron in the previous layer, and each connection has its own weight.	(Yang et al., 2017)
Deep Belief Networks (DBN)	DBNs which are composed of several stacked restricted Boltzmann machines are generative graphical models which learn to extract a deep hierarchical representation of the input data.	(Lee et al., 2009)
Convolutional Neural Networks (CNN)	Neurons in a convolutional layer are only connected to a few adjacent neurons in the previous layer, and the same set of weights is used for every neuron. It can be trained with a particular clustering loss directly without any restrictions on initialization, and a good initialization would greatly increase the clustering performance.	(Chen, 2015; Hsu and Lin, 2017; Chang et al., 2017).
Autoencoders (AE)	One of the most important algorithms in unsupervised learning. It is a powerful method for training a mapping function that ensures the least amount of reconstruction error between the coder and data layer	(Yang et al., 2017; Mrabah et al., 2020b; Xie et al., 2016).
Generative Adversarial Networks (GAN) and Variational Autoencoders (VAE)	A GAN seek an equilibrium between a generator and a discriminator, while VAE seeks to maximize the lower bound of the data log likelihood.	(Dilokthanakul et al., 2016; Makhzani et al., 2015)

reduction and representation learning have been frequently employed to translate the input data into a function space where differentiation is simpler to solve these challenges. Deep neural networks (DNNs) may be used to train non-linear mappings that allow data to be converted into more clustering-friendly representations without requiring manual extraction or

selection of features (Aljalbout et al., 2018). This justifies the enormous interest in deep clustering in other domains.

The majority of current deep clustering methods are based on two simple ideas:

- Deep embedded representations lend themselves well to clustering.
- Clustering assignments can be used as a supervisory signal to learn embedded representations.

Based on these two ideas, clustering algorithms fall into two categories:

1. *Embedded learning and clustering carried out as distinct steps:*
Embedded learning, in particular, comes before clustering. However, after the clustering stage, the newly acquired labels are not used to learn better discriminative features.
2. *Embedded learning and clustering are carried out in conjunction:*
To group the embedded data points into distinct clusters, a traditional clustering algorithm is used. Then, based on the previously acquired clustering assignment, the encoding transformation is trained to push the embedded space to have a more clustering-friendly representation (Yang et al., 2017).

By combining autoencoders with the k-Means algorithm, Deep Embedding Clustering (DEC) (Xie et al., 2016) was one of the first methods to improve the clustering quality. It was then further improved by other novel methods (Mrabah et al., 2020a). Several deep network architectures for clustering, along with references to articles that proposed or used these techniques are presented in Table 2.5. More details of deep clustering methods will be presented in the next section in the context of load profiling.

Occupant Behaviour

Although the main application of deep learning is within supervised learning, it is often utilized in the literature for feature extraction because of its intrinsic non-linear transformation property. Thus, deep learning is potentially a solution for occupant behavior problems. Deep learning architectures suitable for feature extraction are described in Långkvist et al. (2014). Table 2.4 presents different deep network architectures used for this task.

Load Disaggregation

ILM algorithms used for smart meter readings employ a variety of machine learning and deep learning techniques. The different deep architectures used for this task are listed in Alkhulaifi and Aljohani (2020). Deep learning was recently utilized for this job and Table 2.4 summarizes the architectures used.

Fraud Detection

This has been a growing research area that can exploit the availability of smart meter data. Table 2.4 summarizes some of the architectures used for fraud detection.

2.2.3 Deep Learning and Energy Management

Table 2.6 represents different network used in this task. As Table 2.6 shows, although traditional deep learning is not applicable in HEMS, more advanced networks and techniques such as reinforced and adversarial deep learning may be used. Implementing a deep network for Demand response is a new research focus, as indicated in Table 2.3. Table 2.6 shows some of the network architectures used for demand response. In addition to

pure demand response, Long Short Term Memory (LSTM) networks were utilized to improve the accuracy of the forecasting task (Christensen et al., 2019).

Table 2.6: Deep learning research for tasks of energy management domain

	HEMS	Privacy & Security	Demand response
Deep reinforcement learning	Ye et al. (2020)	Shateri et al. (2020)	Patyn et al. (2018), Lu et al. (2018)
Autoencoders		Zhou et al. (2020), Jahromi et al. (2019)	
Recurrent Neural Networks RNN		Nabil et al. (2019)	
Adversarial Deep Learning	Marulli and Visaggio (2019)		
SDAE (Stacked Denoising Autoencoder)	Zhou et al. (2018)		
LSTM (Long Short-Term Memory)		Kong et al. (2020), Liu et al. (2020)	
Hybrid Architectures		CNN-LSTM Niu et al. (2019)	CNN-SVM Jindal et al. (2018)

2.3 Exploratory Data Analysis (EDA)

Although EDA has its roots in statistics, it is not a collection of procedures, as statistics is, but rather a philosophy about how data analysis should be done. (Tukey et al., 1977) described this attitude as a counterbalance to a long-term bias in statistical research toward creating mathematical procedures for hypothesis testing. EDA, according to (Tukey et al., 1977), was a return to statistics' basic goals of discovering and explaining patterns, trends, and correlations in data. EDA is a three-step procedure that starts with an overview, then zooms and filters, and finally details-on-

demand. An analyst must first gain a broad picture of the complete data collection. The analyst identifies “items of interest” in this overview. The analyst zooms in on the items of interest in the second stage and filters away the items that aren’t. The analyst picks an item or collection of things for “drilling down” and getting more details in the third stage. The procedure is iterative once again, with multiple returns to prior steps (Craft and Cairns, 2005). What items will be found “interesting” and deserving of further inquiry is determined by the overall purpose of the investigation.

On this foundation EDA procedure define as follow according to (Andrienko and Andrienko, 2006):

The analyst is motivated by a specific investigative aim, which drives the analysis. The aim is specified as a broad question or a series of general questions. The analyst starts by looking at what’s interesting in the data, with “interestingness” defined as relevance to the investigation’s purpose. When something fascinating is uncovered, more, more specific questions arise, motivating the analyst to investigate further. These questions have an impact on how and what details are seen. As a result, questions are crucial in EDA and can determine the sort of analysis employed. In this era, not only is the selection of a suitable approach critical for effective visualization but so is the accurate parameterization of the selected technique.

On this ground, one of the initial tasks of this thesis is to visualize a data collection to gain a clear comprehension of the data and to aid in the enhancement of the other identified goals. To improve the quality of the modeling and recognize the elements that influence consumer behavior, the data was plotted using various methodologies, and correlation was investigated using several types of plots.

2.4 Load Forecasting

Load forecasting horizon and scope are determining factors for load forecasting techniques. In following sections definition and forecast methods used for each part will be discussed.

2.4.1 Forecasting Horizon

As discussed earlier, the forecasting horizon is an important factor which affects the techniques used and may lead to the use of different parameters. In the following section, different methods of load forecasting are discussed based on the predetermined forecasting horizon

2.4.1.1 Short- term Load Forecasting(STLF)

The output of prediction methods in this time zone is an input for the power system dispatcher, assessing power system security, and generation scheduling. This emphasizes the key role of short-term load forecasting for construction of power system strategies. The accurate forecasting will guarantee reliable, secure and economic strategies. Considering the importance of forecasting in this time range, a variety of methods have been reported for solving the STLF. They can be categorized as Statistical Techniques, Knowledge-Based Expert Systems, Artificial Intelligence (AI) Technique, and Hybrid Techniques (Srivastava et al., 2016). The implementation of fuzzy logic is also reported in some cases in addition to the previous methods (Lee and Hong, 2015).

2.4.1.2 Long and Mid-Term Load Forecasting

Since the applications with long-term and mid-term planning are very similar and essentially the same, they will be discussed jointly. Important applications of these types of forecasting techniques include planning and

finance, as illustrated in Figure 2.5 within these horizons are directly related to load impacting factors, as in STLF this is related to actions taken by the consumer. In long-term forecasts since recognizing the system, need is more important than time, that need is required by system forecast of the needed capacity of the grid is more crucial (Khuntia et al., 2016a).

Although the error in forecasting in all three horizons affects the power system, since in these two horizons the system development is based on this prediction, it is more important than in other horizons. Very recently an accurate forecast in European projects (e.g. GARPUR [<http://www.garpur-project.eu/>]) leads to a better maintenance plan within the mid-term horizon, and the creation and more accurate planning as a result of the long-term horizon (Khuntia et al., 2016a). The accuracy of LTLF has a direct impact on the development of future generation and transmission planning. The outcome of overestimation, as well as underestimation in a long forecast, is that the huge investment done for the construction of new generation units or expanding the present ones, will not be a profitable investment. Thus, LTLF is an important tool for developing and predicting the potential conditions of the power grid. It also helps in calculating the needs regarding planning, designing, and constructing steps of electrical grid and distribution systems. On the other hand, MTLF is the crucial instrument for deciding about the maintenance planning and economic function of power systems. MTLF also allowed power generation companies to decide about the feasible expansion of infrastructure and allocation of available resources (Khuntia et al., 2016b). Mid-term and long-term load forecasting requires a lot of knowledge about other major indicators that may not be related directly to grid-based data or measurements. The MTLF/LTLF is related to reliability analysis so they consider some explicit factors like the prediction of economic indicators such as gross domestic product (GDP) as well as historical weather data,

and demographic data. They also take into account the other factors such as population, age, and appliances in use. Furthermore, any methods on this horizon must be able to handle this large number of factors.

2.4.2 Load Forecasting Scale

The scale is also an important factor that affects the techniques used and may lead to the use of different parameters. In the following section, different methods of load forecasting will be discussed based on the forecasting scale.

2.4.2.1 Meter-level Load Forecasting

Many significant smart grid applications, such as residential load control, renewable energy resource integration, energy storage systems, and demand response management, are closely tied to this scale. Meter level load forecasting which in the literature, refers to it as domestic load, user-level load, and dwelling load was introduced to the literature because of the use of the smart meter. This type of forecasting is very changeable due to the volatile nature of data. The horizon of meter-level load forecasting is either short-term or very short-term. Since customer behavior, is a determinant factor for the shape of the load curve, traditional methods cannot be applied in the short-term case. The early work on the prediction of household load mostly focused on historical data and weather data as the main source of load forecasting. In these researches, traditional load forecasting approaches based on time series or regression analysis were used. However, to the best of our knowledge, only a few studies have been published that leverage the data granularity afforded by smart meters for short-term load forecasting at the meter level (Ghofrani et al., 2011; Kong et al., 2017).

2.4.2.2 Building Load Forecasting

There are two approaches for building load forecasting in the literature, namely data-driven and engineering methods. Engineering methods (white-box models) are out of the scope of this thesis. The first approach is data-related because it implements historical available data for predication of future load (Amasyali and El-Gohary, 2018). The granularity of the data used for forecasting building load is both short-term and mid-term. Load forecasting for commercial buildings, especially buildings with large HVAC (Heating, Ventilation, and Air-conditioning) systems can be complicated and challenging. The complexity gets more for building with a mixture of dissimilar areas such as laboratories, offices, lecture theatres, and event rooms, which have different HVAC and lighting requirements. Besides, building electricity loads vary with internal factors such as occupancy. Moreover, building loads are liable to changes in external weather parameters such as temperature, solar radiation, and humidity. These all show the complexity of building load forecasting. In forecasting commercial building electricity, many strategies such as statistical regression models, thermal models, machine learning, and time series models have been applied. (Yildiz et al., 2017).

2.4.2.3 Integrated Load Forecasting

Residential load forecasting should be aggregated to be used in the grid plan. Three different approaches can be deployed for aggregated load forecasting:

The first one is to forecast load at meter level or building level and then sum them up to calculate the cumulative load forecast.

Another method is to sum the load to create aggregated demand and forecast integrated load. Clustering consumers based on their load profile can improve the accuracy of the forecasting (Da Silva et al., 2013; Wijaya et al., 2015).

Figure 2.5 demonstrates load forecasting approaches in scale and horizon.

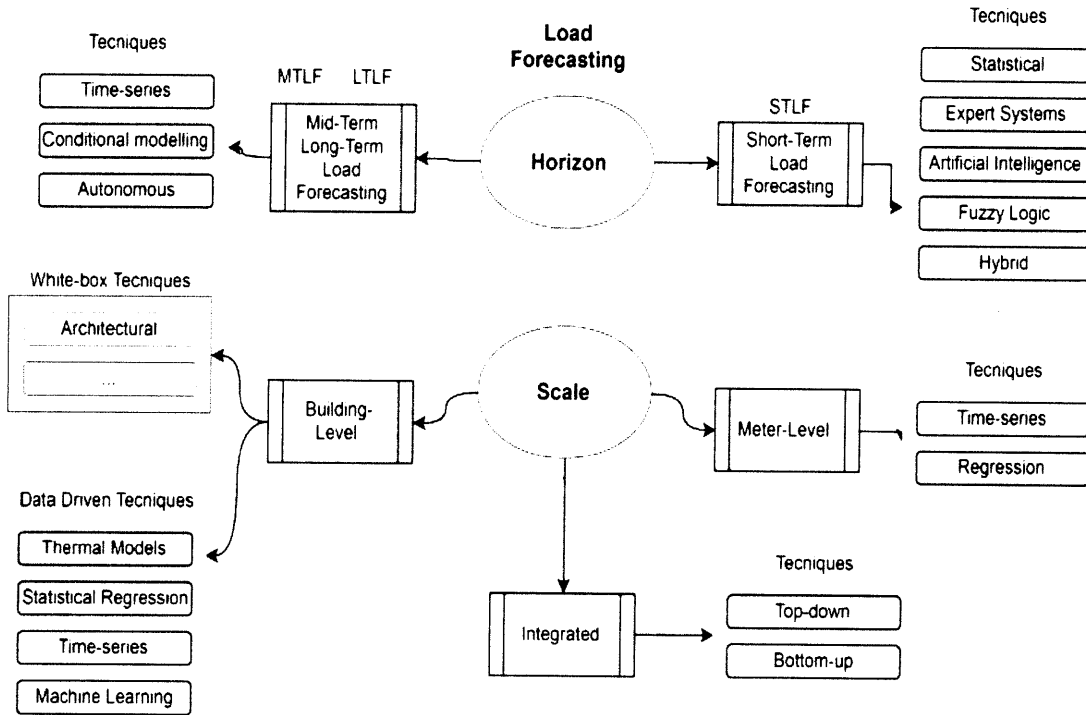


Figure 2.5: Load Forecasting Approach in scale and horizon

2.4.3 Load Forecasting Techniques

The early attempts for load forecasting employed detailed data sets such as anthropology and demo-graphical survey information. However, after the wide implementation of smart meters in recent years, studies consider how to utilize huge amounts of data collected by the smart meter to predicated load. There are different approaches for load forecasting listed in Figure 2.5. However, Load forecasting based on data collected by the smart meter is a data-driven approach.

As with any data-driven approach, data should go through different stages. Figure 2.6 represents these stages graphically. Initially, data will

be pre-processed and cleaned. As mentioned previously, smart meter data is read and stored in different resolutions from second to hourly intervals. If the default smart meter resolution, is higher than the desired forecast resolution, data can be aggregated by averaging(grouping).

Grouping is one step for data selection and it is a well-known method for improving the accuracy of load forecasting. Since forecasting of individual demand often leads to large errors, groups are created before forecasting to reduce forecasting error (Da Silva et al., 2013). When dealing with forecasting smart meter measurement two types of grouping are discussed, longitudinal and cross-sectional grouping. Longitudinal grouping means investigating historical data to recognize periods with similar load patterns (time with the same electricity consumption characteristics). Longitudinal grouping is done in combination with similar-day methods for forecasting. On another hand cross-sectional grouping refers to the aggregating of the customer with a similar load pattern, which means doing load profiling before load forecasting (Quilumba et al., 2014).

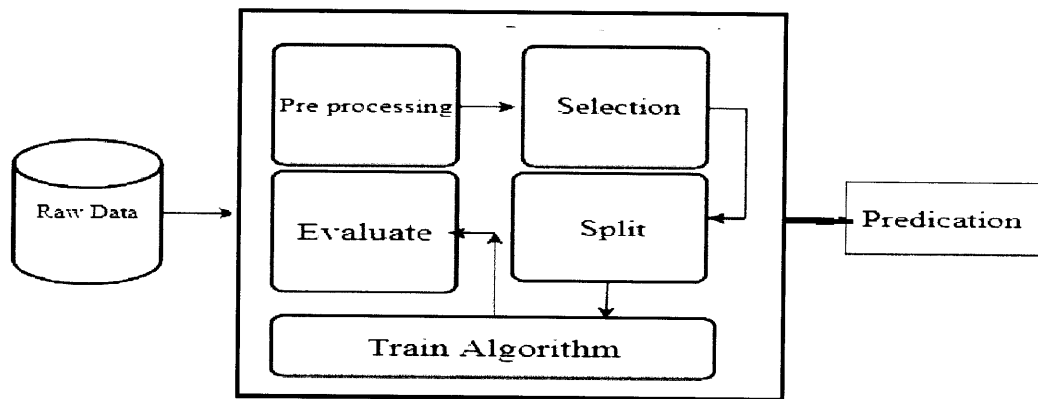


Figure 2.6: Load forecasting as a data science problem

In the following subsection, the different stages of the approach represented in Figure 2.6 are discussed.

2.4.4 Data Preprocessing for Load Forecasting

As with any other data-driven approach, data preprocessing is essential for load forecasting. Furthermore, proper data preprocessing will improve the quality of load forecasting. data preprocessing is a very popular stage in data mining and may include different methods such as: Data cleaning and imputation, data integration(combining multiple data from different sources), and data transformation.

Data cleaning and imputation are about the detecting and imputing of data. Data imputation has different approaches such as completing, replacing, or modifying the outliers. Outliers are any incorrect, incomplete, inaccurate, irrelevant or missing data. Different imputation methods are used in literature to deal with missing data (Kim et al., 2017). If the outlier cannot be smoothed, it will be removed from the data set.

In addition to these, data transformation enhances the representation of data and represents it in better methods. It may include normalization, smoothing, aggregation, disaggregation, and generalization of the data. Different algorithms are implemented for data transformation such as Wavelet transformation or Fourier transformation to represent electricity loads as sinusoidal waves of different frequency and amplitude (Li et al., 2015; Dang-Ha et al., 2017).

Feature selection is also applied to load before forecasting. It is generally done for two reasons; first for finding the most useful input and seconds to make the forecasting computation less expensive. The feature selection step is a common step among load forecasting domain. Integrity is the process of combining multiple data from different sources. AI techniques can be used for efficient feature selection (Wijaya et al., 2015; Hu et al., 2015). An example of this is the use of a genetic algorithm for feature selection (Ghamisi and Benediktsson, 2014).

Statistical methods also can be used, for example, the cross-correlation

was used in (Koprinska et al., 2015), to maintain the input data that exhibit a significant correlation with forecast load. A survey of different feature selection techniques can be found in (Chandrashekar and Sahin, 2014).

Another stage of feature selection is Data reduction. The result of this process is representing data in a smaller space by removing the non-discriminative features. By using data reduction, the same as feature selection methods, the forecast not only will be more accurate but also computationally more efficient by enhancing the performance of the forecasting algorithm. Mutual entropy methods, Principal component analysis (PCA), kernel PCA (KPCA), and cross-correlation are used for data reduction in literature.

2.4.5 Forecasting Algorithm

The solution for solving the load forecasting problem varies in the literature applied forecasting methods can be categorized as Statistical techniques, time series, and Artificial intelligence technique as represented in Table 2.7

Statistical techniques are deterministic models that express the relationship between the load and any input factors.

Some of the models used in statistical techniques are multiple regression methods and exponential smoothing. Although these readings can interpret in other ways, they are naturally time series and hence time series methods are possible for data-driven load forecasting. In this approach, raw data should be represented as a time series. The smart meter records consumption in time lags. Different Auto regression methods such as Autoregressive (AR) model, Autoregressive moving-average (ARMA) model, and an autoregressive integrated moving average (ARIMA) model used for this task.

