Deep Learning based Approaches for Cost Effective Short-term Energy Load Forecasting And Consumer Behaviour Modelling in Households

Ghulam Mohi-Ud-Din

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

2020

Acknowledgements

Firstly, all praise is due to the almighty ALLAH , the most beneficent, the most merciful and the lord of the 'Alamin', for granting me the determination and capability to complete my long journey during this PhD study.

I would like to especially express my sincere gratitude to all of my supervisors Dr. Angelos Marnerides, Prof. Qi Shi, and Dr. Chelsea Dobbins for their timely suggestions, continuous encouragement, inspired guidance, and invaluable supervision that encouraged me to face any challenges during my PhD study and complete it successfully. I would also like to thank Dr. Aine MacDermott for their valuable feedback on my research work which assisted in plublishing the work. I would like to thank them for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis.

Finaly, words cant express my feeling toward my parents, brother and sister who deserve great thanks for their encouragement during my study and supporting me to finish my PhD.

To my parents.

Abstract

Today, there is a lot of enthusiasm to fulfil global energy needs from alternative energy resources. Due to the increasing demand for electricity, the traditional electricity market relies on decisions to plan electricity systems, and to generate and distribute electricity to their consumers to balance demand and supply. The peak demands of electricity highly affect these decisions and often cause system failure and shortage of electricity. By predicting energy requirements, these peak demands and the uncertainties in human behaviour in households are optimised to balance the load through various demand response programmes. A smart grid ecosystem requires intelligent Home Energy Management Systems (HEMSs) to profile highly non-stationary and non-linear measurements and conduct correlations of such measurements with diverse inputs (e.g. environmental factors) in order to improve the end-user experience, as well as to aid the overall demand-response optimisation process.

The huge amount of energy consumption information collected from the individual appliances opens up lots of opportunities to mine the hidden patterns in the data in order to understand the human behaviour related to energy usage. However, processing huge amounts of information for analysis purposes demands lots of resources e.g. time and computational power. Parallelisation techniques allow the processing of large amounts of data while requiring less computational time. However, neural networks widely used in data processing are highly complex to parallelise during the model parallelisation due to their sequential nature. A key challenge here is to

exploit the parallelisation capabilities of hardware as well as software in terms of multicore/multithreaded CPUs and GPUs (Graphical Processing Units).

To overcome these challenges, in this research work, we propose a new unified approach to predict day, week and month-wide energy consumption by reducing the computation resources and model human behaviour in households in order to save scarce energy resources and improve demand response programmes. We go beyond current profiling schemes by proposing Deep COLA; a Deep Competitive Learning Algorithm that addresses limitations of high dimensional data and enables accurate modelling of appliance-level energy consumption. We show that our proposed scheme is far more computationally efficient and scalable data-wise than three conventional clustering approaches namely, K-Means, DBSCAN and SOM, using real household datasets.

This research work includes a number of contributions. The first is a dominant feature selection algorithm from the pool of features to increase the performance of the forecasting model. The second is a prediction model based on deep learning algorithms to improve the forecasting accuracy and processing time. The third is a clustering algorithm based on competitive learning to profile day, week and month-wide energy consumption patterns, using appliance-level data for a given household. The current methods are based on K-Means, DBSCAN and rule mining which require expert knowledge to get improved clustering. However, the proposed concept of competitive learning allows to extract compact and well separated clusters. This approach automatically extracts the optimal number of clusters without using Elbow, Silhouette or Bootstrap methods commonly seen in the existing work. The fourth is the profiling of appliance-level energy consumption in synergy with environmental factors in order to reveal per-household behavioural characteristics under three associations: appliance-to-appliance, appliance-to-time and appliance-to-environment. The last one is a parallelisation approach in the form of data parallelisation to forecast energy consumption by utilizing large amounts of data.

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

The next list describes several abbreviations that will be later used within the body of the document.

$ANFIS$	Adaptive Neuro-based Fuzzy Inference System
$ANOVA$	Analysis of variance
$ARIMA$	Autoregressive Integrated Moving Average Model
$ARIMAX$	Autoregressive Integrated Moving Average with Explanatory Variable
$ARMA$	Autoregressive Moving Average Model
CI	Calinski-Harabasz Index
CNN	Convolutional Neural Network
DBN	Deep Belief Network
$DBSCAN$	Density Based Clustering Algorithm
$DeepCOLA$	Deep competitive learning algorithm
DI	Davies Bouldin Index
DNN	Deep Neural Network
DR	Demand Response

DSM	Demand Side Management
DTW	Dynamic Time Warping
ELM	Extreme Learning Machine
$FF-DNN$	Feed-Forward Deep Neural Network
FNN	Feed-Forward Neural Network
HC	Hierarchical Clustering
$HDFS$	Hadoop Distributed File System
$HEMS$	Home Energy Management System
HMM	Hidden Markov Modeling
$LASSO$	Least Absolute Shrinkage and Selection Operator
$LTLF$	Long Term Load Forecasting
MAE	Mean Absolute Error
$MAPE$	Mean Absolute Percentage Error
MLR	Multiple Linear Regression
MS	Mean Shift
MSE	Mean Squared Error
$MTLF$	Medium Term Load Forecasting
NN	Neural Network
PEV	Plug-in electric vehicle

PSO	Least Square Support Vector Regression
PSO	Particle Swarm Optimization
PV	Photovoltaics
$R-DNN$	Recurrent Deep Neural Network
$ReLU$	Rectified Linear Unit
RES	Renewable Energy Sources
RF	Random Forest
$RMSE$	Root Mean Squared Error
RNN	Recurrent Neural Network
$RSTM$	Long Short-Term Memory
SI	Silhouette Index
SOM	Self Organizing Maps
$SPCC$	Sample Pearson Correlation Coefficient
$STAR$	Smooth Transition Auto-Regressive Model
$STLF$	Short Term Load Forecasting
$STPAR$	Smooth Transition Periodic Auto-Regression
SVM	Support Vector Machine
SVR	Support Vector Regression
$UK-Dale$	UK Domestic Appliance-Level Electricity

$VSTLF$	Very Short Term Load Forecasting
$YARN$	Yet Another Resource Negotiator

List of Publications

- Qublai Khan Ali Mirza, **Ghulam Mohi-Ud-Din**, and Irfan Awan. A cloud-based energy efficient system for enhancing the detection and prevention of modern malware. Proceedings - International Conference on Advanced Information Networking and Applications, AINA, 2016-May: pages 754-761, 2016.

- **Ghulam Mohi-Ud-Din** and Angelos K. Marnerides. Short term power load forecasting using deep neural networks. 2017 International Conference on Computing, Networking and Communications (ICNC), pages 594-598, 2017

- **Ghulam Mohi-Ud-Din**, Andreas U Mauthe, and Angelos K Marnerides. Appliance-level Short-Term Load Forecasting using Deep Neural Networks. IEEE International Conference on Computing, Networking and Communications (ICNC) 2018, 2018.

- Arjmand Naveed, Fun Hu, Tshiamo Sigwele, **Ghulam Mohi-Ud-Din**, Misfa Susanto, and Muhammad Ali. Similarity Analyzer For Semantic Interoperability Of Electronic Health Records Using Artificial Intelligence (AI). Journal of engineering and scientific research (JESR), 2019.

- **Ghulam Mohi-Ud-Din**, Angelos K Marnerides, Qi Shi, Chelsea Dobbins, and Aine Mac Dermott. Deep COLA: A Deep COmpetitive Learning Algorithm for Future Home Energy Management Systems. IEEE Transactions on Emerging Topics in Computational Intelligence, 2020.

Chapter 1

Introduction

1.1 Overview

The prediction of the energy consumption requirements at the macro-level (i.e. distribution level) as well as at the micro-level (i.e. appliance level in the households) plays a crucial role in the realisation of the vision of the Smart Grid (SG). The prediction of the energy consumption for various time intervals such as short-term, medium-term and long-term essentially enable decision makers to plan for the generating, buying or selling of the excess of energy in the market. The following Figure 1.1 indicates an overall worldwide increasing trend in the energy

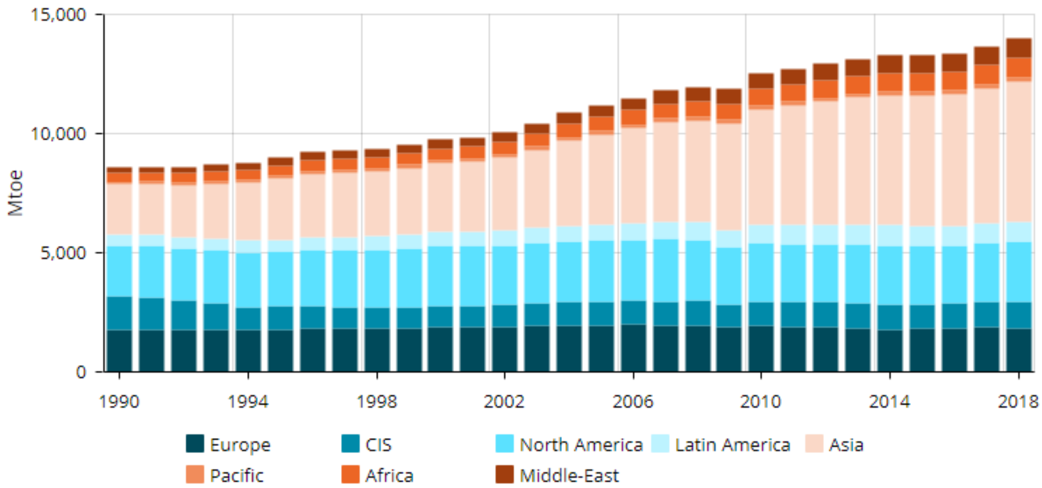

Figure 1.1: Worldwide energy consumption trend over the years [1]

consumption over the years. The author in [1] has stated that the increase in the usage of energy mainly happened due to the uninterrupted economic growth and the growing energy demands. This rising trend, in particular, is observed in North America, Asia, Africa and the Middle-East as shown in Figure 1.1. Various other factors, which affected the energy consumption over the years, were related to environmental conditions such as weather (e.g. hot summer and cold winter) and observed specifically in the United States [1]. The decision-making procedure followed by energy market agents in the energy market relies heavily on accurate electricity load forecasts at all levels including short, medium and long-term time durations.

Essentially, the energy market agents alongside having accurate energy load predictions, also require and make decisions based on, the understanding of individual consumers from domestic as well as corporate sectors [3]. With the introduction of Advanced Metering Infrastructure (AMI), large amounts of energy consumption data will be available from the individual appliances at the consumer premises. The new AMI systems are capable of recording data at a fine-grained level which opens up a new horizon of opportunities in terms of efficient energy management, understanding human behaviour, developing new applications in the energy market etc. The extraction of information from the large amount of consumption data requires high processing power in order to provide inferences to the energy market agents for decision making.

In the early days due to the monopoly, the government-controlled electricity companies ensured consistent energy supply through short-term energy forecasts and planning generation and transmission using long-term energy predictions [4]. This has changed due to the introduction of deregulation and the competitiveness in the energy sector. The electricity across the world is bought and sold based on the policies of spot and derivative markets [5]. The electricity load forecasting is critical in the energy sector for the decision making process. The errors in decision making for buying and selling electricity can be costly for the companies in the balancing market and often result in financial losses. The business requirements of electricity load forecasting and load profiling are summarised as follows:

- Energy demands: The short-term electricity load forecasting is required to plan energy demands and transmission. Without load forecasting, excess of energy cannot be stored using the current available storage technologies and shortage of energy cannot be avoided at the time of use. Therefore, energy utilities forecast electricity demands and purchase only the required energy [6], [7].

- Consumer profiling: The extreme peak usage of energy across the day has a diverse effect on the system planning and purchasing of electricity. The consistent supply of electricity to the consumers requires fewer peaks in the electricity usage. The electricity companies profile their consumers' behaviour and provide them incentives to reduce or shift their electricity consumption to other parts of the day in order to balance the consumption [8].

- Computational cost: With the advancement of technology, the smart meters are capable of recording energy consumption from individual appliances at very small intervals. The large datasets of energy consumption from these appliances allow the suppliers to extract patterns to understand consumer behaviour. However, the large datasets require high computation overhead for processing to get load prediction and consumer profiling [9].

1.2 Problem Statement

Today, there is a lot of enthusiasm to fulfil the global energy needs from alternative energy resources. The alternative energy is an emerging concept, which is the energy produced by means other than from the fossil fuels such as coal, oil and natural gas. Traditionally, sources such as water, wind and solar energy have been the primary ways to produce alternative energy. Our whole infrastructure, from households to commercial industries, depends on electricity. In traditional power systems, electricity is generated centrally and supplied to end-customers. The traditional power systems cannot fulfil the increasing energy requirements with high reliability. Thus, it is essential to transform traditional power systems to SG, which will be able to integrate energy from all kinds of sources and further supply energy based on demand.

The SG allows utilisation of energy resources efficiently with the proper planning of energy generation, integration and distribution. A smart meter in SG not only helps to utilise electricity efficiently by displaying energy consumption but also uses renewable sources at their full potential and reduces power shortage in developing countries. The smart meters and the appliances connected with them generate data at a large scale that eventually possess Big Data properties such as volume, variety, velocity and veracity [10]. The volume refers to the vast amount of data generated and collected each second. In the past, there was mostly structured data but now data is unstructured and has many varieties, e.g. text, images, video and voice. The velocity refers to the speed at which data is generated and propagated across the network. Another big data property, veracity, demands that data must have quality and generate meaningful results, to take right actions after the decision-making process [11].

There is a major challenge for the power industries to analyse data beyond just extracting information for billing purposes. The reason is that smart meter data growth is exponential and contains hidden structures that can provide information about fault detection, theft detection and demand response management [12]. The generation and consumption of energy requires the grid to meet peak demands by the consumers to achieve continuous supply of electricity. The peak demands of electricity occur due to the harmonization of energy consumption activities of the consumers in a particular area. As a result, high cost is required to build grid infrastructure in order to meet peak demands. These costs are obtained from the consumers in terms of high energy prices [13]. The energy systems in many countries are being upgraded with the adoption of energy generation through renewable energy sources (RES). The increased adoption of renewables introduces unpredictable fluctuations in energy generation due to high dependency on the environment and the geographical area. The household consumers are also taking part in utilising RES by installing photovoltaic (PV) facilities for energy generation. The households are critical stakeholders as they have 27.2% of energy demand across Europe [14]. With the help of renewables, households can take part in increasing generation capacity alongside becoming a valuable consumer. The generation and consumption of electricity by the households show a strong influence for the overall energy systems, however, they are not currently properly modelled to tackle peak demands [15]. Through proper modelling of household consumption, the peak demands of electricity can be decreased or shifted across the day to overcome system failures. A great challenge rests on how to improve the accuracy of load forecasting [16]. The accurate load forecasting is a major driving force for many applications in the power industry such as system planning, price forecasting, demand response, fault detection, asset management, contingency analysis and load flow analysis [17].

The task of load prediction and modelling may utilise a number of heterogeneous features such as time, environment, socio-economic, appliances and their characteristics, economic and demographic information, customer behaviour as well as contextual information related to social events in a particular area. However, a big challenge lies in selecting the most influential features which affect the overall energy demand [18]. These factors can have different effects in different areas, new buildings come in, and new appliances are installed, which demand more energy and make the previous predictions less accurate. However, up-to-date appliances are energy efficient than their predecessors. The diversity and volume of datasets invoke high-dimensionality in the SG domain, which in turn triggers complex, non-linear and non-stationary statistical properties that are hard to accurately capture within statistical load forecasting methods.

Additionally, the consumption data available through smart meters at the appliance level possesses big data properties which require lots of computational resources to process in order to make prediction and profiling. As deep learning is providing state-of-the-art performance in various fields such as image processing, healthcare, finance etc., it has a great potential to transform the traditional grid into SG. A great effort has been made in the literature to use large and complex deep neural networks to improve the prediction and profiling. However, due to the deep neural networks taking long processing time, various techniques such as initial weight adjustments, model parameter tuning and weight decay have been adopted to reduce the models' training time. However, these techniques do not increase efficiency of the deep neural networks significantly. Therefore, there is a possibility of utilizing software and hardware capabilities to parallelise deep neural network training to reduce computational resources. Currently, available parallelisation techniques lack in using the full potential of hardware and software capabilities in terms of system architecture, parameter synchronisation and data communication overhead. Therefore, this research work focuses on developing a unified framework for accurate short-term load forecasting, understanding human behaviour related to their day-to-day energy consumption activities and reducing the computation cost for analysing large amounts of data.

1.3 Research Aims

The aims of this research work are as follows:

- To provide a way for electricity companies and their customers to plan for generating electricity, integrating it from other sources, distributing and utilising it more effectively and efficiently. Today, the proper planning of utilising scarce energy resources is a major challenge to fulfil the ongoing energy requirements for our day-to-day life.

- To develop a novel methodology using artificial intelligence techniques for accurate short-term electricity load forecasting for large scale distribution centres as well as for small scale individual households by reducing computational time. On the large scale, electricity load forecasting will play an important role in the decision-making process of electricity companies. On the small scale, it will provide recommendations to customers in individual houses to plan their everyday activities to utilize electricity efficiently and cut down their electricity bills. Further, a model is proposed in order to understand human energy consumption habits in households to better plan demand response programmes for efficient usage of energy consumption.

1.4 Research Objectives

The objectives of this research work are presented as follows:

- Comprehensive revision of existing short-term electricity forecasting trends and techniques to develop the insight and knowledge about the strengths and weaknesses in the area. The critical review of the currently available techniques has been made to highlight research gaps related to load forecasting with respect to statistical as well as artificial intelligence approaches, profiling and parallelisation techniques.

- Extracting features using time and frequency domain analysis in order to model various time and environment related effects and uncertainties in energy consumption which will assist in improving accuracy.

- Designing and implementing a model based on recurrent deep neural network (R-DNN) using the extracted features to improve short-term energy prediction accuracy and comparing with feed forward deep neural network (FF-DNN) to assess the performance.

- Proposing an approach using Apache Spark and Hadoop frameworks to utilise large amounts of consumption data to decrease the processing cost of deep learning algorithms using in-memory processing and lock-free synchronisation.

- Evaluation of the proposed algorithms on the bases of different statistical criteria such as the mean absolute percentage error (MAPE), mean squared error (MSE) and root mean square error (RMSE) [19], and the processing cost.

- Implementation and evaluation of the resulting models in terms of their performance and prediction accuracy using real world datasets.

1.5 Novel Contributions

The novelty of the research work is based on the following aspects:

- A new technique to select the most dominant features from a pool of features using time and frequency domain analyses which assist the proposed models in learning underlying patterns to provide the fewest prediction errors. The frequency domain analysis assists in modelling variations in the energy consumption by smoothing sudden changes in the signal.

- A proposed model based on deep learning algorithms to forecast day and week ahead electricity load in distribution centres and individual homes by exploiting time dependencies in energy consumption. The existing solutions are mostly based on statistical models which are parametric and require expert effort for parameter selection. The existing deep learning algorithms in the literature are mostly developed for particular areas and suffer from generalization capabilities [20]. Design and implementation of a clustering algorithm based on competitive learning proposed in this thesis to profile day, week and month-wide energy consumption patterns, using appliance-level data for a given household.

- A new technique using the Apache Spark and Hadoop frameworks to parallelise the deep learning processing to shorten the time while producing accurate results for large datasets.

In the literature, the researchers proposed neural network architectures based on MapReduce but they are mostly focusing on parallelising the model while trading off the accuracy [21].

1.6 Thesis Outlines

The outlines of each chapter are as follows.

Chapter 1

Chapter 1 describes the problem statement and the motivation for the research work carried out in this thesis with the overview of the business requirements of short-term load forecasting, load profiling and parallelisation. Furthermore, this chapter outlines what we want to achieve through the research aims and indicates the number of steps which have been taken to fulfil these goals through the research objectives. Finally, this chapter particularly highlights the contributions as a result of this research work.

Chapter 2

Chapter 2 reviews the existing work in the literature which highlights the recent research carried out in the energy prediction and the understanding of human behaviour. The discussion is presented about the traditional approaches as well as artificial intelligence based techniques which have been proposed in the past to improve the accuracy of the prediction and consumer modelling. Further, it identifies the research directions in short-term load forecasting, load profiling and parallelisation of the deep neural networks which lay the foundation of the proposed research in this thesis.

Chapter 3

Chapter 3 starts by presenting the discussion about the methodology and the proposed deep neural network-based algorithms for short-term electricity load forecasting. The next step in the proposed approach is the profiling of the energy consumption of the individual appliances in order to model human behaviour. The algorithm based on the concept of competitive learning is proposed for consumer profiling using granular energy usage data from individual households. The short-term load forecasting model based on the parallelization is proposed to reduce the processing time.

Chapter 4

In chapter 4, the description of the datasets has been provided which are used for the experiments and evaluation of the proposed models. The experimental results are highly dependent on the quality of the data, therefore, the energy consumption data collected from the real households has been used to test the applicability of the proposed algorithms. In particular, energy consumption datasets are described in terms of the feature set, duration, recorded time interval and the number of records.

Chapter 5

The chapter 5 provides the implementation of the proposed models. In particular, the experimentation and results of short-term load forecasting from the perspective of distribution centres and individual households are described. Further, the results and analysis of human behaviour modeling have been described using individual appliances' energy consumption. The recent developments in the energy metering technology enable the recording of huge amounts of data from individual appliances at one-second intervals. The processing of large amounts of data increases the analysis time, so parallelisation approaches have been explored to address the issue. Finally, the experimentation and its results on the implemented models are described.

Chapter 6

The evolution in the technology is paving the way for the futuristic smart grid and smart homes. The chapter 6 focuses on providing the directions for the further improvements in approaches for prediction, load profiling and parallelisation techniques. Additionally, it also discusses the limitations and challenges of the proposed work in this thesis and how they can be eliminated.

Chapter 7

Finally the research work has been concluded with the discussion of the problem statement solution and highlighting the benefits of the proposed work.

Chapter 2

Literature Review

2.1 Introduction

The prediction and understanding of end consumers' requirements for products or services is paramount and has significant importance for essentially every business. For instance, the food industry makes food items by estimating their consumption by end consumers and analysing their eating habits as the food cannot be preserved at its best level for long periods of time. Similarly, the transportation industry such as railways, airlines and various travel companies predict numbers of passengers on a day-to-day basis and offer competitive prices and special time-to-time offers to attract customers and improve their businesses. The manufacturing industry develops products based on their analysis of the market and consumer requirements in terms of their interest and geographical locations across the globe.

Likewise, the power grid which is the most complex system on earth made by humans [22], produces the energy at a large scale and delivers to 87% of the population of the world in 2019 as noted in [23]. At this large scale, unlike other industries, it is not feasible for the energy utilities to store large amounts of energy and deliver it to the consumers at a later stage. Therefore, the planning is made to synchronise the process of energy generation and distribution in order to balance its supply chain.

There are a number of challenges which have been observed in the traditional power grid. These challenges relate to the aging infrastructure which is near to expire, the steadily increasing energy demands across the world, high rates of CO2 emission due to degraded efficiency, integration of renewable resources which require intelligent systems for management due to their uncertain nature, transparent energy pricing models and security issues [24]. The advancements in the field of Information and Communication Technologies (ICT) are transforming the traditional power grid into Smart Grid (SG) [25], [26]. The SG improves the reliability, efficiency, resilience and availability of energy to consumers with the help of ICTs [27]. The dynamic nature of SG enables innovations in its operations and services with the help of intelligent control and monitoring, full duplex communication and self healing capabilities. The SG focuses on the involvement of all stakeholders of the power grid, such as governments, regulatory bodies, decision makers, distributors of electricity as well as end consumers. The consumers of electricity in the SG domain become more aware of their energy consumption through smart metering infrastructure and take part in various demand response programmes.

The major benefit of the introduction and implementation of SG in energy markets is the enhanced energy management through optimization and control of the energy supply chain [28]. The reliable operation of the SG energy systems in the presence of increasing penetration in the energy generation through renewable resources requires a timely response from end consumers [29]. As a result, the demand side management has significant importance and is recognised as a key approach in the SG domain [30] [31]. The demand side management (DSM) is a powerful tool and contains a collection of strategies and measures to transform conventional power systems to green systems. The DSM techniques enable decision makers to change the energy generation load of power systems by influencing consumer activities [31]. A major concern is raised about the reliability of the power systems during the peak periods in which most of the consumers utilise electricity at the same time. The various measures in the demand side management are carried out in order to influence the activities of end consumers to shift their usage of electricity to off-peak hours [32], [33] and reduce the extra stress on power systems.

The prediction of energy requirements of end consumers and the understanding of their electricity usage characteristics are an important part of the demand side management programmes [34], [35]. The energy load forecasting at the consumer end takes into account the local knowledge about derivatives such as weather, social events, economy of the area, consumer activities etc. This information is usually not available to the demand response planners during the development of the programmes at such a granular level. The accurate load forecasting and load profiling at the consumer end enables the efficiency and success of the demand side programmes [36]. The prediction and profiling of the energy consumption can assist in the following ways:

- Optimising the consumption of the energy and expected heat losses [37].
- Modelling of the uncertainty and variations due to the integration of renewable resources with predicted demands [36].
- Estimating the impact of power change, power cut-off, peak, off-peak and various incentive-based programmes [38].

In the literature, many models and techniques have been proposed to forecast [39], [40], [41] and model [42] electricity load for distribution centres and individual houses and improve accuracy. Most of the work is done on predicting load on the large-scale system level rather than individual households [43]. A number of techniques have been utilised to analyse large scale SG data for load predictions based on statistics, machine learning and soft computing [44]. In particular, linear regression and stochastic time series analysis are the main statistical approaches that have been broadly used [45]. The traditional models lack in properly representing complex nonlinear relationships between load and a number of other factors [46]. In order to tackle the aspect of nonlinearity, a number of efforts have been made such as multivariate dynamic regression, load forecasting based on semi-parametric additive models as well as rule based triple seasonal methods [47], [48]. In this regard, artificial intelligence based deep learning methods possess excellent ability to analyse this kind of nonlinear relationship and provide favourable performance regarding load forecasting [43]. The literature review in the sub-sections below is based on the following aspects:

- The identification of the electricity load forecasting techniques which have been developed for various intervals such as short, medium and long terms.

- The review of the approaches developed to understand human behaviour related to the energy consumption based on their daily activities which assist in the decision-making process.

- The large energy consumption datasets require huge amounts of processing power to analyse and extract useful information about the energy usage. The review is based on this purpose to evaluate parallelisation approaches.

2.2 Electricity Load Forecasting

The electricity load forecasting is a process of predicting electrical power requirements in order to fulfil demands for various time intervals in the future. The prediction is made based on the energy consumption information collected over the years in the past from traditional as well as smart meters. Accurate load forecasting has been a major challenge for energy system regulators and electricity utilities [49]. Load forecasting enables power grids and utilities to enhance energy efficiency and revenue during generation and distribution to end consumers.

2.2.1 Categories of Electricity Load Forecasting

In the energy supply chain, the process of load forecasting depends on various time intervals based on strategic objectives of a particular utility. Therefore, the prediction of electricity loads is mainly divided into four categories: very short-term load forecasting (VSTLF), short-term load forecasting (STLF), medium-term load forecasting (MTLF) and long-term load forecasting (LTLF) [50], [51] as described in Table 2.1.

Table 2.1: Load forecasting categories and their time intervals

Category	Description
VSTLF	1 up to 30 minutes ahead prediction of electricity loads [52], [43]
STLF	1 hour up to 1 week ahead prediction of electricity loads[53]
MTLF	1 month up to 1 year ahead prediction of electricity loads[54]
LTLF	1 year to several year ahead prediction of electricity loads[55]

The research work presented in this thesis particularly focuses on the prediction of electricity loads in the short term. The STLF, which spans from one hour ahead to one week, is required to make decisions about operation management, switching loads on multiple plants, budgeting and initiating new projects [56]. Although short-term load prediction has fundamental importance in certain applications such as system security, economic generation and management, and planning [57], it also allows for the performance of basic operations such as economic dispatch, fuel scheduling, unit commitment and unit maintenance more efficiently [58]. On the other hand, the duration of long-term forecasting is from 1 year to many years ahead. It helps in strategic planning of new power system installations, purchasing new generation units and developing transmission and distribution systems. However, due to the uncertain nature of the load prediction procedure, it becomes difficult to accurately predict load for such a long duration. Therefore, naturally long-term forecasts are inaccurate [59].

Most of the existing forecasting models are developed for short-term purposes as compared to the long-term forecasts. The STLF is considered as the main pillar of the SG vision as it provides the data required for load flow and contingency analysis [60]. The STLF allows adequate scheduling and operation of power systems for short time periods, therefore most utilities require forecasting short-term loads. On the other hand, long-term forecasting is mostly done by electricity regulating agencies. Due to the short time periods, the STLF is done a lot more times than long-term forecasting and its accurate analysis has made major contribution to the financial success of energy suppliers in retail electricity markets. This makes the STLF much more important than long-term load forecasting and requires special considerations in developing a highly accurate model. The following sub-sections discuss in detail various existing approaches in the literature based on traditional and artificial intelligence-based techniques for the STLF.

