PHD

Smart Meter Data Analytics

Xu, Minghao

Award date:
2019

Awarding institution:
University of Bath

Smart Meter Data Analytics

By

Minghao Xu

BEng, SMIEEE

The thesis submitted for the degree of

Doctor of Philosophy

in

The Department of
Electronic and Electrical Engineering
University of Bath

December 2018

Signature: …… Date: ……

Abstract

In recent years smart meters and advanced metering infrastructures (AMI) have been rolled out to a substantial number of domestic consumers across the world. An unprecedented amount of fine-grained load data have since been generated. Meanwhile, owing to the rapid and ongoing digitalization of the society, a variety of information about the consumers which was unattainable has streamed in and becomes available, such as the consumer's age, education level, income, etc. However, traditional analytics used in power systems are unable to handle the smart meter data efficiently and effectively due to the high volatility, large volume and fast generating speed of domestic load data. The added dimensions to the smart meter data from other sources increase the variety of data and further complicate the processing of the data. This thesis proposes a range of methods to address the challenges from two key aspects:

1) Uncovering the underlying patterns of the smart metering data. This thesis proposes a novel load forecasting methodology that leverages the knowledge learned from one forecasting task to achieve more efficient and accurate forecasting for another by utilizing transfer learning along with deep learning. The adoption of deep neural networks enables the effective modelling of highly complex and nonlinear relationships within smart metering data and has endowed us with strong predictive power. Additionally, transfer learning would further improve the predictive performance and significantly reduce the required amount of data, computational power, and the efforts for hyperparameter optimization.

2) Revealing the interconnection/correlation between data from other sources and smart meter data. Based on the sources, other available data could be classified into two groups, i.e., social-economic/demographic data of consumers and data from the power system. For the social-economic data, this thesis first proposes an ensemble learning framework that could not only predict the social-economic status accurately form smart metering data, but also provide insights into the correlation between the two sets of data due to the model's interpretability. Conversely, a deep Convolutional Neural Network (CNN) based model is proposed to infer the load characteristics from the social-economic data of the consumers. It leverages the convolutional kernel and

deep architecture to overcome the hurdle brought by mixed types of data and infers multiple load characteristics simultaneously. It is validated on real data and demonstrates an improvement in both the learning efficiency and the prediction accuracy compared to predicting each characteristic separately. As for the data from the power system, this research preliminarily focuses on the phase connectivity of a consumer. The phase connectivity is not commonly available, however, keeps gaining increasing attention due to the critical need for Low Carbon Technologies integration and network balancing. A novel Spectral and Saliency Analysis (SSA) method is developed to accurately identify the phases of consumers using their smart metering data.

Contents

Acknowledgement

Firstly, I would like to express my deepest appreciation to my supervisor Prof. Furong Li for her continuous help, support, and guidance throughout the course of my study. She is not only a great supervisor in research but also a good friend and advisor in daily life.

I would like to thank Dr Adrian Evans and Dr Nathan Smith for their advice and guidance. I would like to express my gratitude to Dr Ran Li, who gave me constructive suggestions and generously shared me with his experience and knowledge.

I would like to thank all my current fellow colleagues and friends Mr. Da Huo, Mr. Heng Shi, Miss Heather Wyman-Pain, Miss Wei Wei, Mr. Yuankai Bian, Mr. Hantao Wang, Miss Qiuyang Ma, Miss Chi Zhang, Mr. Xiaohe Yan, Mr. Xinhe Yang, Miss Wangwei Kong, Dr. Zhong Zhang, Dr. Seyed Fazeli, Dr Ignacio Hernando Gil, Mr. Mike Thomas, Dr. Chenghong Gu and Dr. Kang Ma for their suggestions and friendship.

I would also like to thank all my previous fellow colleagues and friends Dr Zhongjian Liu, Dr Chen Zhao, Dr Zhipeng Zhang, Dr Jiangtao Li, Dr Fan Yi, Dr Lin Zhou, Dr Huiming Zhang and Dr Jie Yan for their assistance in my research and life.

I would like to take this opportunity to express my ultimate gratitude to my family for encouraging me in all of my pursuits and inspiring me to follow my dreams. I am sincerely thankful to my parents who are my lifelong role models. I am grateful to my beloved wife, Yanyun, who has been my best friend and companion and helped me get through this special period in the most positive way.