Artificial intelligence techniques are also used for this task. Soft computing, as well as machine learning, are a sub-field of artificial

Table 2.7: Publications of various load forecasting techniques against scale

	Short-Term	Mid-Term	Long-Term
Neural Networks	Liu et al. (9 26) Kong et al. (2017)		Chen and Wang (2012) Chen (2012)
Time-Series	Chaouch (2014) Eskandarnia and AlHammad (2020)	Lee and Hong (2015)	de Oliveira and Cyrino Oliveira (2 01)
Statistical Methods	Ghofrani et al. (2011)	Eskandar et al. (2020)	Hong et al. (2014)
Machine Learning	Lusis et al. (7 11)	(Hu et al., 2015)	
Fuzzy Logic		Chen (2012) Lee and Hong (2015)	Chen and Wang (2012)
AI Methods		Amjady and Keynia (0 01)	

intelligence. Among computational intelligence, the most widely used methods for load forecasting are neural networks and machine learning. A review of artificial intelligence techniques implemented for short-term load forecasting was conducted by (Zor et al., 2017). Different machine learning algorithms were used in prediction such as Support Vector Machine (SVM) and artificial neural network. The selection of the prediction algorithm and machine learning techniques is highly related to the forecasting zone and scope. In this thesis we imply the different time series methods for integrated hourly load forecasting, In following section the background of this techniques are represented

2.4.6 Evaluation Metric For Forecasting

There are different evaluation indexes for assessing the quality of predication. The list of available indexes can be found in (Ghamisi and Benediktsson, 2014). In order to evaluate the forecasting , three index used namely $MAE, r^2 - Score, MAPE$ which stands for Mean Absolute Percentage Error.

$$MeanAbsoluteError(MAE) = \sum_{i=1}^n \left(\frac{|y_i - \hat{y}_i| * 100}{n} \right) \quad (2.1)$$

$$MAPE = \sum_{i=1}^n \left(\frac{|y_i - \hat{y}_i| * 100}{n} \right) \quad (2.2)$$

$$r^2_{score} = \frac{n \sum xy - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}} \quad (2.3)$$

where y = *real value* and $\hat{y}$ = *predicated value*

2.5 Time Series Methods for Load Forecasting

As shown in Table 2.7, time-series techniques include different methods such as ARIMA and Boxconex are used for load forecasting. According to (Deb et al., 2017) different versions of the ARIMA model were utilized to forecast the load demand for countries, buildings or cities. But there is no consistency between the parameters of ARIMA models in these studies. Furthermore, most of these studies use monthly readings for forecasting and the smaller granulate data such as hourly and daily data were not fed to ARIMA. (Joshi and Patel, 2018) shows that box-Jenkins shows better accuracy for forecasting granulated data.

2.5.1 ARIMA(Box-Jenkins) Background

One of the more powerful techniques for time series modeling is the Box-Jenkins procedure (ARIMA modeling). This method is developed by George Box and Gwilym Jenkins in 1970 for finding the autoregressive and moving average parameter and the procedure is described as follow:

- o Model-identification: check for seasonality and if needed use the differencing to create white noise (stationary data), use plot to check for autoregressive or moving average part of the model.

- o Parameter estimation.

o Model checking: to get the residuals that are constant in mean and variance and independent of each other.

According to (Deb et al., 2017), different versions of the ARIMA model were utilized to forecast the load demand for countries, buildings, or cities. But there is no consistency between the parameters of these methods in these studies. Future more, most of these studies use monthly readings for forecasting. ARIMA models are denoted with the notation (p, d, q) (Hyndman and Athanasopoulos, 2018). Three distinct integers (p, d, q) that are used account for seasonality, trend, and noise in data sets. Note that p is the auto-regressive part of the model AR, d is the integrated part of the model (*differencing I*), and q is the moving average part of the model (MA). The general formula of the ARIMA model is represented in Equation 2.4

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_q \varepsilon_{t-q} + \varepsilon \quad (2.4)$$

Since the ARIMA model works best for data with no trends, it cannot capture the trends and seasonality. In order to discover trends and seasonality, a data visualizer and data decomposition were implemented.

In the initial stage, data is examined for stationarity. If the data set is not stationary either the d parameter of the model is set with help of the Visualization or statistical tests such as the dickey fuller test, or the data transform to stationary. The other parameters of the model are set with the use of the Auto-correlation function (ACF) (2.5) and partial auto-correlation function (PACF). The parameter must be set to ensure that residuals have constant mean and variance and are independent to each other.

$$ACF(X, lag) = \frac{E[(x_t - \mu)(X_{t+lag} - \mu)]}{\sigma^2} \quad (2.5)$$

In addition to this, a data visualizer and data decomposition were implemented to discover trends and seasonality. ARIMA model can not

capture trends and seasonality. Seasonal ARIMA (SARIMA) is introduced to solve the limitation of the ARIMA model.

2.5.1.1 Seasonal Autoregressive Integrated Moving Average Model (SARIMA)

ARIMA cannot model time series when there is a trend and the data set have a seasonal pattern. To address this limitation, SARIMA was introduced with the formula (2.6)

$$(p, d, q)(P, D, Q, s) \quad (2.6)$$

where P, D, Q represent Seasonal autoregression, difference and moving average respectively and s is the seasonality length. the SARIMA model may have too many interconnected parameters and hyper-parameters (Shumway and Stoffer, 2017). As a result, selecting a good SARIMA model is usually a difficult task. The complexity of developing a suitable model will be increased by the data set's big data nature.

2.5.2 Prophet Algorithm Background

Prophet is a time-series data forecasting process based on an additive model that fits non-linear trends with yearly, weekly, and daily seasonality, as well as holiday impacts. It works best with time series with substantial seasonal influences and historical data from multiple seasons. Prophet usually handles outliers well (Taylor and Letham, 2018).

Prophet algorithm works based on decomposition. Decomposition is used to investigate the seasonality and trend in time series. Decomposition breaks the time series to four components: $level(L)$, $Trends(T)$, $Seasonality(S)$, $Noise(N)$.

Two types of decomposition are introduced: additive (equation 2.7) and Multiplicative model. either seasonal Extraction in ARIMA Time Series (SEAT) is example of additive decomposition (Phinikarides et al., 2015),

and Seasonal and Trend decomposition using Loss(STL), is example of a multiplicative method (Dagum and Bianconcini, 2016). The capability of sTL to handle all types of trends, control the degree of smoothness and allow the change in seasonality is the advantage of this method. Hence STL is preferable method when the data set is versatile. So, STL is a versatile and robust method suitable for estimating nonlinear relationships (Chatfield, 2018).

$$y(t) = L + T + S + N \quad (2.7)$$

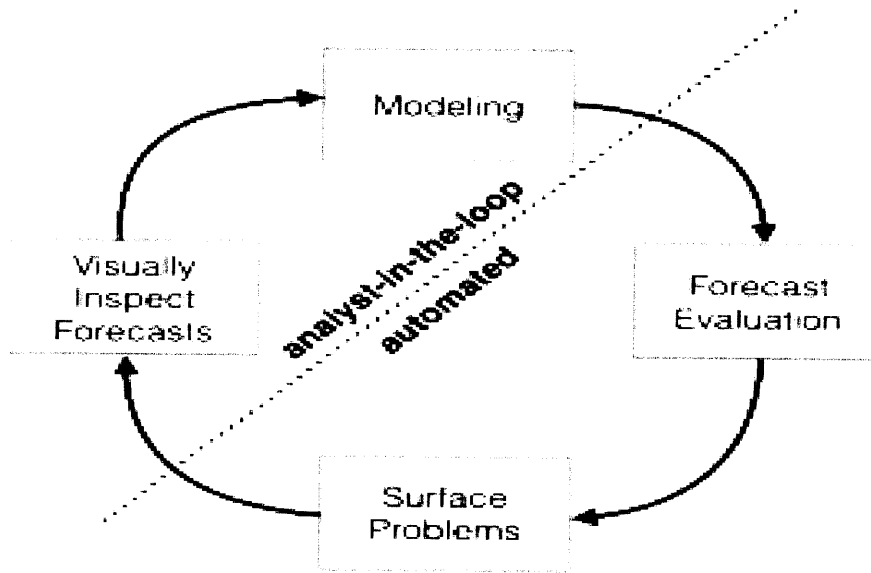


Figure 2.7: Semantic view of prophet, forecast in scale

For large-scale forecasting, Prophet employs an analyst-in-the-loop technique. The semantic perspective of these approaches is depicted in Figure 2.7. This method begins by modeling a time series using the parameters supplied by analysts, then creating and assessing projections. Whenever a performance issue accrued or the need for human intervention is arises, it is alerted to human analysts, who may then evaluate the forecast and perhaps alter the model based on this feedback.

2.6 Chapter Summery

This chapter provided an explanation of smart meter data analytics. A taxonomy was constructed in order to provide a comprehensive survey that covers every area of smart meter analytics. By defining the major domains within the field and the numerous functions that are carried out within each domain, this classification helped scholars better grasp the current research trends and the opportunities that are open to them. Since very few review articles explored novel approaches such as deep learning for smart meter analytics and since deep learning outperforms shallow machine learning methods, A denotative research analysis was performed to determine the deep net architecture used in each task, while all the domains of the taxonomy. In the next chapter, two enhanced methodologies will be provided for two tasks in knowledge discovery and load forecasting domain.

Chapter 3

Methodology

- 3.1 Datasets and Preprocessing
- 3.2 Deep Embedded Clustering Methodology
- 3.3 Methodology of Time Series Forecasting
- 3.4 Load Forecasting
- 3.5 Time Series Methods for Load Forecasting
- 3.6 Chapter Summery

Chapter 3

Methodology

This chapter discusses the methodology and experimental regime used in order to achieve the objectives defined in chapter 1. The experimental result will be discussed in chapter 4. This chapter has three subsections for discussing the methodology of deep embedded clustering, EDA, and Load forecasting. The database used for experimental is also described in this chapter.

3.1 Datasets and Preprocessing

3.1.1 Real-World Datasets

As demonstrated in Figure 2.1 smart meter analysis starts with data. Although in recent years the importance of open data for sustainable development in smart cities was discussed (Kumar, 2021), only two open access data sets are available for real-world smart meter data (<http://www.ucd.ie/issda/>, <https://data.london.gov.uk/dataset/>). The first dataset (Irish) will be used to compare the framework's performance with previous proposed work, and the results will be presented at the end of the next section. However, the latter dataset (London) was used to conduct the

main load profiling experiment presented in this paper. This data set is part of the UK Power Networks Low Carbon London Project, which includes energy consumption readings for 5,562 London households. Readings were collected between November 2011 and February 2014, and the original data file is around 11GB subdivided into 167 CSV files with around 167 million rows.

The London data set has 5 main columns as follows *Consumption* records in KWh recording the consumption of the house; *DateTime* is taken at half-hourly intervals; *LCDID* represents a unique identifier of the household; and *StdorDToU* that shows if the consumer is subjected to Dynamic Time of Use (DToU) tariff. the data type are demonstrated in Table 3.1. Most households have been allocated to an Acorn group classification (see <https://acorn.caci.co.uk/>) which is categorical data that represents demographic group segments of the UK population. The data set included 5767 time series, with each time series containing readings from a single household. Each reading is timestamped every half hour from 24:00 hours of 23/11/2011 to 23:30 of the 28th of February of 2014. The downloaded files will be preprocessed as explain in following section.

3.1.1.1 Preprocessing

Due to the nature of the smart meter readings, preprocessing involved big-data techniques for loading and manipulating the data set. The data set was read and aggregated from 168 files with four different techniques. The memory footprint and speed for each method are recorded in Table 3.2.

As demonstrated, Dask was used because it provides the highest efficiency for dealing with such a big data set. Dask is an open-source parallelization platform that supports the Pandas data frame and Numpy array data structures. Dask dataframe scale Pandas workflows, allowing programs to manage large amounts of data by manipulating data in parallel

Table 3.1: Data description

filed-name	Description	Datatype
LCDID	Unique House id	Text
std or DToU	Tarrif	Text
Date and Time	Date and time of reading	Text
Acorn	The level of social of household(A-K)	Text
Acorn-group	The 5 groups of socioeconomic	Text
consumption	the consumption as <i>Kilowatthour</i>	numeric

through a personal laptop or a cluster of computers.

A smart meter dataset is an $m \times n$ matrix, where m is the number of readings per time and n is the number of households. This data set included 5767 time series, with each time series containing readings from a single household. Each reading is timestamped every half hour from 24:00 hours of 23/11/2011 to 23:30 of the 28th of February of 2014. Thus the dimensions of the resulting data frame are (39972,5567).

Data imputation is the next step in data prepossessing. In this stage, if the consumption readings for the specific house are empty for x readings and $x < \mu$, the house will be excluded from the analysis; where the μ is set according to m .

The formula specified in Equation 3.1 is used to fill missing values. As Equation 3.1 illustrates, next year's (or previous year's) readings for the

Table 3.2: Different Approaches for Dealing with Big-Data

Methods	Map-Reduce		Filter and Aggregate	
	Memory (Mb)	Time (s)	Memory (Mb)	Time (minutes)
chunking	440.0	746.0	500.0	10 /filter
pipeline	190.0	876.5	-	-
Delayed function	600.0	395.0	-	-
Dask function	400.0	559.4	500.0	2.5 /filter
Dask with threads client	200.0	540.7	760.0	2.0 /filter

same time and date are used to fill in the null values. In case this is also missing, it is filled with the average consumption of the same month.

$$x_i = \begin{cases} x_{i+365 \times 48} & \text{if } x_{i+365 \times 48} \neq \text{null}, \\ x_{i-365 \times 48} & \text{if } x_{i-365 \times 48} \neq \text{null}, \\ \frac{\sum_{j=1}^{30 \times 48} x_j}{30 \times 48} & \text{otherwise} \end{cases} \quad (3.1)$$

On other data preprocessing stage is *data normalization* as described by Equation 3.2, after data normalization each consumption reading will be in the range $0 \leq x \leq 1$.

Min-Max normalization is a popular strategy for scaling data that are used to train a deep neural network.

$$x_{ji} = \frac{x_{ji} - \min(x_i)}{\max(x_i) - \min(x_i)} \quad \forall i = 1 \dots m \text{ and } \forall j = 1 \dots n \quad (3.2)$$

The data set was pre-processed furthermore according to the task in hand which will be discussed in each task separately. for example, the dataset

is usually *re-sampled*. Furthermore, and the frequency of the time series is decreased (i.e. down-sampling). *Data reshaping* will also be performed for some task, where the smart meter data are rotated or reshaped before feeding it to the algorithm. The result of this step is a matrix of size $n \times m$. Each row in reshaped matrix contains consumption readings for a specific house.

To improve the quality of the modeling and recognize the elements that influence consumer behavior, the data was plotted using various methodologies, and correlation was investigated using several types of plots. The details of the exploratory data analysis to investigate factors and model the data are presented in sections 4.1 and 4.2.

3.2 Deep Embedded Clustering Methodology

The survey conducted to investigate the first objective of the thesis showed the importance of load profiling in several domains and tasks within the smart meter analysis field. Hence the second objective of this thesis aims at deploying new techniques (such as deep learning) to improve the efficiency of load profiling. This lead to the proposal of algorithm 1, and the following section will outline the newly proposed methodology for implementing deep learning to obtain better load profiles.

3.2.1 Enhancing Load Profiling with Deep Clustering

The enhanced load profiling framework consists of three steps; namely: data preprocessing, clustering, and finally obtaining the load curves. A summary of these steps is illustrated in Figure 3.1 and outlined in Algorithm 1. An expanded description of each step is discussed in the following subsections, followed by the proposed algorithm. The clustering step will be explained in following subsection.

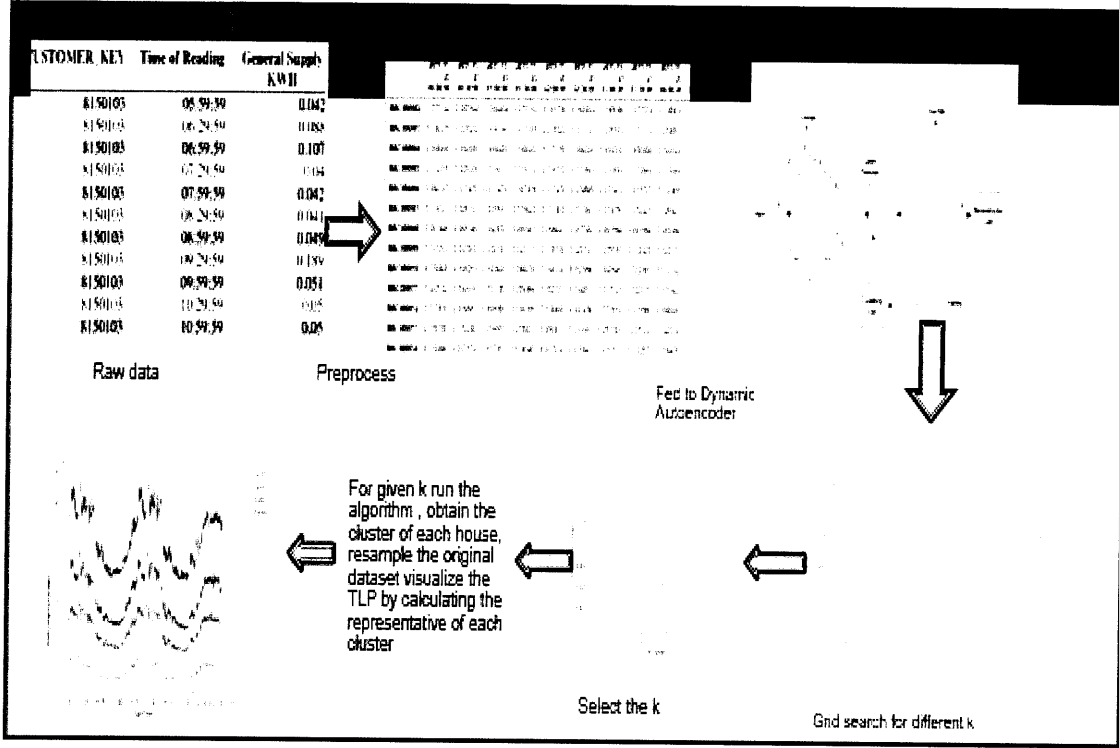


Figure 3.1: Summary of the deep load profiling framework steps

3.2.2 Clustering Step

Three deep autoencoder based clustering algorithms will be adopted for the proposed load profiling framework; namely, Deep Embedded Clustering (DEC) (Xie et al., 2016), Improved Embedded Clustering (IDEC) (Guo et al., 2017), and DynAE (Mrabah et al., 2020b). These are all Encoder based architectures as explained in Table 2.6. All these algorithms require that the number of clusters k is specified as input. In order to obtain the best number of clusters, the algorithm will perform grid search for k and calculate the Silhouette Coefficient (Rousseeuw, 1987). Based on the results obtained, the best k will be selected. The deep clustering algorithms and theory behind them is explained in following sections.

3.2.3 The Deep Clustering Algorithms

This thesis will implement an auto encoder based network for clustering task as mentioned in Table 2.6, namely three deep autoencoder-based clustering algorithms DEC, IDEC, DynAE will be examined for this task. DEC clusters data in feature space Z by continuously learning a set of k cluster centers $\{\mu_j \in Z\}_{j=1}^k$ and parameter θ of the deep network that map data points to Z . This method and its family members IDEC and DynAE work in two stages:

1. parameter initialization with a deep autoencoder (pre-training stage)
2. parameter optimization (i.e clustering), in which the algorithm iterates between computing an auxiliary target distribution and minimizing the loss function.

The loss function of all these algorithms includes minimizing Kullback–Leibler (KL) divergence (Xie et al., 2016). The initializing step of these methods is the stacked DNN autoencoder. In the following sub-section, we introduce the autoencoder loss function for the proposed algorithm.

Autoencoder Loss

The autoencoder is made up of two components: an encoder and a decoder as shown in Figure 3.2. An encoder function $Z = f_\phi(x)$ which maps original data X_i into a latent representation Z_i , and a decoder that produces a reconstruction $X'_i = g_\theta(Z_i)$. The reconstructed representation X'_i is required to be as similar to X_i as possible. In a latent space Z_i , the encoder maps its input X_i to a representation Z_i . During training, the decoder attempts to reconstruct X_i from Z_i , ensuring that no useful information was lost during the encoding phase. The reconstruction objective for a set of data samples $X = \{X_i \in \mathbb{R}^D\}_{i=1}^n$ by Mean of Square Errors (MSE) is shown in Equation 3.3 (Xie et al., 2016).

$$\min_{\theta, \phi} L_r = \min_{\theta, \phi} \frac{1}{n} \sum_{i=1}^n \|X_i - g_{\theta}(f_{\phi}(x_i))\|^2 \quad (3.3)$$

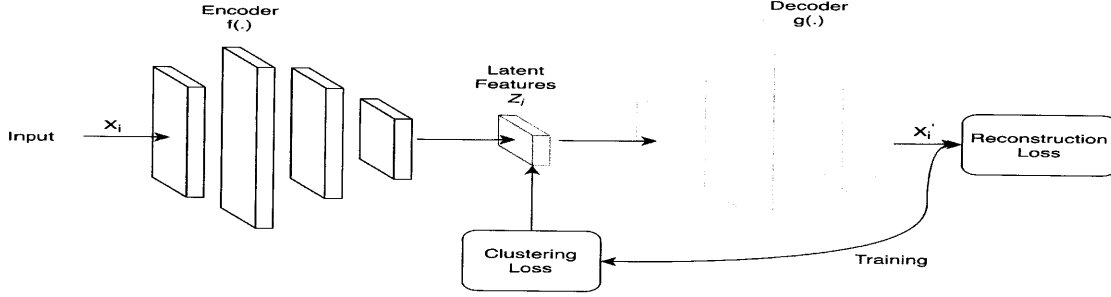


Figure 3.2: Overview of the autoencoder architecture

Clustering with KL Divergence

The algorithms have two inputs: initial cluster centroids $\mu_{j=1}^k$, and non-linear mapping f_{θ} that maps data points to embedded points. Furthermore, they iterate between two steps: the first step is computing a soft assignment between the embedded points and the cluster centroids, and the second step in which learning from current high confidence assignments using an auxiliary target distribution refine the cluster centroids and update f_{θ} . This procedure is repeated until a convergence condition is satisfied.