2.2.2 Statistical Approaches

The regression methods are widely used for load forecasting because of their simplicity with respect to formulation [61]. They are mainly used to find linear relationships between load demand and other factors (e.g. weather, humidity and socio-economic) influencing it. Validation techniques such as data-splitting, cross-validation and bootstrapping are used to find parameters used in this type of method. The accurate load forecasting depends on properly considering the latest daily load change, seasonal load change and annual load growth [62]. The transformation function within such methods consists of translation and reflection methods that allow us to deal with the aforementioned parameters [62]. As the data is generated at a large scale in SGs, it essentially contains several useful statistical components such as trend, cyclic and irregular components. The auto-regression (i.e. AR) or dynamic regression models are used to analyse such components since data might not fit well with traditional regression models. Such models aim to auto correlate errors while estimating load forecasts [63]. Throughout the years, AR models have been improved in order to consider the moving average and differentiated order of data time series. Hence, Box and Jenkins [64] proposed an auto-regressive integrated moving average model (ARIMA), which predicts equally spaced univariate stochastic time series data, intervention data and transfer function data. It requires that the mean, variance and covariance should be constant across the time. The ARIMA model is further improved by taking other time series such as temperature and humidity as input variables. It can handle more complex relationships and provide better results than AR models [65].

In Cui et. al. [66], the authors stated that accuracy of short-term load prediction is greatly affected by temperature effects in the summer season. External factors can change structures in the data. Therefore, the authors proposed an improved ARIMAX model to deal with this changing behaviour of the structures. Their proposed model achieved a mean absolute error (MAE) of 0.0037. The results of the proposed model are compared with other models such as a sigmoid function based artificial neural network (ANN) as well as an auto-regression model and auto-regression moving average model (ARMA). The proposed ARIMAX model was able to get higher accuracy than all other models in the comparison.

In the research work presented by Amaral et. al. [67], the authors predicted load by generalizing a logistic smooth transition auto-regressive model (STAR). They generalized the STAR model by adding a periodic component and called the extended model smooth transition periodic auto-regression (STPAR). The results obtained from STPAR were compared with a simple auto-regressive process (AR), naive benchmark (NAIVE) and feed forward back propagation based ANN. STPAR provides better results than all the other models with MAPE of 3.42.

In the research work presented in [68], the authors focused on forecasting short-term and medium-term electricity demands on a distribution network in France. A dataset was collected by the ERDF (The French manager of its public electricity distribution network) every 10 minutes from 2,260 substations. The load forecasting was required to manage the distribution grid, quantify constraints of the network and optimise the configuration of the network for each substation. The proposed methodology utilises semi parametric additive models for load forecasting. It has the capacity to automatically adapt variations in datasets, analyse complex relationships in the data, require less human intervention during its estimation process and be computationally cost effective. The suggested model addresses the computational issues by realizing a good trade-off between the fit and complexity of the model. This property is achieved by minimizing generalized cross validation criteria [69]. Their proposed model is a single equation which takes certain parameters such as day type (week days), special tariff, theta (smooth temperature), offset (indicating holidays and daylight saving) and toy (time of the year for the observation). The day is divided into 144 instants and for each instant a separate model is used. The authors also tried one model for all 144 instants to capture the time structure of the data and the correlation between the instants. However better results were obtained by using one model per instant in terms of the goodness of fit and computation time. Log transformation was also applied but it did not make any improvement, so instead raw values were utilised.

2.2.3 Artificial Neural Network based Approaches

There are several research studies based on artificial intelligence for load forecasting. The artificial intelligence-based techniques are better in terms of handling mutation behaviours in data structures than regression based methods. In [70], the authors evaluated artificial neural networks, linear and log linear regression methods for annual load forecasting based on the three statistical evaluations: RMSE, MAE and MAPE. They considered two mutated variables, real GDP and population. The real GDP takes into account the inflation or deflation which is necessary for real load forecast. As compared to other models, the ANN model is able to get higher accuracy. The authors in [71] proposed an ensemble STLF model based on an extreme learning machine (ELM). The accuracy of forecasting is improved by combining ELM with

Levenberg-Marquardt in their method. The output in their method is combined based on weight assignments to individual forecasts.

2.2.3.1 Discriminative Models for Load Forecasting

The authors in [72] compared and analysed two most important and widely used techniques, ANN and adaptive neuro-based fuzzy inference system (ANFIS), for load forecasting. Environment conditions were the same for both techniques. According to the authors, the performance of these two techniques was not compared before under the same operational conditions. These methods cannot be ranked by considering only one aspect, as there are many factors on which prediction accuracy depends. These include the number of inputs, configuration of model parameters and software versions. The authors adopted a two-step approach. Firstly, they compared results by considering multiple parameters and then selected most appropriate parameters to improve the performance. Predicting time varying load composition has higher potential for making decisions for demand side management and system planning. The authors evaluated both models based on two metrics, MAPE and processing time. They selected a two-layer ANN for the prediction. The reason behind this is that it can represent almost all input and output relationships. The only requirement is to choose an appropriate number of neurons for the hidden layer in the ANN model.

The ANN has two types, feed forward ANN (FFANN) and cascade forward ANN (CFANN). Either of them can be used for load forecasting. The authors proposed a novel approach to optimising the number of neurons in the hidden layer to avoid errors and showing hidden relationships. ANFIS is a Sugeno fuzzy inference system which consists of two input functions, two membership functions and one output. ANFIS predicts rules by learning from inputs and targets. The data used in this analysis was collected every 30 mins, which totals 48 samples per day. First FFANN and CFANN were compared on the bases of MAPE performance. FFANN provides better accuracy when the number of neurons in the hidden layer are adjusted to 4 and the Bayesian Regularization (trainbr) algorithm is used for training. On the other hand, the same accuracy is achieved by CFANN using the same parameters. However, using 5 hidden layers and Levenberg-Marquardt (trainlm) as a training algorithm, CFANN achieves the results faster. For ANFIS, three input and output member functions, Triangular-shaped (trimf), Generalized bell-shaped (gbellmf) and the difference between two sigmoidal membership functions (dsigmf), were compared. Trainbr, trainlm, trimf, gbellmf and dsigmf are the training functions which update weight and bias during the learning process of the neural network [73]. Out of the five, trimf has less MAPE and processing time than the others.

In the research work by [61], the authors proposed a neural network-based approach to select the best prediction method for the electricity consumption of each customer by just utilising a small subset of the customers' consumption data. Smart meters assisted in the process of demand responses with the help of certain techniques such as voluntary curtailment, direct control and pricing incentives. The power consumption patterns were predicted to facilitate the demand responses but it is often very challenging to select best predictions because of the huge amount of data produced by the smart meters. Their proposed framework provides a solution for this challenge. Two historical averaging methods of power consumption prediction are also employed to assist in continuously updating results using a sliding window technique which takes into account the data variability. The authors claim that their approach does not require retraining and provides the best results for sustainable demand responses. The proposed network is a feed forward back propagation network. The authors focused on capturing the variations in consumer consumption which was achieved by taking daily consumption standard deviation values for the past 24 hours as an input vector. During the training phase, the neural network learns by the comparison of an estimated best method to the known best method. The error is fed back to find the best match to the correct output. The authors compared their results with the oracle method and found that the proposed method is three times faster than the oracle method.

The authors in [74] proposed discriminative models based on deep neural networks to predict load for short-term duration to ensure adequate capacity and better estimate of raw material for energy generation. The authors utilize deep neural networks with and without pre-training using stacked autoencoders and recurrent neural networks along with long short term memory for load prediction. In their research work, the evaluation of these deep learning models is based on MAPE, MPE and the time required to train and evaluate the models. The authors also use baseline models for comparison which include the weighted moving average, multiple linear regression, regression tree, support vector regression, and multilayer perception. In the baseline models, the authors show that the regression tree performs better. However, the prediction accuracy is still less as compared to deep learning models which shows that the problem is non-linear and deep learning models learn non-linear patterns more efficiently. Further, the deep neural network with pre-training and stacked auto-encoders provides the lowest MAPE error of 1.84%.

2.2.3.2 Generative Models for Load Forecasting

In research work [75], the authors designed and implemented a deep learning based discriminative model by considering the training algorithms and the structure of the network including the number of hidden layers and the neurons in each layer as well as the type of the data used. The training algorithms used are bayesian regularization, scaled conjugate gradient, and levenberg-marquardt. The authors state that the architecture of the network has considerable effect on the model training and prediction accuracy. The MAPE error increases for bayesian regularization and levenberg-marquardt as the number of hidden layers increases as opposed to the scaled conjugate gradient. However, bayesian regularization provides higher prediction accuracy than the other two training algorithms when there are fewer number of hidden layers. This relates to the fact that large and complex deep learning models often suffer from overfiting which can be tackled by using large datasets.

The research work presented in [76] proposed a generative model based on a Bayesian neural network for residential energy load forecasting. The traditional Bayesian neural network requires long running time in order to produce the prediction results. Therefore, the authors presented an improved Bayesian neural network model for the prediction. In this paper, the authors observed the effect of the number of hidden layers on the prediction accuracy and errors. The authors found the similar results to [75] where the large and complex deep learning models do not considerably improve the prediction accuracy and produce large errors.

2.2.4 Hybrid Approaches

The hybrid approaches combine the capabilities of multiple algorithms in order to improve the accuracy of their prediction. The individual algorithms applied to load prediction and consumption profiling tasks often provide the best performance in certain aspects but lack in others. Therefore, the integration of more than one algorithm assists in overcoming these limitations. For instance, convolutional neural networks (CNN) are good at extracting hidden features from the data while recurrent neural networks can model the relationships between time steps. The authors in [77] proposed a hybrid model based on CNN and a long short term memory network (LSTM) to predict electricity loads and minimize the forecasting error. Their proposed model basically consists of three parts: a CNN module, a LSTM module and a feature fusion module. They applied forget gates and memory cells to learn long time information and the CNN algorithm to extract and model patterns which appear in multiple regions. Finally, the feature fusion module integrates the output of LSTM and CNN and makes predictions. The proposed model was applied in a real world case where data was collected for three years at a sample rate of one hour. The authors compare their proposed model with random forest (RF), decision tree (DT) and deep energy (DE) algorithms. The proposed model was also compared with individual algorithms CNN and LSTM. The authors demonstrated that their proposed model improves prediction accuracy in terms of improved MAPE, MAE and RMSE errors. In particular, their model improves prediction by 9% as compared to the DE algorithm, 14% to LSTM and 12% to CNN.

The authors in [78] proposed a new approach for the load prediction to include seasonality effect. The authors assessed the applicability of their proposed model based on four case studies in four different seasons (i.e. spring, summer, autumn and winter). They argued that the diverse environmental conditions such as temperature, cloud cover and wind speed in various seasons have a considerable effect on the accuracy of STLF. Therefore, the authors proposed a specific similarity concept with the load prediction process in order to incorporate environmental effects. The proposed hybrid approach is based on two algorithms, firefly and support vector machine (SVM). The firefly algorithm is responsible for assessing and selecting parameters for the SVM algorithm. The authors argued that the existing approaches lack in properly modelling the effect of seasons in the load forecasting and therefore result in a high number of prediction errors. However, by incorporating the seasonality effect using the season specific similarity concept, their proposed approach is able to improve prediction more than five times as compared to the existing methods.

The authors in [79] proposed a hybrid model based on multi-layer LSTM and CNN for STLF at temporal granularities such as minutes, hours and days. The authors proposed that instead of using univariate or multivariate sequences in LSTM and CNN, a dataset is preprocessed to pair power consumption values with the contextual information which provides bivariate sequences. The advantage is that the bivariate sequences properly illustrate critical contextual information during neural network algorithm training. The authors conducted experiments using datasets with and without considering holidays as these have diverse effects on energy consumption in households as well as offices. The comparison of the proposed approach was conducted with ARIMA and Sequence to Sequence (S2S) LSTM based on MAPE and relative root mean squared error (RRMSE) measures. The authors show that the proposed approach produces lower MAPE errors 70% and 45% as compared to ARIMA model and S2S LSTM respectively.

The authors in [80] introduced a hybrid model based on multi-resolution wavelet analysis and least squares support vector regression (LSSVR). The proposed model is optimised using a particle swarm optimisation algorithm. The authors proposed that original consumption sequences are decomposed using wavelet analysis during the preprocessing phase and used as input to the least squares support vector regression. The accuracy of the prediction is highly dependent on the parameters of the least squares support vector regression algorithm. Therefore, the particle swarm optimization (PSO) is used to select values for the kernel parameter and regularization. The proposed hybrid approach was compared with the radial basis function (RBF), elman network and LSSVR with only the PSO algorithm using MAPE, RMSE and MAE error measures. The dataset used in this research has the frequency of half an hour over the period of three months and was collected from New South Wales in Australia. Furthermore, the authors grouped the data based on the days of the week as the consumers' living and working habits correlate with each other. The authors indicate that the proposed hybrid approach improves prediction accuracy with the MAPE error of 0.907 and performs better as compared to other traditional methods such as RBF, elman network and LSSVR with the use of only the PSO algorithm.

The following subsection describes two deep learning algorithms which are utilized in this research work for short-term load forecasting.

Feed-forward Deep Neural Network (FF-DNN)

A neuron is a basic unit of the FF-DNN model, structurally inspired by the human neuron. In the FF-DNN model, an input x is combined with a weight w and a bias b at the neuron using the following equation [6]:

$$\phi = \sum_{n=0}^{N-1} w_n x_n + b \tag{2.1}$$

A nonlinear activation function is applied on computed values, which produces an output and sends it as input to other connected neurons. The basic goal of FF-DNN is to approximate the function $\sigma(\phi)$ which adds non-linearity to the neural network.

Multi-layer Feed-forward Neural Networks are made up of many such neurons interconnected in different layers. The output of the model is fully decided by weights and biases. The network learns by adjusting weights to minimize a loss function with a large set of training data. The loss function is given by:

$$Loss(q|W, B) \tag{2.2}$$

Where q represents each training example in the data, W is the matrix of weights and B is the set of biases.

Recurrent Deep Neural Network (R-DNN)

The multi-layer perceptron is the simplest form of R-DNN in which the output of one hidden layer along with the input is fed back in the network. The following non-linear equations delineate the functionality of the simple structure of R-DNN [81]:

$$Q(t) = f_1(X(t) \times W_i + Q(t-1) \times W_{td})$$
$$Y(t) = f_2(Q(t) \times W_i) \tag{2.3}$$

Where $X(t)$ is input at time t, $Q(t)$ is output at time t, $Q(t-1)$ is the time delay input at time $t-1$, W_i is a weight for the input layer, W_{td} represents a weight for time delay input, f_1 and f_2 are hidden and output layer transfer functions respectively. The time delay unit is required to hold the output and feed back at the next time step.

2.3 Energy Consumption Profiling

Energy consumption profiling in the households is a process of grouping or clustering patterns which show similar characteristics. These individual patterns of energy consumption are represented through load profiles which provide information about the shape of the load, average load, peak and time duration of the peak load [82]. The SG architecture involves an intelligent HEMS which enables the SG to control and monitor the energy consumption of individual appliances at the household level. The HEMSs are the main building blocks at the consumer side in the SG ecosystem. Therefore, effective HEMSs are essential in order to: i) model highly non-stationary and non-linear measurements from individual appliances and ii) conduct correlations of such measurements with diverse inputs (e.g. environmental factors) in order to enhance consumers' understanding of their consumption improve the overall demand-response programmes for energy utilities.

The emergence of the Internet of Things (IoT) based on the acceptance of smart-meters in households in most developed and developing countries has widened the functional requirements of HEMSs [83]. The appliances used in the households for various purposes related to heating and cooling, cooking, entertainment and laundry are now regarded as IoT devices that can assist in energy monitoring, profiling, and controlling through HEMSs. Apart from the directly observed energy consumption measurements, environmental measurements (e.g. humidity, temperature) are also used in order to enrich the view of customer-specific behavioural characteristics and empower the “smart” factor in such systems [83].

It has been stated that the human activities to consume energy vary significantly in households from individual to individual based on their daily preferences. Due to the unpredictable nature of human activities a big challenge behind the design of “smart” properties for HEMSs relates to the fact that household electricity consumption possesses highly non-stationary and non-linear properties. In addition, various household devices and sensors for collecting energy consumption information, usually operate on diverse sampling rates and pragmatic scenarios, which affect the data quality and tend to produce noisy and incomplete measurements [84].

The non-stationary and non-linear properties of the energy consumption of households arise due to the uncertainty of consumers’ day-to-day activities. Traditional profiling algorithms lack the ability to capture these non-stationary and non-linear properties, which affects the decision making process during energy generation and demand response programmes. Currently, utility companies have been focusing on utilising SG technologies to facilitate energy supply in order to fulfil peak energy requirements and reduce cost. In addition, the peak demands of energy consumption in domestic households, as well as in the commercial sector, have contributed to the problem of climate change, as high energy requirements leave a negative impact on the economy and environment of the region. In order to reduce these peak demands, a fundamental requirement is to reduce peak energy consumption by enabling consumers to shift flexible energy consumption to off-peak hours. This can be achieved by understanding human behaviour through profiling individual appliances in the household, without affecting the consumers’ comfort level.

The home energy management systems (HEMSs) rely on profiling techniques and a number of them have been proposed in the past. The clustering algorithms are usually categorised into partition based, hierarchical based and density based clustering algorithms. For instance, the authors in [85] utilise energy consumption, socio-economic and environmental data to analyse the relationships between consumer behaviour and consumption patterns, using spectral clustering under statistical Hidden Markov Modeling (HMM). The authors use Gap Statistics to infer the number of clusters from data and K-medoids as the last clustering step. The limitation of their work is that Gap Statistics increases the processing time and does not scale well for large datasets.

2.3.1 Load Profiling based on Statistical Methods

The authors in [86] proposed a sample Pearson correlation coefficient (SPCC) distance method with a density based mean-shift (MS) algorithm, based on non-parametric mode-seeking clustering for energy consumption load profiling. The MS based clustering is based on one parameter, kernel bandwidth. The drawback of their method is that it requires domain knowledge to select a kernel bandwidth value. Additionally, the MS algorithm is slow in processing and increases the inference time. The work presented in [87] proposed a model to manage energy consumption based on prioritizing home appliances by focusing on appliance usage habits, which is based strictly on per-appliance energy consumption.

The study in [88] correlates appliance-level consumption and daily consumer activities to model the association between the use of appliances at different time intervals by employing a density-based clustering algorithm (DBSCAN), which we have used for the comparison in Section 5.4. Although DBSCAN automatically extracts the number of clusters, domain knowledge is required to optimize the values of DBSCAN's parameters, i.e. neighbouring points and the minimum number of elements in a cluster. DBSCAN is also computationally expensive as shown in our comparison.

The authors in [89] employed a frequent sequential mining algorithm, i.e. window sliding with de-duplication, in order to model hidden patterns and preferences of inhabitants to autonomously achieve energy conservation. This approach has the drawback that it extracts various hidden patterns in terms of rules that often have less usability and comprehensiveness. Conversely, in our proposed method, the hidden features are automatically extracted using deep representational learning instead of requiring deep expert knowledge.

2.3.2 Load Profiling based on Machine Learning Methods

In addition, the work in [90] performs appliance-level energy consumption profiling by utilising meta-features of the Wigner-Ville time-frequency distribution over a simple partition based K-means clustering scheme. The authors selected a number of clusters by applying K-means iteratively which increases the computational time and is not feasible when applied to large datasets.

Understanding the energy consumption of households is crucial in order to develop successful demand response programmes in a SG domain. However, traditional demand response programmes lack consideration and understanding of human behaviour [91]. Furthermore, demand response programmes are often developed by price-based or incentive-based strategies. Efficient demand response programs play an important role in energy efficiency and the low-carbon economy, which are assisted by the development of HEMSs related technologies. Additionally, detailed information about energy consumption is highly desirable to enable consumers to embrace energy efficient behaviours during peak demands [91].

The authors in [92] proposed a model based on a partition based K-means algorithm to find daily load profiles for residential consumers. The authors studied the effect of correlation among the consumption profiles. The authors argued that high correlation among the energy consumption profiles requires a fewer number of clusters, while a large number of clusters are required to explain the similarity of load profiles for lowly correlated energy consumption.

It is evident from the literature that the existing clustering approaches suffer from two main weaknesses. Firstly, they are unable to extract the number of clusters from the data without increasing processing time due to recursive methods used for various numbers of clusters and the compactness of the obtained clusters as described by the dunn index (DI), silhouette index (SI) and Calsinki and Harabasz index (CI). Secondly, the majority of the reported methods rely on traditional clustering algorithms, such as partition or density-based clustering schemes, which are fundamentally not efficient in capturing the uncertainty contained within explicit consumer behaviour. Naturally, the uncertainty invoked by customer-specific habits has a direct impact on appliance-level energy consumption profiling and it unavoidably affects the decision-making process for several control mechanisms over a variety of HEMS applications. The proposed algorithm in this thesis exploits the uncertainty in consumer behavior with the help of three uniquely defined associations to assist in the decision-making process in HEMS applications.

2.4 Parallelisation Approaches

The current energy market is undergoing a wave of transformation of current electricity grids into SGs. The goal of this transformation is to lower the dependency of energy generation from fossil fuels and increase the usage of renewable resources. Additionally, SGs also aim to ensure the efficiency and reliability of these energy systems. The SGs along with smart meters facilitate energy suppliers and consumers to get the valuable information of their consumption. In this regard, the European Union (EU) has aimed by 2020 to rollout smart meters and replace 80% of current electricity meters [93]. This vision is also being adopted and implemented in other countries across the world such as the United States. The upcoming high surge of smart metering technologies will result in an increase in the volume of energy consumption data. Currently, the smart meters are capable of collecting energy consumption data at the rate of one second intervals. The number of electrical appliances in households are also increasing with the advancement of technology. Therefore, huge amounts of data will be collected from these appliances which will possess big data properties. The challenges in analysing big data in the energy industry has attracted the interest of energy research scholars.

The distribution and parallelisation of data and hardware are required to train complex models using huge amounts of data for computation time efficiency. There are various forms of parallelisation for deep learning models such as model parallelism, data parallelism, pipeline parallelism and hybrid parallelism. These forms of parallelisations are discussed below.

2.4.1 Forms of Parallelisation

2.4.1.1 Model Parallelisation

In this form of parallelisation [94], the distributed architecture loads a specific part of the complex deep learning model. The workers such as Nodes or GPUs in the distributed architecture are responsible for different tasks and layers in the deep learning model. In the forward pass during the deep learning model training, the input is passed to the first component which calculates the output and feeds to the next workers for further processing. In the backward pass, the deep learning model computes the gradients and propagates back through the workers in the distributed architecture.

The major strength of this form of parallelisation is that it does not require parameters synchronization for the deep learning model. The parameters are initialized at the start of the training process. The individual operations are performed by various workers in the distributed architecture in a parallelised fashion. The data is transferred between these operations.

The major challenge in this form of parallelisation is related to the reduction of heavy data communication between different workers of the distributed architecture. The operations can lag due to the large delays in communication which will in turn increase the computation time of the deep learning algorithms. Furthermore, the operations in deep learning models are dependent on each other. Therefore, it is challenging to effectively split these models.

The research work presented in [95] proposed a model based on Cannon algorithm for matrix multiplication to parallelise nerual network training. The network is trained using a technique called network parallel training. In this technique, the nodes in each layer of the network are divided on different processes. Each process is responsible to calculate an activation function individually in parallel using the Cannon algorithm. The Cannon algorithm utilizes a defined set of rules and shifts rows and columns of the matrix across various processes. The weights are communicated after calculations to other nodes. The authors showed that model parallelisation through the Cannon algorithm improves the speedup.

2.4.1.2 Data Parallelisation

In data parallelisation, each worker is responsible for running a replica of the deep learning model. However, the parallelisation is achieved through splitting the data into non-overlapping parts and training individual replicas of the model. The output of the individual workers is used to update the parameters of the model. This allows us to reduce computation time through training speedup.

The major strength of this form of parallelisation is that it can be used to parallelise any deep learning model with fewer parameters. The data parallelisation provides optimum performance for the models that require high computation. However, the weakness of this form of parallelisation is the synchronization of model parameters across the whole distributed architecture [96]. The deep learning algorithm computes gradients locally and updates the global parameters stored on the server. Additionally, the performance of the data parallelism is affected for extremely large models which exceed the size of a single node in the distributed environment [97]. In this technique, the only assumption is that the data should be independent and identically distributed.

The authors in [98] proposed a computing framework based on big data analytics tools Spark and Hadoop for load forecasting. The authors first assessed and evaluated the computational time required to build a forecasting model in a distributed environment and then determined the threshold of the amount of data in order to provide considerable performance improvements using the distributed framework. To achieve this goal, the authors proposed their framework based on three algorithms: multiple linear regression (MLR), a least absolute shrinkage and selection operator (LASSO) and RF. The dataset used in the experiments consists of 10 years' energy consumption collected at the rate of one second and utilized as input to the three algorithms without the contextual information such as temperature, humidity, day type or consumer occupancy level. The authors evaluated their results using a standalone computer and a computing cluster respectively. They showed that Spark provides better performance on the computing cluster when the dataset is larger than 6 months. The conclusion is drawn from the experiments that the proposed distributed framework based on Spark and Hadoop improves the load forecasting performance when the data is large and the forecasting model is complex.

The research work in [99] proposes an Apache Spark based STLF platform for urban load analysis using a Dynamic Bayesian Network model and GraghX. The GraphX is an API provided by Apache Spark to accelerate graph and graph-parallel computations to improve the computational performance as compared with Graphlab and Graph. The experiments on the proposed platform were conducted using a dataset which contains electricity consumption along with the contextual information such as the time of the consumption and the location in terms of coordinates. The platform was compared with Bayesian Network and Markov Chain models. The authors illustrated that the proposed distributed load forecasting platform provides low deviation errors and is more fitting with real consumption values. The authors also stated that the forecasting accuracy can be further improved by using more variables and minimizing random errors from signal distribution.

The authors in [100] proposed an approach to parallelise a backpropagation neural network which is based on the MapReduce model using distributed computing for large-scale datasets. The authors argued that though there are existing techniques but it is difficult to trade-off between efficiency and precision. The authors designed a cascading model to improve the performance of the parallelisation method in terms of the precision. The authors split the data and train multiple models. In training, they used ensemble techniques such as bootstrapping and majority voting. The authors argued that the proposed algorithm provides high accuracy when data is large, however, the issue arises due to the overhead encountered by MapReduce processing. This can be tackled by utilizing the capabilities of Spark in-memory processing techniques.

2.4.1.3 Pipeline Parallelisation

The pipeline parallelisation technique is based on the concept of both model and data parallelisation. In this technique, both the model and dataset are split. Each worker thread runs part of the model on part of the data and computes output. The output is passed to the other workers in the sequence in both forward and backward passes. The major strength of this technique is that the performance can be improved without overloading the individual workers as they do not need to hold all the data and work only on part of the data using part of the model.

2.4.1.4 Hybrid Parallelisation

The hybrid parallelisation allows us to use a combination of all the architectures, i.e. model, data and pipeline parallelisations. This is ideal because the deep learning models are complex in terms of hidden layers and hyper parameters and require different forms of parallelisation.

2.4.2 System Architecture

The synchronization of all the worker threads in a network is referred to as system architecture. The poor synchronization of worker threads leads to stalls and delays in the worker output. The common system architectures include centralized, decentralized and federated learning architectures which are explained in following sections.

2.4.2.1 Centralized Architecture

In this architectural design, the worker threads update parameters of the model at a parameter server. The server is centralized which holds the global parameters of the model. This architectural design is most common among different parallelisation architectures.

2.4.2.2 Decentralized Architecture

This form of architecture does not require a parameter server. The parameters of the model are updated individually by worker threads. As a result, every worker thread communicates parameters to all other workers in the network. Therefore, the parallelisation performance mainly depends on the parameter exchange between the workers. The optimization of the communication can be achieved using a topology such as ring topology.

2.4.2.3 Federated Learning

The centralized and decentralized approaches of parallelisation require that the data is balanced and distributed independently and identically to the worker threads in one machine or in a data centre. However, in federated learning, the data does not need to be uploaded to servers for processing. The federated learning architecture is suitable for the scenario where distributed devices access cloud data centres for required services. Traditionally, data is transferred from mobile devices to a cloud, complex deep learning models are trained in the cloud, and necessary services are then provided to the mobile devices. In federated learning, the data resides on mobile devices and a model obtained from a data centre is trained locally in the mobile devices. The output of the model is summarized and sent to the cloud to improve the model globally.

The strength of federated learning is that the privacy of the data is preserved as it is not transferred to the cloud. Additionally, there is less communication between the distributed devices and cloud centers. The parallelisation based on federated learning provides better computational performance than the centralised architecture due to improved communication between worker threads.