List of Figures

List of Tables

List of Abbreviations

Commission for Energy Regulation	CER
Computer Vision	CV
Convolutional Neural Network	CNN
Customer Behaviour Trials	CBTs
Demand Side Management	DSM
Demand Side Response	DSR
Distribution Network Operator	DNO
Energy Management System	EMS
False Negative	FN
False Positive	FP
High Performance Computing	HPC
k-Nearest Neighbours	KNN
Linear Discriminant Analysis	LDA
Long Short Term Memory	LSTM
Mean Absolute Error	MAE
Mean Absolute Percentage Error	MAPE
Mean Squared Error	MSE
Multi-Layer Perceptron	MLP
Normalised Root-Mean-Square Error	NRMSE
Office of Gas and Electricity Markets	Ofgem
Peer-to-Peer	P2P
Principal Component Analysis	PCA
Random Forest	RF
Recurrent Neural Network	RNN
Residual Network	ResNet
Root-Mean-Square Error	RMSE
Short-Term Load Forecasting	STLF
Smart Grid	SG

Support Vector Machine	SVM
True Negative	TN
True Positive	TP

Chapter 1

Introduction

This chapter briefly describes the research background, motivation, challenges, and contributions of this work. It also provides an overview of the thesis.

1.1 Research Background

1.1.1 Climate Change and Low Carbon Technologies

Growing population and industrialization of the societies are creating a tremendous need for energy use [1], which have also led to a rapid increase in greenhouse gas emissions. In response, over 190 parties have signed the Kyoto Protocol aiming to reduce the greenhouse emissions [2]. Specifically, in the UK, the Parliament of the United Kingdom announced the Climate Change Act 2008 [3]. The act sets its goal as 34% reduction of CO2 emission by 2020 and at least 80% greenhouse gases emissions reduction by 2050. For the energy sector in the UK, as shown in Figure 1-1, it generated the most amount of CO2, accounting for about 40% of the total emission in 2008 [4], and would make a major contribution to achieving the targets in the act. In the following year, the Office of Gas and Electricity Markets (Ofgem) introduced the Low Carbon Network (LCN) fund [5] to encourage Distribution Network Operators (DNOs) to get prepared for and facilitate the transition to a low carbon future.

Under the circumstances, Low Carbon Technologies (LCTs) such as renewable generations have been significantly promoted [6, 7]. However, the existing distribution network will face unprecedented stresses mainly due to the inherently volatile and uncertain output of renewable generations, such as solar power, wind power and biomass energy. Consequently, a portfolio of projects emerged with the goal of

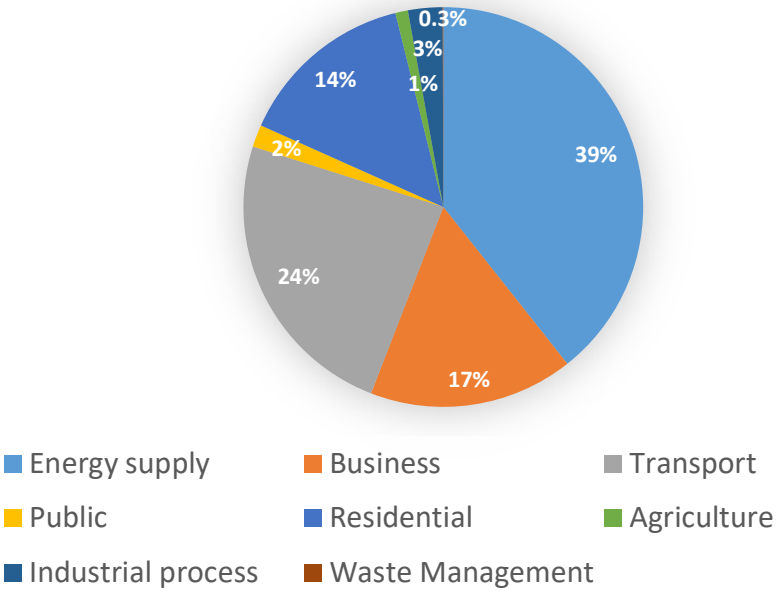

Figure 1-1 UK greenhouse gas emission by source sector in 2008

understanding how the existing network will need to change to accommodate a series of LCTs.

1.1.2Smart Grid and Smart Meters

To achieve a greener future, delivering an affordable and low-carbon supply of electricity has become the main concern of electricity utilities. The evolution of the Smart Grid (SG) [8, 9] provides a potentially cost-effective way to reduce demand and make better use of LCTs, which is a key step towards the goal of low carbon emission.

The SG is the next generation power grid. In contrast to the traditional power grid, it uses the two-way flow of both electricity and information [10]. The anticipated improvement brought by the SG includes the improved power quality and reliability, the enhanced capacity and efficiency of the existing power networks, accommodating distributed energy resources and renewable generations, automated maintenance and operation, etc. [11].

The transmission network is regarded as the most important element in the chain of energy supply due to its vital role to the overall system security. Hence, it has previously received more attention regarding the installation of advanced monitoring, protection and control equipment [12]. However, considering the fact that nearly 90% of all the power outages and disturbances take place in distribution network [9], the evolution of current grid should start from its bottom which is the distribution network. Specifically, the deployment of smart meters and other Advanced Metering Infrastructure (AMI) for the distribution networks have been the focus in the past several years. Smart meters have been rolled out to a large number of domestic households across the world. The UK government has established a central programme to roll out smart meters across the UK. Approximately 53 million smart meters will be deployed to all the UK homes and small businesses by the end of 2020 [13]. The unprecedented visibility on individual consumers attained through these smart meters could provide crucial information for the consumers to save energy and choose better tariff, for the energy suppliers to develop tailored tariff and more effective Demand Side Response (DSR) schemes, for the DNOs to better facilitate LCTS and achieve more efficient operations.