Soft Assignment

For embedded points $z_i = f_{\theta}(x_i) \in Z$ corresponding to $x_i \in X$, a quantity q_{ij} is introduced as the similarity between embedded point z_i and centroid μ_j , which is measured by the t -distribution where the degree of freedom is set to $\alpha = 1$. Consequently, q_{ij} can be interpreted as the probability of assigning data point i to cluster j . In other words it is the similarity between embedded point z_i and cluster center μ_i , and is defined as follows:

$$q_{ij} = \frac{(1 + \|z - \mu_j\|^2 / a)^{-\frac{a+1}{2}}}{\sum_j (1 + \|z - \mu_j\|^2 / a)^{-\frac{a+1}{2}}} \quad (3.4)$$

In this equation the α is degree of freedom and as mentioned set to 1.

The Loss Function

The objective is to minimize the loss function defined by equation (3.5) (Min et al., 2018):

$$L = \lambda L_r + (1 - \lambda) L_c \quad (3.5)$$

where $\lambda \in [0, 1]$ is a coefficient used to balance these two loss function components, namely the reconstruction loss L_r defined in Equation 3.3, and the clustering loss L_c defined in Equation 3.8.

The DEC, IDec and DynAE networks are trained by matching the soft assignment to the target distribution so they iteratively refine the clusters by learning from their high confidence assignment, with the assistant of an auxiliary target distribution p_{ij} defined as in Equation 3.6.

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_j q_{ij})} \quad (3.6)$$

The clustering loss of Equation 3.5 can be defined as in (Aljalbout et al., 2018):

$$L_c = \sum_{i=1}^N \sum_{k=1}^K s_{ik} \|z_i - \mu_k\|^2 \quad (3.7)$$

where z_i is an embedded data point, μ_k is a cluster center and s_{ik} is a Boolean variable for assigning z_i to μ_k . Minimizing this loss with respect to the network parameters assures that the distance between each data point and its assigned cluster center is small.

The clustering loss can also be measured using the Kullback-Leibler (KL) divergence Xie et al. (2016); Li et al. (2018) shown in Equation 3.5, and the same loss function was calculated in the proposed framework.

$$L_c = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (3.8)$$

Note that q_{ij} is as defined in Equation 3.4 and p_{ij} is the target distribution defined in Equation 3.6.

Deep Learning Optimization

Mini-batch stochastic gradient descent along with back propagation are used to optimize the loss function. There are, in particular, three types of parameters to optimize or update: autoencoder parameter weights, target distribution P , and cluster centers. The optimization of each parameter was adopted as described in (Guo et al., 2017).

Dealing with Feature Randomness and Feature Drift

The efficiency of global optimization suffers significantly as a result of feature drift. A neural network with a multi-objective cost function is vulnerable to feature drift, which takes place when different objective functions compete inappropriately or unfairly against each other (Mrabah et al., 2020a). Even after a large number of iterations, if the two objective functions are not balanced correctly, the network will not meet its goal and minimize the loss (Mrabah et al., 2020a). If the competition is not carefully considered, it can be highly hazardous as it may restrict convergence, cause divergence, or result in an insufficient solution. As a result, it is preferable to prevent any implicit rivalry within the same network. As illustrated in Equation 3.5, the objective function of deep clustering consists of two parts which is naturally subject to feature drift.

On the other hand, the biggest challenge of clustering with the absence of real labels is that semantic labels should be used in order to define an objective function that can learn high-level comparisons. Self-supervision and pseudo supervision are two principles that underpin existing unsupervised representation learning techniques. Pseudo-supervision entails training a model with pseudo-labels that are predicted

based on a presumptive similarity metric rather than the real ones. Therefore, some of the pseudo-labels mismatch the real ones (Rasmus et al., 2015).

Self-supervision aims to teach general-purpose features by completing a pretext problem. A strong pretext problem necessitates the acquisition of high-level features. The chosen pretext affects the effectiveness of self-supervision. The popular pretext tasks include reconstruction pretext and the adversarial objective (Goodfellow et al., 2014). Compared to self-supervision, the process of using a pseudo-supervised objective function can identify target categories in spite of the use of a proxy. The generated pseudo-labeling can then be automatically generated without any human intervention, similar to self-supervised learning.

Although deep neural networks can perfectly match semantic labels when the number of parameters exceeds the number of data points (Zhang et al., 2021), feature randomness occurs as a neural network is trained with pseudo-supervised labels. After extracting pseudo-labels, feature randomness is defined as the deviation of the gradient of the real supervised objective function compared with the pseudo-supervised gradient.

There are certain disadvantages to training a neural network with objective functions that work in a highly opposing and counterbalancing way. In reality, by updating the learning weights concerning the pretext objective L_r , the features learned through the optimization of the main function L_c can easily suffer from the drifting effect. This may lead to the failure of the global optimization process. In other words, the model becomes very sensitive to the trade-off parameter λ between L_c and L_r .

Hence, by increasing the balancing coefficient, the model will suffer from more feature randomness. By contrast, decreasing λ will cause feature drifts. In general, the pseudo-supervised objective function dominates the optimization process when the balancing coefficient is increased, and this

causes feature randomness to increase.

Essentially, DynAE not only decreases the feature randomness but also achieves a stronger trade-off between feature randomness and feature drift, by steadily and seamlessly converting a self-supervised target into a pseudo-supervised one. It also describes a dynamic loss function to solve the clustering-reconstruction trade-off. The detailed explanation of the loss function for embedded clustering on this methods can be found in (Mrabah et al., 2020b)

Evaluation of the Clustering Performance

Because the data from smart meters is unlabeled, assessing the quality of the clustering findings can be challenging (Vercamer et al., 2015). Cluster Validity Indices (CVIs) are commonly used for load profiling task assessment and validation, When the number of clusters and other clustering parameters are unknown. A huge number of CVIs have been proposed and assessed for the clustering task in general (Kim and Ramakrishna, 2005). Among different CVIs, the Silhouette Validity Index is widely used (Wang et al., 2015; Sun et al., 2020). The silhouette coefficient is calculated by

$$SI = \frac{\beta - \alpha}{\max(\alpha, \beta)}. \quad (3.9)$$

where β is the distance between a sample and the nearest cluster that the sample is not a part of, and α is the mean of intra-cluster distance.

However, in the load profiling task, as in other real-world clustering scenarios with no known ground truth, determining the optimal number of clusters will become more difficult. Breakpoints on the CVI curve (Knee Technique) are widely recognized as viable options for addressing this problem (Zhou et al., 2017). A break point of the silhouette coefficient was used to determine the best number of clusters in different papers (Yang et al., 2013; Sun et al., 2020).

Algorithm 1 Proposed Enhanced Load Profiling Framework

```
1: Pre-process( $A$ ) and return  $B$ 
    $A \leftarrow B$   $\triangleright A$  is smart meter readings and  $B$  is matrix of size  $n \times m$ 
2: procedure CLUSTERING( $B$ )  $\triangleright$  the output is a vector  $K$  with size  $n \times 1$ 
3:   for  $j \leftarrow 3$  to 10 do
4:      $k \leftarrow j$ 
5:     for Deep Clustering Algorithm  $\leftarrow$  [DEC, IDEC, DynAE] do
6:       set hyperparameters of network (learning rate, no of iterations,  $\phi, \dots$ )
7:       Deep Clustering Algorithm( $B, k$ )
8:       register Silhouette Coefficient (SI)
9:     end for
10:  end for
11:  Visualize the SI and select number of cluster  $k$ 
12:  Evaluate the performance of different algorithm for  $k$ 
13:  Run the best clustering algorithm (according to silhouette) with  $k$  and find cluster
    of each house
14: end procedure
15: procedure OBTAIN LOAD PROFILE( $B, K$ )
16:    $O = (B|K)$   $\triangleright$  merge the two matrices
17:    $k \leftarrow$  number of clusters
18:   Initialize Matrix  $T$  with size  $m \times k$ 
19:   for  $l \leftarrow 1$  to  $k$  do
20:     for  $i \leftarrow 1$  to  $m$  and  $j \leftarrow 1$  to  $n$  do
21:        $T_{ij} \leftarrow \text{Means}(O_{ji})$  where  $O[m+1] = l$ 
22:     end for
23:   end for
24:   for  $j \leftarrow 1$  to  $k$  do
25:     Load profile of  $cluster_j = T[j]$ 
26:     Re-sample time series  $T[j]$  to day-wise and visualize
27:     evaluate the load profiling task and select the best performance
28:   end for
29: end procedure
```

3.3 Methodology of Time Series Forecasting

According to conducted survey, time series methods can be conduct for STLF in meter level. Time series methods also can be used for MTLF. The primary task of time series modeling is the identification of the high-level properties of data in order to explore and formulate a closed form formula that describe the time series. Two main factors for obtaining an appropriate time series modeling are degree of fit and the complexity. The model should be as much as possible simple and in same time fit the historical data well. The trade off between these two factors depends on the application. Although modeling is possible for non-stationary time series but If the data set is stationary the modeling will be much easier, so in literature the first task in this discipline is to find the statistical properties of time series in order to check the stationarity of time series and future do the mathematical transformation to transform data to stationary form if needed . Some statistical test like Augmented Ducky Fuller can be implement in order to check for stationarity in data set.

For obtaining objective three of this thesis two time series methods explained in chapter 2 has been used, namely Prophet and Sarimax.

3.3.1 Seasonal ARIMA (SARIMA)

Figure 3.3 represents the proposed framework. Choosing of ARIMA or its extension SARIMA depends on the nature of the data set and the existence of seasonality and trend in the data. Furthermore,proposed framework implemented enhance Sarima for model selection, which is described in algorithm 2.

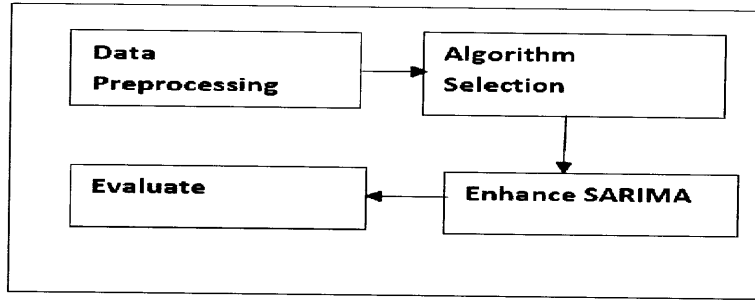


Figure 3.3: Proposed framework for SARIMA

3.3.1.1 Proposed Framework

As Figure 3.3 shows raw data first go through the pre processing to change to univariate time series.

This methodology utilizes SARIMA, an extension of the Box-Jenkins model which considers seasonality in data. The following subsections discuss the different parts of the proposed methodology.

Data Preprocessing

As explained in section 3.2 data set first goes through the initial preprocessing, Thus the initial matrix $167,000,000 * 5$ changes to the data frame with size $(39972, 5562)$. Furthermore, 5 houses with less than 20% readings throughout the analysis were removed. (The threshold was set to 80%). These created the x_i matrix with dimensions $(39972, 5557)$. Following that, the number of null values in the data set was 21.9% of the total data. The remaining null values were filled according to Equation 3.1.

The horizon and scope of this task are STL and integrated load, which is forecasting of one day advance for the city of London. Data re-sampling and data aggregation were applied to the data set. The output of this level

is the matrix of 168×1 which is a time-series data representing the daily consumption of London in KWH for the period of 2011-11-23 to 2014-02-28. This univariate time series is then feed to algorithm 2.

Algorithm Selection

As explained in chapter 2 the first stage in time series modeling is to obtain statistical properties of time series. These properties for daily and hourly univariate data set is represented in Table 3.4.

Table 3.4: Static Properties of Data Set

Statistics of hourly data	Statistics of daily data
Count 19864	count 829
mean 1790.065259	mean 42895.909132
std 1034.331235	std 20152.110830
min 0.000000	min 90.385000
25% 1066.618750	25% 34439.920000
50% 1748.852000	50% 45972.851997
75% 2438.727250	75% 58860.151998
max 5176.137998	max 82650.492003

Since choosing of ARIMA or its extension SARIMA depends on the existence of seasonality and trend in the data, first the existence of the seasonality and trend in time series will be investigated by seasonal

decomposition (Ma and Li, 2016). These types of decomposition breaks the time series to four components: *level(L)*, *Trends(T)*, *Seasonality(S)*, *Noise(N)*. Two types of decomposition are introduced in literature : Additive (equation 3.10) and Multiplicative model. Seasonal Extraction in ARIMA Time Series (SEAT) decomposition, which is additive, was used for data set (Phinikarides et al., 2015).

$$y(t) = L + T + S + N \quad (3.10)$$

The seasonality of the data set is checked with Seasonal Extraction in

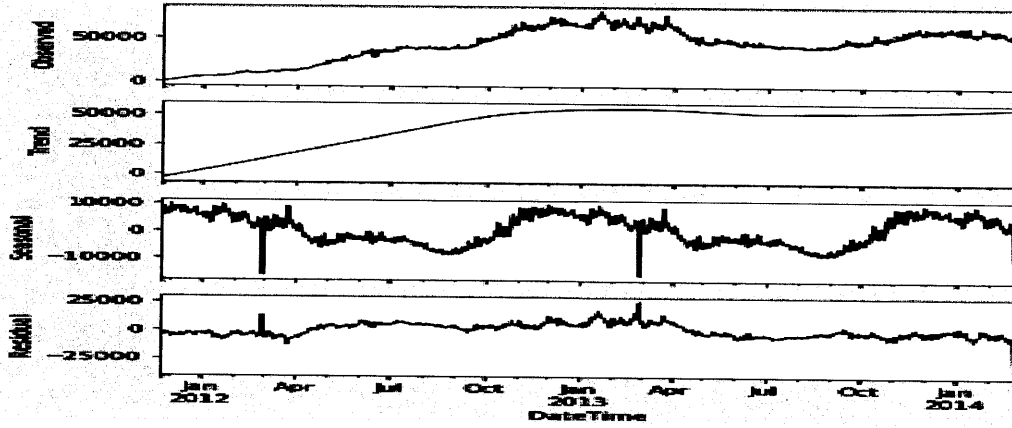


Figure 3.4: Decomposition of data set

ARIMA Time Series (SEAT) as done by Phinikarides et al. (2015)). Statistic test Augmented Dickey-Fuller (ADF), was also used to check for stationary, and both tests were implemented using the stats library of python. The Dickey-fuller test result is shown in Table 3.5. The P-value of the test is $p - value = 0.345486$ so the null hypothesis is denied and it is approved that the data set is not stationary.

Figure 3.4 shows the result of decomposition in the data set. The figure clearly shows the trend and a seasonal pattern in the data set, hence the best method to model the data is SARIMA which is a seasonal version of Box Jenkins. Since the SARIMA has a huge search space due to the big

number of parameters and hyperparameters, the enhanced SARIMA model was proposed and explained in the next section.

Table 3.5: ADF Test in daily Data

Results of Dickey-Fuller Test:
Test Statistic -1.871707
p-value 0.345486
#Lags Used 7.000000
Number of Observations Used 821.000000

Enhanced SARIMA

The model selected for this data set is SARIMA, as it potentially has a wide search space. For this reason, the parameters and hyperparameter setting is a challenging and expensive task. This limitation is addressed in the literature by two methods. First by defining a function to set the parameters automatically (grid search) as described in Shumway and Stoffer (2017). Secondly, by allowing the system to utilize the optimizing algorithm to initial the parameter (in python this is available by set enforce-stationary and enforce-invertibility to true). According to the result of our experiments, it is computationally cheaper and more efficient to use intuition from ACF and PACF graphs and the history of the data set, to hard code some parameters and limit the domain of grid search.

Algorithm 2 Enhance SARIMA

```
D, d, s  $\leftarrow$  Hard code base of decomposition
range of p, q  $\leftarrow$  Intuition from ACF plot
range of P, Q  $\leftarrow$  Intuition of AFC of seasonal
AIC1  $\leftarrow$  0
while do p, d, P, D in range
    model  $\leftarrow$  SRAIMAX(p, d, q)(P, D, Q, s)
    calculate AIC of Model
    if AIC1 > AIC then
        Model  $\leftarrow$  model
        AIC1  $\leftarrow$  AICofmodel
    end if
end while
run the diagnostic test
if residual  $\neq$  whitenoise then Select the previous model and go to 16
end if
Tune hyper parameter for the selected Model
Save the parameter of Model
```

This is reflected in the enhanced algorithm proposed in Algorithm 2. To decide about degree of differencing (parameter d of modeling), ARIMA model fit with zero MA and AR and check for standard deviation. This way the parameter d which is degree of differences can be set as shown in 3.5. The other methods is to use differences on time series till we get a stationary time series.

The autoregressive and moving average in the seasonal and non-seasonal part of the model are the parameters of the model as described in Equation 2.6, which is set with grid search along with ACF and PCAF plot intuitions. The hyperparameters of the model are :

Cover type: the outer product of gradient estimators (opg), observed information matrix estimator, numerical approximation of the Hessian matrix, and robust.

Solver: newton, bfgs, lbfgs, powell, cg, ncg, and basinhopping.

Since BIC penalizes the model more heavily AIC was used to select the model the enhance the algorithm. To apply the algorithm, python programming language and its data package such as NumPy, panda, dask, and stat were implemented on a personal computer. The psutil is used to print the CPU configuration and the CPU has 8 cores with 12 GB of virtual memory when 8 GB is free. so, 58% of virtual memory is available. The results of this algorithm will be presented in next chapter.

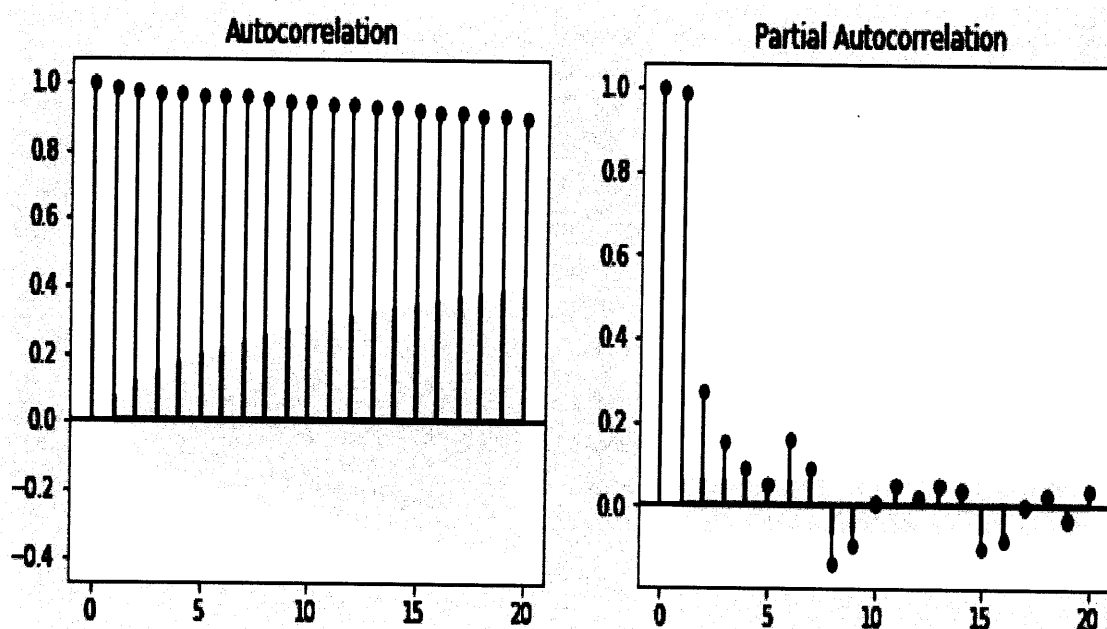


Figure 3.5: Auto correlation and PCAF graph

Evaluation

For evaluation of the model, the diagnostic tools of the Python SARIMAX model were run, and the result was interpreted.

3.3.2 Prophet

Prophet is a tool created by Facebook, and it anticipates time series data using an additive model that fits non-linear patterns to daily, weekly, or annual seasonality. Special events, such as holiday effects, are also taken into account. The Facebook data science team created it and released it as open-source software. The time series should have substantial seasonal effects and record several seasons of data in order to get a satisfactory result from this method.