2.4.3 Synchronization in Parallelised Architecture

The synchronization of the model parameters during training in parallelised architecture across all the worker threads is a major concern to achieve state-of-the-art performance. The synchronization approaches proposed in the literature are described below.

2.4.3.1 Synchronous Approach

In this approach, the worker threads update their parameters after every iteration during training of the deep learning model [101], [102]. The advantage of this approach is the transparency between parameter updates. However, the problem of staleness can occur if some of the worker threads are slower than others.

2.4.3.2 Asynchronous

In this approach, all the worker threads in a distributed environment update deep learning model parameters independently from each other [103].

2.4.3.3 Bounded Asynchronous

In the bounded asynchronous approach, the worker threads update the model parameters freely. However, the delay between the updates is bounded.

2.4.3.4 Speculative Synchronization

An asynchronous approach implies that the worker threads do not wait for the latest model parameters and proceed to the next steps during the model training to avoid stallness. However, this compromises the model training quality. The authors in [39] proposed an approach named speculative synchronization which speculates about new parameter updates. The proposed approach favours updating the parameters if required based on the speculation from other worker threads.

In the literature, the researchers have proposed solutions for big data analytics based on cloud computing [104]. Furthermore, the analysis of big data is not the only challenge which is faced by the energy industry, and the volatility in the process of electricity generation is also a major concern for the utilities. This volatility in the process of energy generation is caused by renewable energy resources [105] such as sunlight, wind, biomass, rain, tides and waves. Therefore, it becomes challenging for utilities to balance the demand and supply of electricity to their consumers. The parallelisation of load forecasting algorithms in order to reduce computational cost can provide solutions to analyse huge amounts of the data, i.e. big data.

2.5 Summary

All the research work presented above, which is based on statistics and artificial intelligence techniques, is effective and provides higher accuracy for STLF in certain conditions. However, the problem with these models is that they are not generalized enough to consider all the factors from various domains such as environmental, socio-economic, customer behaviour and scheduling. The existing load forecasting techniques are not ideal for practical operations to consider, due to the limitations of measurement and monitoring capabilities [66]. The conventional time series methods lack in considering external factors and mostly focus on temporal solutions that strictly deal with the time factor. For example, these time series methods require a large number of observations and assume linearity relations in the data [106]. Furthermore, although the progress is being made to understand human behaviour using profiling techniques to improve building sustainability, it still lacks in fulfilling the requirements of increasing demand for energy consumption in the household sector [107]. This chapter focuses on providing an extensive review of the existing techniques and approaches in the literature for electricity load forecasting, human behaviour modelling and parallelisation techniques. In the following chapter, we describe our research methodology for short-term load forecasting, energy consumption profiling to understand human behaviour and parallalizing forecasting in order to rectify the aforementioned weaknesses of the existing work.

Chapter 3

Research Methodology

3.1 Introduction

The STLF and human behaviour understanding greatly influence the process of energy systems' capacity planning, scheduling and maintenance during their operations over the time. Additionally, the STLF also enables residential, commercial, industrial and municipal consumers to understand their energy consumption and plan their activities accordingly. However, on the one hand, the availability of large amounts of energy consumption information on individual appliances enhances the accuracy of STLF and load profiling algorithms while on the other hand, it increases the demand for high computational resources. This chapter presents the research methodology for predicting load for short-term, modelling human behaviour through load profiling and reducing processing time using parallelisation.

3.2 Short-term Load Forecasting

Figure 3.1 depicts the diagram of our proposed system for electricity load prediction to improve the accuracy while consuming less processing power. The system consists of task specific modules and components which are interlinked with each other. The module 2 performs critical functions of parallelisation and training of the proposed deep neural network. In module 1, data is obtained from the repository and processed for feature engineering. The repository contains

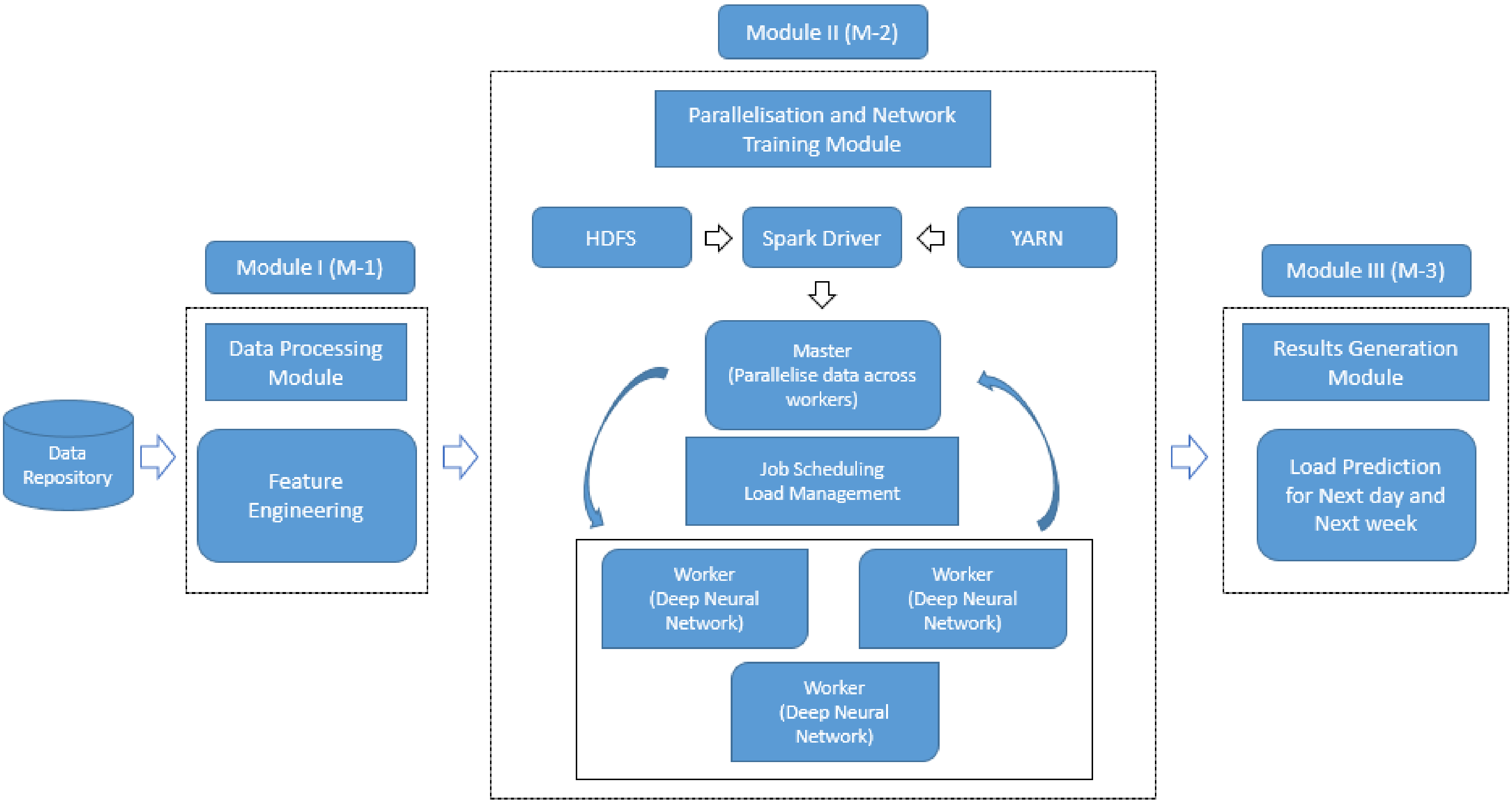

Figure 3.1: System Architecture for efficient and cost effective electricity prediction

energy consumption information from the distribution side as well as from the consumer side. The energy consumption information at the consumer side is collected from the individual appliances. In order to include granularity in the energy consumption data, the raw data is recorded at various time intervals that are 6 seconds, 1 minute, 15 minutes and 30 minutes. This provides most dominant features to module 2 as input.

The individual components in each module show the operations that take place in the whole system. Module 2 is the main module and performs parallelisation, which will allow the deep neural network to train and produce the electricity load forecasting accurately and in a short period of time. Module 2 consists of major components which include Spark Driver, Hadoop Distributed File System (HDFS), Yet Another Resource Manager (YARN), Master and Worker nodes. Module 2 also optimizes the network by tuning different parameters. Spark

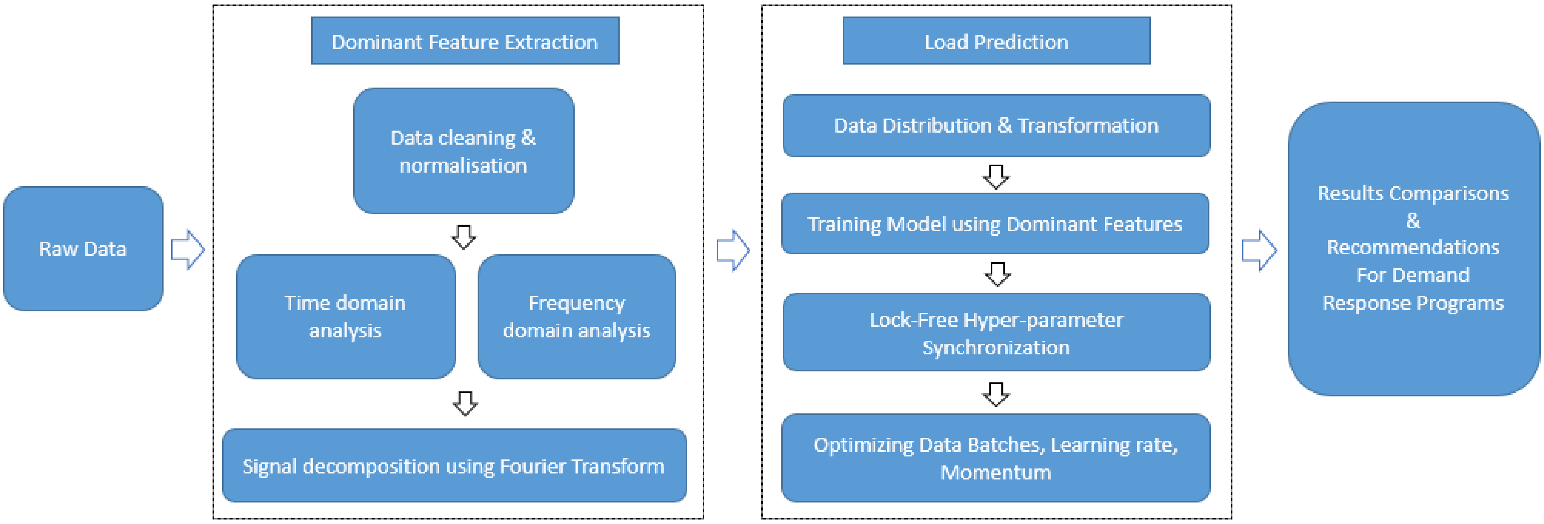

Figure 3.2: Data flow in the proposed system

Driver creates SparkContext which sends data analysis jobs to the worker nodes in the cluster. The YARN manages the resource and schedule jobs on the nodes. Data is distributed across worker nodes and a deep neural network performs processing on the part of the data. All the workers communicate with the master node for the purpose of parameter sharing. The Module 3 acquires output from Module 2 and performs load prediction.

Figure 3.2 shows a general data flow and the functions performed on the data in the proposed system. After data cleaning and normalisation, the first step in data flow is to extract features which model the variations in electricity consumption. To achieve this, the dataset is analysed in both time and frequency domains. In time domain, the temperature, time, holidays, lagged load and data distribution effects are found to be the dominant factors affecting load. In frequency domain, energy consumption time series are decomposed using Fourier Transform. In the second step, the data is distributed across workers in the cluster and transformed to make it suitable for the deep learning models. Each worker node contains a copy of the deep neural network and performs processing on part of the data. The hyper-parameters of the deep neural networks in each worker node are initialised and communicated to the master node for synchronisation. The hyper-parameters are updated using a Lock-Free method which does not affect the processing of individual workers due to the slower nodes in the network. Finally, the results are obtained from the individual worker nodes and presented in visualised forms.

3.2.1 Frequency Domain Feature Extraction

The analysis of data in the time domain provides potential features which express the changing behaviour of electricity consumption. However, the features obtained using the time domain analysis do not provide higher accuracy due to the lack of expressing complex patterns hidden in the dataset. Thus the data needs to be further analysed in the frequency domain to expose these patterns. The sudden changes in the energy consumption of the appliances introduce peaks in the data. These peaks and the sudden variations make it complex for the models to learn the patterns. The frequency domain analysis allows decomposing time domain signals into multiple sinusoids of various frequencies. The time domain signals often show randomness. However, frequency domain analysis converts random signals into different frequencies which are stable and easily predictable and considerably improve the accuracy. The higher and lower frequencies assist in identifying the relationships between dependent and independent features. The Fast Fourier analysis is performed on the electricity load values to determine dominant frequencies. The following equation 3.1 converts load values into an un-normalized univariate discrete Fourier transformed sequence [108]:

$$y_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \tag{3.1}$$

Here, x_n represents a time domain signal (n=0 to N-1), y_k depicts a signal transformed into the frequency domain (k=0 to N-1), i= (-1) and N is the length of the input signal. The figure 3.3 represents Fourier coefficients extracted from the electricity load values.

In figure 3.3 , the x-axis represents the frequency index and y-axis represents the magnitude of those frequency indices. The Fourier coefficients with higher magnitudes represent dominant frequencies. The dominant frequency components describe the higher variation in the electric-

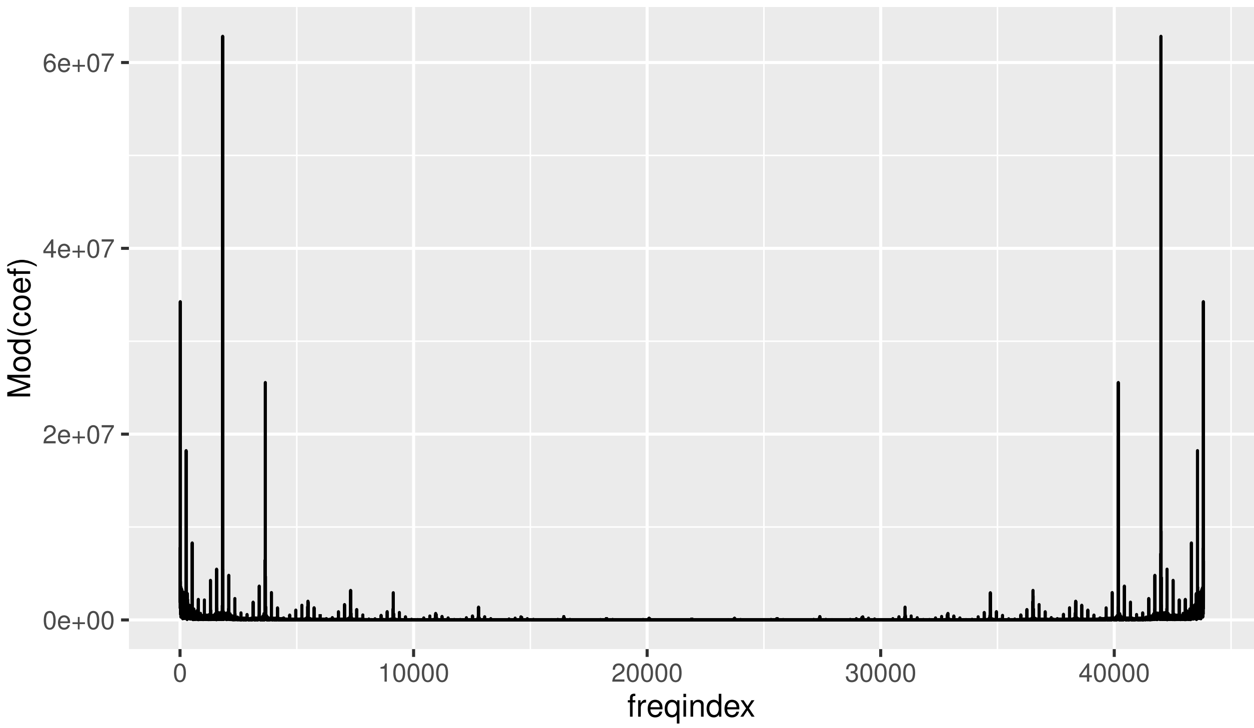

Figure 3.3: Frequency domain analysis

ity load signal. A high pass filter is designed to filter these dominant frequencies from low frequencies. The filtered signals are converted back to the time domain. The figures 3.4-3.7 highlight the filtered signals obtained from frequency domain analysis.

The signals presented in figures 3.4-3.7 are the most dominant signals in the electricity load. These signals contain higher amplitude than the other signals present in the main load signal. The signals with low magnitude are filtered out by the low pass filter because they show fewer variations in the original signals and thus have less effect on the performance of the model.

The following subsections describe a novel approach to select the hyper-parameters and train the FF-DNN and R-DNN models to improve prediction accuracy and tackle the problems such as vanishing or exploding gradients.

3.2.2 Feed-forward Deep Neural Network (FF-DNN)

The hyper-parameters of FF-DNN include activation function, number of hidden layers and neurons in each layer. We employ the Rectifier activation function (ReLU) in the model because

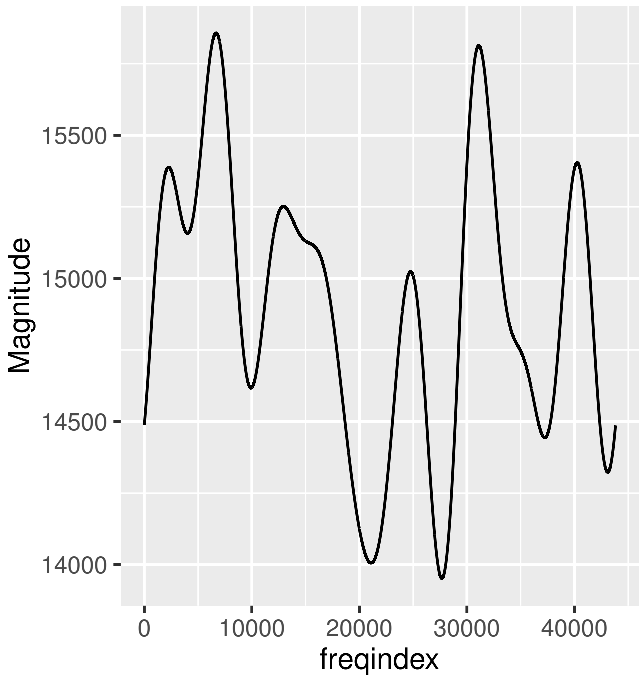

Figure 3.4: Frequency components: subsignal 1

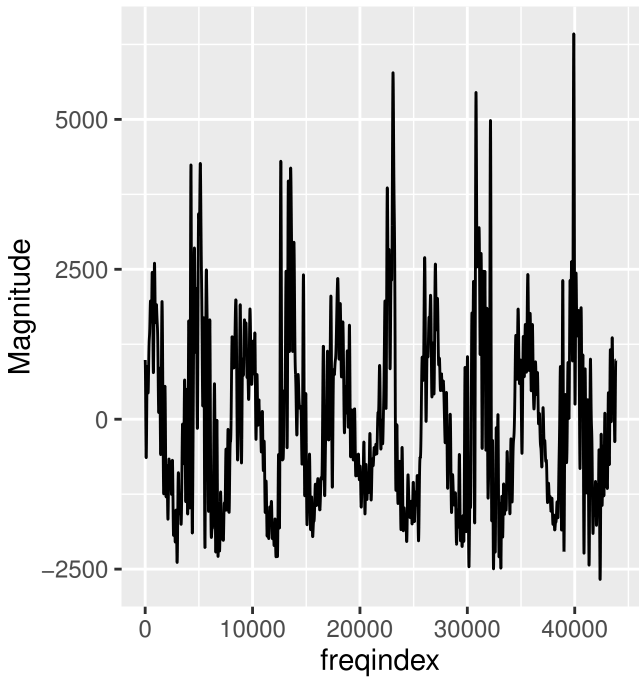

Figure 3.5: Frequency components: subsignal 2

it is biologically accurate and shows high performance in image processing. It is defined as:

$$f(x) = max(0, x) \quad \text{where} \quad f(x) \in R \tag{3.2}$$

ReLU takes less processing time as it works on un-normalized data. As compared to Sigmoid

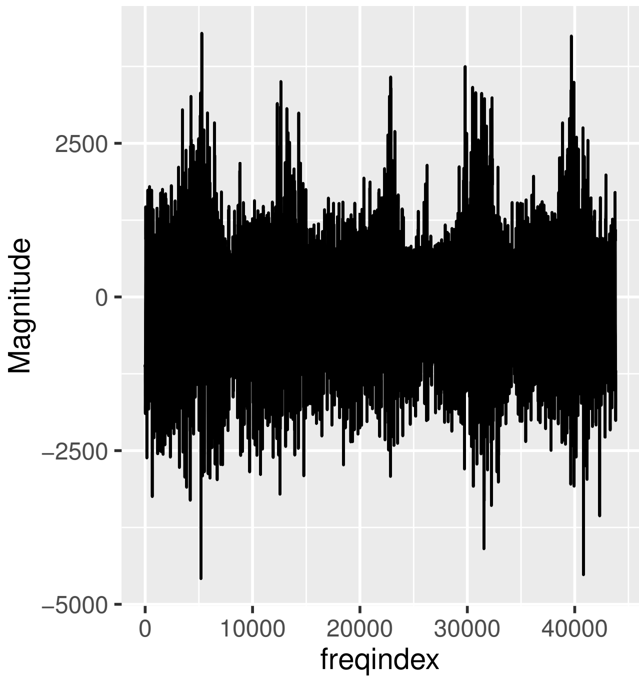

Figure 3.6: Frequency components: subsignal 3

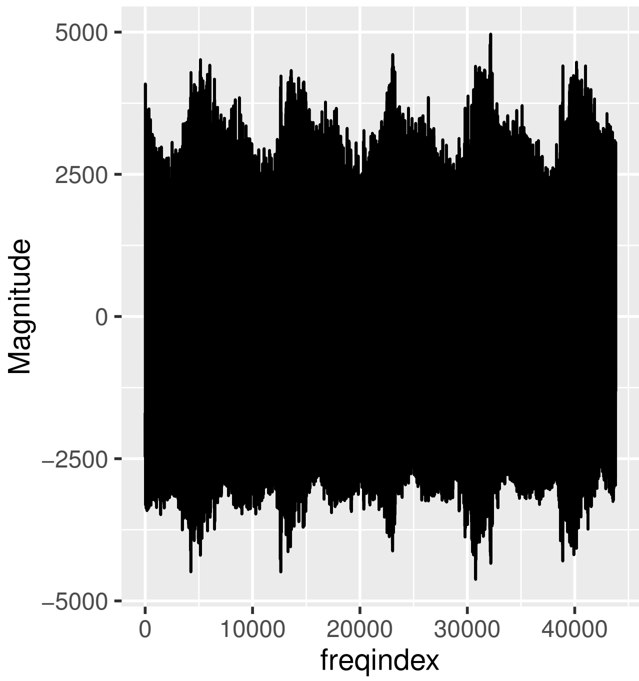

Figure 3.7: Frequency components: subsignal 4

or Tanh, exponential computation is not required in ReLU. Further, ReLU does not suffer from the vanishing gradient problem. To include the effects of all factors in the model, time and frequency domain analysis suggests the use of more features. This improves the prediction accuracy but at the cost of complexity. This makes the model complex which causes the problem of over-fitting. This problem is addressed using regularization techniques. We employ L1 (Lesso) regularization that adds an extra term in the loss function to minimize error:

$$Loss'(q|W, B) = Loss(q|W, B) + \Gamma(q|W, B) \tag{3.3}$$

Where Γ is a regularized term and it is selected using cross validation. The selection of number of hidden layers and neurons in each layer is critical as large networks become complex and difficult to train. The proposed approach is to select hidden layers and neurons using a Grid Search technique to reduce complexity of the network and improve performance.

3.2.3 Recurrent Deep Neural Network (R-DNN)

R-DNN requires the selection of hyper-parameters for load prediction. However, training of R-DNN is more complex as compared to FF-DNN because of the cyclic connections in the R-DNN. R-DNN has the capabilities to learn the context across sequences and predict sequences in the future. The R-DNN network can be developed using many different kinds of architectures. The vanilla R-DNN algorithm suffers from two major problems which are vanishing or exploding gradients. The R-DNN based on the Long Short Term Memory (LSTM) architecture overcomes the problems of vanishing or exploding gradients by an efficient, gradient based algorithm using internal states of network units. This type of R-DNN contains gates such as forget gate, input gate and internal memory cells. As the energy consumption information collected at macro and micro levels is essentially time series based, therefore, the novel approach is to exploit the time dependencies using R-DNN. The numbers of neurons in input and output layers are selected based on the number of features in the dataset and the outcome respectively. The hidden layers are selected in order to model the non-linearity in the energy consumption.

3.3 Energy Consumption Profiling

In the decision making and planning process of energy generation and distribution to the end consumers in the SG infrastructure, understanding end users' behaviour during electricity usage through consumption profiling has fundamental importance. Customer segmentation based on the load profiling is one of several factors (e.g. environment and geographical variations, granularity and quality of the recorded energy consumption) which affect the accuracy of the load prediction [109].

Therefore, in this research work, we have proposed a Deep Competitive Learning Algorithm (Deep COLA) based on the concepts of Deep Neural Networks and competitive learning in order to capture these non-stationary properties of diverse household-related measurements. Our novel contribution lies with our proposition that the impact on the energy consumption in a typical household can be explained in terms of three unique associations, i.e. appliance-to-appliance, appliance-to-time and appliance-to-environment. The automatic disclosure of these associations can assist in modelling consumer energy usage to serve the desired Demand-Response (DR) business model for utilities [110], improving health-care through collaboration with hospitals, lowering energy consumption and sustaining the environment by maintaining (or reducing) the level of CO_2 [89].

As the smart meters in the HEMS are capable of recording energy consumption information at very small intervals (i.e. 1 second), a large amount of computational resources is required to process and analyse this information. The proposed Deep COLA algorithm is computationally efficient, which extracts meaningful day, week and month-wide energy consumption profiles using a competitive learning approach. Under a synergistic approach, Deep COLA utilises appliance-level energy consumption information with environmental factors and models uncertainties and variations in the consumer behaviour by extracting appliance-to-appliance, appliance-to-time and appliance-to-environment associations in each household. Moreover, it eliminates the need to specify a-priori the number of clusters, k, which is common in the majority of schemes proposed in the past. This work particularly focuses on the following objectives:

- Developing a clustering algorithm based on competitive learning to profile day, week and month-wide energy consumption patterns, using appliance-level data for a given household. Current methods [88],[89] are based on K-means, DBSCAN and rule mining, which require expert knowledge to achieve adequate clustering. However, the proposed concept of competitive learning allows the extraction of compact and well-separated clusters, without the direct involvement of experts.

- A new approach to automatically extract the optimal number of clusters without using Elbow, Silhouette or Bootstrap methods, which are commonly used methods in the existing work.

- Profiling of appliance-level energy consumption in synergy with environmental factors in order to reveal per-household behavioural characteristics under three new associations : i) appliance-to-appliance, ii) appliance-to-time and iii) appliance-to-environment. This is opposed to existing approaches that only profile consumption based on time.

- Comparison and evaluation of Deep COLA against three commonly used clustering approaches in terms of computational cost and clustering accuracy over two real datasets.

3.3.1 Deep COLA Description & Formulation

The novel Deep COLA algorithm is based on the concept of competitive learning [111] and operates by applying dimensionality reduction on energy consumption datasets that contain a large number of features. In brief, in the paradigm of deep neural networks, the competitive learning enables nodes in the network to compete with each other when they receive signals from the nodes in the previous layer and form clusters. As also evidenced by the Deep COLA pseudocode in Algorithm 1, its functionality operates on two basic procedures that involves a number of smaller steps: (i) learning procedure and (ii) retrieval & association procedure. In the following sections, we describe the formulation of these procedures.

3.3.2 Learning Procedure

The first step of the Deep COLA learning procedure involves encoding high dimensional input patterns into a set of low dimensional templates during the feed-forward phase. Following the encoding step is the consolidation of the set of templates towards input patterns during the feedback phase. The final step relates to retrieving and associating inputs with saved patterns. The set of learned templates are formed when the input is propagated through the hidden and output neurons and multiplied with the weights between them. In the following sections, these steps are explained in detail.

Encoding of Input Patterns

Let $X = \{x_1, x_2, x_3, \dots, x_m\}$ be an input sample from the dataset with the dimensions (1 * m). W^1_{ij} $(m * n)$ is a weight matrix having the dimensions $(m * n)$ between the input and hidden layers, and W^2_{ij} $(n * m)$ is a matrix indicating weights between the hidden and the output layers, where m represents the number of features and n is the number of neurons in the hidden layer. The two matrices are expressed as:

$$\begin{gathered} W^1_{ij} = (w_{ij(m\times1)}, w_{ij(m\times2)}, w_{ij(m\times3)}, \dots, w_{ij(m\times n)}) \\ W^2_{ij} = (w_{ij1}, w_{ij2}, w_{ij3}, \dots, w_{ijn})^T \end{gathered} \tag{3.4}$$

The choice of the initial weights can originate from random values, informed decisions (Xavier initialization), or pre-trained weights in a methodology called Transfer Learning [112]. In our algorithm, we employ the Xavier weight initialization process [113] to select W^1_{ij} and W^2_{ij}. As such, weights are randomly initialized and multiplied with the square root of the

Algorithm 1 Deep COLA pseudocode for consumer behaviour modelling

procedure ALGORITHM
  $X \leftarrow$ Data matrix
  $W^l_{ij} \leftarrow$ *Weights for node (i) in layer (l)*
  $C \leftarrow$ *saved pattern*
  $\hat{X} \leftarrow$ *reconstructed output*
  $d \leftarrow$ distance
  $pd \leftarrow$ previous distance
  $t1 \leftarrow$ temp variable
  $q \leftarrow$ control variable for threshold
  initialize:
  $\theta \leftarrow 0$
  $d \leftarrow 0$
  $q \leftarrow 0.3$
  $W \leftarrow$ Xavier initialization
  perform reconstruction of input:
  for input vector x in X **do** ▷ Loop for total no. of input vectors
    $Y^1_i \leftarrow \sum_{j=1}^{n} W^1_{ij} x_j$ ▷ calculate signal at hidden layer for node (i)
    $S^1_i \leftarrow 1/(1 + e^{Y^1_i})$ ▷ apply activation
    $Y^2_i \leftarrow \sum_{j=1}^{n} W_{ij} S^1_j$ ▷ calculate signal at output layer
    $S^2_i \leftarrow 1/(1 + e^{Y^2_i})$ ▷ apply activation
    $C \leftarrow W^2_{ij} x_j$ ▷ find correlated node
    sort vector in descending C
    for vector C_i **do**
      ▷ measure (dis)similarity
      $d \leftarrow \frac{1}{N} \sum_{n=1}^{N} (\hat{X}_n - X_n)^2$
      ▷ calculate error for output layer
      $e^2_i \leftarrow \frac{1}{N} \sum_{n=1}^{N} (\hat{X}_n - X_n)^2$
      $ed^2_i \leftarrow e^2_i W^2_{ij}$ ▷ calculate error delta
      ▷ calculate error for hidden layer
      $e^1_i \leftarrow S^l_i - W^1_{ij}$
      $ed^1_i \leftarrow e^1_i W^1_{ij}$ ▷ calculate error delta
      ▷ update weight layer 1
      $W^1_{ij} \leftarrow W^1_{ij} + S^1_i * ed^1_i$
      ▷ update weights layer 2
      $W^2_{ij} \leftarrow W^2_{ij} + S^2_i * ed^2_i$
    minimize d *for reconstructed output* Y
  perform clustering:
  for vector Y_i **do**
    ▷ find similar patterns in the reconstructed output
    $C_1 \leftarrow \frac{1}{N} \sum_{n=1}^{N} (\hat{Y}_n - Y_n)^2$

    calculate vector which will track outside patterns
    assign new cluster based on threshold for outside patterns

current neural network layer dimensions. This prevents the weights from reaching a large value or vanishing to a minimum value. Hence, the Xavier process assists in maintaining the same variance of weights across the layers. The Xavier process is defined as:

$$VAR(W_{ij}) = 2/(n_{in} + n_{out}) \tag{3.5}$$

The parameters n_{in} and n_{out} are the number of current and next layer neurons, respectively.

Next, the input matrix X is multiplied with the weights and summed at each neuron of the hidden layer to obtain the input signal received at the hidden layer neuron denoted as Y_i^1 and the output signal to the next layer denoted as Y_i^2 (see lines 17 and 18 in algorithm 1).

$$\begin{aligned} Y_i^1 &= \sum_{j=1}^{n} W_{ij}^1 x_j \\ S_i^1 &= 1/(1 + e^{Y_i^1}) \end{aligned} \tag{3.6}$$

At each output neuron, the output of the hidden layer neurons is multiplied with the weights and summed as follows:

$$\begin{aligned} Y_i^2 &= \sum_{j=1}^{n} W_{ij}^2 S_j^1 \\ S_i^2 &= 1/(1 + e^{Y_i^2}) \end{aligned} \tag{3.7}$$

At the output layer, the algorithm learns the hidden representation and reconstructs the original input. In order to establish a decision on whether to activate or deactivate neurons in the preceding layer, the Sigmoid and ReLU activation functions are applied at each neuron of the hidden and output layers (see lines 18 and 20 in algorithm 1). In particular, both activation functions perform transformations to the weights and inputs, to learn complex patterns of energy consumption habits, with respect to consumer behaviour.

Consolidation of the Hidden Representations

During the consolidation step, the cumulative error signals, e_i and e_j, are computed (see lines 27 and 30 in algorithm 1) using the input pattern X and outputs S_i^1 and S_i^2. The resulting computation aims to indicate the level of similarity between the input patterns X and the templates $\hat{X}$ saved in the network. The templates in the network are saved in the form of weights of the neurons at the output layer. As the number of clusters increases, the number of neurons in the hidden layer also increases. The error signal at the output layer is:

$$e_i^2 = \frac{1}{N}\sum_{n=1}^{N}(\hat{X}_n - X_n)^2 \tag{3.8}$$

Where the cumulative error signal at the output layer is:

$$ed_i^2 = e_i^2 W_{ij}^2 \tag{3.9}$$

Using the chain rule during the back propagation of the error signal, the derivative of the error with respect to weights of the output to the hidden layer Y_i^2 is:

$$\frac{\partial ed_i^2}{\partial W_{ij}^2} = \frac{\partial ed_i^2}{\partial e_i^2} * \frac{\partial e_i^2}{\partial Y_i^2} * \frac{\partial Y_i^2}{\partial Y_i^{2\prime}} * \frac{\partial Y_i^{2\prime}}{\partial W_{ij}^2} \tag{3.10}$$

The derivation process is repeated for the hidden to the input layer Y_i^1 as:

$$\frac{\partial ed_i^1}{\partial W_{ij}^1} = \frac{\partial ed_i^1}{\partial e_i^1} * \frac{\partial e_i^1}{\partial Y_i^1} * \frac{\partial Y_i^1}{\partial Y_i^{1\prime}} * \frac{\partial Y_i^{1\prime}}{\partial W_{ij}^1} \tag{3.11}$$

After calculating the error derivatives, the weights of the hidden and output layers are updated with respect to the computed errors. Consequently, there will be a shift of the weights towards the input pattern. Eventually, the resulting shift strengthens the saved patterns such that the weights between the layers show similarity with the input pattern X.

The new weights are calculated as:

$$\begin{aligned} \Delta W^1_{ij+1} &= \Delta W^1_{ij} - \eta \frac{\partial ed^1_i}{\partial w^1_{ij}} \\ \Delta W^2_{ij+1} &= \Delta W^2_{ij} - \eta \frac{\partial ed^2_i}{\partial w^2_{ij}} \end{aligned} \tag{3.12}$$

Where η represents the learning rate. The encoding and consolidation process provides the output, which has reduced the dimensions, compared to the input.

3.3.3 Retrieval & Association

As explained earlier, the data is passed through the network and weights are adjusted to minimize the error surface during the reconstruction of the input at the output layer. These steps assist in saving the patterns in the network in terms of adjusted weights and building the Deep COLA learning model. In the retrieval and association procedure, the process of pattern search and matching is initiated. Hence, the sole goal is to assess the similarity of the input pattern to the saved patterns, by utilising a similarity score.

Objective functions

There are two objective functions that are used in the process of retrieval and association. The first objective function (mean squared error) is used to find the similarity and dissimilarity of the input vector with the reconstructed output. The level of similarity is calculated between a given input pattern and the reconstruction $\hat{\mathrm{X}}$ using the normalized version of sum of squared differences, as follows (see line 25 in algorithm 1):

$$d = \frac{1}{N} \sum_{n=1}^{N} (\hat{X}_n - X_n)^2 \tag{3.13}$$

Where N represents the total number of patterns in X. The second objective function is used to find the patterns which are outside the threshold. In this procedure, the feature space is considered as probability distribution. The dense regions in the feature space correspond to be similar and construct a cluster. The process is repeated until all the points in the feature space are associated with the same cluster. The mode of the density function in the window is determined by (see line 40 in algorithm 1):

$$C_1 = \frac{1}{N}\sum_{n=1}^{N}(\hat{Y}_n - Y_n)^2 \tag{3.14}$$

Epoch Selection in Deep COLA Algorithm

The Deep COLA algorithm performs clustering in multiple epochs. The suggested number of epochs for the Deep COLA is 100 in order to achieve the best accuracy. This number of epochs is selected using a technique called early stopping. Early stopping is a type of generalization, which is used to avoid overfitting during training of the algorithm. After the 100 epochs, the improvements in the training error of the algorithm becomes minimal and therefore, 100 epochs are selected during the training. In the first epoch, the algorithm finds the number of clusters using three steps: 1) encoding, 2) consolidation and 3) retrieval and association. The first epoch retrieves clusters using competitive learning. However, after the first epoch, the Deep COLA algorithm refines the clusters using the retrieved cluster centres. At the second epoch, data is shuffled to prevent the gradient descent reaching a local minimum and passing through the network. For instance, if the cost function is not convex then it will be made up of a multitude of peaks and troughs. Some of the troughs will be deeper than the others. The troughs that are not the lowest are regarded as local troughs and their corresponding minimum points as local minimum points. Subsequently, the distance between the weights and all the nodes created in the first epoch is calculated at the output layer. The node having the minimum distance from the input data is selected and the data is assigned to it. In the first epoch, the threshold parameter is used for cluster assignment, while in the upcoming epochs the minimum distance between the new data points and the existing patterns is selected for cluster formation.

Network Architecture of Deep COLA Algorithm

There are other parameters in the network architecture of the Deep COLA algorithm which are required for the training such as hidden layers and the number of neurons in each layer, learning rate, the activation and loss functions and the optimizer. The accuracy and validity of the extracted load profiles highly depend on the best selection of hyperparameters of the Deep COLA algorithm. These parameters can be selected by I) tuning an individual parameter and observing the model performance, II) searching in the parameter intervals with various zoom levels, III) performing a grid search over a complete set of parameters [114]. We employ a grid search technique to tune the model parameters. The network architecture of the Deep COLA algorithm consists of 3 hidden layers and the optimizer function used is Adadelta. The hidden layer activation functions are Sigmoid and ReLU which assist in learning the underlying structures in the data. In order to update the weights of the parameters, a mean squared error is used. A large learning rate affected the training of the Deep COLA algorithm by taking big steps and provided inaccurate load profiles. Therefore, the learning rate of 0.01 is used as a hyperparameter for efficient training. Table 3.1 provides a summary of the Deep COLA parameters.

Table 3.1: Summary of the Deep COLA parameters

Parameter	Value
Layers and Neurons	[43 5 43]
Optimizer	Adadelta
Hidden layer activation	Sigmoid and ReLU
Loss function	Mean Squared Error
Learning rate	0.01
No. of Epochs	100

3.4 Short-term Load Forecasting through Parallelisation

The traditional energy systems in generation grids and households are being transformed into smart grids and smart meters respectively. The household appliances which are utilised for the activities such as cooling, warming, cooking, entertainment etc will be able to generate a huge amount of data at a very granular level which will contain big data properties. The efficiency and validity of the machine learning models is highly dependent on the amount of data. The accuracy of the prediction and consumption profiling can be increased with the help of developing complex models to process large amounts of data. As more data is available, it facilitates the complex models to learn hidden patterns and relationships between energy consumption and various factors such as consumer behaviour, temperature variations, humidity levels, geographical locations, energy prices and occupancy effects. However, the current data analytics technologies are highly affected in terms of computational cost to process large amounts of data. Additionally, the volatility of the generation systems, especially energy generation through renewable resources, adds additional challenges for current data analytics technologies.

The current machine learning technologies mostly rely on sequential processing of data and the processing time is directly proportional to the amount of data. However, parallel computing is achieved through parallel processing in terms of model parallelism and data parallelism. The deep learning technologies are recently attracting the interest of research communities due to their superior performances in object detection [115], speech recognition [116] and character recognition [117]. The success of deep learning is mainly influenced by three major drivers [97]. The first driver is related to the size and complexity of the deep learning algorithms used for the analysis of data. The complex algorithms, trained properly using large amounts of data, often provide better performance as compared with shallow deep neural networks. The second driver is the availability of huge amounts of data due to the advancements of technology. The accuracy of the deep learning algorithms can be improved by using more data. The third driver is related to the available resources in terms of distributed systems for deep learning algorithms training. In particular, the reduction in the computational time of these algorithms to process large amounts of data is due to the parallel computing hardware such as GPUs, which is the major factor for the success of deep learning algorithms. The following sections provide descriptions of the different forms of parallelisation, system architectures and synchronizations which are utilized for big data analytics in the SG domain.

3.4.1 Short-term Load Forecasting Model Description based on Parallelisation

Although there are various approaches available in the literature to improve the processing time and accuracy of the deep learning models for large amounts of data, the improvement in the performance highly depends on three major factors: the type of parallelisation used, the method for synchronizing parameter updates (the central or distributed paradigm) and the communication overhead across the cluster nodes. The main deep learning model components are depicted in Figure 3.8.

Our proposed parallelisation model is based on the concept of data parallelisation. The data is distributed across the clusters, and the individual deep learning model on each node performs prediction of the energy consumption. The data parallelisation approach provides a good performance when the deep learning models have less complexity. On the other hand, the model parallelisation approaches improve the performance of large and complex deep learning models [118].

The centralized architecture consists of a client-server model. At the end of each batch processing, the worker nodes calculate the gradient and update the server which contains global model parameters. In the centralised architecture, the model parameters are updated asyn-

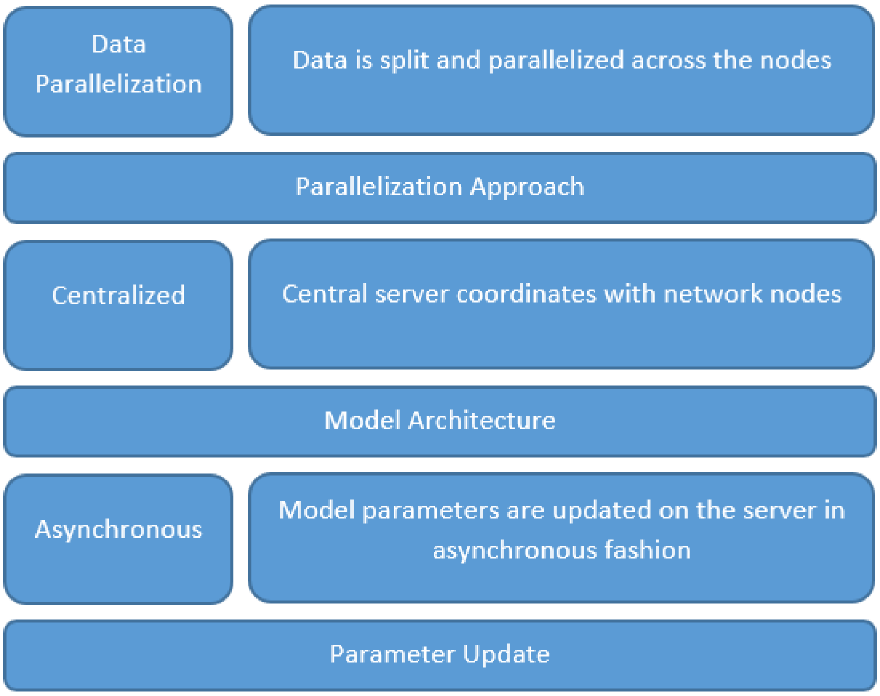

Figure 3.8: Parallelisation model components

chronously which means all the nodes send the gradient without requiring the lock. In this way, the worker nodes do not have to wait for the slower worker nodes. In the centralised architecture, the slower worker nodes are the main bottleneck because they increase the overall computational cost of the deep learning model [119].

In the distributed learning, there are certain requirements which are fulfilled with the help of big data processing frameworks responsible for the management of data storage and processing. Our deep learning parallelised model is implemented using a big data processing framework called Apache Hadoop and Spark.

3.4.1.1 Apache Hadoop

The Hadoop Distributed File System (HDFS) is an open source software framework which is developed to store and process large amounts of data efficiently and reliably [2] in a distributed environment. The Hadoop framework provides cheap, scalable and fast computation [120] to manage the data produced at a high rate by smart meters in households. Table 3.2 describes the main components such as HDFS, Yarn [121], HBASE, Pig [122], Hive [123], Zookeeper [124], Cuhkwa and Avro in the Hadoop echosystem.

Table 3.2: Components of Hadoop Architecture [2]

Component	Description
HDFS	Distributed file system
Yarn	Distributed resource manager
HBASE	Column-oriented table service
Pig	Dataflow language and parallel execution framework
Hive	Data warehouse infrastructure
ZooKeeper	Distributed coordination service
Chukwa	System for collecting management data
Avro	Data serialization system

The HDFS component distributes large amounts of electricity consumption data across multiple nodes in a cluster. The data is indexed in the HDFS file system which improves the processing and reduces the computational time of the deep learning algorithms. In HDFS, the storage management in the cluster is performed by two entities, namenode and datanode. The namenode maintains metadata about the storage in all the worker nodes in the cluster. The files are stored as blocks typically of size 128 megabytes and replicated to the datanodes in the cluster. The namenode keeps the metadata related to the blocks in random access memory (RAM) in order to reduce access time during computations. The datanode, on the other hand, resides in worker nodes acting on the commands for deleting, adding or replacing blocks received from the namenode.

3.4.1.2 Apache Spark

The Apache Spark provides a unified analytics engine to process large amounts of data with high scalability [125]. Initially, a model was developed to process data-parallel computations in a cluster which was called MapReduce [126]. The MapReduce became widely popular due to providing efficient fault tolerance, load sharing, scalability and data-aware scheduling. The

MapReduce programming model allows us to create acyclic data flow graphs which utilize a set of operators in order to pass data. This enables MapReduce to automatically achieve scalability and fault tolerance without user intervention [125]. The programming model based on acyclic data flow graphs is particularly suitable for certain types of applications which do not rely on reusing data. However, deep learning algorithms are based on the paradigm of processing large amounts of data recursively to enhance the processing of learning. Therefore, Apache Spark provides support to implement deep learning models by inheriting scalability and fault tolerant capabilities from MapReduce.

3.4.1.3 Parallelised Deep Learning Model

The pseudocode in algorithm 2 illustrates the steps taken to parallelise the deep learning model. Mainly the data is partitioned across all the nodes in the architecture and the parameters such as the learning rate, number of iterations, weight initialisation and weight decay are initialised. The model performs learning on individual nodes in a parallelised fashion which assists in reducing the overall processing cost. At the end of each iteration, the parameters and errors are communicated to the master node and also passed to all the nodes to update the weights.

Algorithm 2 Parallelised Deep Learning Model for short-term load forecasting

procedure ALGORITHM
    initialize:
    $P \leftarrow$ Set parameters of the model such as weights and learning rate
    $X \leftarrow$ Shuffle data
    $N \leftarrow$ Set subset of data X on each Node N
    perform learning through parallelisation:
    for each subset x in X **do** ▷ Loop for total no. of partition of data
        on each node N, perform gradient descent to reduce RMSE
        Synchronise params with all the nodes N
    Aggregate the RMSE errors and finish learning

3.5 Summary

In this chapter, we provide the description of our research methodology for predicting energy consumption for short-term, proposed Deep COLA algorithm to profile individual appliances energy usage in households and a parallelisation model based on Apache Spark to forecast energy consumption in a parallelisation paradigm to reduce computational time. The view of the energy consumption information for day, week and month alongside human behaviour modeling for their day-to-day activities in households provides enormous opportunities to utility companies in their decision making processes such as demand response programmes. Furthermore, consumers in households also benefit from their detailed analysis of energy consumption for individual appliances and manage their activities for cost effective energy usage. In the following chapter, we will describe the energy consumption data collected at appliance level and its properties, which are used to assess the applicability of our proposed approaches in this chapter.

Chapter 4

Data Description and Pre-Processing

4.1 Introduction

The characteristics of the energy consumption data used during the development of the models for load prediction and understanding human behaviour have special importance to achieve required outcomes. These characteristics of the data are associated with the granularity of energy consumption, the number of appliances in a house, geographical factors, the overall energy consumed in a particular area under a distribution centre, the overall energy consumed by a house, and the amount of energy consumed by individual appliances. Therefore, to assess the performance and validity of our proposed models, three energy consumption datasets have been used for experiments in this research. The datasets consist of the electricity consumption recorded at a distribution centre and the consumption in individual households with a large set of electrical appliances for their daily activities. The dataset collected from the distribution centre is categorised as a state level dataset, while the dataset collected from individual households is categorised as a house level dataset. The description of each dataset is provided in the following sections.

4.2 State-Level Dataset

The state level dataset is obtained from the ISO New England (ISO-NE) electricity distribution centre in the USA for the duration from 2007 to 2012 as summarized in Table 4.3. ISO-NE is an independent, non-profit Regional Transmission Organization, serving 6 states, namely Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont.

ISO-NE oversees the operation of New England's bulk electric power system and transmission lines with the power generated and transmitted by its member utility providers, as well as Hydro Quebec, NB Power, the New York Power Authority and utilities in New York State, when the need arises. ISO-NE is responsible for reliably operating New England's 32,000 megawatts [MW] (43,000,000 hp) bulk electric power generation and transmission system. One of its major duties is to provide tariffs for the prices, terms and conditions of the energy supply in New England. The load consumption values are recorded at the end of each hour in a day. The whole dataset consists of 52,600 records. These consumption values represent the load for the 6 states combined. As this dataset contains energy consumption information from the distribution side, it enables the deep learning model to learn patterns at higher energy consumption level. Therefore, the requirement is to select the energy consumption data to evaluate proposed models at higher level (distribution side) as well as lower level (consumer side) in order to improve model generalization capabilities. Table 4.1 lists the original features in the dataset and their description.

4.3 House-Level Dataset

4.3.1 Smart Star Dataset

The house level dataset Smart Star contains the electricity load of individual houses at the appliance level like microwaves and dish washers. This dataset is quite comprehensive and

Table 4.1: ISO New England dataset features and description

Features	**Description**
Date	Date (MM/DD/YYY)
Hour	Hour of the day (24 load values in a day)
ElecPrice	Price of the electricity (MW/h)
DryBulb	Dry bulb temperature (Fahrenheit)
DewPnt	Dew point temperature (Fahrenheit)
SYSLoad	NEPOOL system load = [generation - pumping load + net interchange] as determined by metering (MWh)

it is collected for the duration of 3 years from 2014 to 2016 as summarized in Table 4.3. It is obtained from the Smart* project [127], containing both electricity loads at the appliance level and weather conditions. In the dataset, the appliance-level power consumption values have been collected by sensors with varying measurement samples in each household (e.g. 1, 15, 30 minutes). This dataset assisted in evaluating proposed models at appliance level. At appliance level the uncertainty and variations in energy consumption are higher as compared to the energy consumption recorded at this distribution side. Therefore, these variations provide a chance to evaluate the proposed deep learning models. The dataset has the following features as shown in Table 4.2:

Table 4.2: Smart Star dataset features

Features			
Date	Time	Furnace HRV	Microwave
Dish Washer	Fridge	Dryer	Bedroom Lights
Temperature	Condition	Humidity	Visibility
Pressure	Wind Speed	Cloud Cover	Wind Bearing
Precipitation	Dew Point	Precip Probability	Total Consumption

4.3.2 UK-Dale Dataset

The house-level dataset UK-Dale originally contains 46 features including appliance level electricity consumption information and weather conditions. The electricity consumption values at the appliance level are collected using multiple sensors in a home. The different sensors collect data at different time intervals, i.e. 1 and 6 seconds as summarized in Table 4.3. Therefore, it is essential to extract data at the same time interval and aggregate it into one file. Some of the features in the dataset contain categorical values such as the weather condition which can be clear, cloudy, rain, wind, or fogy. These are converted to numerical values 1 to 10 using direct mapping. After converting all the values to numeric ones, the min-max normalisation is applied. The normalisation process converts all the feature values to those between 0 and 1. The reason for applying the normalisation is that large values in the dataset will require large weights during the weight initialisation process in neural networks. The Smart Star dataset contains 15 appliances which do not represent the range of appliances that are used inside residential houses. Therefore, the UK-Dale dataset is used to provide a facility to evaluate the models using a larger number of appliances which is 53. Additionally, the sampling rate of energy consumption in the UK-Dale dataset is also lower as compared with the Smart Star dataset. Table 4.3 summarise the Smart Star, UK-Dale and ISO New England datasets:

Table 4.3: Smart Star, UK-Dale and ISO New England Datasets Description

Dataset	**Duration (years)**	**Households**	**Appliances**	**Sampling rate**	**Country**
Smart Star	3	4	15	1, 15 (min)	UK
UK-DALE	4	1	53	6 (sec)	UK
ISO New England	6	-	-	1 (hour)	USA

4.4 Data Pre-processing & Feature Extraction

The electricity consumption datasets require pre-processing to make them suitable for deep learning algorithms. This pre-processing consists of handling empty records and normalising data values. The Smart Star dataset originally contains 6 features. Initially, this dataset contained incorrect sequences of date/hour data and missing values in some attributes. As the data was collected using multiple sensors at very small intervals, they generated a large amount of information that was aggregated at a common point. Therefore, networking issues and delays in circuits resulted in incorrect sequences of date/hour and missing values. The records with above 30% missing values have been removed, while the records with fewer missing values are filled using an attribute mean method that allows us to insert missing values using the mean of the neighbour values. The updated dataset contains only a small proportion of the input data, which does not affect the validity of the revised dataset.

4.4.1 Extracting Features from ISO New England Dataset

The ISO New England dataset contains features which do not properly highlight the effect of various factors that make electricity consumption profiles of household users highly non-linear and complex in nature. The factors such as weather, time, holidays, lagged load and load distribution in different time periods are the most affecting factors to electricity consumption on a day-to-day basis. The data is analysed in time and frequency domains to extract features to represent these effects. This approach is described in the following sections.

4.4.2 Time Domain Feature Extraction

The original ISO New England dataset is pre-processed in the time domain to analyse the variations in consumers' behaviour to consume electricity in the households with respect to time. The following effects are captured using the time domain analysis of the data.

4.4.2.1 Effect of Temperature on Electricity Consumption

There is a higher impact of climate on the consumption of electricity. The consumers utilise more electricity when there are extreme lower or higher temperature values in a certain area [128]. Figure 4.1 highlights that the daily consumption of electricity over a week is affected by the temperature. Depending on the season, when the temperature is higher during the

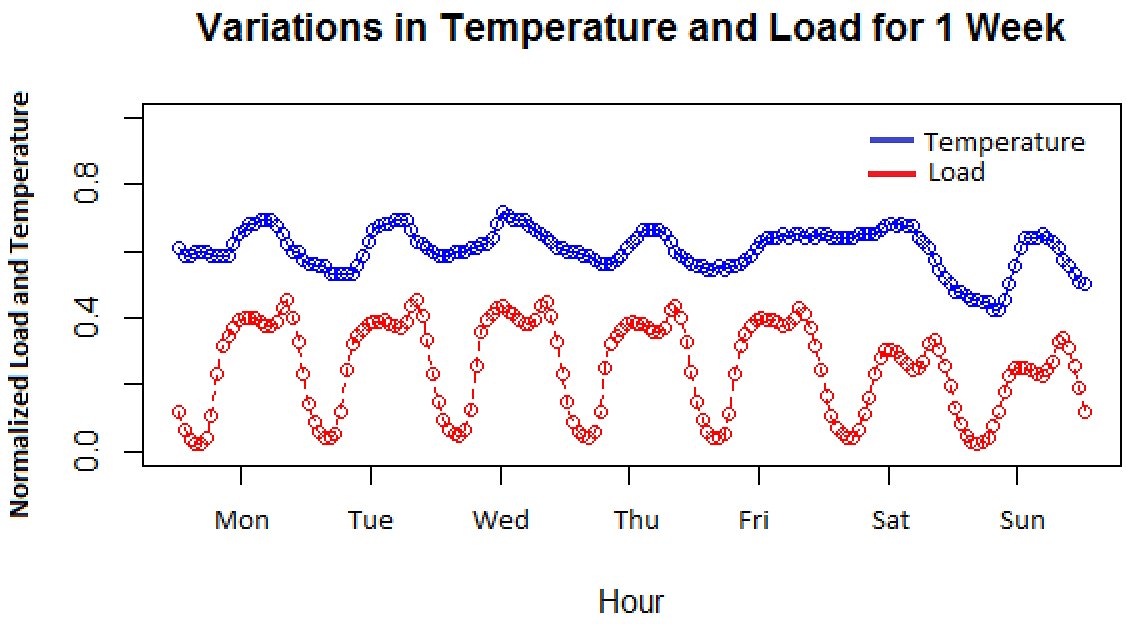

Figure 4.1: Effect of temperature on electricity load

day time, more electricity is consumed to make the indoor temperature suitable according to the environment. The dry bulb and dew points are used to determine the humidity in the air. Different amounts of electricity are consumed to maintain the humidity in summer and winter seasons. The dry bulb and dew point features have been selected to represent the temperature effect on electricity consumption. The temperature provides a cumulative effect on the electricity consumption with the other factors such as consumer behaviour and social events. Figure 4.1 reveals that although the temperature is higher on Friday and Saturday, it has little effect on consumption. This occurs because there are fewer activities in the commercial sector on Saturday than on Friday.