3.3.2.1 Methodology

The methodology utilized for the load forecasting task with prophet is shown in 3.6. Despite the fact that the task is identical to SARIMA, the tasks techniques in each section are distinct. Although feature engineering is the first stage in both approaches, in Prophet, the output of this phase is a matrix that the prophet algorithm can accept. The subsections that follow go over the various aspects of the suggested methodology.

3.3.2.2 Data Prepossessing

The result of initial prepossessing stage explained in section 3.1.1 is matrix with size (39972,5562). This matrix goes through re-sampling to translate data to the daily data set. for that, data index changes and convert to a time slab so data represent as time series then the sum of the daily load was calculated. The output of this level is the matrix of 168×1 which is a time-series data representing the daily consumption of London in KWH for the period of 2011-11-23 to 2014-02-28. Since Prophet accepts two columns of a dataframe in which the first column is a time-series data type with the name ds, and the second one (the y-column) has a number format, the data set needs to be processed more to be converted to a matrix that can

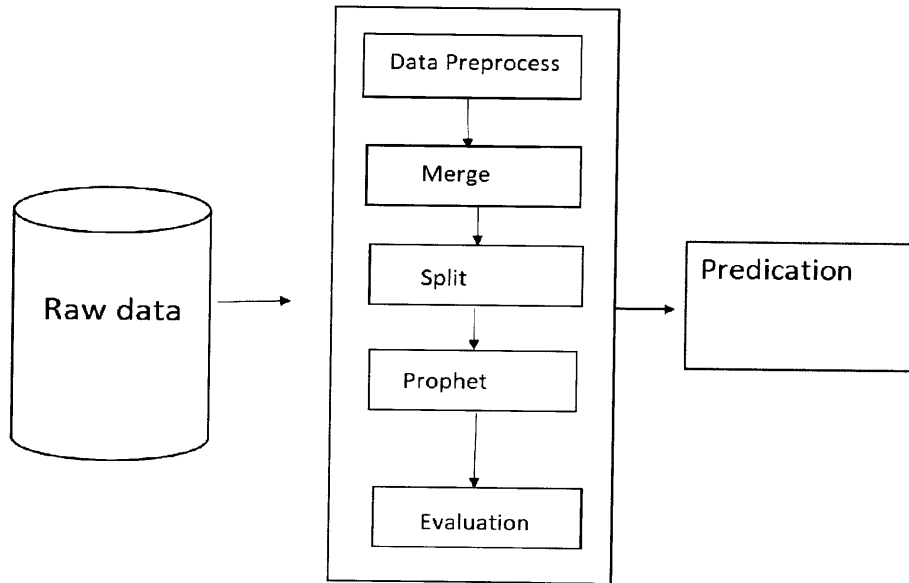


Figure 3.6: Proposed methodology for Prophet

be accepted by the algorithm.

3.3.2.3 Merge and Split

The important event also can be fit to the Prophet algorithm. In this thesis, public holidays and weekends were extracted from the UK calendar and a data frame was created from this is demonstrated in Figure 3.7, and it was merged with step 1's data frame to investigate their effect on consumption patterns.

3.3.3 Forecasting Evaluation

We commonly divide our data into a train and a test set in applied machine learning: the training set is used to construct the model, and the test set is used to evaluate it. We could even employ k-fold cross validation, which repeats the procedure by dividing the data into k groups and giving each one a chance to be a held out model. The data set is divided into train and test to

	ds	holiday
0	2012-12-26	Boxing Day
1	2012-12-25	Christmas Day
2	2012-08-27	Summer bank holiday
3	2012-05-06	Queen's Diamond Jubilee (extra bank holiday)
4	2012-04-06	Spring bank holiday (substitute day)
5	2012-07-05	Early May bank holiday
6	2012-09-04	Easter Monday
7	2012-06-04	Good Friday
8	2012-02-01	New Year's Day (substitute day)
9	2013-12-26	Boxing Day
10	2013-12-25	Christmas Day
11	2013-08-26	Summer bank holiday
12	2013-05-27	Spring bank holiday
13	2013-06-05	Early May bank holiday
14	2013-01-04	Easter Monday
15	2013-03-29	Good Friday

Figure 3.7: Holidays data set

quantify the model's skill on unknown data. The model is chosen based on how well it performs on data that was not used during training. When using K fold, a sort of cross-validation, overfitting is avoided. With time series data, these strategies aren't applicable.

This is due to their assumption that there is no correlation between the observations and that each observation is independent, and selects the train and test data sets by shuffling.

This isn't the case with time series data, because the time dimension of the observations prevents us from randomly grouping them. Rather, we must divide the data and adhere to the chronological order in which it was collected. Time series splitting was employed to solve this problem. This cross-validation version is a K-Fold variant. It returns the first k folds as train set and the (k+1)th fold as test set in the kth split. Unlike traditional cross-validation approaches, consecutive training sets are supersets of the previous ones. The data set was divided with $k = 7$ as demonstrated in Figure 3.8, and data chunks were used for train and test when test data

always have fix number. This division was used for evaluating forecasting algorithms, prophet, and SARIMA. The performance was evaluated with an index explained in section 2.4.6. The result is demonstrated in next chapter in Table 4.4.

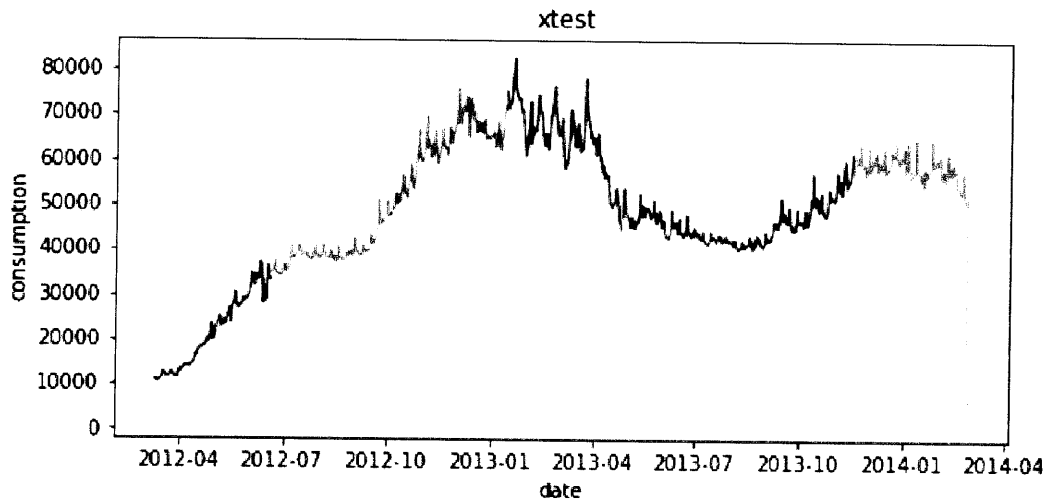


Figure 3.8: Time series Splitting with $k=7$

3.4 Chapter Summery

This chapter explained in detail the database used to test the proposed methodology. Dask has the best performance for reading and dealing with such a big data set. This chapter also discussed initial pre-processing and pre-processing before each task. This chapter also provides a methodology for exploratory data analysis, to improve the quality of the modeling and recognize the elements that influence consumer behavior. Furthermore, the correlation was investigated using several types of plots. In addition, in this chapter, we describe an enhanced load profiling framework which consists of three steps, namely, data preprocessing, clustering, and finally obtaining the load curves, illustrated in Figure 3.1 and explained in Algorithm 1. This chapter also described two enhanced methodologies for STFL at meter

level.

Chapter 4

Results and Discussion

4.1 Results and Discussion for EDA

4.2 Results and Discussion for Embedded Load Profiling

4.3 Experimental Result for STLF

4.4 Chapter Summery

Chapter 4

Results and Discussion

This chapter described the experimental regime and results of the proposed methodology in the previous chapter. EDA will be employed to look into key database aspects as well as the relationship between weather and consumption. The embedded load profiling also will be tested and DEC, IDEC, and DynAe networks will be compared with traditional clustering algorithms namely K-mean, Spectral, Hierarchical, and GMM. The number of clusters will be investigated by the *knee method* as described in Algorithm 1. The framework will be evaluated with the Irish data set and the result will be compared with new proposed hybrid methods. Furthermore, Algorithm 2 will be tested for load forecasting with enhanced SARIMA. In addition, the framework proposed in Figure 3.6 will be evaluated.

4.1 Results and Discussion for EDA

As demonstrated in Figure 4.1, the initial data exploration was conducted to demonstrate the distinct socio-demographic classifications as one data categories of this data collection. The figure also shows the count of consumers in each category. Another data category included in the database

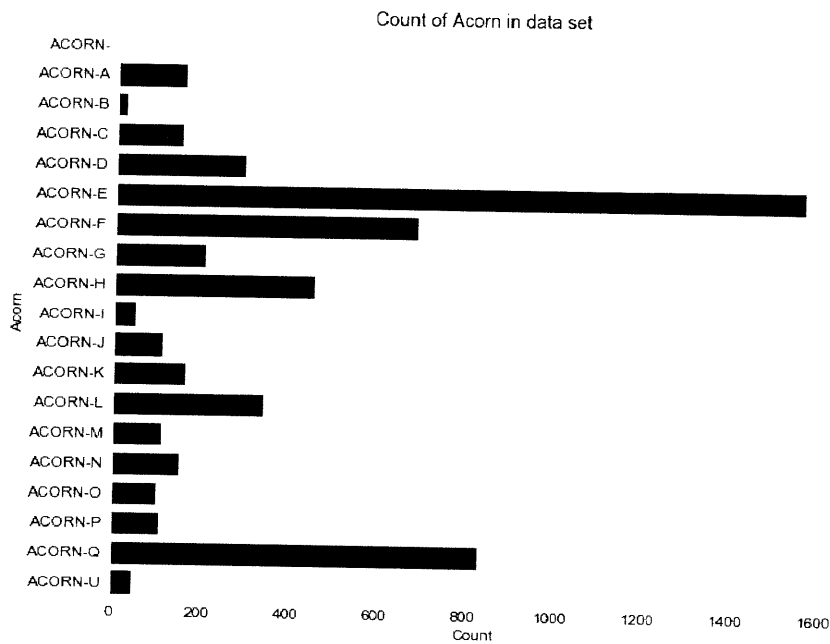


Figure 4.1: Acorn Count

is for tariffs as STD with flat rate tariffs and DTOU with Dynamic Time of Use (DTOU) energy prices for the 2013 calendar year.

The Smart Meter IHD (In-Home Display) or a text message to the consumer's mobile phone were used to provide tariff costs one a day in advance. Figure 4.2 demonstrated this attribute of data set as box plot, and it also shows different acorn-group with fix or not fix tariff.

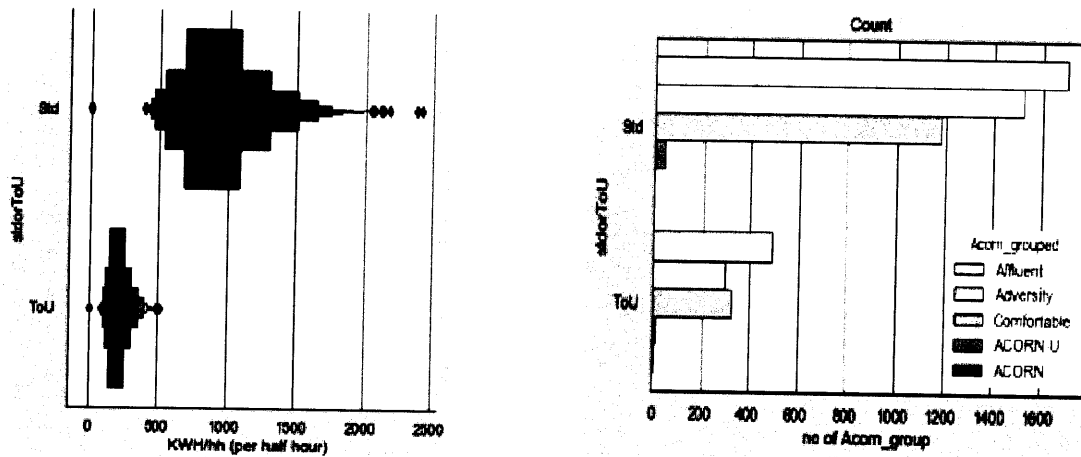


Figure 4.2: The tariff demonstration

The next exploratory data task is to model the consumption of each group to explore the effect of socioeconomic class on consumption patterns. Before this visualization, data was re sampled monthly to demonstrate information more coherent, as it is shown in Figure 4.3.

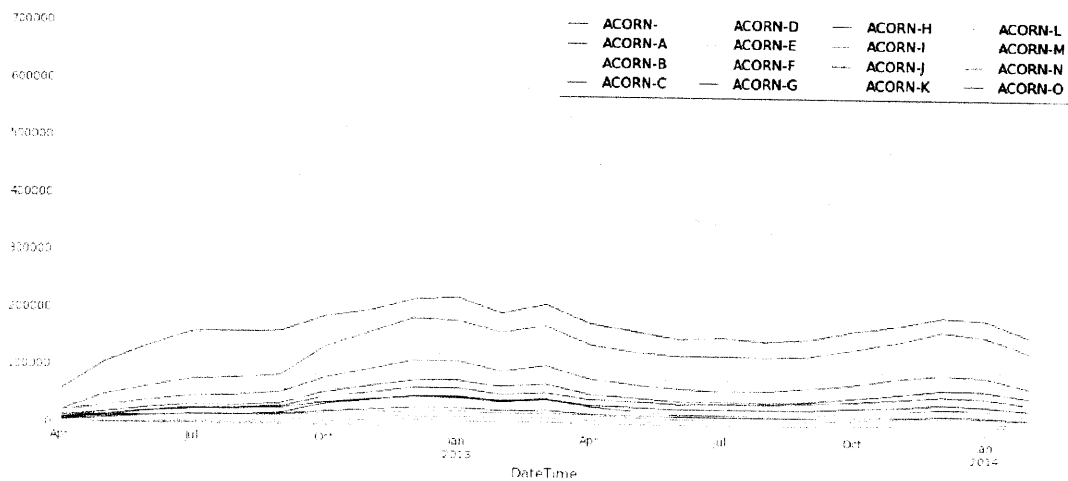


Figure 4.3: The consumption per month

The next investigation is to model daily and monthly consumption as shown in Figure 4.4. Figure 4.5 represents the average consumption per month in study period.

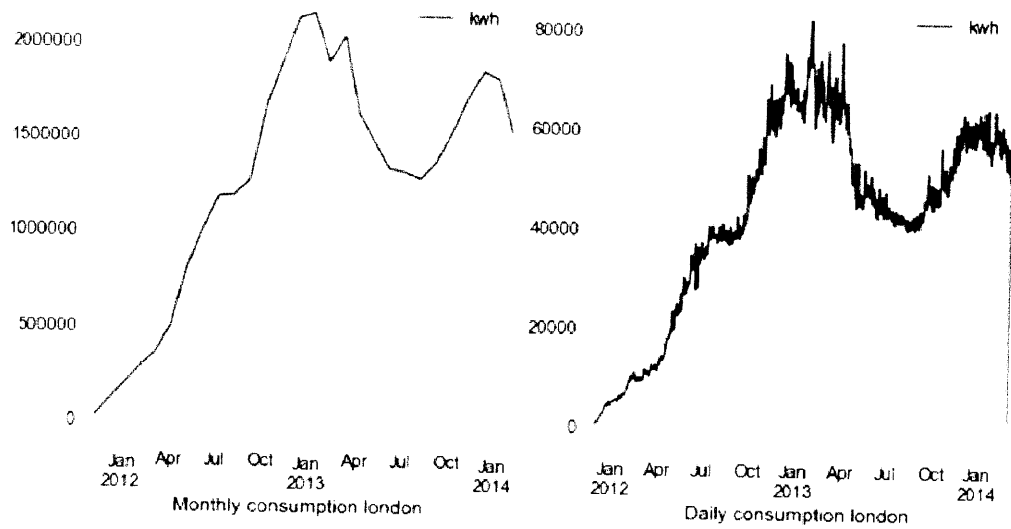


Figure 4.4: The consumption per month and Day

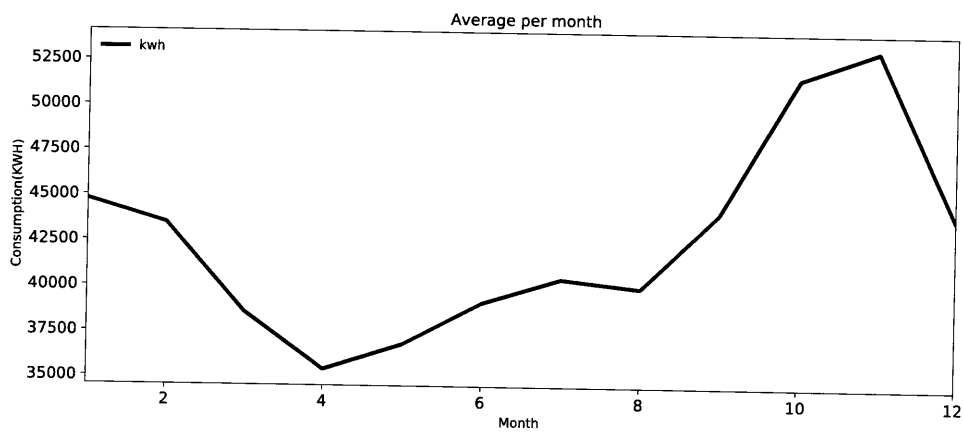


Figure 4.5: The average consumption per month

The other investigation is the yearly separation of consumption as presented in Figure 4.6.

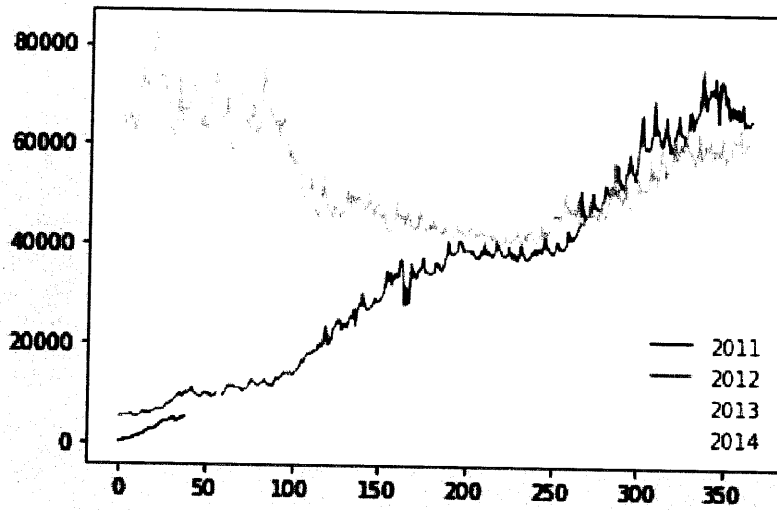


Figure 4.6: Consumption per year (2011-2014)

A better presentation of this relation with more details is plotted by the use of a heat map in figure 4.7. This plot also represents the temperature of each day in the study periods. In the next section, the effect of temperature along with other factors on consumption will be investigated further.

4.1.1 Investigation of Factors Affecting Consumption

In order to investigate other factors, the data set was merged with weather data set and then re sampled as described in preprocessing stage. Figure 4.8 shows the daily temperature in study time.