4.4.2.2 Effect of Working and Non-Working Days on Electricity Consumption

Figure 4.2 illustrates that the electricity consumption is different in working and non-working days. A higher amount of electricity is normally consumed during week days in households, schools, offices, industries etc. as compared to weekends. Therefore, the electricity demands are not the same for the whole week. In Figure 4.2, the electricity consumption values are plotted for three adjacent weeks. From days 1 to 5, electricity consumption is higher, while on days 6 and 7 electricity consumption is lower. Therefore, to add the effect of working and non-working days in the load forecasting based on these observations, a new feature "Is Working Day" has been introduced in the dataset.

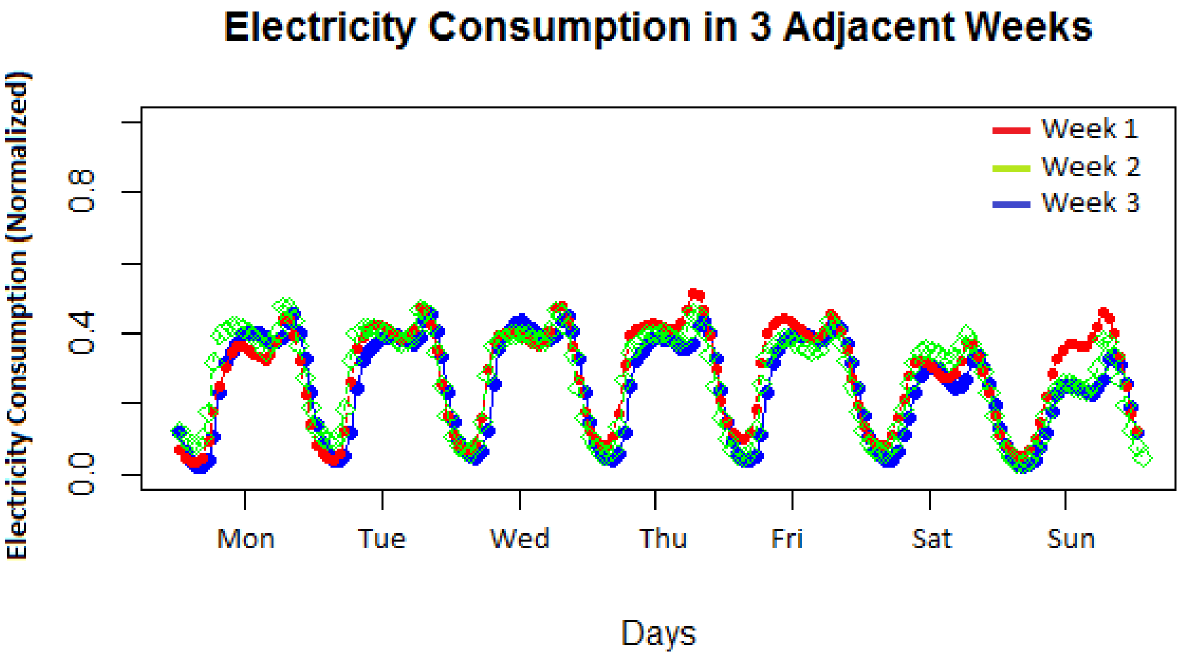

Figure 4.2: Working and non-working days effect on electricity load

4.4.2.3 Effect of Time on Electricity Consumption

Alongside the temperature as well as working and non-working day effects, time also has an effect on the amount of electricity consumed. Figure 4.2 indicates that the electricity consumption is highly dependent on the time as the consumption of electricity increases and decreases before and after the midday respectively. Two features “Hour” and “Week Day” are derived to introduce the time dependency in the dataset. Most of the activities are performed during the week days, so the consumption of electricity remains higher than the weekends. However, sometime social events are planned during the weekends, resulting in the increased consumption of electricity and hence different weekend consumption patterns. This can be observed in Figure 4.2 where week 1 has higher consumption on Sunday. These small changes in the consumption patterns occur with random behaviour and contribute relatively more to load prediction errors.

4.4.2.4 Lagged Load Effect

Lagged hourly peak load values contain vital information about specific patterns of users' electricity consumption. This information can be extracted using multiple lagged load values with the features: previous day same hour load, previous 24 hours average load and previous week same hour load. The lagged load values of the electricity load incorporate the information regarding the behaviour of the consumers. These lagged values are extracted from the electricity load, and consumer activity is analysed based on the consumption patterns on the same day, the previous day and the previous week.

4.4.2.5 Data Distribution Effect

Data distribution effects are captured over the past 24 hours using skewness, kurtosis, variance and periodicity. Skewness represents the symmetry of the load values. It provides information about the lower or higher trends in consumption values over a certain time period. Kurtosis illustrates heavy-tailed or light-tailed characteristics of data which highlights outliers. The variance shows the dispersion of individual points in a set of data. These features uncover hidden patterns in the dataset efficiently and assist deep neural networks to provide highest prediction accuracy. Further, Figure 4.3 depicts the density of the electricity demand over the period of one year (Smart Star dataset [127]). This figure illustrates that the electricity demand

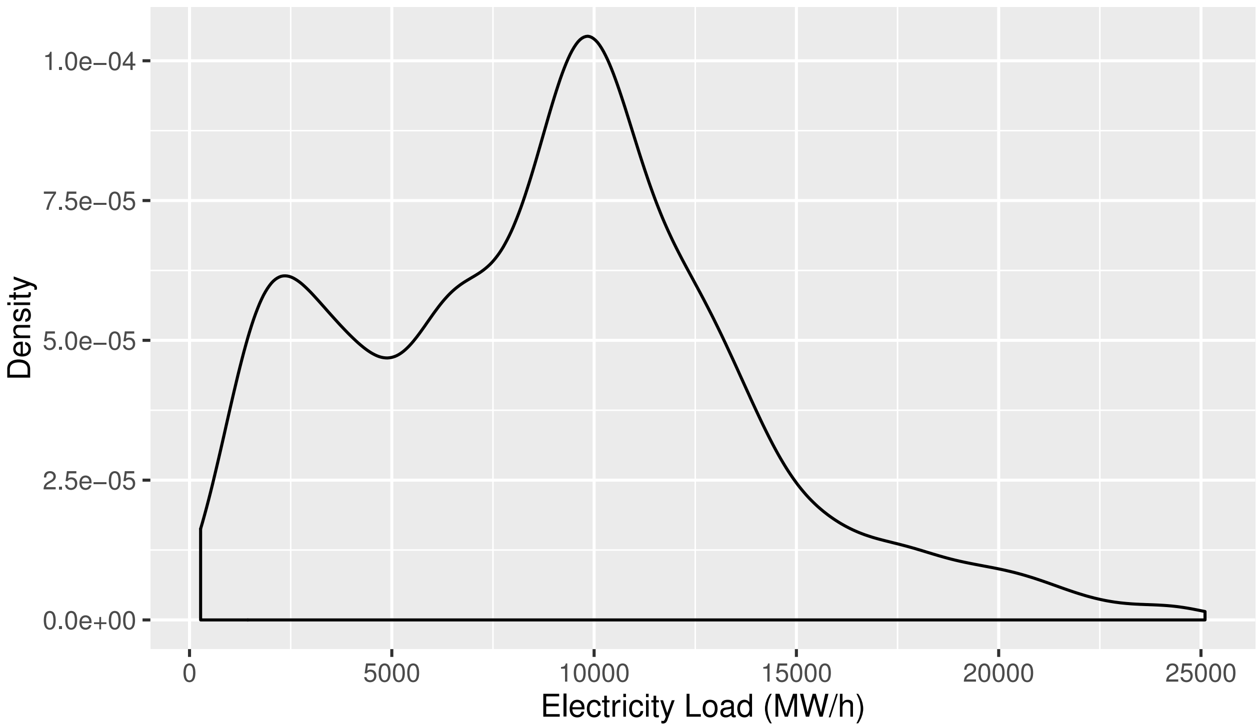

Figure 4.3: Density of electricity load

is not constant over the year because of various factors. It shows that over the year, most of the electricity demand was around 10,000 MW/m, which is helpful in planning electricity generation for the next year.

4.5 Summary

The applicability of the proposed approach for load forecasting and profiling highly depends on the granularity and quality of energy consumption information. The high frequency consumption data allows us to implement variable cost related demand response programmes in households [129]. In this chapter, we have provided the description of the datasets used in this research. To fully assess our proposed algorithms, the energy consumption data is used at various levels such as state and household levels using individual appliances' consumption.

Chapter 5

Experiments and Result Evaluation

5.1 Introduction

In order to improve prediction accuracy, extracting meaningful load profiles and efficient utilization of resources, a research methodology has been proposed in Chapter 3. This Chapter evaluates the proposed algorithms through experimentation by using real world energy consumption datasets.

5.2 Deep Neural Networks for Short-term Load Forecasting

Neural networks which are biologically inspired from human brain functionality possess the capabilities to learn hidden non-linear and complex structures in the data. In the late nineties, learning was difficult in large densely connected networks [130]. Hinton et al. [130] proposed Deep Belief Network (DBN) and showed that deep architectures can be trained using greedy layer wise pre-training techniques. With the success of DBN, DNN with many hidden layers are being utilized to tackle problems in the area of artificial intelligence. DBN and DNN refer to the class of neural networks that contain multiple hidden layers. However, the difference between them lies in the fact that the DBN contains some undirected layers called restricted boltzmann machines (RBM). These layers are trained using a technique called contrastive divergence which is an unsupervised learning algorithm. In this thesis, two models based on FF-DNN and R-DNN were proposed in Chapter 3 for day and week ahead short-term load forecasting. FF-DNN contains connections in the layers in the forward direction while R-DNN contains cyclic connections in the layers. The cyclic connections in R-DNN enables the network to form directed graphs along temporal sequences, which makes R-DNN suitable for the time series related applications [131].

5.3 Short-term Load Forecasting Results Analysis

In this section, we evaluate our proposed FF-DNN and R-DNN models for different seasons in 2012 using training data from ISO New England Dataset covering the years from 2007 to 2011. The training dataset contains 43,824 records while the test dataset contains 24 and 168 records for day and week ahead forecasting respectively. The new features extracted from the original 5 features using the frequency domain analysis introduced in Section 4.4.3 increase the accuracy of the models. For the evaluation of the models, RMSE, MAE and MAPE have been calculated for each forecasting type in the four seasons of 2012 which are summarized in Table 5.1. The existing approaches based on the shallow learning methods provide higher errors during load prediction. The MAPE error obtained from the proposed approach is compared with the research work in [132]. The load prediction method developed in [132], is based on multiple linear regression. The multiple linear regression produced the MAPE error of 3.99% which is higher than our two proposed models FF-DNN and R-DNN for the whole year MAPE error in Table 5.1. In the time domain, the MAPE error is 1.42% and 1.30% for FF-DNN and R-DNN respectively. While in the frequency domain, the MAPE error is 0.067% and 0.057% for FF-DNN and R-DNN respectively. As we will explain next, the predictions are made on the basis of two case studies.

5.3.1 Case Study 1

In case study 1, we use only time domain features and make predictions for day and week ahead using both FF-DNN and R-DNN models. Due to temperature variations in different seasons,

Table 5.1: Summary of the prediction errors using Time and Frequency domain features with FF-DNN and R-DNN models (TD=Time Domain, FD=Frequency Domain)

Season	**Forecast Type**	**MAPE(%)**				**RMSE(MW/h)**				**MAE(MW/h)**			
		TD		**TD+FD**		**TD**		**TD+FD**		**TD**		**TD+FD**	
		FF-DNN	R-DNN	FF-DNN	R-DNN	FF-DNN	R-DNN	FF-DNN	R-DNN	FF-DNN	R-DNN	FF-DNN	R-DNN
Winter	Day	1.00	0.97	0.019	0.016	202	194	3.84	3.55	152	141	2.81	2.63
	Week	1.01	0.98	0.035	0.029	188	178	6.19	5.73	149	138	4.98	4.54
Spring	Day	0.94	0.82	0.030	0.023	159	144	4.78	4.35	139	124	4.40	4.21
	Week	0.80	0.76	0.026	0.020	135	126	4.27	4.02	104	95	3.35	3.10
Summer	Day	1.26	1.16	0.060	0.045	209	202	11.86	9.06	103	98	10.67	9.60
	Week	1.03	1.01	0.078	0.056	284	366	16.52	15.33	292	277	13.47	11.97
Autumn	Day	0.91	0.82	0.036	0.026	190	175	6.36	5.86	133	116	5.00	4.42
	Week	0.97	0.83	0.039	0.029	179	162	7.28	6.33	131	118	5.14	4.81
Whole Year	Year	1.42	1.30	0.067	0.057	306	287	131	119	210	197	6.99	5.96

electricity consumption varies considerably and we forecast loads in winter, spring, summer and autumn separately. The time domain MAPE, RMSE and MAE errors are listed in Table 5.1 and shown in Figure 5.1 for FF-DNN and R-DNN. For example, the proposed models are able to get the fewest MAPE errors in the spring season while the highest MAPE errors are found in the summer season as depicted in Figure 5.1.

This is due to unexpected variations in electricity consumption because of high temperatures and social events in the summer season. FF-DNN networks are related to the simpler class of nerual networks where data flows in one direction from input to output. However, R-DNN due to their feedback connections, works as a memory based on temporal sequences and model complex patterns. Therefore, R-DNN provides higher accuracy with lower MAPE errors in both day and week-wide load forecasting as compared to FF-DNN networks. Electricity load consumption at consumer premises is mainly dependent on the time where occupants perform their day-to-day activities in domestic as well as in commercial buildings. Therefore, R-DNN outperforms FF-DNN for load prediction by learning temporal sequences.

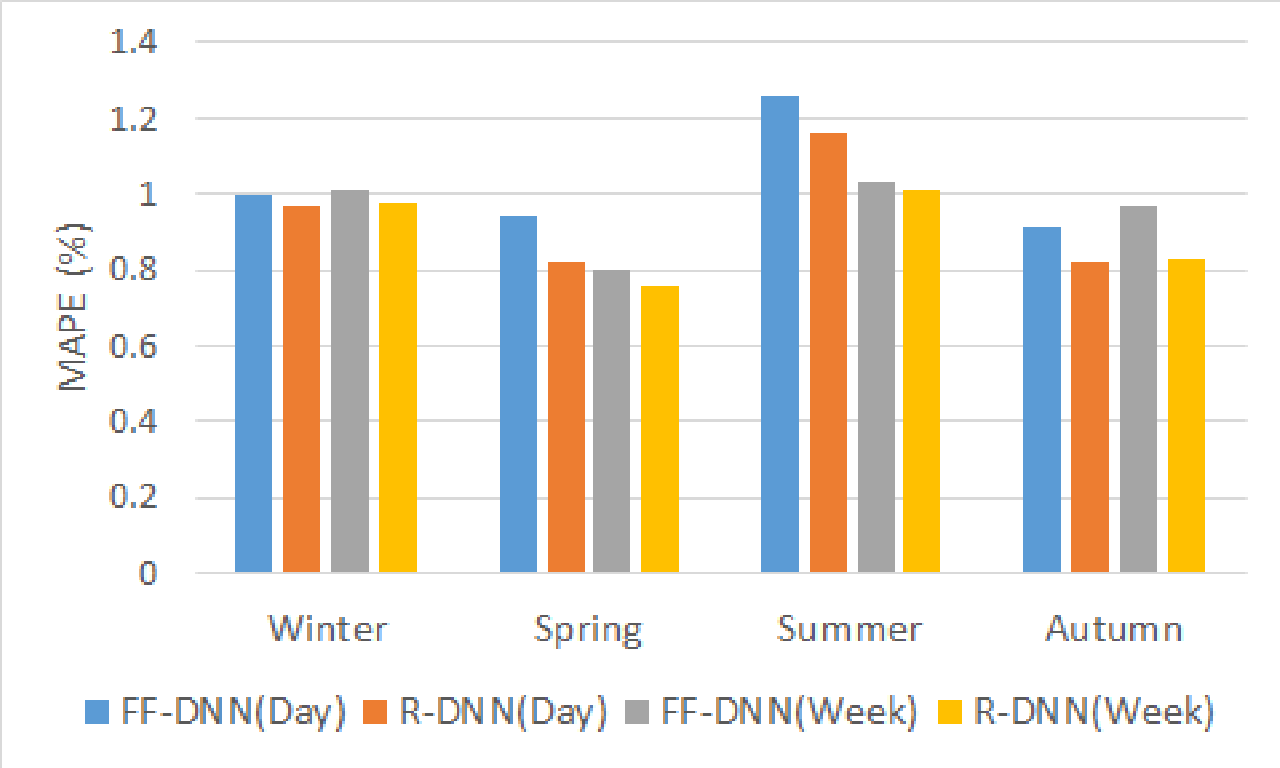

Figure 5.1: MAPE error comparison using time domain features

5.3.2 Case Study 2

In case study 2, we utilise composite features from both time and frequency domain analyses to forecast day and week ahead electricity loads using FF-DNN and R-DNN. The experiment results in terms of MAPE errors are shown in Figure 5.2. The errors are listed under the combined time domain and frequency domain (TD+FD) column in table 5.1 for both models. Clearly, the accuracy is improved with the errors being much lower than those found in case study 1 and illustrated in Figure 5.1.

The frequency domain features enable FF-DNN and R-DNN networks to learn variations and complex interactions between dependent and independent features and improve accuracy. During the spring season, the MAPE error for the next day forecasting with FF-DNN and R-DNN decreases from 0.96 to 0.03 and 0.81 to 0.024 respectively. Similarly during the summer season, the error with FF-DNN and R-DNN reduces from 1.26 to 0.06 and 1.16 to 0.04 respectively for the next day load forecasting.

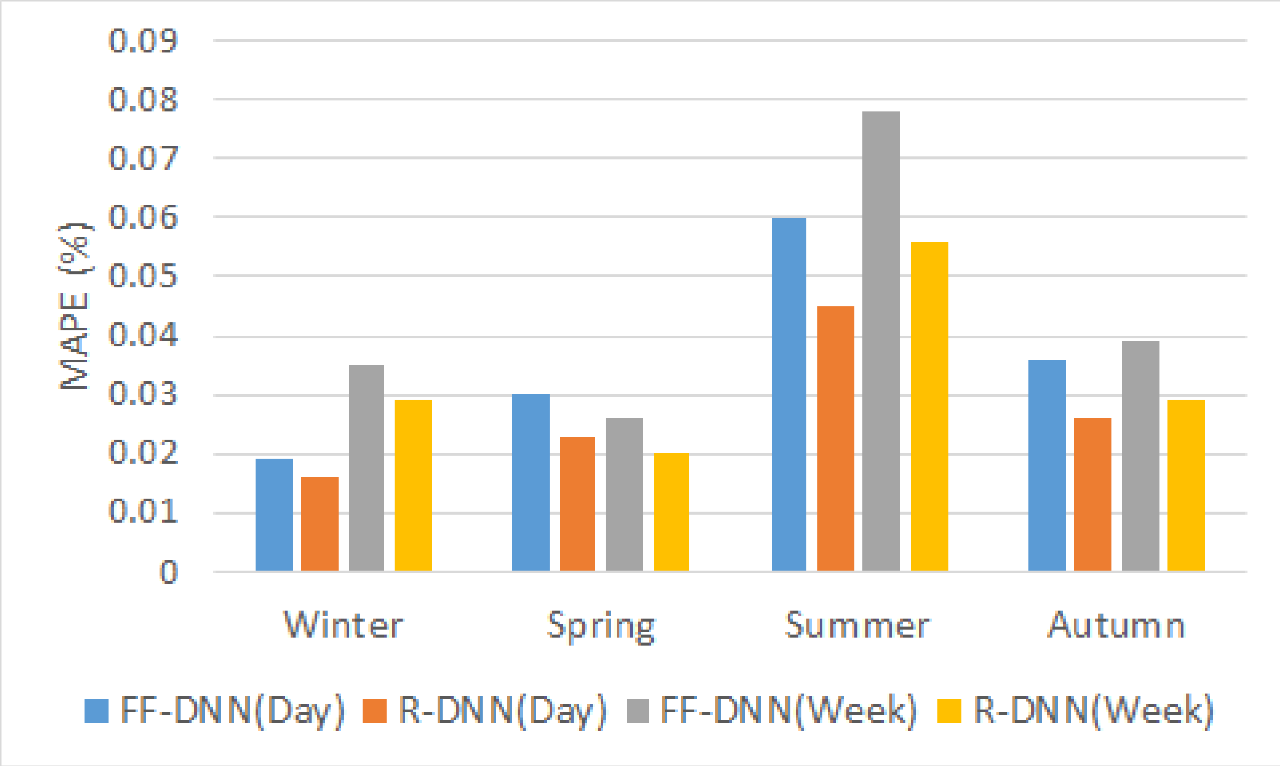

Figure 5.2: MAPE error comparison using both time and frequency domain features

5.4 Deep COLA Results Evaluation

The proposed Deep COLA algorithm has been assessed using the two real world electricity consumption datasets introduced in Section 4.3, Smart Star [1] and UK Domestic Appliance-Level Electricity (UK-DALE) [133], in terms of composing meaningful day, week and month-wide appliance-level energy utilisation profiles. These datasets contain energy consumption information at appliance level along with contextual information (e.g. time and environmental factors). The variety of devices and sampling intervals used in the Smart Star dataset led our initial pre-processing step to employ a scaling method for transforming power consumption time series and further hold a similar sampling frequency with non-power measurements. For instance, environmental measurements (e.g. humidity, fog, visibility, etc.) were converted to numerical values from 1 to 10 using direct mapping and subsequently scaled to get mean 0 and variance 1. The reason for scaling the features is that large inputs to our algorithm would essentially require large weights during the weight initialization process, which would degrade

[1] The Smart Star project dataset: http://traces.cs.umass.edu/index.php/Smart/Smart

the algorithm's efficiency.

The UK-Dale dataset contains a large number of features that are based on the consumption of 53 appliances. These features increase the dimensionality of the dataset, which increases the overhead and processing cost of the clustering algorithm. Another caveat is that it can also increase the complexity of the clustering algorithm, which causes a reduction in accuracy. Moreover, since each dataset had a different number of appliances, we selected the most utilised appliances out of each dataset. By the manual inspection of power consumption timeseries, we ended up selecting 6 appliances: furnace HRV, microwave, dishwasher, fridge, dryer and bedroom lights, from the Smart Star dataset and 38 from the UK-Dale dataset. The appliances, such as switches and outlets, which had 0 or a minimal amount of energy consumption, were excluded from the datasets.

As we will show in Section 5.4.4, Deep COLA is also compared, in terms of computational time and clustering accuracy, against commonly used clustering approaches: K-means, DBSCAN and Self Organizing Maps (SOM). The K-means and DBSCAN algorithms are used directly without reducing the dimensionality of the energy consumption data, while SOM is used to reduce the dimensionality of the data and then the K-means algorithm is used to profile the energy consumption of individual appliances.

5.4.1 Day-wide and Week-wide Energy Consumption Profiling (UK-Dale Dataset)

Figure 5.3 shows the consumption profiles extracted from the UK-Dale dataset for one day time duration[2]. Through this analysis, we demonstrate that Deep COLA adequately extracts patterns from high-volume data sampled at 6-second intervals and bearing high granularity. The purpose is to find flat, medium and high consumption profiles that represent the electricity used in a day by grouping individual appliances. In contrast with the Smart Star dataset, we assessed a large number of appliances for multiple homes. This resulted in different profiles in

[2]Similarly with Figure 5.3 we highlight the low, medium and high consumption profiles.

both one day and one week characterisations. The Deep COLA algorithm extracts three load profiles in the dataset as shown in Figure 5.3.

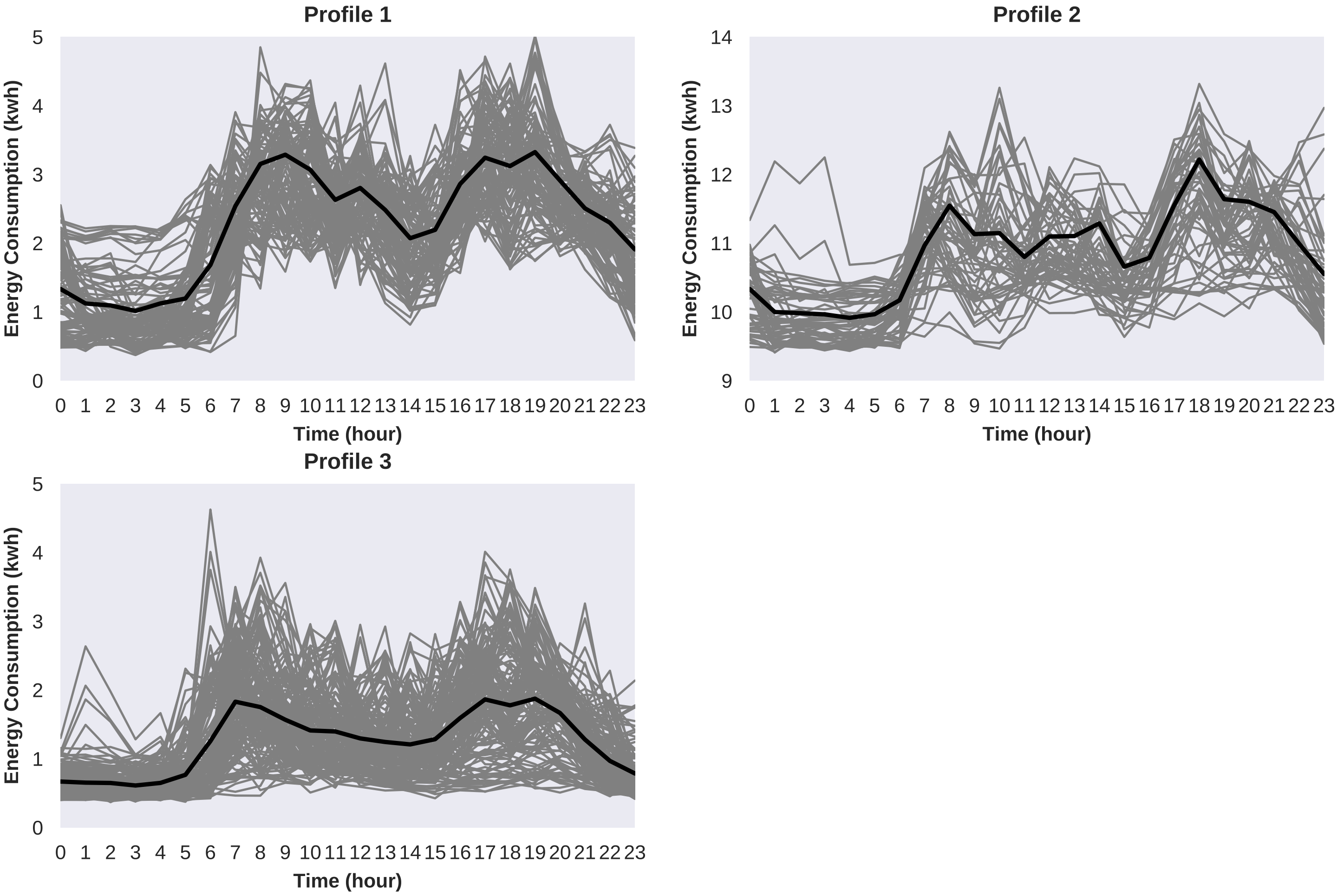

Figure 5.3: One day consumption profiles from Home 1 (UK Dale Dataset)

These three profiles indicate the most common patterns in the energy consumption dataset. The gray lines in Figure 5.3 show the actual energy consumption of appliances for 24 hours while the black lines highlight their mean energy as to represent a common pattern. These three profiles categorise a household energy consumption patterns into low, medium and high energy consumption patterns. Therefore, the extracted profiles are evaluated based on their shape as well as their average consumption.

It is evident from all the three load profiles that most of the energy consumption activities in the household are concentrated during morning and evening times. However, the major difference lies in the variation of the energy consumption. For instance, load profile 1 has high variations during consumption as compared to load profile 3 which has less sudden changes in its average energy consumption. These variations are further estimated using statistical analysis of the extracted load profiles. These statistics include measures of spread and central tendency. The purpose for the measures of spread is to estimate how much the energy consumption values differ from the average values and they include standard deviation. On the other hand, central tendency measures calculate the mean of average energy consumption values and these include mean, median and quartiles. The quartiles are estimated in 25, 50 and 75 percentiles. Table 5.2 provides values for the statistical measures for day wide load profiles such as the standard deviation, mean, median and quartiles.

Table 5.2: Day-wide load profiles statistical estimation

Profile	**Standard Deviation**	**Mean**	**Median**	**Quartiles (25, 50, 75)**
Load Profile 1	206.9	2.164	2.211	[1.075, 2.211, 2.982]
Load Profile 2	225.9	10.681	10715.4	[9.812, 10.715, 11.412]
Load Profile 3	237.2	1.250	1193.1	[0.595, 1.193, 1.979]

By analysing the energy consumption of individual appliances in each load profile, it is possible to categorise the activities of the energy consumers. Table 5.3 provides additional information about the consumers activities in each load profile by listing individual appliances and average energy consumption. The energy consumption dataset contains information about the on and off events of the individual appliances. These events show how many times and for how long an appliance was used during the day. Therefore, Table 5.3 indicates the percentage use of appliances in each profile. The activities associated with appliances are categorised as cooking, ICT, laundry, heating/cooling, home lighting and entertainment. Over the recorded period, the occupants in the household performed a range of activities instead of few particular ones based on energy consumption. For instance, profile 1 has the highest share of the usage percentage of the appliances and is associated with most of the activities while consuming less average energy.

Table 5.3: Grouping of appliances found in extracted load profiles (UK Dale Dataset)

Activity	**Appliances**	**Profile 1**	**Profile 2**	**Profile 3**
Cooking	Fridge	48.9	16.6	34.5
	Kitchen lights	60.7	18.8	20.5
	Dishwasher	34.2	15.1	50.7
	kettle	100.0	0.0	0.0
	Microwave	71.4	14.3	14.3
ICT	Htpc	69.4	8.3	22.2
	Data logger pc	67.7	32.3	0.0
	Office pc	33.3	0.0	66.7
	Lcd office	100.0	0.0	0.0
Laundary	Washing machine	53.7	12.8	33.6
	Iron	7.7	30.9	61.3
	Hoover	20.5	58.9	20.5
Heating/Cooling	Boiler	84.1	6.3	9.5
	Solar thermal pump	65.2	30.4	4.3
	Fridge	48.9	16.6	34.5
Lighting	Lighting circuit	45.7	13.9	40.4
	Bedroom lamp	100.0	0.0	0.0
	Livingroom lamp	60.0	0.0	40.0
Entertainment	Tv	71.4	14.3	14.3
	Amp livingroom	100.0	0.0	0.0
Avg. Consumption (kwh)		1.8	10.7	3.7

In addition, profiles 2 and 3 contain activities associated with the appliances having less shares of their usage in overall consumption. However, profile 2 has highest amount of average energy consumption although it has the lowest usage percentage share. The reason for the increased energy consumption is because the appliances ran for certain time periods with the maximum operating mode. Figure 5.4 illustrates the distribution of the energy consumption profiles over the days in a week. The figure shows that each profile is active over a certain period of the week. For example, load profile 1 shown with the black line is more active at the start of the week while load profile 3 shown with the green line increases power consumption during the weekend. However, both profiles behave similarly during the week. On the other

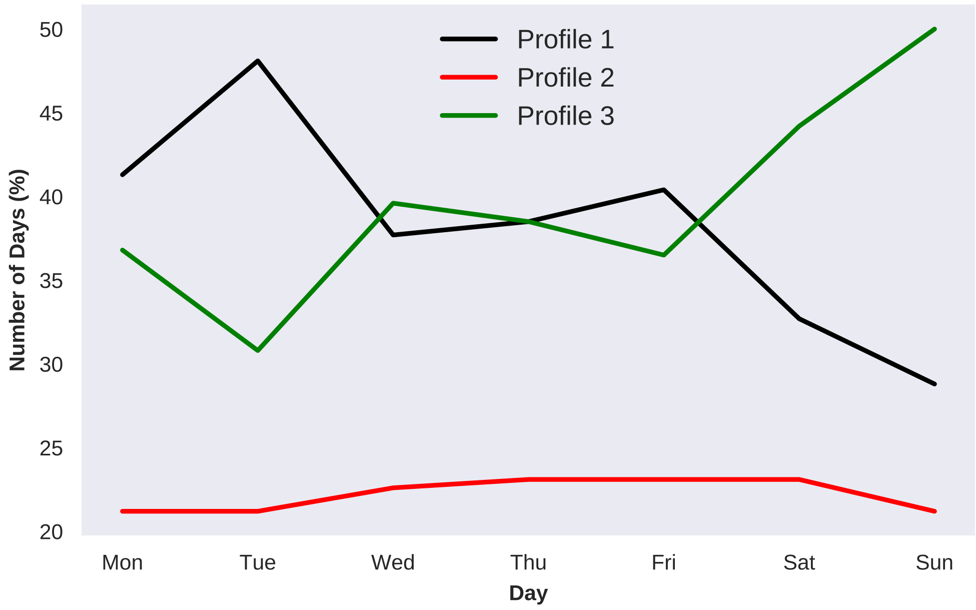

Figure 5.4: Distribution of consumption profiles over the week from Home 1 (UK Dale Dataset)

hand, load profile 2 shows an active trend during the weekdays while having a low percentage around the weekend.

The Deep COLA week-wide profiling revealed more consistent usage of profiles in terms of appliance-to-appliance associations over time as shown in Figure 5.5. Similarly, in this figure, the actual energy consumption over a week is represented by the gray lines and their average energy consumption is represented by the black lines. These profiles show commonly occurring patterns. Similarly, week-wide load profiles are also used for the statistical analysis which includes the measures such standard deviation, mean, median and quartiles. The following Table 5.4 describes these statistical measures for week-wide consumption profiles.

Table 5.4: Week-wide load profiles statistical estimation

Profile	**Standard Deviation**	**Mean**	**Median**	**Quartiles (25, 50, 75)**
Load Profile 1	2435.2	17.279	11.440	[6.326, 11.440, 26.019]
Load Profile 2	6516.5	127.9	125.01	[115.66, 125.01, 135.42]
Load Profile 3	2706	27.375	22.262	[9.92, 22.26, 38.99]

The activities associated with the week-wide profiles are illustrated in Table 5.5. The appliances contained in the day-wide profile 1 show a higher percentage of usage in the week-wide profile 1. Overall, the week-wide energy demand from various appliances varied and there

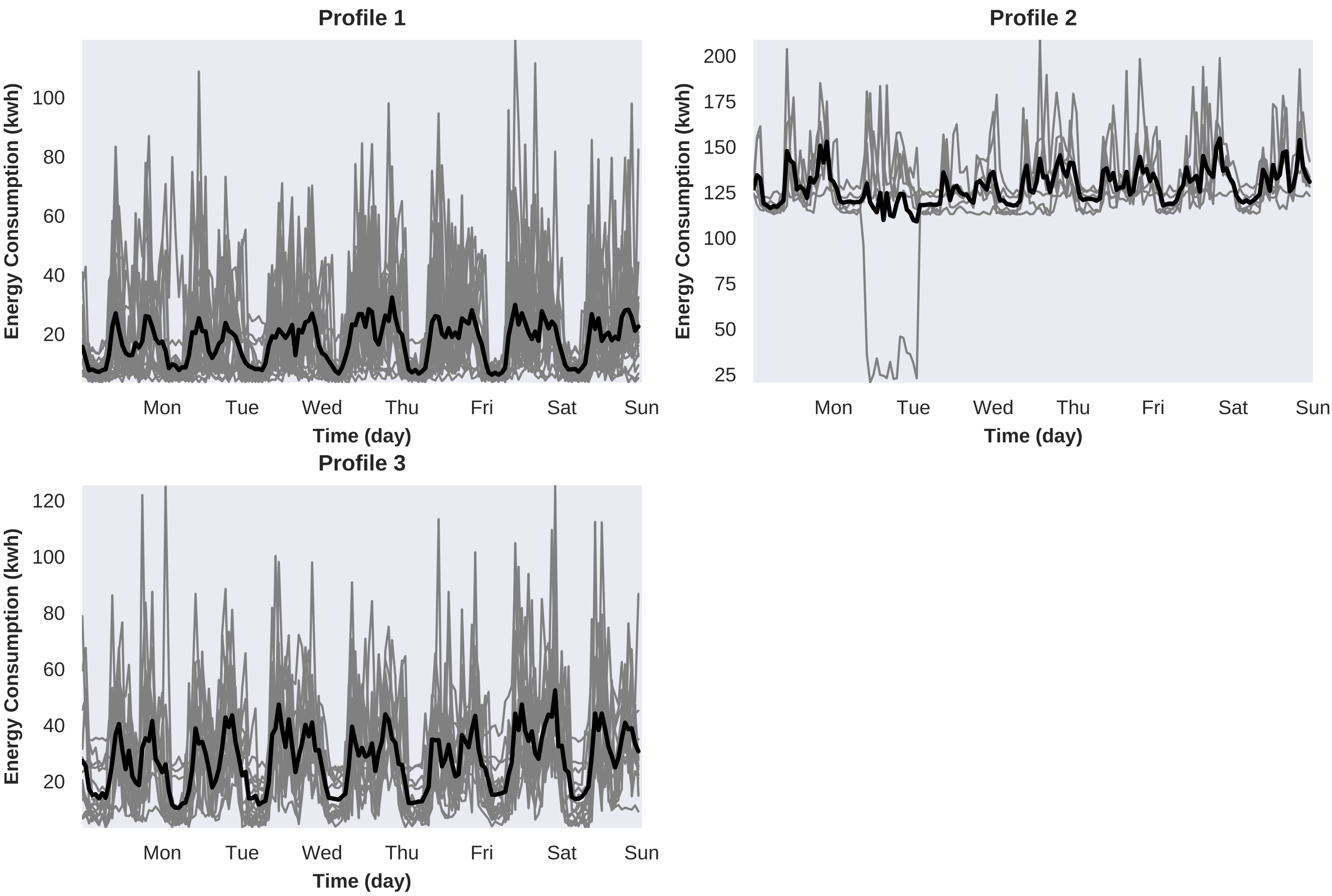

Figure 5.5: One week consumption profiles from Home 1 (UK Dale Dataset)

was a slightly different view of the weekly behavioural activities. Profile 2 has a larger demand than any of the other profiles because its activities were related to the laundry that operated during the peak hours over particular days of the week. Moreover, the constant use of the fridge, in synergy with the use of a home boiler system, contributed significantly to the higher energy demand. On the other hand, the lowest consumption out of profile 1 was related to low energy demand appliances, mainly lamps, throughout various rooms in the house.

5.4.2 Month-wide Energy Consumption Profiling (Smart Star Dataset)

Figure 5.6 provides a month-wide representation of the consumption profiles for each household in the Smart Star dataset, as computed by the Deep COLA clustering approach. As evidenced, there are 4 profiles that can be clearly distinguished in Homes A, B, C and D in terms of power

Table 5.5: Grouping of appliances found in week-wide load profiles (UK Dale Dataset)

Activity	**Appliances**	**Profile 1**	**Profile 2**	**Profile 3**
Cooking	Fridge	57.7	11.4	31.9
	Kitchen lights	67.2	0.0	32.8
	Dishwasher	56.4	7.4	36.2
	kettle	67.3	2.0	30.6
	Microwave	82.4	0.0	17.6
ICT	Htpc	52.6	15.8	31.6
	Data logger pc	75.0	15.9	9.1
	Office pc	50.0	0.0	50.0
	Lcd office	81.8	9.1	9.1
Laundary	Washing machine	49.1	12.9	38.0
	Iron	29.4	49.4	21.2
	Hoover	37.1	42.9	20.0
Heating/Cooling	Boiler	66.0	10.0	24.0
	Solar thermal pump	80.4	2.3	9.3
	Fridge	56.7	11.4	31.9
Lighting	Lighting circuit	47.0	9.5	43.5
	Bedroom lamp	45.9	0.0	54.1
	Livingroom lamp	40.0	0.0	60.0
Entertainment	Tv	81.2	6.2	12.5
	Amp livingroom	0.0	0.0	100.0
Avg. Consumption (kwh)		17.3	127.9	27.4

consumption (i.e. Kwh).

In general, the distribution of the samples for each profile, as well as the resulting average energy consumption ranges throughout all the households, suggests 4 prevalent profiles, where profiles 1 is observed to be similar in all households A, B, C and D. However, for each house, the most dominant profile varies, since the behavioral usage of individual appliances by the household occupants is different in most cases. In parallel, each profile for a given household results from a combined view over the appliances that contributed towards a given energy consumption value over the time, which may be correlated with some environmental aspects (e.g. humidity). In particular, Home A and C belongs to Profile 1 and 3. These Homes

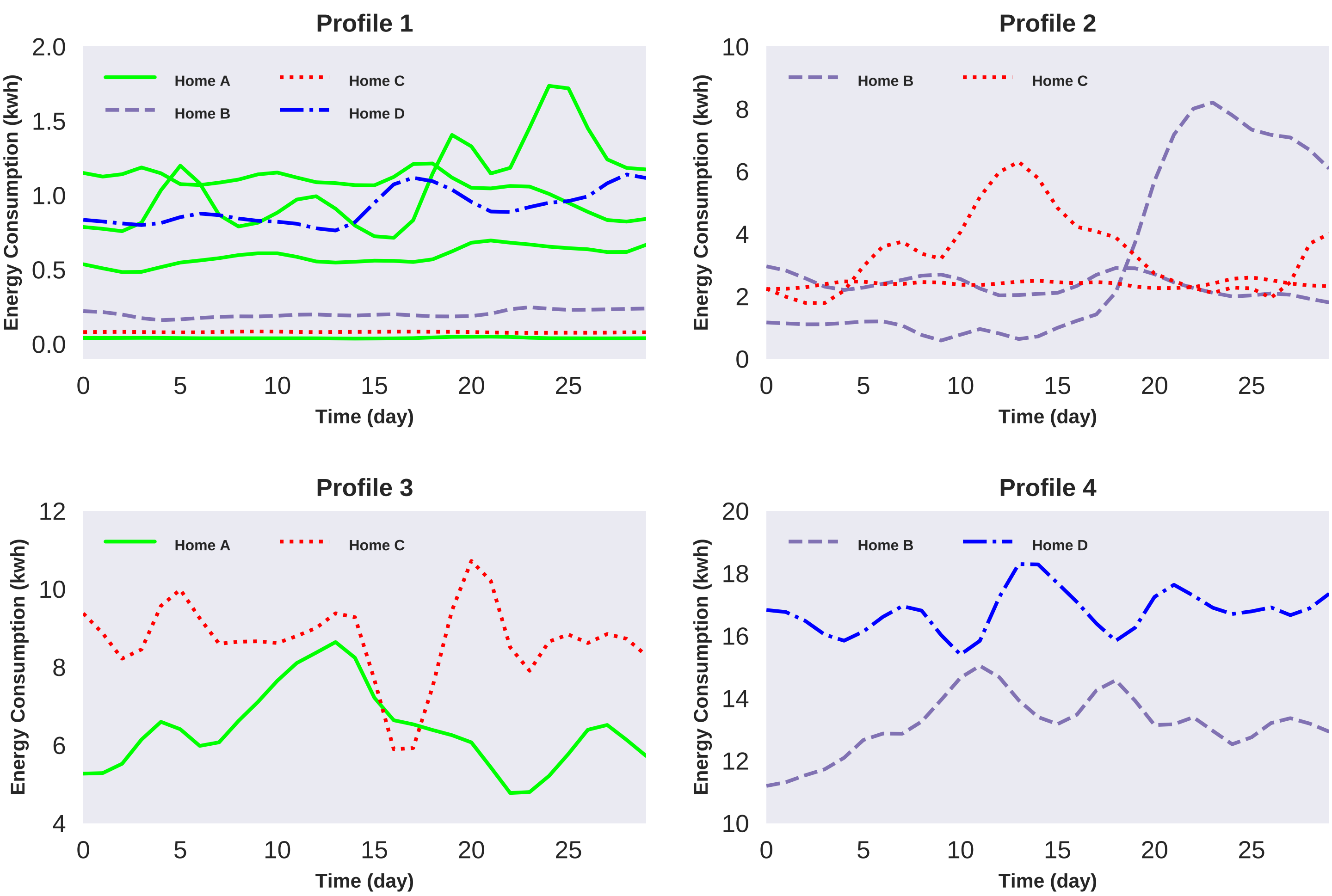

Figure 5.6: Profiles in 4 Smart Star homes based on the lowest, medium and highest energy consumption clusters.

appear to share similar energy consumption characteristics. On the other hand, Home B is slightly different as it follows Profile 4 instead for Profile 3. These Profiles show various levels of average energy consumption and assist in planning demand response programmes for particular households. In the same manner, statistical measures are calculated for the month-wide energy consumption load profiles. These statistical measures are standard deviation, mean, median and quartiles which assist in better understanding of spread and central tendency. The load profile 2 has the highest values for standard deviation, mean and median due to the large energy consumption. The following Table 5.6 illustrates the values of these measures.

Hence, we demonstrate next that the produced profiles can assist in composing associations (see Table 5.7), with respect to appliance-to-appliance, appliance-to-time, as well as appliance-

Table 5.6: Week-wide load profiles statistical estimation

Profile	Standard Deviation	Mean	Median	Quartiles (25, 50, 75)
Load Profile 1	0.091	0.564	0.5806	[0.131, 0.580, 1.059]
Load Profile 2	0.138	14.96	14.98	[13.65, 14.98, 16.34]
Load Profile 3	0.0006	7.53	7.50	[6.41, 7.50, 8.60]

to-environment.

Table 5.7: Month-wide appliance-to-appliance association in the Smart Star Dataset

Profile	House	Appliances				
1	Home A	FurnaceHRV BedroomLights	Microwave	DishWasher	Fridge	Dryer
	Home B	FurnaceHRV BedroomLights	Microwave	Dishwasher	Fridge	Dryer
	Home C	FurnaceHRV BedroomLights	Microwave	Dishwasher	Fridge	Dryer
	Home D	FurnaceHRV BedroomLights	Microwave	Dishwasher	Fridge	Dryer
2	Home B	BedroomLights Dryer	Fridge	Dishwasher	Microwave	FurnaceHRV
	Home C	Dryer FurnaceHRV	BedroomLights	Microwave	Fridge	Dishwasher
3	Home A	Dishwasher FurnaceHRV	Fridge	Dryer	Microwave	BedroomLights
	Home C	Dryer FurnaceHRV	Microwave	Dishwasher	BedroomLights	Fridge
4	Home B	Dishwasher FurnaceHRV	Microwave	Dryer	BedroomLights	Fridge
	Home D	Dishwasher FurnaceHRV	Microwave	Dryer	BedroomLights	Fridge

5.4.3 Energy Consumption Associations

5.4.3.1 Appliance-to-Appliance

Table 5.7 demonstrates the relationship between appliance-to-appliance, which is associated with each profile generated from the month-wide clustering presented in Figure 5.6. Each sequence of appliances from left to right in the table depicts the order in which they are most likely to be operated in each home. Profile 1 bears similar characteristics for each home in which the appliances have the same associations. This indicates that this profile is associated with a consumer who often performs daily activities in a similar fashion. For instance, in Home

A, B, C and D, the consumer is likely to spend more time throughout the whole month in the kitchen, due to the appliance associations (e.g., microwave, dishwasher and fridge). Based on the appliance-to-time analysis associated with this profile, it was also revealed that when the microwave is being operated it is highly likely that simultaneously, or within a few minutes, the consumer may use the dishwasher and during this time the fridge is operating at high mode as well. We have also identified that the parallel and highly frequent use of the FurnaceHRV appliance, which is responsible for kitchen ventilation purposes, indicates the cooking habits of a consumer. On the other hand, the homes associated to profiles 3 and 4 have a better internal environment due to the less use of the FurnaceHRV, so there is the minimal use of ventilation while cooking.

Moreover, throughout all the other identified profiles, consumer behaviour is relatively varied in comparison to profile 1. Thus, there is less similarity in terms of domestic habits.

5.4.3.2 Appliance-to-Time

Figure 5.6 complements Table 5.7 since it demonstrates a high-level view of the month-wide maximum consumption value distribution over the four profiles that are mapped as energy consumption during the month for each household. Hence, a more general appliance-to-time association may be profiled based on the most prevalent profiles in the four aforementioned consumption categories. For instance, profile 1 is present in home A, B, C and D, and is regarded as a very low energy profile, with consumption around $0.8kw/h$ for the whole month. Similarly, homes A, and C share similar consumption characteristics of around $8kw/h$ and are associated with profile 3. The homes B and D that consume energy of $16kw/h$ and beyond belong to a high energy profile. Figure 5.6 indicates that Profile 1 belongs to all four Homes (A, B, C and D). These profiles illustrate variations in energy with respect to time.

5.4.3.3 Appliance-to-Environment

The appliance-to-environment association is identified by correlating environmental inputs (e.g. humidity, temperature) with the actual energy utilisation values in each profile. Through our clustering algorithm, we have identified that profile 1 has a higher relevance with the environment and can provide an insight in terms of internal household environmental conditions. We have identified that FurnaceHRV has more significance as compared to profile 3 and 4. FurnaceHRV is used mainly for ventilation purposes while a consumer cooks as it directly balances the humidity level, which is inversely proportional to the air temperature in a given household. Naturally, extremely low or high temperatures will cause an interruption in the humidity level, which will require FurnaceHRV to consume more energy to adjust the humidity level.

5.4.4 Deep COLA Performance Comparison with Other Techniques

In this section, Deep COLA performance is compared with existing techniques which are K-means, DBSCAN and SOM. These are used to extract load profiles from the energy consumption dataset. These algorithms have been implemented in Python and their parameters have been selected using grid search criteria. In particular, the K-means algorithm is implemented using the Python Sklearn package. The K-means algorithm stops training when centroids have stabilised or a given number of iterations have reached. The number of centroids, maximum number of iterations and initial algorithm are selected as 4, 1000 and k-means++ respectively. The DBSCAN algorithm is a density based clustering technique which does not require a number of clusters in advance. The other parameters of the DBSCAN algorithm are Eps and Min Samples. The parameter Eps defines the maximum distance between two sample points in order to make a cluster. The parameter Min Sample represents the minimum number of sample points in a cluster. The Eps and Min Sample in DBSCAN algorithm are selected as 0.5 and 5 respectively. The SOM algorithm is also based on the concept of competitive learning, however, the number of clusters is fixed. The SOM algorithm is implemented using the Python MiniSOM package. The size of the map, learning rate and radius in the MiniSOM package are selected as (4,4), 0.01 and 0.1 respectively. For the purpose of comparison, the proposed algorithm Deep COLA, and the other algorithms that are K-Means, DBSCAN and SOM have been evaluated on the same hardware. The system is Linux based with Python 3 as the programming language and has the specifications as: 64 GB RAM, Intel Core i9 3.60 GHz, and Graphic Card: NVIDIA Quadro 4GB.

Table 5.8 presents a comparison between Deep COLA against the commonly used K-means, DBSCAN and SOM schemes, in term of their computational costs over time. It is evident from the table that, in comparison to the rest, Deep COLA requires less time to compose profiles across all the datasets. In contrast with our scheme where profiles are created adaptively, K-means and DBSCAN algorithms require additional processing to interpret and validate the clusters' consistency (e.g. with the use of k-dist, Elbow or Silhouette metrics) in order to specify the number of clusters. Moreover, Deep COLA inherits deep neural network properties, thus it is much more efficient at handling larger datasets than the other approaches.

Table 5.8: Computational time (seconds) comparison among Deep COLA, K-means, DBSCAN and SOM

Dataset	Homes	Consumption Period	No. of Records	Deep COLA	K-means	DBSCAN	SOM
UK-DALE	1	1 Year	12560	164	196	1500	230
UK-DALE	1	4 Year	84168	750	900	2100	1120
Smart Star	4	3 Year	17521	200	260	430	540

Evaluation of Deep COLA performance using Dunn, Silhouette and Calinski-Harabasz indexes

The quality of the clusters produced by an algorithm mainly depends on higher inter-cluster separation and lower intra-cluster separation. For the evaluation of the Deep COLA performance, three commonly used validity indexes are selected: Dunn index (DI), Silhouette index

(SI) and Calinski-Harabasz index (CI). The Dunn index is used to identify groups of clusters that maximise inter-cluster separation and minimise intra-cluster separation. The Silhouette index allows us to compare the cohesion and separation of clusters. The Calinski-Harabasz index is based on the idea of ANOVAs that allows us to estimate variations among and between clusters. The Calinski-Harabasz index is mostly suitable for spherical clusters.

Figure 5.7 illustrates a comparison of distance measures and index metrics. The x-axis represents the experiment no. and y-axis represents the index value. The sub-figures in Figure 5.7 depict the graphs of the three indexes, DI, SI and CI, using the Euclidean distance and Dynamic Time Warping (DTW). The values of indexes DI, SI and CI mainly depend on the random initial weights selected at the beginning of the algorithm training. The drops of index values at 4th and 5th experiments in first and second sub-figure represent that the initial weights in the algorithm did not provide the ability to lower the optimisation error during training and as a result the clusters are not formed properly. The internal architecture of the deep neural network affects the accuracy of the clusters and the indices. Therefore, initializing the weights using Xavier technique and avoiding overfitting through number of nodes assists in improving indices.

Figure 5.8 provides a visual representation in terms of bar graph of Deep COLA's comparison against the other algorithms selected earlier, based on clustering accuracy and computational cost. On the x-axis of the top chart in Figure 5.8, the index metrics (DI, SI and CI) are shown, while the y-axis represents the values of the indexes. A higher value of the indexes DI, SI and CI represents better clustering. The CI index values are normalised to show them on the graph. The blue colour represents the index values obtained using the Deep COLA algorithm. The index metrics DI, SI and CI show the better clustering performance of Deep COLA, compared with the other algorithms. These indices are internal evaluation techniques for the clusters, which are based on the dataset itself. The internal evaluation is carried out using two

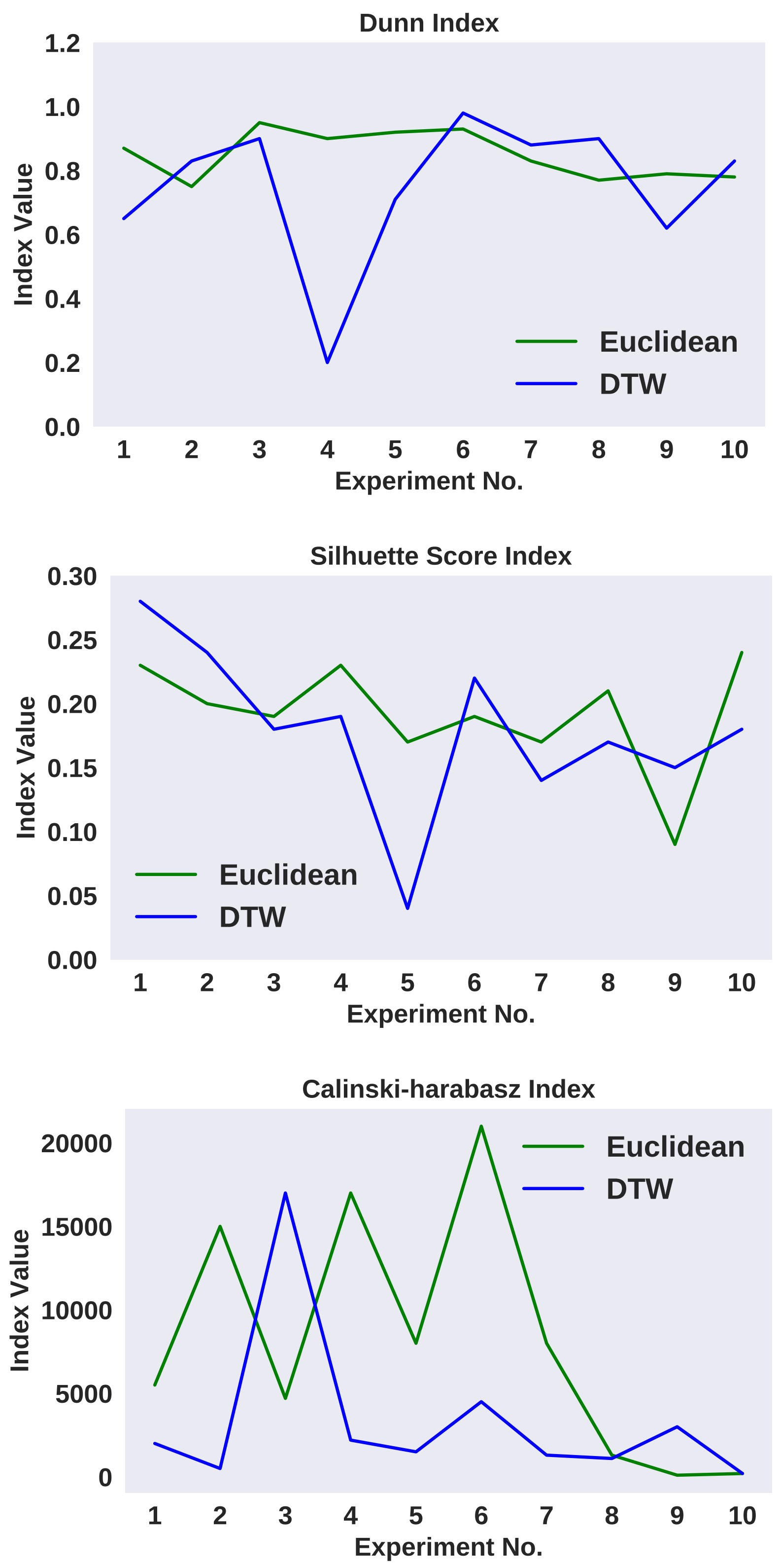

Figure 5.7: Distance measures and index metric comparison

characteristics of the clusters, i.e. the compactness and good separation of the clusters.