In order to investigate correlation between daily temperature and daily consumption the scatter graph is plotted (4.9) and correlation coefficient is calculated= -0.17. To investigated more the relation between temperature and consumption Ordinary Least Squares regression (OLS) Test was run and the result is shown in Figure 4.10 As the test shows for each degree increase in temperature, consumption is decreased by almost 640 Kwh, and intersect is 40K(kwh). Since the p-value is 0 null hypothesis is rejected and there is a regression relation between daily consumption and daily temperature

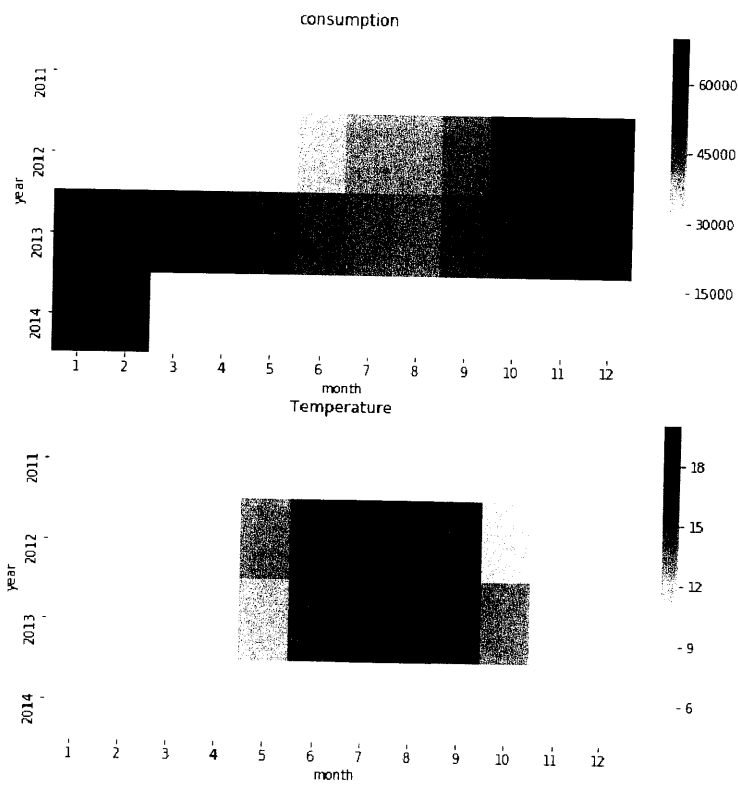


Figure 4.7: Consumption per month of each year (2011-2014)

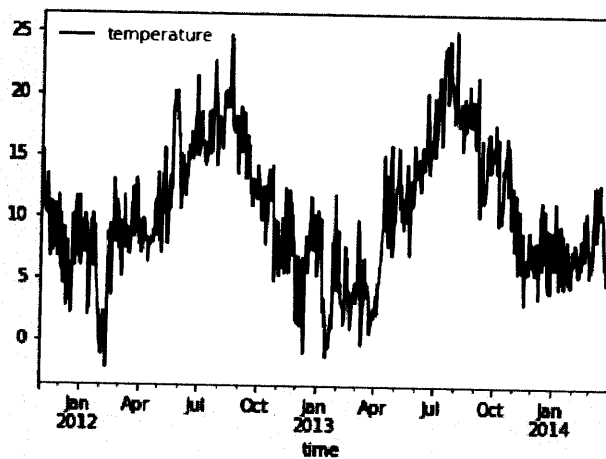


Figure 4.8: London weather temperature

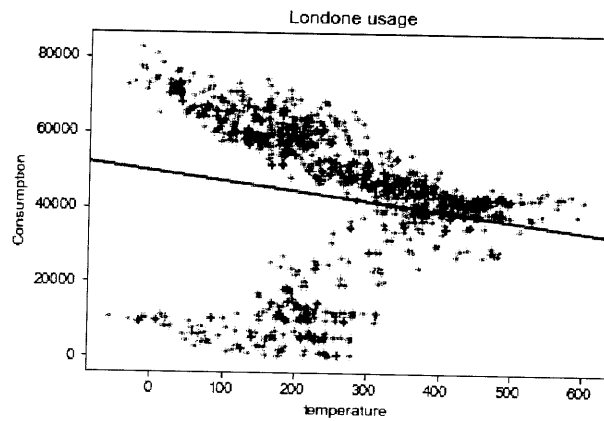


Figure 4.9: Temperature consumption correlations

although the regression factor is smaller than 0.2.

Since the weather data set has other factors, the Seaborn advance function cluster map was used to demonstrated the correlation between each feature of the weather data set and consumption pattern was demonstrated in Figure 4.11.

```

=====
                        OLS Regression Results
=====
Dep. Variable:          kwh      R-squared:                0.030
Model:                  OLS      Adj. R-squared:             0.029
Method:                 Least Squares      F-statistic:           25.45
Date:                  Thu, 07 Mar 2019    Prob (F-statistic):     5.60e-07
Time:                  21:19:22      Log-Likelihood:        -9366.4
No. Observations:      828          AIC:                  1.874e+04
Df Residuals:          826          BIC:                  1.875e+04
Df Model:              1
Covariance Type:       nonrobust
=====

```

	coef	std err	t	P> t	[0.025	0.975]
const	4.967e+04	1500.042	33.112	0.000	4.67e+04	5.26e+04
temperature	-639.2021	126.716	-5.044	0.000	-887.925	-390.479

```

=====
Omnibus:                98.676      Durbin-Watson:           0.019
Prob(Omnibus):          0.000      Jarque-Bera (JB):        132.724
Skew:                   -0.973      Prob(JB):                1.51e-29
Kurtosis:               2.751      Cond. No.:               25.9
=====

```

Figure 4.10: OLS Result

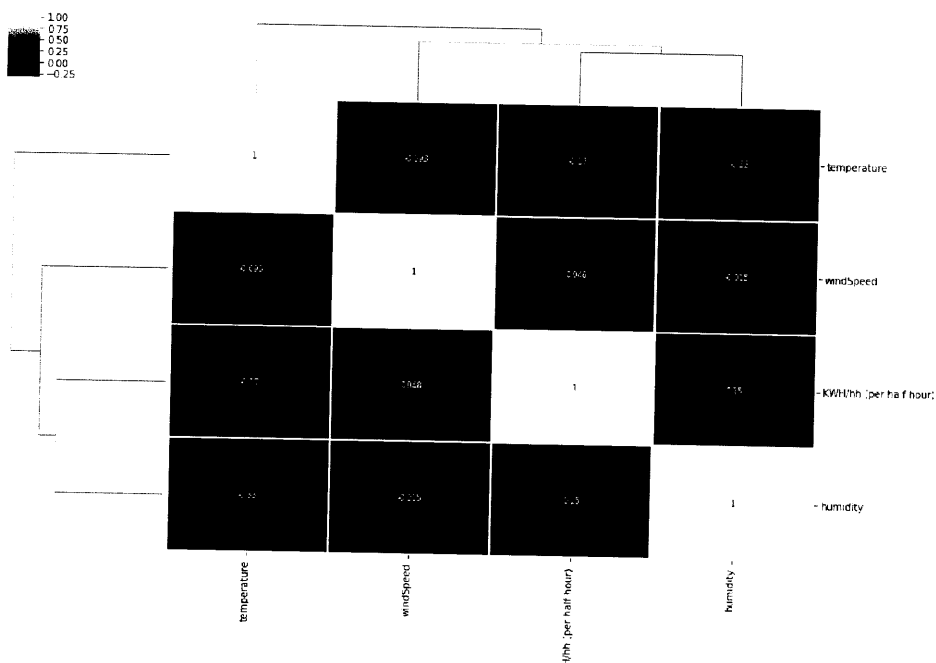


Figure 4.11: Weather factors correlation

By use of Seaborn package cluster map, the correlation coefficient of each part of the calendar view is demonstrated in Figure 4.12.

The coreletion between calender events and consumption

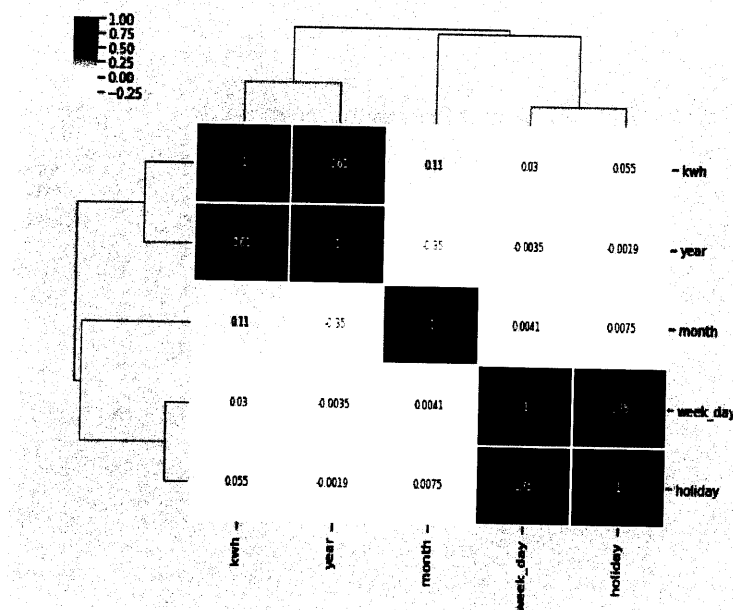


Figure 4.12: Calendar Event Regression

Figure 4.13 demonstrates the relation between weather and consumption in each month of the different year as well as total usage and average temperature for each month.

[illegible]

Figure 4.13: Calendar Analysis

4.2 Results and Discussion for Embedded Load Profiling

In this section embedded load profiling framework proposed in Algorithm 1 will be tested, and the results will be shown for the selected datasets. We start the chapter by describing the experimental protocol.

4.2.1 Experimental Protocol

All deep learning algorithm implementations were carried out on the Google Colaboratory Pro platform with high RAM and GPU-enabled, while the preprocessing was done on a personal computer with an 8-core CPU and 12 GB virtual memory.

Preprocessing

As explained in section 3.1 data set first go through the initial reprocessing. Thus the dimension of the resulting data frame is (39972,5562). In Algorithm 1 step 1 this data frame is represented by the matrix A . Then the data frame was rotated and reshaped such that the house number is the primary key of the database. This is done because the houses with identical patterns should be grouped in the same category as explained in sub-section 3.1.

Furthermore, 5 houses with less than 20% readings over the course of the analysis were removed.(The threshold was set to 80%). These created the x_i matrix with dimensions (5557,39792). Following that, the number of null values in the data set was 21.9% of the total data. The remaining null values were filled according to Equation 3.1.

Then by applying Equation 3.2 all readings will be normalized. Note that because using deep learning the proposed framework avoids many

complex pre-processing tasks (such as feature selection and dimensionality reduction) as described above.

Clustering

In order to find the best clustering algorithm for performing load profiling, the dataset was fed to DEC, IDEC, and DynAE deep clustering networks. To identify the best number of clusters the Silhouette Index (SI) was registered for k from 2 to 10. The elbow method (Kodinariya and Makwana, 2013) was then used to find the best number of clusters k as described in step 9 of Algorithm 1.

All network architectures were adapted from the original papers (Xie et al., 2016; Guo et al., 2017; Mrabah et al., 2020b). The experimental results showed that a custom dimension network with 11 fully connected layers of 37972, 2000, 1000, 500, 365, 100, 365, 500, 1000, 2000, 37972 neurons perform best for this database. All the layers except the bottleneck and final layers used the ReLU activation function.

In the pretraining stage, the autoencoder was trained adversarially end-to-end with the fully connected network. For this phase, Adam optimization was used to update all the learn-able parameters with a learning rate of 0.0005. It is worth mentioning that even though in the original papers a learning rate was attempted at 0.001 for pre-training, our experiment showed that a learning rate of 0.005 converged better. This is demonstrated in Figure 4.14 which shows a faster and smoother convergence.

After pretraining, the autoencoder weights were fine-tuned by optimizing a dynamic loss function. K-means was used for initializing the embedded clustering centers as well as updating them. As described in Equation 3.5, the loss function has two parts: the L_r reconstruction cost, as defined in Equation 3.3, and L_c the clustering loss defined in Equation 3.8.

The loss function is optimized using Stochastic Gradient Descent

(SGD), with a learning rate of 0.001 and a momentum of 0.9. Both the pre-training and clustering steps are run in 256-piece batches. The optimization process runs for different number of iterations.

For learning rate 0.005, the autoencoder weights had been saved. For a fair comparison, the saved weights of the pre-training stage were loaded to the clustering layer of the three deep embedded clustering networks and the number of iterations for clustering training was set to 600.

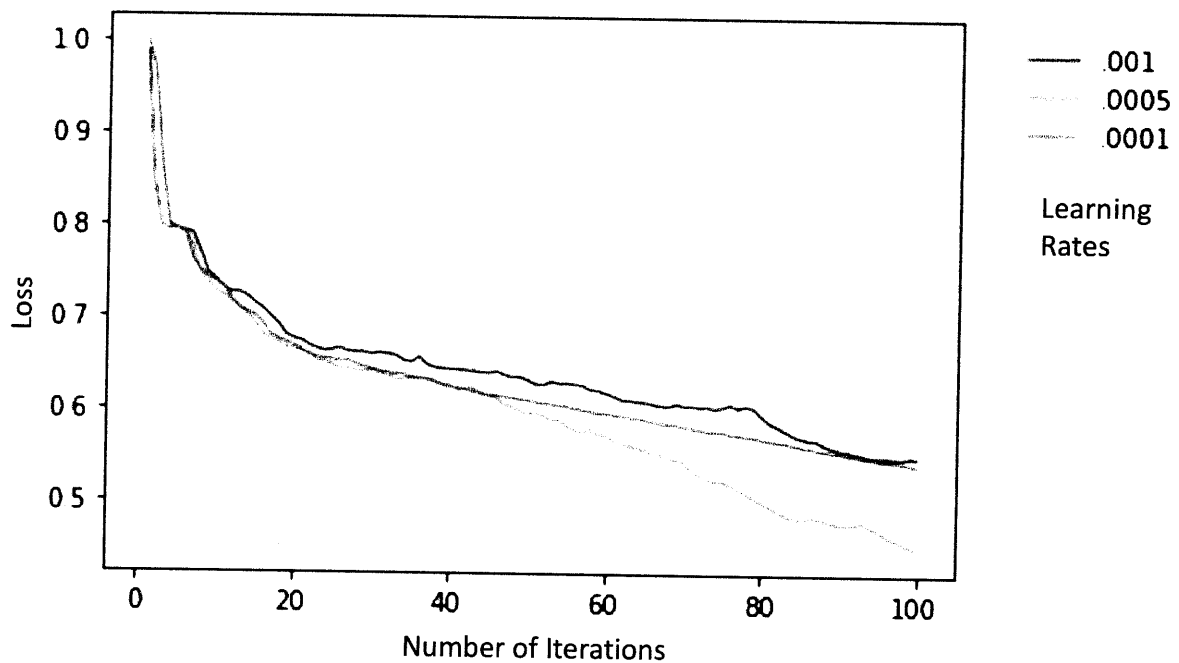


Figure 4.14: Loss for different learning rates

Evaluation and Obtaining the Load Profiles

Based on clustering results, several assessment criteria may be used to evaluate the performance of load profiling algorithms to aid in selecting an acceptable number of clusters. As explained in Algorithm 1 the proposed framework utilizes the Silhouette Index for this purpose. Furthermore, the purpose of the load profiling task is to obtain distinct groups of consumers.

The load curve quality can also be further visually assessed by

examining the load curve for each group as explained in Algorithm 1 steps 19-22.

As the centroids of each cluster are calculated in order to obtain the load profile for each cluster, using the Python Datetime package, four empty time series of size 39792 were created, with timestamps every half hour from 00:00 AM on November 23, 2011, to 23:30 to February 28, 2014. The average consumption of all houses in each cluster at each timestamp is then filled into these time series as describe in Algorithm 1 steps 12-18. These time series were re-sampled to the day, and each cluster's normal load profile was visually presented (see Figure 4.17).

4.2.2 Results of Embedded Deep Load Profiling

Table 4.1 shows the Silhouette Coefficients for various numbers of clusters for three deep learning networks DEC, IDEC, and DynAE. For evaluating the proposed enhanced load profiling method, the deep learning clustering algorithms were compared with different classical clustering algorithms, namely, k-Means, hierarchical clustering, Gaussian mixture models (GMM), and spectral clustering. As Table 4.1 shows when the number of clusters increases, the traditional clustering algorithms are not able to perform clustering..

The classical clustering methods were run with different parameters and the highest obtained SI was recorded for each method as demonstrated in Table 4.1. As explained in Algorithm 1 step 9, the elbow method was used to obtain the best number of clusters.

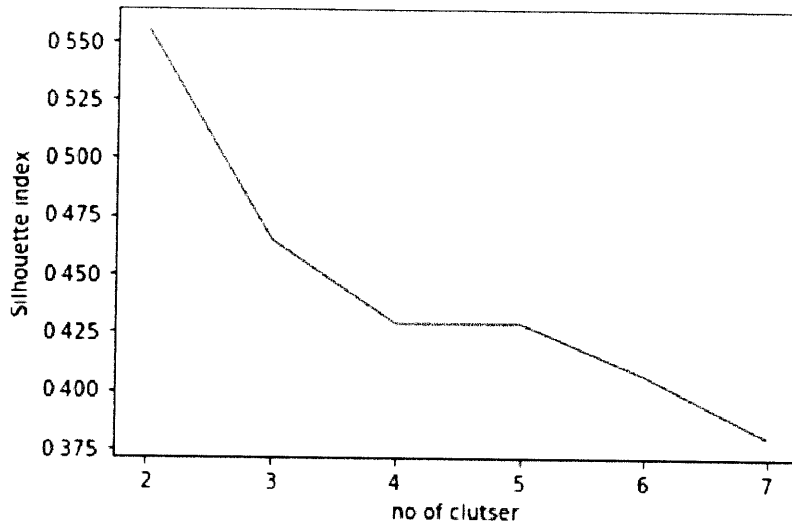


Figure 4.15: Elbow methods on Silhouette Index for Hierarchical clustering on sample data.

Figure 4.15 shows the elbow method graph for hierarchical clustering, which clearly shows a break at $k = 4$. Experimental results show that for all six clustering algorithms used for validation and implementation of load profiling, the most appropriate number of clusters was $k = 4$.

Table 4.1 shows that deep clustering increased the SI index to 0.59 using Deep Embedded Clustering (DEC) and 0.63 for DynAE, which was the best performance. This demonstrates the effectiveness of the proposed framework for load profiling.

According to the experimental findings, the number of iterations for the clustering process should be set when the network achieves convergence as shown in Figure 4.16, which for this data set was about 600 iterations.

Although the SI index can improve with further iterations, this may affect the quality of obtained load profiles. As described in Zhu et al. (2019); Panapakidis and Christoforidis (2017), the load profiling task apart from clustering includes the selection of pattern representation techniques and the extraction of representative consumers. The task aims to separate consumer groups and provide a meaningful load curve for each group. Hence, as

Table 4.1: Silhouette Index (SI) results for classical vs deep clustering methods

no. of clusters	Classical Clustering				Deep Clustering		
	k-Means	Hierarchical	GMM	Spectral	DEC	IDEC	DynAE
3	0.0815	0.5540	0.1610	0.0025	0.3303	0.2010	0.6243
4	0.084	0.4644	-0.0029	-0.1280	0.5929	0.2022	0.6343
5	0.0357	0.4288	-0.1130	-0.0128	0.5891	0.1796	0.5821
6	0.0321	0.4068	-0.0144	-0.1761	0.4384	0.1568	0.5321
7	0.0146	0.3803	-0.0185	NA	0.4125	0.1421	0.4366
8	0.0112	NA	-0.0899	NA	0.3471	0.0863	0.4415
9	0.0059	NA	NA	NA	0.3175	0.0958	0.4443
10	0.0009	NA	NA	NA	0.3192	0.0933	0.4313

discussed in algorithm 1, the resulting SI value along with visualization of HEC are used to choose the best algorithm for the load profiling task.

The typical load profile of each group was extracted and plotted as explained in 4.2.1. The load profiles obtained by DynAE and the classical hierarchical clustering are shown in Figure 4.17 and Figure 4.18. While hierarchical clustering with linkage parameter set to ‘average’ and affinity parameter set to ‘Euclidean’ achieved $SI = 0.4$, the quality of the load profiles of this approach were not satisfactory. This can be visually deduced while comparing the clearly separable load curves in Figure 4.17 with the load profile curves of Figure 4.18. Note that DynAE strikes a balance between feature randomness and feature drift as described in section 3.2.3. This functionality, as demonstrated by the experimental results that were based on a real-world data set, qualifies DynAE as the most suitable method for load profiling tasks.