The DeepCOLA algorithm identifies these two characteristics using the capabilities of deep representational learning, in which data representations are extracted on a layer-by-layer basis.

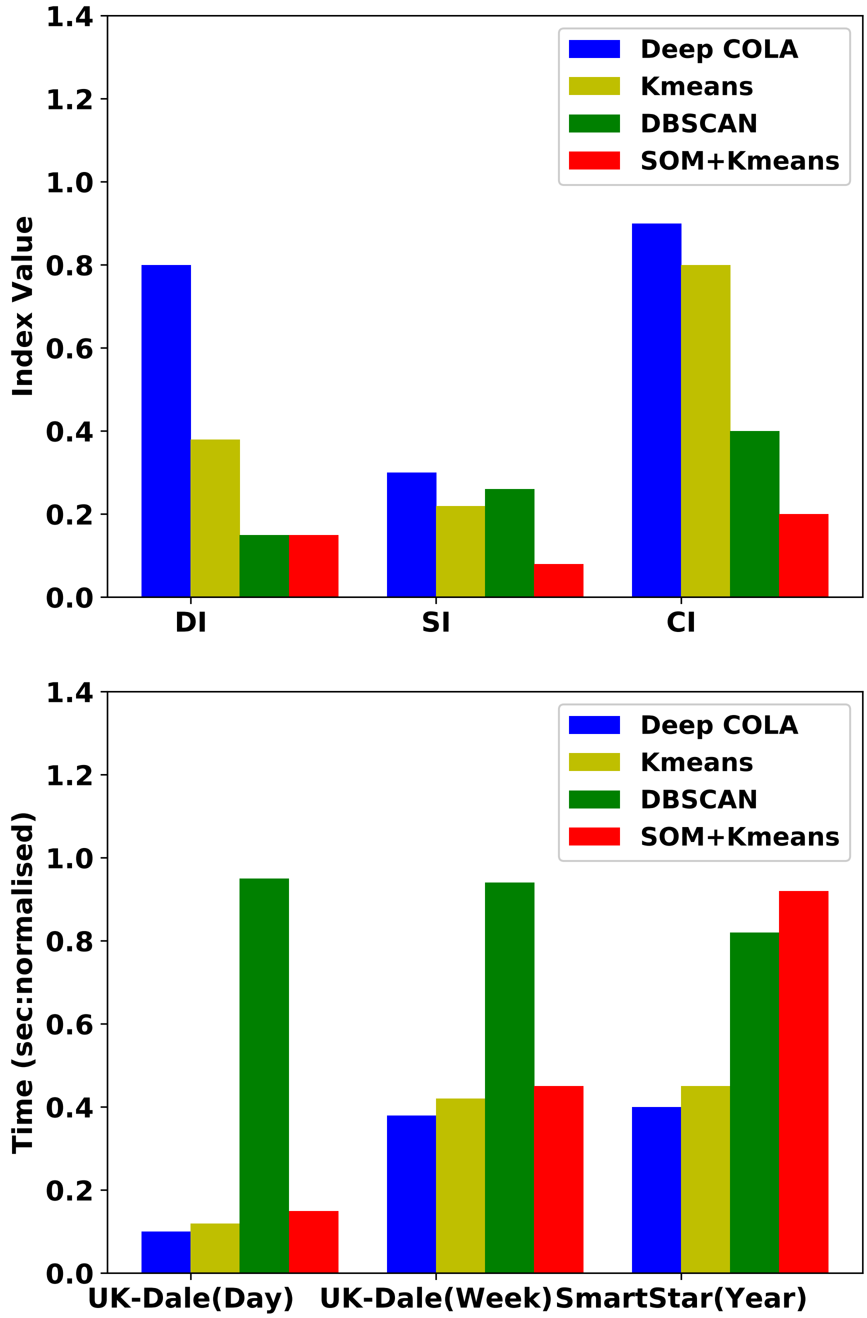

Figure 5.8: Clustering accuracy and computational time comparison of Deep COLA, K-means, DBSCAN and SOM

The compactness and separation characteristics are essential for the efficient profiling of energy consumption of individual appliances. The KMeans algorithm performs clustering on the original dataset, which contains a large number of features. Due to the high dimensionality of the data, KMeans requires a high computational cost in order to extract load profiles. Additionally, in the high dimension, all the data points become similar to each other. However, the Deep COLA algorithm is based on the concept of first reducing the dimensionality of data and then applying competitive learning to find clusters. Therefore, the dimensionality reduction and clustering loss are performed and thus better clusters are found in low dimensional datasets. The bottom chart in Figure 5.8 describes the processing time taken by the algorithms. The Deep COLA algorithm presented by bar with blue color in Figure 5.8 required less processing time as compared to other algorithms for load profiling.

Table 5.9: Deep COLA evaluation

Exp.	Layers	No. of Neurons	No. of Clusters	Distance Measure	DI	SI	CI
1	3	[43, 50, 1]	7	Euclidean	0.87	0.24	6230
			7	DTW	0.66	0.27	1451
2	4	[43, 45, 45, 1]	7	Euclidean	0.77	0.20	15146
			2	DTW	0.83	0.24	4.60
3	4	[43, 50, 50, 1]	11	Euclidean	0.95	0.19	4317
			20	DTW	0.89	0.18	17072
4	4	[43, 55, 45, 1]	9	Euclidean	0.89	0.24	16873
			12	DTW	0.34	0.18	1633
5	4	[43, 55, 50, 1]	12	Euclidean	0.93	0.18	8504
			2	DTW	0.73	0.05	3.2
6	4	[43, 55, 55, 1]	6	Euclidean	0.93	0.19	20856
			5	DTW	0.99	0.21	4837
7	5	[43, 30, 30, 30, 1]	22	Euclidean	0.82	0.18	10258
			4	DTW	0.86	0.15	788
8	5	[43, 40, 40, 40, 1]	4	Euclidean	0.74	0.20	1616
			6	DTW	0.89	0.17	610
9	5	[43, 45, 45, 45, 1]	3	Euclidean	0.74	0.09	11
			6	DTW	0.62	0.15	3066
10	5	[43, 50, 50, 50, 1]	5	Euclidean	0.72	0.23	80
			5	DTW	0.81	0.17	170

Table 5.9 represents the parameter configurations and index values only for the Deep COLA algorithm. Our experiments have highlighted the effect that a different number of layers and neurons in each layer has on the accuracy of the clusters. Initially, the experiments were conducted using 3 layers in the Deep COLA algorithm and index metrics (DI, SI and CI) were calculated. The three indices show a decreasing trend, while increasing the number of layers. The maximum DI, SI and CI are required to obtain compact clusters. Therefore, experiment

no. 6 provides the most accurate clusters with four layers in the network.

Further Comparison of Deep COLA with other Algorithms

In this section, the clustering performance of the Deep COLA algorithm is compared further against KMeans, DBSCAN and SOM. SOM is used as a hybrid algorithm with the KMeans for the purpose of dimensionality reduction.

The process of clustering energy consumption using these algorithms involves first applying an algorithm to the original dataset, which consists of all feature vectors, in order to obtain a low dimensional feature vector. The purpose of applying the SOM algorithm for dimensionality reduction is that in the higher dimensions, the KMeans performance starts degrading due to the use of the Euclidean distance. As a result, the distance converges between two data points in extremely high dimensions and the minimum and maximum distances become equal. Therefore, clustering is performed on a low dimensional feature vector using an algorithm such as KMeans or Hierarchical clustering. The KMeans algorithm is used with the SOM algorithm to cluster individual household energy consumption. SOM requires more parameters to specify the dimensionality of the final reduced feature vector. Additionally, the KMeans and Hierarchical clustering schemes also require the number of clusters to be specified. Table 5.10 illustrates a comparison between Deep COLA and the other algorithms. For the Deep COLA algorithm, both distance measures (Euclidean distance and DTW) are used, while other algorithms are assessed using the Euclidean distance only.

Table 5.10: Deep COLA comparison with other algorithms

Algorithm	Clusters (#)	Distance	DI	SI	CI
Deep COLA	5	Euclidean	0.80	0.35	1746
	5	DTW	0.99	0.21	4837
KMeans	9	Euclidean	0.39	0.23	1676
DBSCAN	3	Euclidean	0.17	0.30	691
SOM + HClust	9	Euclidean	0.03	0.03	396

The results in Table 5.10 illustrate that the DTW slightly improves the DI and SI index metrics. However, SI shows reduced performance while using DTW. The load profiles do not show a linear trend over time and depend upon various factors in the household, such as user daily activities, environment and economic factors. The distance measures, i.e. Euclidean and DTW, compute the similarity between load profiles in different ways. For instance, the Euclidean distance metric is based on the assumption that the data points in the load profiles are linearly identical and bear an exact match. On the other hand, the DTW distance metric does not assume a linear match between the load profiles and learns the underlying patterns for calculating the similarity. In order to group similar appliances on a day-to-day basis, the DTW distance measure provides an efficient way to calculate correlation among daily activities. For example, the time of use of appliances, e.g. toaster, microwave oven, TV, laptop, etc., may vary every day in the household based on the user's preferences. As a result, this will have an effect on the energy consumption in terms of shifting and translating load profiles. The DTW distance measure reduces the effect of shifting and translation of energy consumption in time. The DTW performs elastic transformations in order to classify the load profiles with similar characteristics.

5.5 Parallelization Experiments and Results Analysis

The experimental set-up consists of a deep neural network with multiple hidden layers which is trained using the Apache Spark unified analytics framework. The experiments were conducted using four systems to form a data analytic cluster. The Linux operating system was used on all these four systems to implement the deep learning models. The specifications of the four systems are (System 1: 16 GB RAM, Intel Core i7 3.60 GHz, Graphic Card: NVIDIA GeForce 4GB, System 2: 16 GB RAM, Intel Core i7 3.40 GHz, Graphic Card: NVIDIA Quadro 1GB, System 3: 64 GB RAM, Intel Core i9 3.60 GHz, Graphic Card: NVIDIA Quadro 4GB, System 4: 16 GB RAM, Intel Core i7 3.60 GHz, Graphic Card: 2GB). The Figure 5.9 illustrates the master and workers setup in the cluster for parallelisation. The Apache Spark framework works

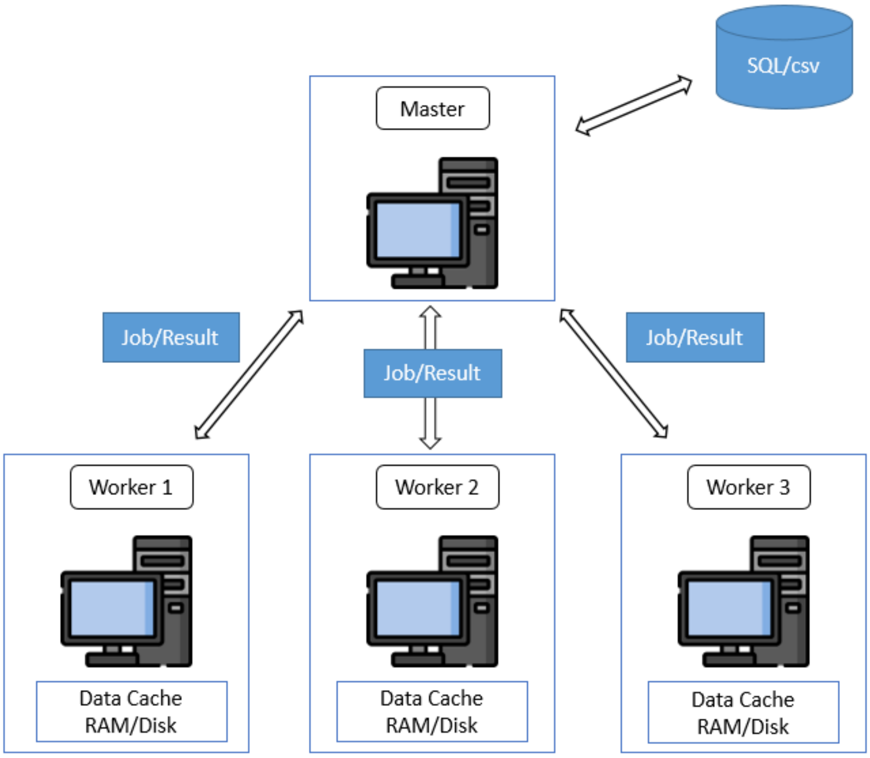

Figure 5.9: Hardware setup of cluster for deep neural network parallelisation

in a client-server architecture. Therefore, one system in the experimental setup acts as a master node while the other three systems work as slave nodes. The proposed deep learning model has been implemented on these systems using Python 3.

The household electricity consumption dataset (UK-Dale) is used to predict energy consumption for the next day. The dataset consists of 4 years of energy consumption at a 6 seconds rate. The major challenges in the existing parallelisation approaches are related to the computational complexity of deep neural networks and the synchronisation of the hyper-parameters which affect their performance. The performance of a deep neural network depends upon the time required to access the data and perform computations. The proposed deep neural networks solved these challenges using novel techniques. The data is distributed using HDFS, which efficiently stores the data in the Hadoop-native file format and parallelizes in a cluster. The HDFS architecture performs processing on the data while keeping it in the storage which requires increased input-output (I/O) queries. The data access from storage is usually slower as compared to the in-memory data access. Therefore, the proposed approach utilized the Apache Spark for designing our deep neural network. The Apache Spark performs analysis on the data while keeping it in memory which improves performance as compared to the disk-based Hadoop map-reduce model [134]. Another challenge is related to updating weights of the network after performing certain training iterations. The slower worker nodes in the cluster cause delay during updating weights. Therefore, our proposed deep neural network algorithm is based on the lock-free synchronization for the parameters' updates. This improves the performance of the algorithm and reduces the convergence time during the inference as shown by the authors in [135]. Table 5.11 indicates the parameter configuration of the model.

Table 5.11: Deep Neural Network Configurations

Parameter	Configuration
Hidden layers	3
Optimization	Adam
Learning rate	0.01
Objective function	ReLU

5.5.1 Results and Evaluation

This section explains the results obtained during the experiments conducted using the deep neural network and Apache Spark. The energy consumption data is partitioned across multiple worker threads and each worker thread performs processing on the data and computes RMSE for the next day load forecasting. For comparison, Table 5.12 indicates the processing time in seconds and the accuracy of the deep neural network in terms of RMSE for a single partition (i.e. no partition). From this table, it is evident that without parallelization, the proposed algorithm takes long processing time for load forecasting and encounters a high RMSE error.

Table 5.12: Processing time and accuracy with single partition

Partition	Processing time (sec)	RMSE
1	255	10.80

In contrast, Table 5.13 indicates the processing time and accuracy of the deep neural network with 3 partitions.

Table 5.13: Processing time and accuracy with 3 partitions

Partition	Processing time (sec)	RMSE
1	154	7.30
2	120	6.10
3	135	7.40

It can be observed from Table 5.13 that the RMSE values are improved when the data is parallelized across multiple worker threads during the training of the deep learning algorithm. The time required to process and predict energy consumption has also reduced. The reason is that each worker thread has to process less information in a parallelized fashion as compared to the single partition. The worker threads perform processing on the data without waiting for the other worker threads to finish their jobs and update the model parameters in a lock-free loosely synchronized way. Moreover, Table 5.14 indicates the processing time and accuracy of the deep neural network with 5 partitions.

Table 5.14: Processing time and accuracy with 5 partitions

Partition	Processing time (sec)	RMSE
1	98	5.20
2	120	3.40
3	89	3.20
4	78	4.20
5	80	2.30

In this case, the data has been split across 5 worker threads in order to assess the improvements in the processing time and RMSE. The algorithm provides improved RMSE and processing time.

To further investigate the trend of the algorithm performance vs. the number of partitions, Table 5.15 shows the case of 7 partitions. Clearly, the efficiency of the algorithm for load forecasting depends upon the trade-off between the number of partitions and the amount of

Table 5.15: Processing time and accuracy with 7 partitions

Partition	Processing time (sec)	RMSE
1	65	4.10
2	55	3.70
3	67	3.40
4	45	2.90
5	57	3.50
6	78	3.30
7	72	4.60

data. From Table 5.15, it can be observed that though processing time has reduced, there are fewer improvements in the RMSE values. The reason is that individual worker threads get less data that does not possess all the characteristics of the whole dataset and are thus unable to untangle hidden patterns. A similar situation is observed in Table 5.16 with 9 partitions.

Table 5.16: Processing time and accuracy with 9 partitions

Partition	Processing time (sec)	RMSE
1	50	6.30
2	45	7.30
3	55	6.50
4	67	5.20
5	44	8.80
6	39	9.90
7	37	10.60
8	34	11.10
9	38	12.70

5.6 Summary

STLF is an important task in the context of optimal energy management for both energy providers as well as individual consumers [136]. A granular and accurate approach for composing STLF is to consider the measurements gathered by each appliance in a given household.

Nonetheless, the dynamic, customer-specific behaviour of power utilisation over different appliances as well as their indirect relationships with exogenous features (e.g. weather, humidity, etc.) pose a challenge to any STLF scheme and a method to capture these properties is required. This chapter introduces the applicability of the two models based on the FF-DNN and R-DNN algorithms for short term bulk power load forecasting. A new method is also presented in chapter 3 that enables the extraction of features from the original raw power measurements by exploiting the joint time-frequency (TF) representation of the load signals. Consequently, as shown in the experimental outcomes of this work, our proposed method allows us to model the most dominant factors that affect the power consumption patterns. Based on the two case studies presented herein, we have shown that the weather, time, holidays, lagged load and data distribution over the consumption period are found to be the most dominant factors.

The efficient profiling of appliance-level energy consumption in residential households is anticipated as a fundamental building block in future HEMSs [137]. In this chapter we have stated that the profiling should be done in the context of appliance-to-appliance, appliance-to-time and appliance-to-environment associations. We have thus applied Deep COLA to profile appliance-level energy consumption. Through our evaluation over the real datasets introduced in Chapter 4, we have demonstrated the superiority of Deep COLA in comparison with the commonly used methods. This is achieved by avoiding the a-priori selection of the cluster number in our scheme, whilst being capable of composing accurate and computationally optimal day, week and year-wide appliance-level consumption profiles. Given the clustering enabled by Deep COLA, we have also demonstrated the feasibility of identifying end-user habitual insights over the three aforementioned associations to aid in developing various demand response programmes. We hope that Deep COLA will become a core element within future HEMSs.

The parallelization of deep learning models is able to process large amounts of information using less computational resources and processing time, which is essential to meet future SG demands. The proposed algorithm utilizes a data-based parallelization technique and predicts energy consumption in terms of processing time and RMSE. The data is partitioned across 3, 5, 7 and 9 worker threads, and the improvements in time and RMSE (in most of these partition cases) have been observed. The analysis indicates that the parallelization approach provides better performance and increases the efficiency of the deep learning model. However, a large number of worker threads reduce processing time but provide higher error rates (i.e. RMSE values) for load forecasting. In the following chapter, we will highlight future research directions for load forecasting and human behaviour modelling with the perspective of parallelization and briefly describe limitations of the proposed research.

Chapter 6

Conclusion

With the technological advancements, a SG as a new kind of power grid provides a promising environment for the vast operations of control and conditioning of energy production, distribution and consumption with the help of smart meters, smart appliances at consumer premises and efficient integration and utilisation of renewable resources. The process of energy production and distribution in the power grid is a complex task which requires advance planning and important decision making in a stochastic environment. The decisions are made to improve the performance of the energy systems in order to deliver sufficient, stable and reliable electricity to end consumers.

The power system operators and energy market agents make the planning of the power systems by assessing the consumers' energy demands and understanding of their usage characteristics. The consumers' energy requirements are assessed through the prediction of the electricity usage for various time horizons and understanding the consumption characteristics with the help of load profiling. The uncertainty caused by the complex interaction of the processes related to energy production through traditional means, integration of renewable energy resources, increasing trend of electrical vehicles charging as well as the contribution of socio-economic and environmental conditions, heavily affect the accuracy and applicability of the prediction of electricity and consumer profiling. The decision making process in the power grid not only relies on the forecasting of energy demands but system operators and energy market agents also require clear understanding of the end consumers' day-to-day activities related to the energy usage.

Another interlinked challenge which arises in the process of decision making by various stakeholders of the power grid is due to the processing of large amounts of energy consumption data in nearly real-time. Currently, most of the smart meters are capable of recording energy consumption at 6 second intervals, and advanced smart meters are even able to record energy consumption at 1 second intervals. The traditional algorithms for the prediction of electricity usage and consumer behaviour modelling require exponential amounts of processing to analyse huge amounts of data and model underlying interactions of energy generation and consumption.

This research work focused on providing a unified framework which enables energy forecasting for the short-term time horizon and the characterisation of the energy consumption by end consumers in order to reveal complex interactions of their activities. The large amount of energy consumption data collected at the appliance level with 6 second intervals has been utilized for energy prediction and profiling, and the processing time of our proposed approach has been reduced through the parallelisation.

The recently employed demand-response model enabled by the transformation of traditional power grids to the SGs allows energy providers to have a clearer understanding of the energy utilisation of each individual household within their administrative domain. Nonetheless, the rapid growth of IoT-based domestic appliances within each household in conjunction with the varying and hard-to-predict customer-specific energy requirements is regarded as a challenge with respect to accurately forecasting the day-to-day or week-to-week energy consumption demand. Such a forecast is considered essential in order to compose a granular and accurate aggregate-level power consumption forecast for a given household. Therefore, this research work has proposed a new approach in Chapter 3 to assess the applicability of Deep Neural Network for short-term energy forecasting at the distribution level as well as the household level. The proposed Deep Neural Networks are FF-DNN and R-DNN which learn hidden structure in the dataset. The energy consumption data is essentially a time series, therefore, the R-DNN exploits the time dependencies in the energy consumption and accurately predicts load for the short time duration. We have demonstrated its superiority over the past heavily used models in terms of prediction accuracy and computational performance with experiments conducted with the real time energy consumption datasets.

Furthermore, to understand the human behaviour, the adequate profiling of appliance-level energy consumption in residential households is considered a fundamental building block for HEMSs. Thus, we have argued that profiling should be in the context of appliance-to-appliance, appliance-to-time and appliance-to-environment associations. We have thus introduced Deep COLA, a novel Deep Competitive Learning-based Algorithm, in Chapter 3 to profile appliance-level energy consumption. The Deep COLA algorithm is based on the concept of competitive learning. In competitive learning, nodes in the output layer compete with each other and the node with the highest activation wins and updates its weights. The functionality of Deep COLA mainly depends upon two steps: encoding of the higher dimensional input to lower dimensional features and retrieval & association. Through our evaluation over the real datasets, we have demonstrated the superiority of our scheme in comparison with commonly used methods. This is due to our algorithm not requiring the a-priori selection of the cluster number, whilst also being capable of composing accurate and computationally optimal day, week and year-wide appliance-level consumption profiles. Given the clustering enabled by Deep COLA, we have also demonstrated the feasibility of identifying end-user habitual insights over the three aforementioned associations. We hope that Deep COLA will act as a core element within future HEMSs.

The advanced smart meters in SG are capable of collecting huge amounts of energy consumption information from individual appliances at the consumer premises. The analysis of huge amounts of information requires exponential processing time. The proposed approach utilizes data based parallelization to speedup the processing of deep learning models to predict load in the future. In this parallelization scheme, the data is distributed across multiple nodes in the cluster. The deep learning model performs training on each node and updates network parameters on a centralised node. The parameters are updated using a lock-free synchronization procedure which does not slowdown the learning of slower nodes. The data is distributed across 3,5,7 and 9 nodes in the cluster. The results showed that the processing time reduces while increasing nodes, however, there is a trade-off between the number of nodes and the data size. Excessively distributing data across nodes affects the performance of deep learning models and reduces prediction accuracy.

Our unified approach for predicting energy consumption, understanding human behaviour through profiling the energy usage during day-to-day activities and parallelising the processing of large amounts of data enables system operators and energy market agents to efficiently plan the energy generation, distribution and consumption life cycle.

Chapter 7

Future Research Directions and Limitations

7.1 Introduction

Accurate STLF and human behaviour understanding by analysing large amounts of data efficiently plays an important role in developing better demand response programmes and achieving the goals of SG. In previous chapters, we presented the research methodology and the experimentation based on real world datasets which provided high prediction accuracy and assisted in modelling human activities related to energy consumption in households. This chapter provides future research directions which can be carried out to make a step forward for the vision of SG.

7.2 Research Directions for Future Work

The future advanced metering infrastructure in households will be capable of collecting huge amounts of information about various aspects such as energy consumption at appliance level, energy loss and restoration, socio-economic, geographical and environmental information [138]. The activities of consumers in households vary significantly in terms of energy consumption and therefore, it becomes challenging to model the variation and uncertainty in consumer behaviour in order to properly predict and capture the profiles of energy usage [139]. The energy generation through renewable resources is becoming popular due to their low cost and eco-friendly advantages in order to reduce the consumption of fossil fuel [140]. However, the problem arises due to the fact that the energy generation through renewable resources is highly irregular and unpredictable, which increases the complexity of balancing the energy demand and supply [3]. Furthermore, the introduction of electric vehicles and their popularity with the passage of time will significantly increase the energy consumption. Therefore, the research work presented in this thesis can be extended to further improve prediction accuracy by accommodating the information related to uncertainty introduced by renewable resources and charging of electric vehicles. The inclusion of various features has a significant role in improving the accuracy of the load prediction. The research work proposed an approach to extract features using time and frequency domain analysis to model the uncertainty and the variations in the energy consumption information. The future research direction can be to further utilize the signal decomposition methods to identify the peaks in the energy consumption and include them as features in the dataset. The electricity load forecasting and human behaviour understanding in households are an important research area due to large variations in consumer activities.

Additionally, the decisions in the energy market not only rely on the energy load forecasting but also take into account end consumer types and their consumption profiles. The energy consumers have huge diversity across their age, gender, education, energy awareness, socio-economic status, willingness to utilize renewable resources, and eco-friendly environments. Further, the end consumers of the energy can be categorised into domestic, industrial, agriculture, governmental, transportation, and commercial. Due to the large amount of energy consumption in households, the domestic sector is a key to reducing the carbon emissions, fuel dependency and planning of the capacity of energy generation systems [141]. Therefore, in this research work, the focus was on extracting consumption profiles of domestic consumers and understand their behaviour towards energy usage during their day-to-day activities. The utilities make the trading of the energy by incorporating energy consumers from all sectors. Therefore, the information related to energy consumers from other sectors such as commercial, manufacturing and industrial, agricultural and governmental can be included in the profiling to further improve the accuracy of the algorithm. The inclusion of energy consumption information from all sectors will introduce heterogeneity and provide a vast amount of data for the deep learning algorithm to efficiently train and improve generalizability.

With the technological advancements, lots of business processes and day to day human activities rely on highly accurate short-term energy prediction and load profile modelling, e.g. energy generation and distribution, price adjustments, demand response programmes, consumer awareness, etc. The MAPE, MAE and RMSE of the load forecasting can be further improved by adding additional information related to human activities, and building characteristics and social events in the area. The accurate energy load forecasting depends on many factors, therefore, the characteristics related to the buildings and the social events around the consumer premises add the context during the training of deep learning algorithms.

There are various parallelization techniques in the literature such as model-based parallelization, data-based parallelization, pipeline parallelization and hybrid parallelization. In our research work, we have adopted data-based parallelization for short term energy consumption prediction which increases the model performance. This approach requires data partitioning so that each worker thread can process in parallel and update model parameters. In this parallelisation, each worker thread gets a part of the data and performs processing to learn hidden structures. However, the model-based parallelization approach does not require data partitioning and predicts the load based on the whole dataset, due to the fact that the model based parallelisation consumes the whole dataset which enables the model to learn hidden structures in the data efficiently. Currently, model-based parallelization approaches are complex and due to the sequential nature of the deep learning models, it remains challenging to develop model-based algorithms. Therefore, further research work can be carried out to parallelize the deep learning network across multiple nodes in the cluster and perform activation and the backpropagation of the errors in order to update the weights. Additionally, the hybrid parallelization techniques utilize data and model parallelization together and further reduce the processing time. The hybrid parallelization splits data in batches and the network across nodes and overcome their individual limitations.