To further strengthen our claim that the performance of deep clustering is superior to classical methods and to provide a bench-marking result, the framework was tested on the Irish dataset used by (Sun et al., 2020). As mentioned in Table 4.2 the researchers have used a concurrent k-means

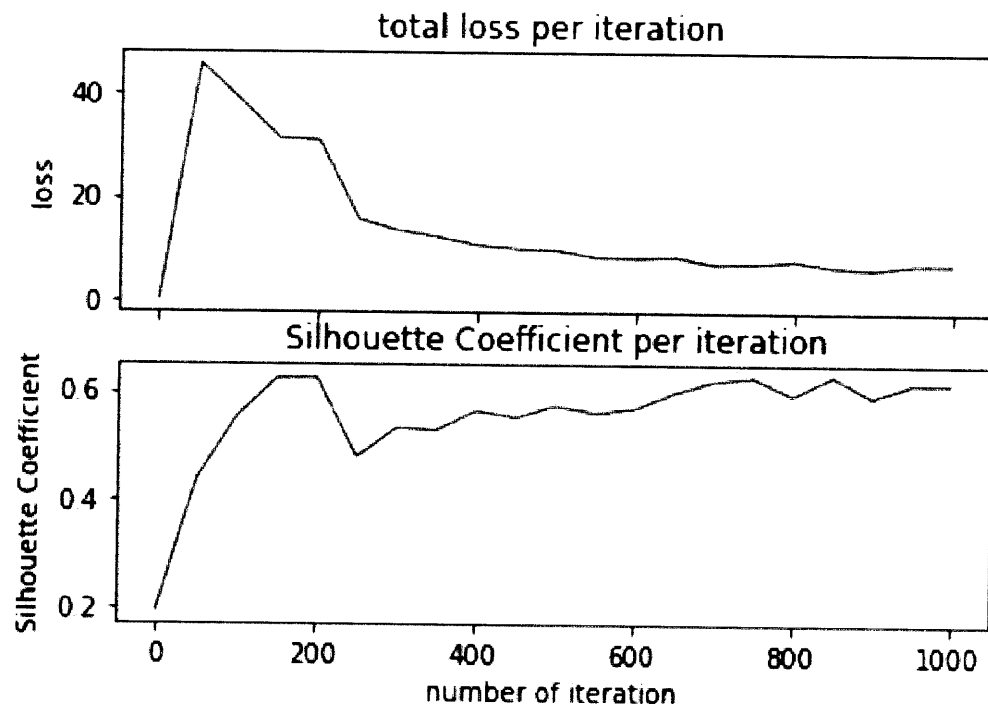


Figure 4.16: Loss function and SI value per iteration.

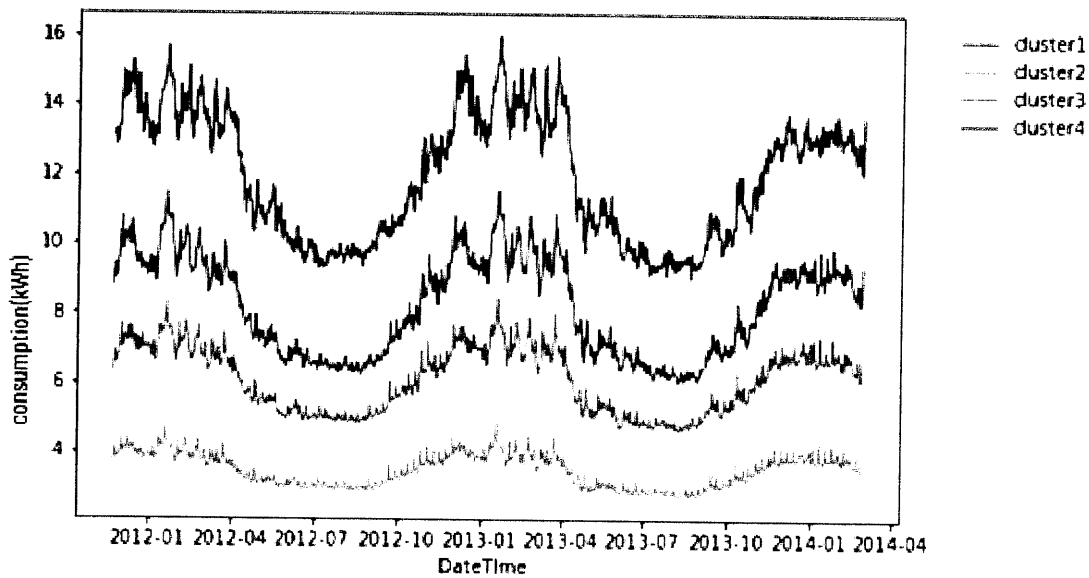


Figure 4.17: Load profiles of each cluster obtained by DynAE

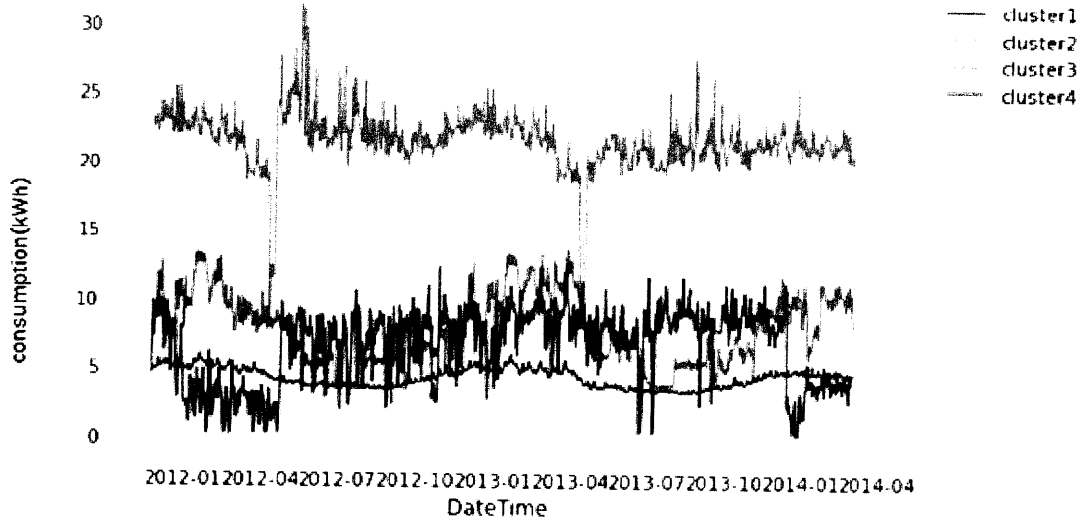


Figure 4.18: Load profiles of each cluster obtained by hierarchical clustering

algorithm for dimensionality reduction followed by spectral clustering to obtain the clusters. The comparison of the Silhouette values obtained by the researchers and our proposed method in Table 4.2 shows that the deep clustering framework achieved superior results. This proves our hypothesis that the dimensionality reduction and the deep features extracted by the autoencoder are more effective than the similarity matrix extracted by k-means and spectral clustering reported by (Sun et al., 2020). Figure 4.19 presents the results in a chart, which shows again that based on the elbow method $k = 4$ presents a suitable number of clusters. It also shows the improved performance by the deep clustering framework.

4.3 Experimental Result for STLF

This section presents the results of the short-term load forecasting experiments conducted using SARIMAX and Profet.

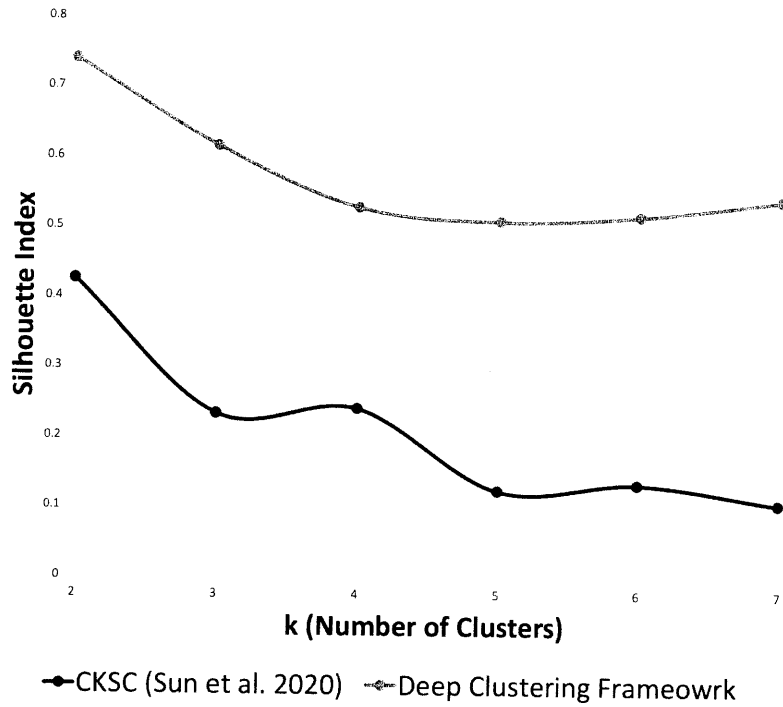


Figure 4.19: Comparison of Silhouette Index for the Proposed Framework and Sun et al. 2020 on the Irish dataset

Table 4.2: Silhouette Index (SI) results for CKSC and the proposed framework for the Irish dataset

no. of clusters	CKSC (Sun et al., 2020)	Deep Clustering Framework (DynAE)
2	0.425	0.7395
3	0.233	0.6158
4	0.240	0.5285
5	0.123	0.5090
6	0.133	0.5167
7	0.105	0.5404

4.3.1 Experimental Result for SARIMAX

SARIMAX (Seasonal Auto-Regressive Integrated Moving Average with eXogenous factors) of stats library was used to run algorithm described on

Algorithm 2. The experiment was run on a personal laptop with four cores processor and 8MB RAM. The model attains is $is(0,1,3)(1,1,2,7)$. The coefficient of model represented in Figure 4.20 The diagnostic tools were

	coef	std err	z	P> z	[0.025	0.975]
ma.L1	-0.1208	0.027	-4.553	0.000	-0.173	-0.069
ma.L2	-0.1308	0.028	-4.665	0.000	-0.186	-0.076
ma.L3	-0.0592	0.029	-2.051	0.040	-0.116	-0.003
ar.S.L7	-0.6681	0.504	-1.325	0.185	-1.657	0.320
ma.S.L7	-0.2592	0.521	-0.496	0.619	-1.279	0.761
ma.S.L14	-0.5739	0.473	-1.213	0.225	-1.501	0.353
sigma2	2.714e+06	9.58e+04	28.341	0.000	2.53e+06	2.9e+06

13149.490183657155

Figure 4.20: Model Coefficients

run to assess the quality of the model. As Figure 4.21 shows the residual of model is clearly a white noise, which means the algorithm models the data set well, also the trend and seasonality of data set is captured with algorithm.

The model set then was used for predication of electricity load in 2014, The predicated value from model and real value is demonstrated in Figure 4.22.

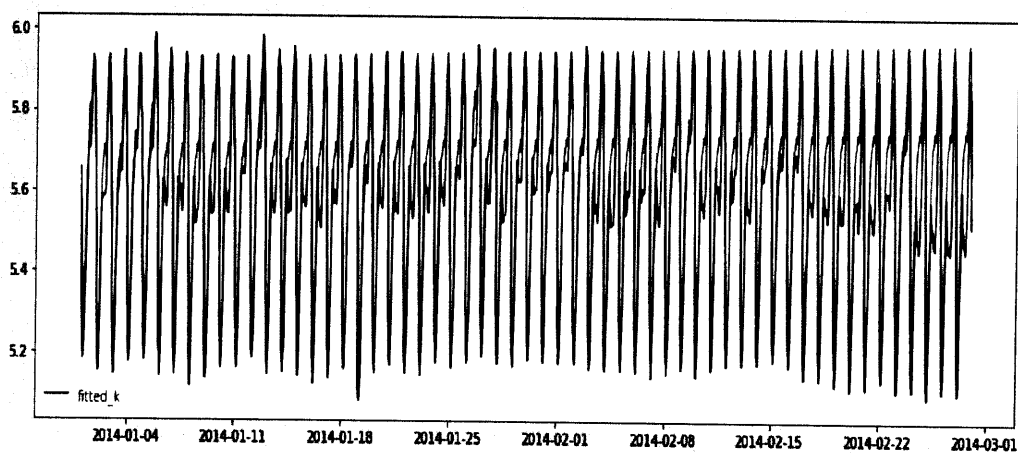


Figure 4.22: Predicate vs real DATA

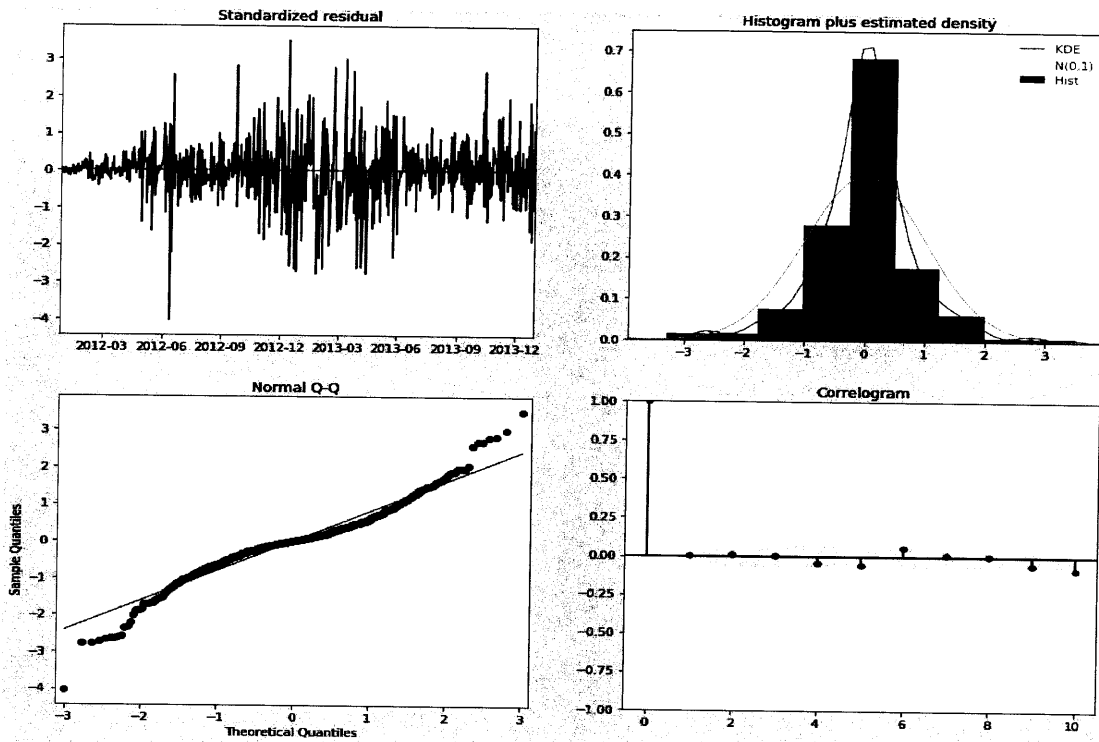


Figure 4.21: Diagnostic tool with $s=7$ in daily data set

4.3.1.1 Evaluation

To evaluate our model, The approach introduced with Wang et al. (6 09) was used. In his approach he used the ARIMA model coupled with decomposition. The data set was decomposed with The Ensemble Empirical Mode Decomposition proposed with Wu and Huang (2009) before the ARIMA fitting, then the ARIMA model was applied to each component separately. This approach was adopted and two separate data set were created as demonstrated in Figure 4.23 . For each components, Arima procedure was applied and the model obtained is $(3,1,3)(0,1,3,7)$, In addition, we predicates the consumption of February 2014 with our model and model was explained above.

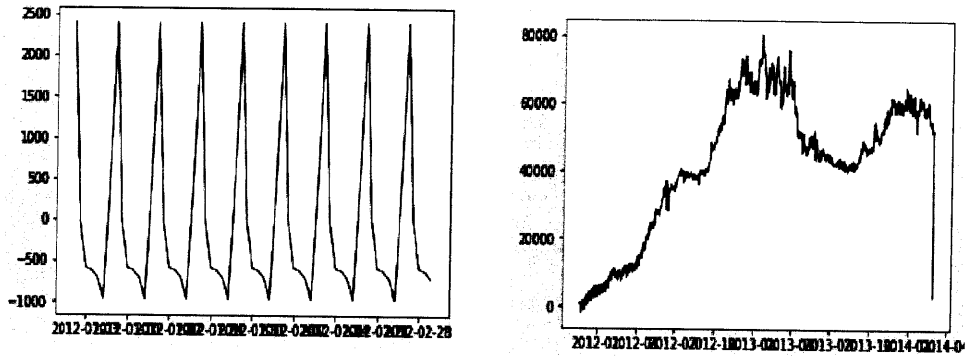


Figure 4.23: Ensemble Empirical Mode Decomposition result

The result is demonstrated in Table 4.3 and it shows that SARIMAX performs better.

Table 4.3: The SARIMAX Evaluation

	AIC	MAE
SARIMAX	13334	7514
Wang et al. (2015)	13697	7796

4.3.2 Experimental Result for Prophet

Prophet was used to perform decomposition of the data set in order to find the trend for the week of day and month of the year in the data set as shown in Figure 4.24. This is done by decomposition. The Figure clearly shows that consumption increased during the weekend and cold season. This representation justifies that holidays and weekends be fed as important factors to the prediction algorithm.

The Bank holidays of the UK were converted to dataframe along with weekend data extracted from daily dataset and fed to algorithm as holiday. Furthermore, since Prophet would be used to predict daily consumption for February 2014, the data set was split into training and test.

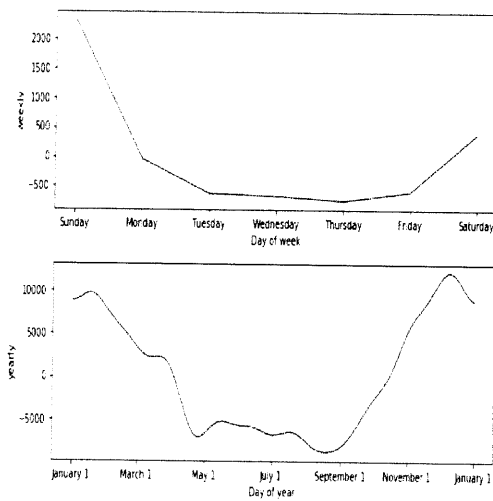


Figure 4.24: Seasonality on week and month

Figure 4.25 show the result of forecasting for test periods. The reading from 2011 to the end of January 2014 was used to train the model. The trained model was then used to predicate the reading for February 2014. The reading has shown with a scatter plot, and predicate values show with a blue line. by using Automatic changepoint detection in Prophet, the figure also indicates the changepoint in the dataset with vertical lines.

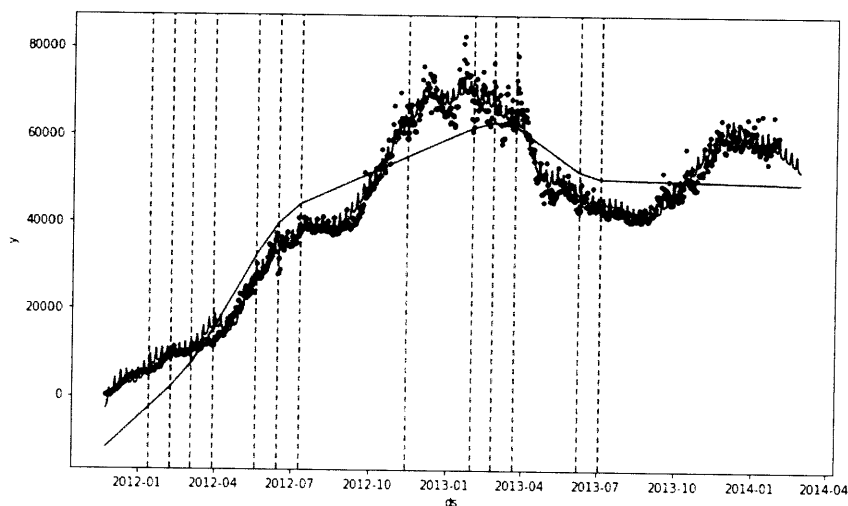


Figure 4.25: Forecasting with Prophet

To measure the efficiency of the algorithm the time series folding

demonstrated in Figure 3.8 along with three evaluation metrics explained in section 2.4.6 were implemented and demonstrated in Table 4.4.

4.3.3 Forecasting Evaluation

Whether time series modeling is applied, the trend, seasonality, and cyclicity of time series are critical to capture. As a result, data must be separated in such a way that the key attributes are preserved. To obtain this goal as explained in chapter 3, the “TimeSeriesSplit” was used for dividing the data set to train and test and the result of this evaluation for two proposed methods is illustrated in Table 4.4.

Then selected methods were used to predicate values from 1/1/2013 to 28/2/2014 and then the quality of the method was quantified with an evaluation index.

4.4 Chapter Summery

This chapter described the experimental regime and results. By using EDA and statistical tests key database aspects such as holidays were investigated and correlation coefficient between weather and consumption is calculated= -0.17. The embedded load profiling also will be tested and DEC, IDEC, and DynAe networks were compared with traditional clustering algorithms namely K-mean, Spectral, hierarchical, and GMM. Experimental results show that for all six clustering algorithms used for validation and implementation of load profiling, the most appropriate number of clusters was $k = 4$. Table 4.1 shows that deep clustering increased the SI index to 0.59 using Deep Embedded Clustering (DEC) and 0.63 for DynAE, which was the best performance. This demonstrates the effectiveness of the proposed framework for load profiling.

Table 4.4: Forecasting for 7 fold

train	test	SARIMA	Prophet
2011-11-23 to 2012-03-09,	2012-03-10 to 2012-06-20	MAE:5738.57 r2_score:0.12 MAPE: 21.61	MAE:6323.811 r2_score: -0.068 MAPE: 23.58
2011-11-23 to 2012-06-20 ,	2012-06-21 to 2012-10-01	MAP:3570.36 r2_score: -0.82 MAPE: 9.061	MAE:11339.27 r2_score: -14.99 MAPE: 28.30
2011-11-23 to2012-10-01	2012-10-02 to 2013-01-12	MAP:4583.69 r2_score:0.33 MAPE: 7.07	MAP:14649.35 r2_score: -4.25 MAPE: 22.85
2011-11-23 to2013-01-12	2013-01-13 to 2013-04-25	MAP:13358.42 r2_score: -3.60 MAPE: 22.755	MAP:16286.92 r2_score: -5.53 MAPE: 27.608
2011-11-to 2013-04-25	2013-04-26 to2013-08-06	MAP:11643.391 r2_score: -19.59 MAPE: 26.42	MAPE:5884.69 r2_score: -3.93 MAPE: 13.03
2011-11-23 to 2013-08-06	2013-08-07 to 2013-11-17	MAP:11561.14 r2_score: -6.22 MAPE: 22.90	MAP:19699.42 r2_score: -17.069 MAPE: 39.783
2011-11-23 to 2013-11-17	2013-11-18 to 2014-02-28	MAP:7885.505 r2_score: -2.427 MAPE: 45.166	MAP:4900.66 r2_score: -0.482 MAE: 33.72

Algorithm 2 was tested for load forecasting with enhanced SARIMA. The model attains were (0, 1, 3)(1, 1, 2, 7), and the coefficient of the model is represented. For evaluation time series splitting with 7 folds was used and SARIMA and prophet were used to predicate the day base load for February 2014 for the city of London.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

5.2 Future Work

Chapter 5

Conclusions and Future Work

Finally, we present the thesis conclusions based on the research questions as well as some ideas on possible future work and open questions.

5.1 Conclusions

The first question of this thesis was “How to use vast amounts of smart meter data to promote and improve the electricity grid’s efficiency and sustainability.” This was addressed with the creation of a comprehensive and current taxonomy for smart meter analysis. The suggested taxonomy was then published in two scholarly publications (Eskandarnia and Al-Ammal, 2022; Eskandarnia et al., 2022a). As the taxonomy describes, there are three major domains in smart meter analytics; namely, load forecasting, knowledge discovery, and power management.

The second question of this thesis was “What are computational methods to segment consumer base of multivariate big data produced by smart meter efficiently?”. First, this thesis highlighted the necessity within

the field for more advanced load profiling algorithms. Accordingly we developed an improved load profiling framework that makes use of deep clustering techniques. The proposed framework performed better than other classical hybrid frameworks such as (Sun et al., 2020), as shown by the results in Figure 4.19. The proposed framework was also published in a scientific journal (Eskandarnia et al., 2022b).

An extensive literature study revealed that load profiling is one of the most important tasks in smart grid analytics and sustainable energy practice. Load profiling divides the consumer into groups and generates a typical load profile for each group.

Because the results of load profiling are frequently considered as input to many other smart meter analysis activities (such as load analysis and demand response) this is a significant priority in smart meter analysis. Load profiling is also paramount to improving the performance of other sustainable energy management tasks such as short-term load forecasting and home energy management system (HEMS). All of these factors point to the necessity for more advanced load profiling algorithms.

Traditional clustering-based load profiling algorithms, on the other hand, necessitate a time-consuming feature reduction step to lower the dimensionality of smart meter readings, which are often a big temporal data set. The consistency of the produced load profile is frequently harmed as a result of the high complexity/dimensionality of the data provided by the feature reduction process. While deep learning has recently been applied for other realms of smart meter analysis, such as load forecasting, load disaggregation, and dynamic pricing; it has rarely been used for improving the performance of load profiling. By defining load profiling as a clustering problem and solving it using various deep learning clustering methods, this thesis contributed to the literature with a new technique, and subsequently answered the second research question.

The thesis compared three deep clustering algorithms based on autoencoders for generating load profiles of a highly dimensional data: DEC, IDEC, and DynAE. These deep neural networks learn non-linear mappings that enable data to be transformed into more *clustering-friendly representations* without the need for dimensionality reduction or feature selection steps.

To evaluate the proposed method, load profiling was also performed using three classical machine learning algorithms to benchmark their results with the results of the proposed autoencoder-based algorithms. According to the experimental results presented in chapter 4, DEC raised the SI index to 0.59. However, by mitigating feature randomness and feature drift, DynAE increased the index to 0.63. Furthermore, the results show that the proposed deep clustering methods can effectively segment houses according to their consumption and improve the quality of the load curve obtained for each cluster/group. Our proposed methods also address the hyperparameter values for the networks such as the learning rate and the number of iterations suitable for the load profiling task.

Comparing the load curves obtained from the proposed framework with classical machine learning algorithms (Figure 4.17 and Figure 4.18) demonstrates the ability of the proposed method to obtain clear and separable load curves for different groups of consumers.

The second research question was addressed and answered with running the proposed framework. As demonstrated in Figure 4.17, the Dynamic Autoencoder was able to group consumers into four distinct groups and provide a load profile for each group. The second question answer was then answered through grouping consumers to four groups and provided a TLP for each group. Furthermore, our proposed methods improved SI index with 25%.

The third research question was formulated as forecasting of integrated

load forecasting in STFL horizon. To answer this question and attain the last objective of the research, the prediction of big temporal data was investigated and two time series-based load forecasting methods namely SARIMA and prophet method were applied.

Electrical load forecasting was formulated as a data-driven topic. Furthermore, data were transferred as time-series data. Data then were re-sampled through aggregation techniques. Secondly, data were translated to uni variant time series. The Seasonal AutoRegressive Integrated Moving Averages (SARIAM) was chosen to model and predict the data since the seasonality and trend of the data set were demonstrated through visualization and statistical tests. The SARIMA process is computationally intensive due to a large number of parameters and hyperparameters. The upgraded technique, which combines a statistical parameter selection with the auto grid, was developed to address this problem. The best model was chosen using the AIC. After that, the model is fitted to the data set and tested using diagnostic tools. The model was trained and then used to predicate the consumption in the test data set. The proposed model was compared and evaluated with the model proposed with Wang et al. (2015). Our proposed enhanced SARIMA was published at a peer-reviewed conference (Eskandarnia and AlHammad, 2020).

Prophet is the second algorithm utilized to achieve this goal. Prophet was chosen because it can handle large-scale forecasting. The prophet's data visualization depicts the data set's weekday pattern. Bank holidays and weekends were extracted and fed to data as an important point to address this aspect of time series. The outcome was then assessed using three metrics and visualization. This solution was presented and published at the Smart Cities Conference (Eskandar et al., 2020). The third question was answered by predicating the electrical load required in London for February 2014.

5.2 Future Work

There are several deep learning techniques that can help in load profiling. Deep clustering followed by load profile generation using Generative Adversarial Networks (GANs) may produce better profiles and may be useful in generating typical loads for data augmentation purposes. The same technique may be used for potentially generating individual load profiles for new customers for testing or forecasting purposes.

Furthermore, the interplay between load profile quality and feature randomness and feature drift deserves to be further investigated, especially for cases where the evaluation dataset has been labeled. Also, by considering the time-series nature of smart meter readings, deep temporal clustering approaches as well as incremental methods are highly valuable in practice, and have not been investigated for the load profiling task.

References

- Aghabozorgi, S., Shirkhorshidi, A. S., and Wah, T. Y. (2015). Time-series clustering—a decade review. *Information Systems*, 53:16–38.
- Al-Wakeel, A. and Wu, J. (2016). K-means based cluster analysis of residential smart meter measurements. *Energy Procedia*, 88:754–760.
- Aljalbout, E., Golkov, V., Siddiqui, Y., Strobel, M., and Cremers, D. (2018). Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*.
- Alkhulaifi, A. and Aljohani, A. J. (2020). Investigation of deep learning-based techniques for load disaggregation, low-frequency approach. *Int. J. Adv. Comput. Sci. Appl*, 11:701–706.
- Amasyali, K. and El-Gohary, N. M. (2018). A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 81:1192–1205.
- Ambreen, K., Khalid, R., Maroof, R., Khan, H. N., Asif, S., and Iftikhar, H. (2017). Implementing critical peak pricing in home energy management using biography based optimization and genetic algorithm in smart grid. In *International Conference on Broadband and Wireless Computing, Communication and Applications*, pages 560–569. Springer.

- Amjady, N. and Keynia, F. (2008-10-01). Mid-term load forecasting of power systems by a new prediction method. *Energy Conversion and Management*, 49(10):2678–2687.
- Andrienko, N. and Andrienko, G. (2006). *Exploratory analysis of spatial and temporal data: a systematic approach*. Springer Science & Business Media.
- Asghar, M. R., Dán, G., Miorandi, D., and Chlamtac, I. (2017). Smart meter data privacy: A survey. *IEEE Communications Surveys & Tutorials*, 19(4):2820–2835.
- Assaf, T., Osman, A. H., and Hassan, M. (2016-04-06). Fair autonomous energy consumption scheduling based on game-theoretic approach for the future smart grid. In *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim)*, pages 235–239.
- Azaza, M. and Wallin, F. (2017). Evaluation of classification methodologies and features selection from smart meter data. *Energy Procedia*, 142:2250–2256.
- Basu, K., Debusschere, V., Bacha, S., Maulik, U., and Bondyopadhyay, S. (2015). Nonintrusive load monitoring: A temporal multilabel classification approach. *IEEE Transactions on industrial informatics*, 11(1):262–270.
- Bouktif, S., Fiaz, A., Ouni, A., and Serhani, M. A. (2020). Multi-sequence lstm-rnn deep learning and metaheuristics for electric load forecasting. *Energies*, 13(2):391.
- Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

- Chang, J., Wang, L., Meng, G., Xiang, S., and Pan, C. (2017). Deep adaptive image clustering. In *Proceedings of the IEEE international conference on computer vision*, pages 5879–5887.
- Chaouch, M. (2014). Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves. *IEEE Transactions on Smart Grid*, 5(1):411–419.
- Chatfield, C. (2018). *Introduction to multivariate analysis*. Routledge.
- Chen, G. (2015). Deep learning with nonparametric clustering. *arXiv e-prints*, pages arXiv–1501.
- Chen, H., Wang, Y.-H., and Fan, C.-H. (2020). A convolutional autoencoder-based approach with batch normalization for energy disaggregation. *The Journal of Supercomputing*, pages 1–18.
- Chen, T. (2012). A collaborative fuzzy-neural approach for long-term load forecasting in taiwan. *Computers & Industrial Engineering*, 63(3):663–670.
- Chen, T. and Wang, Y.-C. (2012). Long-term load forecasting by a collaborative fuzzy-neural approach. *International Journal of Electrical Power & Energy Systems*, 43(1):454–464.
- Christensen, M. H., Nozal, D. C., Kavadas, I., and Pinson, P. (2019). Data-driven learning from dynamic pricing data-classification and forecasting. In *2019 IEEE Milan PowerTech*, pages 1–6. IEEE.
- Craft, B. and Cairns, P. (2005). Beyond guidelines: what can we learn from the visual information seeking mantra? In *Ninth International Conference on Information Visualisation (IV'05)*, pages 110–118. IEEE.

- Da Silva, P. G., Ilić, D., and Karnouskos, S. (2013). The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading. *IEEE Transactions on Smart Grid*, 5(1):402–410.
- Dagum, E. B. and Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer.
- Dang-Ha, T.-H., Bianchi, F. M., and Olsson, R. (2017). Local short term electricity load forecasting: Automatic approaches. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4267–4274. IEEE.
- de Oliveira, E. M. and Cyrino Oliveira, F. L. (2018-02-01). Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods. *Energy*, 144:776–788.
- Deb, C., Zhang, F., Yang, J., Lee, S. E., and Shah, K. W. (2017). A review on time series forecasting techniques for building energy consumption. *Renewable and Sustainable Energy Reviews*, 74:902–924.
- Dedinec, A., Filiposka, S., Dedinec, A., and Kocarev, L. (2016). Deep belief network based electricity load forecasting: An analysis of macedonian case. *Energy*, 115:1688–1700.
- Devlin, M. A. and Hayes, B. P. (2019a). Load identification and classification of activities of daily living using residential smart meter data. In *2019 IEEE Milan PowerTech*, pages 1–6. IEEE.
- Devlin, M. A. and Hayes, B. P. (2019b). Non-intrusive load monitoring and classification of activities of daily living using residential smart meter data. *IEEE Transactions on Consumer Electronics*, 65(3):339–348.

- Dilokthanakul, N., Mediano, P. A., Garnelo, M., Lee, M. C., Salimbeni, H., Arulkumaran, K., and Shanahan, M. (2016). Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv e-prints*, pages arXiv–1611.
- Esa, N. F., Abdullah, M. P., and Hassan, M. Y. (2016). A review disaggregation method in non-intrusive appliance load monitoring. *Renewable and Sustainable Energy Reviews*, 66:163–173.
- Eskandar, E. M., Alammal, H., and Almadany, W. (2020). What prophet says about electrical consumption - forecasting techniques for big temporal data. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 543–548.
- Eskandarnia, E., Al-Ammal, H., Ksantini, R., and Hammad, M. (2022a). Deep learning techniques for smart meter data analytics: A review. *SN Computer Science*, 3(3):1–16.
- Eskandarnia, E., Al-Ammal, H. M., and Ksantini, R. (2022b). An embedded deep-clustering-based load profiling framework. *Sustainable Cities and Society*, 78:103618.
- Eskandarnia, E. and AlHammad, M. (2020). Predication of future energy consumption using sarimax. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 657–662.
- Eskandarnia, E. M. and Al-Ammal, H. M. (2022). A taxonomy of smart meter analytics: forecasting, knowledge discovery, and power management. *International Journal of Computing and Digital Systems*, 11(1):851–859.
- Fahiman, F., Erfani, S. M., Rajasegarar, S., Palaniswami, M., and Leckie, C. (2017). Improving load forecasting based on deep learning and k-

- shape clustering. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4134–4141. IEEE.
- Farkas, M. and Lacko, P. (2017). Using advanced audio generating techniques to model electrical energy load. In *International Conference on Engineering Applications of Neural Networks*, pages 39–48. Springer.
- Fekri, M. N., Patel, H., Grolinger, K., and Sharma, V. (2021). Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Applied Energy*, 282:116177.
- Garcia, F. C. C. and Macabebe, E. Q. B. (2017). Inferring appliance energy usage from smart meters using fully convolutional encoder decoder networks. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2075–2078.
- Ghamisi, P. and Benediktsson, J. A. (2014). Feature selection based on hybridization of genetic algorithm and particle swarm optimization. *IEEE Geoscience and remote sensing letters*, 12(2):309–313.
- Ghofrani, M., Hassanzadeh, M., Etezadi-Amoli, M., and Fadali, M. S. (2011). Smart meter based short-term load forecasting for residential customers. In *2011 North American Power Symposium*, pages 1–5. IEEE.
- Gong, X., Tang, B., Zhu, R., Liao, W., and Song, L. (2020). Data augmentation for electricity theft detection using conditional variational auto-encoder. *Energies*, 13(17):4291.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial

nets [c]//advances in neural information processing systems. *New York: ACM*, 26722680.

Gungor, V. C., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., and Hancke, G. P. (2011). Smart grid technologies: Communication technologies and standards. *IEEE transactions on Industrial informatics*, 7(4):529–539.

Guo, X., Gao, L., Liu, X., and Yin, J. (2017). Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759.

Hassani, H., Farajzadeh-Zanjani, M., Razavi-Far, R., Saif, M., and Palade, V. (2019). Design of a cost-effective deep convolutional neural network-based scheme for diagnosing faults in smart grids. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1420–1425. IEEE.

He, W. and Chai, Y. (2016). An empirical study on energy disaggregation via deep learning. In *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*, pages 338–342. Atlantis Press.

Hong, T., Wilson, J., and Xie, J. (2014). Long term probabilistic load forecasting and normalization with hourly information. *IEEE Transactions on Smart Grid*, 5(1):456–462.

Hosseini, S. S., Agbossou, K., Kelouwani, S., and Cardenas, A. (2017). Non-intrusive load monitoring through home energy management systems: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 79:1266–1274.

- Hsu, C.-C. and Lin, C.-W. (2017). Cnn-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, 20(2):421–429.
- Hu, Z., Bao, Y., Chiong, R., and Xiong, T. (2015). Mid-term interval load forecasting using multi-output support vector regression with a memetic algorithm for feature selection. *Energy*, 84:419–431.
- Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. OTexts.
- Ibrahim, M. S., Dong, W., and Yang, Q. (2020). Machine learning driven smart electric power systems: Current trends and new perspectives. *Applied Energy*, 272:115237.
- Iskandarnia, E., Al-Amal, H., and Al-Madany, W. (2020). Load forecasting in different scale and horizon-a review. In *3rd Smart Cities Symposium (SCS 2020)*, volume 2020, pages 464–470. IET.
- Jahromi, A. N., Sakhnini, J., Karimpour, H., and Dehghantanha, A. (2019). A deep unsupervised representation learning approach for effective cyber-physical attack detection and identification on highly imbalanced data. In *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pages 14–23.
- Jindal, A., Aujla, G. S., Kumar, N., Prodan, R., and Obaidat, M. S. (2018). Drums: Demand response management in a smart city using deep learning and svr. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE.
- Joshi, P. A. and Patel, J. J. (2018). Computational analysis and intelligent control of load forecasting using time series method. In *Proceedings*

of the International Conference on Intelligent Systems and Signal Processing, pages 297–306. Springer.

- Kakran, S. and Chanana, S. (2018). Smart operations of smart grids integrated with distributed generation: A review. *Renewable and Sustainable Energy Reviews*, 81:524–535.
- Kelly, J. and Knottenbelt, W. (2015). Neural nilm: Deep neural networks applied to energy disaggregation. In *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*, pages 55–64.
- Khan, Z. A., Hussain, T., Ullah, A., Rho, S., Lee, M., and Baik, S. W. (2020). Towards efficient electricity forecasting in residential and commercial buildings: A novel hybrid cnn with a lstm-ae based framework. *Sensors*, 20(5):1399.
- Khuntia, S. R., Rueda, J. L., and van der Meijden, M. A. (2016a). Forecasting the load of electrical power systems in mid-and long-term horizons: a review. *IET Generation, Transmission & Distribution*, 10(16):3971–3977.
- Khuntia, S. R., Tuinema, B. W., Rueda, J. L., and van der Meijden, M. A. (2016b). Time-horizons in the planning and operation of transmission networks: an overview. *IET Generation, Transmission & Distribution*, 10(4):841–848.
- Kim, M., Park, S., Lee, J., Joo, Y., and Choi, J. K. (2017). Learning-based adaptive imputation method with knn algorithm for missing power data. *Energies*, 10(10):1668.
- Kim, M. and Ramakrishna, R. (2005). New indices for cluster validity assessment. *Pattern Recognition Letters*, 26(15):2353–2363.

- Kodinariya, T. M. and Makwana, P. R. (2013). Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95.
- Kong, W., Dong, Z. Y., Hill, D. J., Luo, F., and Xu, Y. (2017). Short-term residential load forecasting based on resident behaviour learning. *IEEE Transactions on Power Systems*, 33(1):1087–1088.
- Kong, X., Kong, D., Yao, J., Bai, L., and Xiao, J. (2020). Online pricing of demand response based on long short-term memory and reinforcement learning. *Applied Energy*, 271:114945.
- Koprinska, I., Rana, M., and Agelidis, V. G. (2015). Correlation and instance based feature selection for electricity load forecasting. *Knowledge-Based Systems*, 82:29–40.
- Kumar, B. P. (2021). Open data for smart cities. In *Solving Urban Infrastructure Problems Using Smart City Technologies*, pages 185–211. Elsevier.
- Kuster, C., Rezgui, Y., and Mourshed, M. (2017-11-01). Electrical load forecasting models: A critical systematic review. *Sustainable Cities and Society*, 35:257–270.
- Kwac, J., Flora, J., and Rajagopal, R. (2014). Household energy consumption segmentation using hourly data. *IEEE Transactions on Smart Grid*, 5(1):420–430.
- Långkvist, M., Karlsson, L., and Loutfi, A. (2014). A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 42:11–24.
- Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical

- representations. In *Proceedings of the 26th annual international conference on machine learning*, pages 609–616.
- Lee, W.-J. and Hong, J. (2015). A hybrid dynamic and fuzzy time series model for mid-term power load forecasting. *International Journal of Electrical Power & Energy Systems*, 64:1057–1062.
- Li, F., Qiao, H., and Zhang, B. (2018). Discriminatively boosted image clustering with fully convolutional auto-encoders. *Pattern Recognition*, 83:161–173.
- Li, J. and Wang, F. (2020). Non-technical loss detection in power grids with statistical profile images based on semi-supervised learning. *Sensors*, 20(1):236.
- Li, S., Wang, P., and Goel, L. (2015). Short-term load forecasting by wavelet transform and evolutionary extreme learning machine. *Electric Power Systems Research*, 122:96–103.
- Li, Y., Guo, P., and Li, X. (2016). Short-term load forecasting based on the analysis of user electricity behavior. *Algorithms*, 9(4):80.
- Liu, C., Jin, Z., Gu, J., and Qiu, C. (2017-09-26). Short-term load forecasting using a long short-term memory network. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6.
- Liu, M., Liu, D., Sun, G., Zhao, Y., Wang, D., Liu, F., Fang, X., He, Q., and Xu, D. (2020). Deep learning detection of inaccurate smart electricity meters: A case study. *IEEE Industrial Electronics Magazine*, 14(4):79–90.