7.3 Limitations and Challenges of Proposed Research

The proposed unified approach in this thesis for the prediction of energy demands by processing large amounts of data with reduced processing and the enhanced understanding of end consumers' behaviour based on the deep learning techniques provides assistance to system operators and energy market agents in the decision-making process, such as developing demand response programmes. However, there are certain limitations and challenges to be addressed to fully realise the applicability of the proposed approach.

The assessment and evaluation of the proposed approach has been made by using three publicly available datasets. These datasets contain a large number of appliances in a household. The datasets provide energy consumption information for the time duration from 1 to 5 years. In this research work, the proposed approach is based on the applicability of deep learning approaches for load forecasting which require a huge amount of data for their efficient training and high accuracy. The advancements in the smart meter technologies are quite recent and therefore pose challenges in collecting large amounts of energy consumption information from individual appliances at a very granular interval. Therefore, the publicly available datasets have limits in terms of a large number of houses. By collecting large amounts of data, the proposed models will be able to further improve the accuracy. The data-based parallelization approaches improve the load forecasting accuracy. However, their performance starts degrading if there are large number of partitions of data. Therefore, it requires a trade-of between amount of data availability and the number of partitions. On the other hand, due to the sequential nature of the deep learning networks, model-based parallelization do not allow to ideally parallelize deep neural networks.

7.4 Summary

This chapter highlighted the areas which require further research in order to improve the load prediction accuracy and modelling of human behaviour by consuming less resources. It has been observed that many factors affect energy consumption at consumer premises such as domestic households, commercial buildings and industries. These factors include environment, socio-economic, building characteristics, events in the given area etc. These can be used as features in the dataset to provide context for deep learning algorithms and improve prediction accuracy and profiling consumption. Furthermore, large amounts of data collected through smart meters in SG domain introduce computational challenges which can be tackled through further research in data, model and hybrid parallelization. The following chapter concludes the research work presented in this thesis.

Bibliography

[1] Enerdata, “Global energy statistics yearbook 2019,” 2019.

[2] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, “The Hadoop distributed file system,” in *2010 IEEE 26th Symposium on Mass Storage Systems and Technologies, MSST2010*, IEEE Computer Society, 2010.

[3] F. L. Quilumba, W. J. Lee, H. Huang, D. Y. Wang, and R. L. Szabados, “Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities,” *IEEE Transactions on Smart Grid*, vol. 6, pp. 911–918, mar 2015.

[4] R. Weron, “Electricity price forecasting: A review of the state-of-the-art with a look into the future,” *International Journal of Forecasting*, vol. 30, pp. 1030–1081, oct 2014.

[5] B. Derek W, “Modelling Prices in Competitive Electricity Markets,” *Wiley: General Finance Investments*, p. 358, feb 2004.

[6] F. Rodrigues, C. Cardeira, and J. M. Calado, “The daily and hourly energy consumption and load forecasting using artificial neural network method: A case study using a set of 93 households in Portugal,” in *Energy Procedia*, vol. 62, pp. 220–229, Elsevier Ltd, 2014.

[7] H. Chitsaz, H. Shaker, H. Zareipour, D. Wood, and N. Amjady, “Short-term electricity load forecasting of buildings in microgrids,” *Energy and Buildings*, vol. 99, pp. 50–60, apr 2015.

[8] Y. Wang, Q. Chen, C. Kang, M. Zhang, K. Wang, and Y. Zhao, “Load Profiling and Its Application to Demand Response: A Review,” Tech. Rep. 2, 2015.

[9] R. Talavera-Llames, R. Pérez-Chacón, A. Troncoso, and F. Martínez-Álvarez, “Big data time series forecasting based on nearest neighbours distributed computing with Spark,” *Knowledge-Based Systems*, vol. 161, pp. 12–25, dec 2018.

[10] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, “Deep learning applications and challenges in big data analytics,” *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[11] I.-d. Briefing, “DATA MANAGEMENT AND ANALYTICS FOR UTILITIES In-depth Briefing,” 2013.

[12] W. Luan, D. Sharp, and S. Laroy, “Data traffic analysis of utility smart metering network,” *IEEE Power and Energy Society General Meeting*, pp. 1–4, 2013.

[13] I. Khan, M. W. Jack, and J. Stephenson, “Identifying residential daily electricity-use profiles through time-segmented regression analysis,” *Energy and Buildings*, vol. 194, pp. 232–246, jul 2019.

[14] Eurostat, “Energy consumption and use by households,” 2019.

[15] M. Hayn, V. Bertsch, and W. Fichtner, “Electricity load profiles in Europe: The importance of household segmentation,” *Energy Research and Social Science*, vol. 3, pp. 30–45, sep 2014.

[16] A. Ghanbari, A. Naghavi, S. Ghaderi, and M. Sabaghian, “Artificial neural networks and regression approaches comparison for forecasting iran’s annual electricity load,” *International Conference on Power Engineering, Energy and Electrical Drives*, pp. 675–679, 2009.

[17] Z. Fan, P. Kulkarni, S. Gormus, C. Efthymiou, G. Kalogridis, M. Sooriyabandara, Z. Zhu, S. Lambotharan, and W. H. Chin, "Smart Grid Communications: Overview of Research Challenges, Solutions, and Standardization Activities," pp. 1–19, 2011.

[18] E. a. Feinberg and D. Genethliou, "Load Forecasting," *Applied Mathematics for Restructured Electric Power Systems*, pp. 269–285, 2005.

[19] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, pp. 679–688, oct 2006.

[20] Ö. Ö. Bozkurt, G. Biricik, and Z. C. Tayi, "Artificial neural network and SARIMA based models for power load forecasting in Turkish electricity market," *Plos One*, vol. 12, no. 4, p. e0175915, 2017.

[21] L. Liu, X. Jin, G. Min, and L. Xu, "Real-Time Diagnosis of Network Anomaly Based on Statistical Traffic Analysis," *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 264–270, 2012.

[22] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *International Journal of Forecasting*, vol. 30, no. 2, pp. 357–363, 2014.

[23] International Renewable Energy Agency, "More People Have Access to Electricity, but World Is Falling Short of Sustainable Energy Goals," 2019.

[24] M. E. El-Hawary, "The smart grid - State-of-the-art and future trends," *Electric Power Components and Systems*, vol. 42, pp. 239–250, mar 2014.

[25] IEEE, "IEEE Grid Vision 2050 Roadmap," tech. rep., 2050.

[26] Department of Energy and C. C. UK, "Smart Grid Vision and Routemap Smart Grid Forum," tech. rep., 2014.

[27] P. Bensal and A. Singh, "Smart Metering in Smart Grid Framework: A Review," in *Fourth international conference on Parallel, Distributed and Grid Computing*, 2016.

[28] S. N. Fallah, R. C. Deo, M. Shojafar, M. Conti, and S. Shamshirband, "Computational intelligence approaches for energy load forecasting in smart energy management grids: State of the art, future challenges, and research directions," *Energies*, vol. 11, mar 2018.

[29] T. Samad, E. Koch, and P. Stluka, "Automated Demand Response for Smart Buildings and Microgrids: The State of the Practice and Research Challenges," *Proceedings of the IEEE*, vol. 104, pp. 726–744, apr 2016.

[30] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, pp. 381–388, aug 2011.

[31] J. Corbett, K. Wardle, and C. Chen, "Toward a Sustainable Modern Electricity Grid: The Effects of Smart Metering and Program Investments on Demand-Side Management Performance in the US Electricity Sector 2009-2012," *IEEE Transactions on Engineering Management*, vol. 65, pp. 252–263, may 2018.

[32] C. Gellings, "The Concept of Demand-Side Management for Electric Utilities," in *Proceedings of the IEEE*, 1985.

[33] B. P. Esther and K. S. Kumar, "A survey on residential Demand Side Management architecture, approaches, optimization models and methods," jun 2016.

[34] Y.-M. Wi, S. Kong, J. Lee, and S.-K. Joo, "Demand-Side Management Program Planning Using Stochastic Load Forecasting with Extreme Value Theory," *J Electr Eng Technol*, vol. 11, pp. 0–718, 2016.

[35] M. Glavan, D. Gradišar, S. Moscariello, . Juričić, and D. Vrančić, “Demand-side improvement of short-term load forecasting using a proactive load management a supermarket use case,” *Energy and Buildings*, vol. 186, pp. 186–194, mar 2019.

[36] M. Saviozzi, S. Massucco, and F. Silvestro, “Implementation of advanced functionalities for Distribution Management Systems: Load forecasting and modeling through Artificial Neural Networks ensembles,” *Electric Power Systems Research*, vol. 167, pp. 230–239, feb 2019.

[37] N. Perez-Mora, V. Martinez-Moll, and V. Canals, “DHC Load Management Using Demand Forecast,” in *Energy Procedia*, vol. 91, pp. 557–566, Elsevier Ltd, 2016.

[38] W. Huang, N. Zhang, C. Kang, M. Li, and M. Huo, “From demand response to integrated demand response: review and prospect of research and application,” *Protection and Control of Modern Power Systems*, vol. 4, dec 2019.

[39] J. Zhang, Y. M. Wei, D. Li, Z. Tan, and J. Zhou, “Short term electricity load forecasting using a hybrid model,” *Energy*, vol. 158, pp. 774–781, sep 2018.

[40] M. Q. Raza, M. Nadarajah, D. Q. Hung, and Z. Baharudin, “An intelligent hybrid short-term load forecasting model for smart power grids,” *Sustainable Cities and Society*, vol. 31, pp. 264–275, may 2017.

[41] Z. Guo, K. Zhou, X. Zhang, and S. Yang, “A deep learning model for short-term power load and probability density forecasting,” *Energy*, vol. 160, pp. 1186–1200, oct 2018.

[42] S. Bedingfield, D. Alahakoon, H. Genegedera, and N. Chilamkurti, “Multi-granular electricity consumer load profiling for smart homes using a scalable big data algorithm,” *Sustainable Cities and Society*, vol. 40, no. December 2017, pp. 611–624, 2018.

[43] Y. H. Hsiao, "Household electricity demand forecast based on context information and user daily schedule analysis from meter data," *IEEE Transactions on Industrial Informatics*, vol. 11, pp. 33–43, feb 2015.

[44] C. Xia, M. Zhang, and J. Cao, "A hybrid application of soft computing methods with wavelet SVM and neural network to electric power load forecasting," *Journal of Electrical Systems and Information Technology*, vol. 5, pp. 681–696, dec 2018.

[45] K. R. Cheepati and T. Nageswara Prasad, "Performance comparison of short term load forecasting techniques," *International Journal of Grid and Distributed Computing*, vol. 9, no. 4, pp. 287–302, 2016.

[46] F. L. Meng, X. J. Zeng, and Q. Ma, "Learning customer behaviour under real-time pricing in the smart grid," *Proceedings - 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013*, pp. 3186–3191, 2013.

[47] P. Gaillard, Y. Goude, and R. Nedellec, "Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting," *International Journal of Forecasting*, vol. 32, no. 3, pp. 1038–1050, 2016.

[48] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," 2017.

[49] N. Amjady and F. Keynia, "Short-term load forecasting of power systems by combination of wavelet transform and neuro-evolutionary algorithm," *Energy*, vol. 34, no. 1, pp. 46–57, 2009.

[50] C. Kuster, Y. Rezgui, and M. Mourshed, "Electrical load forecasting models: A critical systematic review," nov 2017.

[51] L. Suganthi and A. A. Samuel, “Energy models for demand forecasting - A review,” feb 2012.

[52] E. Verband der Elektrotechnik, Energietechnische Gesellschaft im VDE., and Institute of Electrical and Electronics Engineers., “Markov Chain based Very Short Term Load Forecasting realizing Conditional Expectation,” in *International ETG Congress 2017*, 2017.

[53] K. Ahmed, M. Al Amin, and M. Rahman, “Application of Short Term Energy Consumption Forecasting for Household Energy Management System,” in *2015 International Conference on Green Energy & Technology (ICGET)2015 International Conference on Green Energy & Technology (ICGET)*, 2015.

[54] R. Torkzadeh, A. Sedighi Anaraki, and A. Mirzaei, “Medium Term Load Forecasting in Distribution Systems Based on Multi Linear Regression & Principal Component Analysis: A Novel Approach,” in *Electrical Power Distribution Networks (EPDC), 2014 19th Conference on*, p. 122, 2014.

[55] S. Kumar Panda, S. Nandan Mohanty, and A. Kumar Jagadev, “Long Term Electrical Load Forecasting: An Empirical Study across Techniques and Domains,” *Indian Journal of Science and Technology*, vol. 10, no. 26, pp. 974–6846, 2017.

[56] Y. Yang, J. Wu, Y. Chen, and C. Li, “A new strategy for short-term load forecasting,” *Abstract and Applied Analysis*, vol. 2013, 2013.

[57] T. G. Nikolaou, “Short-term load forecasting based on the Kalman filter and the neural-fuzzy network (ANFIS),” *IASME/WSEAS International Conference on Energy & Environmental Systems, Chalkida, Greece*, vol. 2006, pp. 189–193, 2006.

[58] T. Senjyu, P. Mandal, K. Uezato, and T. Funabashi, "Next Day Load Curve Forecasting Using Hybrid Correction Method," *IEEE Transactions on Power Systems*, vol. 20, pp. 102–109, feb 2005.

[59] H. M. Al-Hamadi and S. A. Soliman, "Long-term/mid-term electric load forecasting based on short-term correlation and annual growth," *Electric Power Systems Research*, vol. 74, no. 3, pp. 353–361, 2005.

[60] C. E. Borges, Y. K. Penya, and I. Fernández, "Optimal combined short-term building load forecasting," *2011 IEEE PES Innovative Smart Grid Technologies, ISGT Asia 2011 Conference: Smarter Grid for Sustainable and Affordable Energy Future*, pp. 1–8, 2011.

[61] M. Frincu, C. Chelmis, M. U. Noor, and V. Prasanna, "Accurate and Efficient Selection of the Best Consumption Prediction Method in Smart Grids," *Proc. of the IEEE BigData - Intl. Conf. on Big Data*, pp. 721–729, 2014.

[62] S. kumar Singh, *Database Systems: Concepts, Design and Applications.* Pearson Education India, 2011.

[63] R. Zhang, "OLS Regression ? Auto-Regression ? Dynamic Regression ? A Practical Modeling Example in Financial Industry," *SAS Global Forum; Data Mining and Predictive Modeling*, pp. 1–12, 2007.

[64] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel, *Time Series Analysis.* Wiley Series in Probability and Statistics, Wiley, jun 2008.

[65] . Peter and P. Silvia, "ARIMA vs . ARIMAX which approach is better to analyze and forecast macroeconomic time series ?," *Proceedings of 30th International Conference Mathematical Methods in Economics*, no. 2, pp. 136–140, 2012.

[66] H. Cui and X. Peng, "Short-Term City Electric Load Forecasting with Considering Temperature Effects : An Improved ARIMAX Model," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[67] L. F. Amaral, R. C. Souza, and M. Stevenson, "A smooth transition periodic autoregressive (STPAR) model for short-term load forecasting," *International Journal of Forecasting*, vol. 24, no. 4, pp. 603–615, 2008.

[68] Y. Goude, R. Nedellec, and N. Kong, "Local short and middle term electricity load forecasting with semi-parametric additive models," *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 440–446, 2014.

[69] P. Craven and G. Wahba, "Smoothing Noisy Data with Spline Functions," *Numerische Mathematik*, vol. 31, pp. 377–403, 1979.

[70] a. Ghanbari, E. Hadavandi, and S. Abbasian-Naghneh, "Comparison of Artificial Intelligence Based Techniques for Short Term Load Forecasting," *Business Intelligence and Financial Engineering (BIFE), 2010 Third International Conference on*, no. January, pp. 6–10, 2010.

[71] Q. Yang, H. Wang, T. Li, and Y. Yang, "Deep belief networks oriented clustering," *Proceedings - The 2015 10th International Conference on Intelligent Systems and Knowledge Engineering, ISKE 2015*, pp. 58–65, 2016.

[72] Y. Xu, J. Cai, and J. V. Milanovic, "On accuracy of demand forecasting and its extension to demand composition forecasting using artificial intelligence based methods," in *IEEE PES Innovative Smart Grid Technologies Conference Europe*, vol. 2015-January, IEEE Computer Society, jan 2015.

[73] Bhavna Sharma, 1, and Prof. K. Venugopalan, and 2, "Comparison of Neural Network Training Functions for \nHematoma Classification in Brain CT Images," *Http://Www.Iosrjournals.Org/Iosr-Jce*, vol. 16, no. 1, pp. 31–35, 2014.

[74] S. Hosein and P. Hosein, "Load Forecasting using Deep Neural Networks," *2018 IEEE International Conference on Energy Internet (ICEI)*, 2017.

[75] M. Easley, L. Haney, J. Paul, K. Fowler, and H. Wu, "Deep neural networks for short-term load forecasting in ERCOT system," in *2018 IEEE Texas Power and Energy Conference, TPEC 2018*, vol. 2018-February, pp. 1–6, Institute of Electrical and Electronics Engineers Inc., mar 2018.

[76] S. Zheng, Q. Zhong, L. Peng, and X. Chai, "Erratum: A simple method of residential electricity load forecasting by improved Bayesian neural networks," *Mathematical Problems in Engineering*, vol. 2018, 2018.

[77] C. Tian, J. Ma, C. Zhang, and P. Zhan, "A Deep Neural Network Model for Short-Term Load Forecast Based on Long Short-Term Memory Network and Convolutional Neural Network," *Energies*, vol. 11, p. 3493, dec 2018.

[78] M. Barman and N. B. Dev Choudhury, "Season specific approach for short-term load forecasting based on hybrid FA-SVM and similarity concept," *Energy*, pp. 886–896, may 2019.

[79] M. Kim, W. Choi, Y. Jeon, and L. Liu, "A Hybrid Neural Network Model for Power Demand Forecasting," *Energies*, vol. 12, p. 931, mar 2019.

[80] Z. Li and L. Li, "A Hybrid Model of Least Squares Support Vector Regression Optimized by Particle Swarm Optimization for Electricity Demand Prediction," pp. 91–103, Association for Computing Machinery (ACM), 2019.

[81] A. Rahman, V. Srikumar, and A. D. Smith, "Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks," *Applied Energy*, vol. 212, pp. 372–385, feb 2018.

[82] M. Li, D. Allinson, and M. He, "Seasonal variation in household electricity demand: A comparison of monitored and synthetic daily load profiles," *Energy and Buildings*, vol. 179, pp. 292–300, nov 2018.

[83] S. Zhai, Z. Wang, X. Yan, and G. He, "Appliance flexibility analysis considering user's behavior in home energy management system using smart plugs," *IEEE Transactions on Industrial Electronics*, pp. 1–1, 2018.

[84] C. Magazzino, "Is per capita energy use stationary? time series evidence for the emu countries," *Energy Exploration Exploitation*, pp. 1–1, 2017.

[85] A. Albert and R. Rajagopal, "Smart Meter Driven Segmentation : What Your Consumption Says About You," vol. 28, no. 4, pp. 4019–4030, 2013.

[86] kim et al, "Load profile extraction by mean-shift clustering with sample pearson correlation coefficient distance," *Energies*, pp. 1–1, 2018.

[87] X. Zhang, T. Kato, and T. Matsuyama, "Learning a context-aware personal model of appliance usage patterns in smart home," *2014 IEEE Innovative Smart Grid Technologies - Asia, ISGT ASIA 2014*, pp. 73–78, 2014.

[88] S. Rollins and N. Banerjee, "Using rule mining to understand appliance energy consumption patterns," *2014 IEEE International Conference on Pervasive Computing and Communications, PerCom 2014*, pp. 29–37, 2014.

[89] D. Schweizer, M. Zehnder, H. Wache, H. F. Witschel, D. Zanatta, and M. Rodriguez, "Using consumer behavior data to reduce energy consumption in smart homes: Apply-

ing machine learning to save energy without lowering comfort of inhabitants," *Proceedings - 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA 2015*, pp. 1123–1129, 2016.

[90] A. K. Marnerides, P. Smith, A. Schaeffer-Filho, and A. Mauthe, "Power consumption profiling using energy time-frequency distributions in smart grids," *IEEE Communications Letters*, vol. 19, pp. 46–49, Jan 2015.

[91] J. Lai, H. Zhou, W. Hu, D. Zhou, and L. Zhong, "Smart Demand Response Based on Smart Homes," *Mathematical Problems in Engineering*, vol. 2015, 2015.

[92] A. Al-Wakeel and J. Wu, "K-means based cluster analysis of residential smart meter measurements," in *Energy Procedia*, vol. 88, pp. 754–760, Elsevier Ltd, jun 2016.

[93] The European Smart Metering Industry Group, "Position Paper Smart Metering for Europe," tech. rep., 2009.

[94] J. Dean, G. S. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. A. Ranzato, A. Senior, P. Tucker, K. Yang, and A. Y. Ng, "Large Scale Distributed Deep Networks," *Nips*, pp. 1–11, 2012.

[95] L. Ericson and R. Mbuvha, "On the Performance of Network Parallel Training in Artificial Neural Networks," jan 2017.

[96] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl, "Measuring the Effects of Data Parallelism on Neural Network Training," nov 2018.

[97] R. Mayer and H.-A. Jacobsen, "Scalable Deep Learning on Distributed Infrastructures: Challenges, Techniques and Tools," mar 2019.

[98] J. Á. González Ordiano, A. Bartschat, N. Ludwig, E. Braun, S. Waczowicz, N. Renkamp, N. Peter, C. Düpmeier, R. Mikut, and V. Hagenmeyer, "Concept and benchmark results

for Big Data energy forecasting based on Apache Spark," *Journal of Big Data*, vol. 5, dec 2018.

[99] C. Yang, Y. Song, J. Qian, H. Zhao, W. Jiang, H. Tang, and J. Wu, "Apache spark based urban load data analysis and forecasting technology research," in *2017 IEEE Conference on Energy Internet and Energy System Integration, EI2 2017 - Proceedings*, vol. 2018-Janua, pp. 1–6, Institute of Electrical and Electronics Engineers Inc., jan 2018.

[100] Y. Liu, W. Jing, and L. Xu, "Parallelizing backpropagation neural network Using MapReduce and cascading model," *Computational Intelligence and Neuroscience*, vol. 2016, 2016.

[101] X. Zhao, A. An, J. Liu, and B. X. Chen, "Dynamic Stale Synchronous Parallel Distributed Training for Deep Learning," *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, aug 2019.

[102] D. Das, S. Avancha, D. Mudigere, K. Vaidynathan, S. Sridharan, D. Kalamkar, B. Kaul, and P. Dubey, "Distributed Deep Learning Using Synchronous Stochastic Gradient Descent," feb 2016.

[103] S. Zheng, Q. Meng, T. Wang, W. Chen, N. Yu, Z.-M. Ma, and T.-Y. Liu, "Asynchronous Stochastic Gradient Descent with Delay Compensation," sep 2016.

[104] Y. Simmhan, S. Aman, A. Kumbhare, R. Liu, S. Stevens, Q. Zhou, and V. Prasanna, "Cloud-based software platform for big data analytics in smart grids," *Computing in Science and Engineering*, vol. 15, no. 4, pp. 38–47, 2013.

[105] M. H. Athari and Z. Wang, "Modeling the Uncertainties in Renewable Generation and Smart Grid Loads for the Study of the Grid Vulnerability," in *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2016.

[106] U. Yolcu, E. Egrioglu, V. R. Uslu, M. A. Basaran, and C. H. Aladag, "A new approach for determining the length of intervals for fuzzy time series," *Applied Soft Computing*, vol. 9, no. 2, pp. 647–651, 2009.

[107] B. Nepal, M. Yamaha, H. Sahashi, and A. Yokoe, "Analysis of building electricity use pattern using k-means clustering algorithm by determination of better initial centroids and number of clusters," *Energies*, vol. 12, no. 12, 2019.

[108] C. S. Burrus, M. Frigo, S. G. Johnson, M. Pueschel, and I. Selesnick, "Fast Fourier Transforms," 2012.

[109] T. Hong, "Load Forecasting Case Study," tech. rep., 2015.

[110] J. Ma, H. Chen, S. Member, L. Song, S. Member, Y. Li, H. Chen, Y. Li, and C. Cost efficiency, "Residential Load Scheduling in Smart Grid: A Cost Efficiency Perspective Total demand for appliance a to fulfill a task in cycle Upper bound for," *Ieee Transactions on Smart Grid*, vol. 71, no. 2, pp. 19–21, 2016.

[111] D. Rumelhard and D. Zipser, "Feature discovery by competitve learning," *Cognitive Science*, vol. 9, pp. 75–112, 1985.

[112] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, dec 2016.

[113] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 9, pp. 249–256, 2010.

[114] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," jun 2012.

[115] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature 521*, vol. 13, pp. 436–444, dec 2015.

[116] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[117] F. Borisyuk, A. Gordo, and V. Sivakumar, "Rosetta: Large scale system for text detection and recognition in images," tech. rep., 2018.

[118] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," apr 2014.

[119] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran, "Speeding Up Distributed Machine Learning Using Codes," *IEEE Transactions on Information Theory*, vol. 64, pp. 1514–1529, mar 2018.

[120] T. Cowsalya and S. R. Mugunthan, "Hadoop architecture and fault tolerance based hadoop clusters in geographically distributed data center," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 7, pp. 2818–2821, 2015.

[121] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler, "Apache hadoop YARN: Yet another resource negotiator," in *Proceedings of the 4th Annual Symposium on Cloud Computing, SoCC 2013*, Association for Computing Machinery, 2013.

[122] A. F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava, "Building a highlevel dataflow system on top

of Map-Reduce: The Pig experience," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1414–1425, 2009.

[123] A. Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony, H. Liu, P. Wyckoff, and R. Murthy, "Hive - A warehousing solution over a map-reduce framework," *Proceedings of the VLDB Endowment*, vol. 2, no. 2, pp. 1626–1629, 2009.

[124] F. P. Junqueira and B. C. Reed, "The life and times of a zookeeper," p. 46, Association for Computing Machinery (ACM), 2009.

[125] M. Zaharia, M. Chowdhury, M. J. Franklin, and S. Shenker, "Spark: Cluster Computing with Working Sets," in *Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing (HotCloud'10)*, pp. 1–7, 2010.

[126] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, pp. 107–113, jan 2008.

[127] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart*: An Open Data Set and Tools for Enabling Research in Sustainable Homes," *SustKDD*, no. August, p. 6, 2012.

[128] S. Jovanović, S. Savić, M. Bojić, Z. Djordjević, and D. Nikolić, "The impact of the mean daily air temperature change on electricity consumption," *Energy*, vol. 88, pp. 604–609, aug 2015.

[129] K. Gajowniczek and T. Zabkowski, "Electricity forecasting on the individual household level enhanced based on activity patterns," *PLoS ONE*, vol. 12, apr 2017.

[130] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[131] Q. Zhang, H. Wang, J. Dong, G. Zhong, and X. Sun, "Prediction of Sea Surface Temperature Using Long Short-Term Memory," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, pp. 1745–1749, oct 2017.

[132] S. Ahmed Yousuf and A. AKM Rezaul, "Short Term Load Forecasting using Multiple Linear Regression for Big Data," *2017 IEEE Symposium Series on Computational Intefligence (SSCI)*, 2017.

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

[134] S. Salloum, R. Dautov, X. Chen, P. X. Peng, and J. Z. Huang, "Big data analytics on Apache Spark," *International Journal of Data Science and Analytics*, vol. 1, pp. 145–164, nov 2016.

[135] F. Niu, B. Recht, C. Re, and S. J. Wright, "HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent," *Advances in Neural Information Processing Systems*, no. 1, p. 21, 2011.

[136] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, "Short-Term Load Forecasting with Deep Residual Networks," *IEEE Transactions on Smart Grid*, vol. 10, pp. 3943–3952, jul 2019.

[137] R. Teng and T. Yamazaki, "Load Profile-Based Coordination of Appliances in a Smart Home," *IEEE Transactions on Consumer Electronics*, vol. 65, pp. 38–46, feb 2019.

[138] A. Ghosal and M. Conti, "Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 2831–2848, mar 2019.

[139] Qinwei Duan, "Price-Based Demand Response Scheduling Model in Day-Ahead Electricity Market," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 2016.

[140] Q. Zhang, J. Lu, Z. Yang, and M. Tu, "A Deep Learning Based Real-time Load Forecasting Method in Electricity Spot Market," in *Journal of Physics: Conference Series*, vol. 1176, Institute of Physics Publishing, mar 2019.

[141] Y. Liu, "Role of a forward-capacity market to promote electricity use reduction in the residential sectora case study of the potential of social housing participation in the Electricity Demand Reduction Pilot in the UK," *Energy Efficiency*, vol. 11, pp. 799–822, apr 2018.