- Lu, R., Hong, S. H., and Zhang, X. (2018). A dynamic pricing demand response algorithm for smart grid: reinforcement learning approach. *Applied Energy*, 220:220–230.
- Lusis, P., Khalilpour, K. R., Andrew, L., and Liebman, A. (2017-11). Short-term residential load forecasting: Impact of calendar effects and forecast granularity. *Applied Energy*, 205:654–669.
- Ma, C. and Li, Y. (2016). Improving forecasting accuracy of annual runoff time series using rbfnn based on eemd decomposition. *DEStech Transactions on Engineering and Technology Research*, (emme).
- Ma, W., Fang, S., Liu, G., and Zhou, R. (2017). Modeling of district load forecasting for distributed energy system. *Applied Energy*, 204:181–205.
- Madhure, R. U., Raman, R., and Singh, S. K. (2020). Cnn-lstm based electricity theft detector in advanced metering infrastructure. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6. IEEE.
- Mahmud, A. A. and Sant, P. (2017). Real-time price savings through price suggestions for the smart grid demand response model. In *Smart Grid and Cities Congress and Fair (ICSG), 2017 5th International Istanbul*, pages 65–69. IEEE.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. (2015). Adversarial autoencoders. *arXiv e-prints*, pages arXiv–1511.
- Marino, D. L., Amarasinghe, K., and Manic, M. (2016). Building energy load forecasting using deep neural networks. In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pages 7046–7051. IEEE.

- Marulli, F. and Visaggio, C. A. (2019). Adversarial deep learning for energy management in buildings. In *SummerSim*, pages 50–1.
- McLoughlin, F., Duffy, A., and Conlon, M. (2013). Evaluation of time series techniques to characterise domestic electricity demand. *Energy*, 50(C):120–130.
- Messinis, G. M. and Hatziargyriou, N. D. (2018). Review of non-technical loss detection methods. *Electric Power Systems Research*, 158:250–266.
- Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., and Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514.
- Mohassel, R. R., Fung, A., Mohammadi, F., and Raahemifar, K. (2014). A survey on advanced metering infrastructure. *International Journal of Electrical Power & Energy Systems*, 63:473–484.
- Mrabah, N., Bouguessa, M., and Ksantini, R. (2020a). Adversarial deep embedded clustering: on a better trade-off between feature randomness and feature drift. *IEEE Transactions on Knowledge and Data Engineering*.
- Mrabah, N., Khan, N. M., Ksantini, R., and Lachiri, Z. (2020b). Deep clustering with a dynamic autoencoder: From reconstruction towards centroids construction. *Neural Networks*, 130:206–228.
- Muratori, M. and Rizzoni, G. (2016). Residential demand response: Dynamic energy management and time-varying electricity pricing. *IEEE Transactions on Power systems*, 31(2):1108–1117.
- Nabil, M., Ismail, M., Mahmoud, M., Shahin, M., Qaraqe, K., and Serpedin, E. (2019). Deep learning-based detection of electricity theft cyber-

- attacks in smart grid ami networks. In *Deep Learning Applications for Cyber Security*, pages 73–102. Springer.
- Nafi, N. S., Ahmed, K., Gregory, M. A., and Datta, M. (2016). A survey of smart grid architectures, applications, benefits and standardization. *Journal of Network and Computer Applications*, 76:23–36.
- Niu, X., Li, J., Sun, J., and Tomsovic, K. (2019). Dynamic detection of false data injection attack in smart grid using deep learning. In *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–6. IEEE.
- Ozawa, A., Furusato, R., and Yoshida, Y. (2016). Determining the relationship between a household’s lifestyle and its electricity consumption in japan by analyzing measured electric load profiles. *Energy and Buildings*, 119:200–210.
- Panapakidis, I. P. and Christoforidis, G. C. (2017). Implementation of modified versions of the k-means algorithm in power load curves profiling. *Sustainable Cities and Society*, 35:83–93.
- Paresh, S., Thokala, N. K., and Chandra, M. G. (2019). Electrical load disaggregation using a two-stage deep learning approach. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, pages 366–367.
- Patyn, C., Reymond, M., Rădulescu, R., Deconinck, G., and Nowé, A. (2018). Reinforcement learning for demand response of domestic household appliances. *Adaptive Learning Agents 2018 Proceedings*, pages 1–7.
- Phinikarides, A., Makrides, G., Zinsser, B., Schubert, M., and Georghiou,

- G. E. (2015). Analysis of photovoltaic system performance time series: Seasonality and performance loss. *Renewable Energy*, 77:51–63.
- Quilumba, F. L., Lee, W.-J., Huang, H., Wang, D. Y., and Szabados, R. L. (2014). Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities. *IEEE Transactions on Smart Grid*, 6(2):911–918.
- Rasmus, A., Valpola, H., Honkala, M., Berglund, M., and Raiko, T. (2015). Semi-supervised learning with ladder networks. In *IEEE Conference on Neural Information Processing Systems*, pages 3546–3554. Neural Information Processing Systems Foundation.
- Razavi, S. E., Arefi, A., Ledwich, G., Nourbakhsh, G., Smith, D., and Minakshi, M. (2020). From load to net energy forecasting: Short-term residential forecasting for the blend of load and pv behind the meter. *IEEE Access*.
- Rodriguez Fernandez, M., Garcia, A. C., Alonso, I. G., and Casanova, E. Z. (2016-01). Using the big data generated by the smart home to improve energy efficiency management. *Energy Efficiency*, 9(1):249–260.
- Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.
- Shafie-khah, M., Javadi, S., Siano, P., and Catalão, J. (2017). Optimal behavior of smart households facing with both price-based and incentive-based demand response programs. In *PowerTech, 2017 IEEE Manchester*, pages 1–6. IEEE.
- Shateri, M., Messina, F., Piantanida, P., and Labeau, F. (2020). Privacy-cost management in smart meters using deep reinforcement learning.

In *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pages 929–933. IEEE.

Shuai, Z., Hu, Y., Peng, Y., Tu, C., and Shen, Z. J. (2017). Dynamic stability analysis of synchronverter-dominated microgrid based on bifurcation theory. *IEEE Transactions on Industrial Electronics*.

Shumway, R. H. and Stoffer, D. S. (2017). *Time series analysis and its applications: with R examples*. Springer.

Singh, S. and Majumdar, A. (2017). Deep sparse coding for non-intrusive load monitoring. *IEEE Transactions on Smart Grid*, 9(5):4669–4678.

Singhal, V., Maggu, J., and Majumdar, A. (2018). Simultaneous detection of multiple appliances from smart-meter measurements via multi-label consistent deep dictionary learning and deep transform learning. *IEEE Transactions on Smart Grid*, 10(3):2969–2978.

Soykan, E. U., Bilgin, Z., Ersoy, M. A., and Tomur, E. (2019). Differentially private deep learning for load forecasting on smart grid. In *2019 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE.

Srivastava, A., Pandey, A. S., and Singh, D. (2016). Short-term load forecasting methods: A review. In *2016 International Conference on Emerging Trends in Electrical Electronics & Sustainable Energy Systems (ICETEESES)*, pages 130–138. IEEE.

Sun, L., Zhou, K., and Yang, S. (2018-01-10). Regional difference of household electricity consumption: An empirical study of jiangsu, china. *Journal of Cleaner Production*, 171:1415–1428.

Sun, L., Zhou, K., and Yang, S. (2020). An ensemble clustering based framework for household load profiling and driven factors identification. *Sustainable Cities and Society*, 53:101958.

- Sun, M., Zhang, T., Wang, Y., Strbac, G., and Kang, C. (2019). Using bayesian deep learning to capture uncertainty for residential net load forecasting. *IEEE Transactions on Power Systems*, 35(1):188–201.
- Tariq, M., Khalid, A., Ahmad, I., Khan, M., Zaheer, B., and Javaid, N. (2017). Load scheduling in home energy management system using meta-heuristic techniques and critical peak pricing tariff. In *International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pages 50–62. Springer.
- Tasdighi, M., Ghasemi, H., and Rahimi-Kian, A. (2014-01). Residential microgrid scheduling based on smart meters data and temperature dependent thermal load modeling. *IEEE Transactions on Smart Grid*, 5(1):349–357.
- Taylor, S. J. and Letham, B. (2018). Forecasting at scale. *The American Statistician*, 72(1):37–45.
- Tukey, J. W. et al. (1977). *Exploratory data analysis*, volume 2. Reading, MA.
- Ullah, A., Haydarov, K., Ul Haq, I., Muhammad, K., Rho, S., Lee, M., and Baik, S. W. (2020). Deep learning assisted buildings energy consumption profiling using smart meter data. *Sensors*, 20(3):873.
- Vercamer, D., Steurtewagen, B., Van den Poel, D., and Vermeulen, F. (2015). Predicting consumer load profiles using commercial and open data. *IEEE Transactions on Power Systems*, 31(5):3693–3701.
- Villar-Rodriguez, E., Del Ser, J., Oregi, I., Bilbao, M. N., and Gil-Lopez, S. (2017). Detection of non-technical losses in smart meter data based on load curve profiling and time series analysis. *Energy*, 137:118–128.

- Wang, Y., Chen, Q., Gan, D., Yang, J., Kirschen, D. S., and Kang, C. (2018a). Deep learning-based socio-demographic information identification from smart meter data. *IEEE Transactions on Smart Grid*, 10(3):2593–2602.
- Wang, Y., Chen, Q., Hong, T., and Kang, C. (2018b). Review of smart meter data analytics: Applications, methodologies, and challenges. *IEEE Transactions on Smart Grid*.
- Wang, Y., Chen, Q., Kang, C., Zhang, M., Wang, K., and Zhao, Y. (2015). Load profiling and its application to demand response: A review. *Tsinghua Science and Technology*, 20(2):117–129.
- Wang, Y., Chen, Q. X., Kang, C. Q., and Xia, Q. (2016-09). Clustering of electricity consumption behavior dynamics toward big data applications. *IEEE Transactions on Smart Grid*, 7(5):2437–2447.
- Wei, Y., Zhang, X., Shi, Y., Xia, L., Pan, S., Wu, J., Han, M., and Zhao, X. (2018). A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews*, 82:1027–1047.
- Wijaya, T. K., Vasirani, M., Humeau, S., and Aberer, K. (2015). Cluster-based aggregate forecasting for residential electricity demand using smart meter data. In *2015 IEEE international conference on Big data (Big data)*, pages 879–887. IEEE.
- Wu, Z. and Huang, N. E. (2009). Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in adaptive data analysis*, 1(01):1–41.
- Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding

- for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR.
- Yang, B., Fu, X., Sidiropoulos, N. D., and Hong, M. (2017). Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International conference on machine learning*, pages 3861–3870. PMLR.
- Yang, S.-l., Shen, C., and Zhaou, K.-L. (2013). A review of electric load classification in smart grid environment. *Renewable and Sustainable Energy Reviews*, 24:103–110.
- Yang, Y., Hong, W., and Li, S. (2019a). Deep ensemble learning based probabilistic load forecasting in smart grids. *Energy*, 189:116324.
- Yang, Y., Li, W., Gulliver, T. A., and Li, S. (2019b). Bayesian deep learning-based probabilistic load forecasting in smart grids. *IEEE Transactions on Industrial Informatics*, 16(7):4703–4713.
- Ye, Y., Qiu, D., Wu, X., Strbac, G., and Ward, J. (2020). Model-free real-time autonomous control for a residential multi-energy system using deep reinforcement learning. *IEEE Transactions on Smart Grid*, 11(4):3068–3082.
- Yildiz, B., Bilbao, J. I., Dore, J., and Sproul, A. B. (2017). Recent advances in the analysis of residential electricity consumption and applications of smart meter data. *Applied Energy*, 208:402–427.
- Yildiz, B., Bilbao, J. I., and Sproul, A. B. (2017-06-01). A review and analysis of regression and machine learning models on commercial building electricity load forecasting. *Renewable and Sustainable Energy Reviews*, 73:1104–1122.

- Yin, X., Li, X., Zhang, Y., Zhang, T., Lu, C., Ai, Q., Li, Z., and Sun, Z. (2020). A survey of deep learning and its application in distribution network. In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC)*, pages 643–646. IEEE.
- Yu, M. and Hong, S. H. (2017). Incentive-based demand response considering hierarchical electricity market: A stackelberg game approach. *Applied Energy*, 203:267–279.
- Yuan, L., Ma, J., Gu, J., Wen, H., and Jin, Z. (2020). Featuring periodic correlations via dual granularity inputs structured rnns ensemble load forecaster. *International Transactions on Electrical Energy Systems*, 30(11):e12571.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.
- Zhang, P., Cheng, H., Zou, B., Dai, P., and Ye, C. (2019). Load data mining based on deep learning method. In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering*, pages 1–5.
- Zhou, K., Yang, S., and Shao, Z. (2017). Household monthly electricity consumption pattern mining: A fuzzy clustering-based model and a case study. *Journal of cleaner production*, 141:900–908.
- Zhou, L., Ouyang, X., Ying, H., Han, L., Cheng, Y., and Zhang, T. (2018). Cyber-attack classification in smart grid via deep neural network. In *Proceedings of the 2nd International Conference on Computer Science and Application Engineering*, pages 1–5.

- Zhou, X., Canady, R., Li, Y., Koutsoukos, X., and Gokhale, A. (2020). Overcoming stealthy adversarial attacks on power grid load predictions through dynamic data repair. In *International Conference on Dynamic Data Driven Application Systems*, pages 102–109. Springer.
- Zhu, J., Shen, Y., Song, Z., Zhou, D., Zhang, Z., and Kusiak, A. (2019). Data-driven building load profiling and energy management. *Sustainable Cities and Society*, 49:101587.
- Zor, K., Timur, O., and Teke, A. (2017). A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. In *2017 6th International Youth Conference on Energy (IYCE)*, pages 1–7. IEEE.

المخلص

انت تقنيات الشبكة الذكية مع وعود كثيرة بإدارة وترشيد أفضل للطاقة الكهربائية وأنظمتها ، وهو ما يعد خطوة أساسية نحو تحقيق الطاقة المستدامة. توفر أجهزة الشبكة الذكية مثل العدادات الذكية بيانات هائلة تمكن من اتخاذ تدابير مستدامة إضافية في قطاع إدارة الطاقة. ويُنظر إلى تحليلات العدادات الذكية على أنها اللبنة الأساسية للحصول على طاقة مستدامة لأنها تحسن من الكفاءة التشغيلية ، وتعزز الثقة بشبكة الطاقة، وتتيح إدارة أفضل لمواردها. علماً بأن هذا مجال بحثي متعدد التخصصات ويجذب خبراء من تقنية الكمبيوتر وعلوم البيانات والهندسة الكهربائية ومجالات أخرى، وقد نما مؤخراً الاهتمام به بسبب ثروة البيانات المتولدة من العدادات الذكية.

تهدف هذه الأطروحة إلى تصنيف موحد للمجالات المختلفة لتحليل بيانات العدادات الذكية ووظائفها وتقنياتها الأساسية. والهدف من ذلك هو فهم أشمل لاتجاهات البحث العلمي الحالية وأساليبها والفرص التي تتيحها بشكل أفضل. ويعتبر تنميط الأحمال الكهربائية - وهي مهمة ضمن نطاق اكتشاف المعرفة - خطوة أساسية في العديد من مهام تحليل بيانات العدادات الذكية ، مثل التنبؤ والتخطيط، والتي تؤثر بشكل مباشر على إدارة الطاقة المستدامة. ونظراً لأن معظم بيانات العدادات الذكية في العالم الحقيقي غير معنونة أو مصنفة مسبقاً ، فإن نهج تنميط الأحمال يعد من التعلم الغير مشرف عليه ويتم عبر خوارزميات التكتل clustering حيث يهدف إلى تجميع العملاء المرتبطين بناءً على أنماط الاستخدام. ومع ذلك ، وبسبب ما يسمى بلعنة تعدد الأبعاد ، فإن خوارزميات التكتل التقليدية تعمل بشكل سيئ وغالباً ما تؤدي إلى منحنيات أحمال غير مناسبة أو غير مجدية. وعلاوة على ذلك ، يلزم إجراء معالجة مسبقة شاملة للبيانات ، بما في ذلك تقليل عدد معالم البيانات وتحويلها. من ناحية أخرى ، تتخفف موثوقية نتائج أنماط الأحمال بسبب التعقيد الكبير للبيانات الناتجة عن طرق تقليل عدد المعالم feature reduction. تبحث هذه الدراسة في البنى المختلفة للتعلم العميق المستخدمة لكل مجال ، ثم تقترح إطاراً جديداً مضمناً التكتل العميق deep clustering لمهمة تنميط الأحمال الكهربائية.

ويتفرد إطار العمل الذي تقترحه الدراسة لتحديد أنماط الحمل الكهربائي من خلال شقين. أولاً ، باستخدام أداة التشفير التلقائي ، يمثل إطار العمل البيانات كتسلسل هرمي داخل طبقات الشبكة العميقة ، مما يسمح بتقليل الأبعاد وفصل القرار اللاخطي بشكل فعال بين المجموعات على مستوى عنق الزجاجة للمشفّر التلقائي autoencoder . يتم تحقيق ذلك من خلال تنفيذ التجميع المستند إلى وحدة التشفير التلقائي والذي يقوم تلقائياً بتحويل بيانات العداد الذكي إلى معالم أكثر ملاءمة للتجميع والتي تحتفظ بخصائص البيانات الأصلية. ثانياً ، يدمج إطار العمل تقليل الأبعاد والتكتل في إطار عمل تعليمي واحد غير خاضع للإشراف في جميع مراحل العمل. بالإضافة إلى ذلك ، تم التحقق من النموذج القائم على السلاسل الزمنية للتنبؤ بالحمل.

ويتم بعد ذلك تقييم واختبار كل من إطار عمل التكتل العميق وطرق التنبؤ باستخدام مجموعات بيانات حقيقية للعدادات الذكية ، وتظهر النتائج التجريبية أن الإطار المقترح ينتج أنماط أفضل للأحمال وبشكل ملحوظ ، خصوصاً عند مقارنته بخوارزميات التكتل الكلاسيكية وكذلك الأطر المختلفة السابقة المقترحة في أبحاث تحليل البيانات.



استمارة إيداع البحث

الهام محمد اسكندر اسكندرنيا			اسم الباحث:
فاعلية أساليب التعلم العميق لتحليل بيانات العدادات الذكية			عنوان البحث:
2022			سنة النشر:
هندسة علوم الحاسوب	القسم:	كلية تقنية المعلومات	الكلية:
39815815	رقم الاتصال:	19927274@stu.uob.edu.bh	البريد الإلكتروني:

إقرار الباحث بالموافقة على الشروط التالية:

يحق لجامعة البحرين إتاحة وتوفير الرسالة أو الأطروحة المذكورة أعلاه في مكتبة الجامعة أو من خلال موقع الإسهامات البحثية للجامعة أو على نطاق أوسع في الخارج؛ كما ويمكن إعادة طباعة الرسالة إذا اقتضت الحاجة لذلك أو رقميتها مع إتاحتها على شبكة الأنترنت وذلك حسب تقدير إدارة الجامعة.

تنطبق هذه الشروط على جميع النسخ بما في ذلك النسخ الإلكترونية

تعتبر هذه الرسالة أو الأطروحة من المواد المحمية والتي تخضع إلى حقوق النشر والتأليف، ولا يجوز نشر أي اقتباس من الرسالة بدون توثيق المصدر.

التاريخ: 20 أبريل 2022

توقيع الباحث: الهام محمد اسكندرنيا

جامعة البحرين
كلية تقنية المعلومات



فاعلية أساليب التعلم العميق للتحليل بيانات العدادات الذكية

أطروحة مقدمة كجزء من متطلبات الحصول على
درجة الدكتوراة في علوم الحوسبة والمعلومات

اعداد

الهام محمد اسكندر اسكندرنيا

19927274

اشراف

د. رياض قسطنطيني

(أستاذ مشارك)

د. هشام العمال

(أستاذ مساعد)

جامعة البحرين

مملكة البحرين

يونيو 2022