1.1.3 Customer Information Digitization

As the development of the internet and the Internet of Things (IoT) [14], a vast amount of data are being created by the customers. Consequently, a variety of information about the consumers which was traditionally unattainable has streamed in and becomes available, such as the consumer's age, education, and other social-economic or demographic information.

The smart metering data of the consumers demonstrate the consumption patterns of the consumers directly and intuitively, whereas the social-economic or social-demographic status of consumers provides deeper insights into the consumers. A deeper understanding of the two streams of the data would provide insights into the energy behaviours of the consumers. Utilization of the insights would then help develop better Energy Management System and better facilitate the LCTs.

Apart from the social-economic information and smart metering data of the consumers, other forms of data from the power systems have also become available or are seeking methods to be monitored and collected. For example, the voltage profiles of the consumers are being monitored by the smart meters as well, though not commonly being recorded. In contrast, the phase connectivity of a consumer is not commonly available and cannot be monitored by the smart meters. The identification of the phases is receiving increasing attention recently due to the need for balancing network and accommodating LCTs.

1.2 Research Motivation and Challenges

1.2.1 Uncovering the Underlying Patterns of the Smart Metering Data

Traditionally, most meters and sensors are installed in the transmission networks in the power systems. Electricity meters in distribution networks are installed for billing purpose and are normally read on sites by electricians on a monthly basis. The emergence of smart meters brings a compelling opportunity to increase the distribution network's visibility and accessibility. Up to now, a substantial number of smart meters have been deployed at domestic households across the world.

The influx of exceedingly large smart metering datasets presents unprecedented opportunities for uncovering the underlying patterns of the energy behaviours of the consumers. One crucial aspect of uncovering the patterns is load forecasting. For energy consumers, better forecasting of their energy consumption patterns would help them save energy and choose better tariff. For energy suppliers, accurate forecasting would help them design targeted tariff and reduce the cost of purchasing electricity from the wholesale market [15]. For DNOs, the better forecasting of their consumers' energy consumption patterns would help them better facilitate the integration of LCTs, achieving more efficient operation of the networks, and make better planning and investment.

However, traditional forecasting methods used in the power systems are developed for more aggregated loads, such as transmission network load, regional load and national load. Load aggregation naturally helps reduce uncertainties [16]. Consequently, aggregated loads are much smoother, less uncertain and volatile, and are easier to forecast compared to individual household's load. In addition, smart metering data are large in volume and fast in generating speed which poses tremendous challenges to traditional load forecasting methods [17-20].

Recently, with the development of deep learning, the prediction performance of domestic load forecasting has been elevated to an acceptable level [16, 21-24]. Nonetheless, the improved accuracy by implementing deep learning is often accompanied by other inconvenience, some of which are: 1) building and training deep learning models are quite difficult and time-consuming; 2) not all consumers have enough data for training deep models; 3) different deep models are trained for different datasets or consumers, which is repetitive and complicated. Therefore, there is an imperative need for developing efficient and accurate load forecasting models to utilize the smart metering data.

1.2.2 Revealing the Interconnection/Correlation between Data from Other Sources and Smart Metering Data

Apart from the smart metering data, a variety of information about the consumers has become available due to the ongoing digitization of the societies. Based on the sources, the available information or data of interest could be classified into two

categories, i.e., social-economic/demographic data of consumers and data from the power system.

For the social-economic data, such as the social class, age, and education level of the consumer, they depict the energy consumers from a different angle. A better understanding of them would equip the suppliers with the necessary information for developing personalized services on targeted consumers [25, 26]. In addition, provided with the social-economic status of the target consumer, more effective DSR and EMS could then be implemented. Social-economic data are as equally important as the smart metering data. However, energy consumers have varying degrees of availability of data. Not all of them have both the smart metering data and the social-economic data. This drives the need for understanding the underlying correlation between the two streams of data and developing models to infer one source of data given the other.

As for the consumer data from the power systems, a wide range of data are becoming available, e.g., the voltage profiles at the consumer end, the received frequency of the consumer, and the phase connectivity of the consumer. The voltage and frequencies can be monitored by most smart meters, whereas the identification of phase connectivity of a consumer requires the instalment of extra equipment. Given the fact that knowing the phase connectivity of the consumers in a network would help balance the network and accommodate LCTs, a cost-effective and efficient way to identify phases should be developed to help move towards a greener and efficient distribution network.

1.3 Research Contributions

This research aims to fully utilize the smart metering data for load forecasting and linking the smart metering data to other sources of data in a big data context. The main contributions of the research are summarized as follows:

- Development of transfer learning based short-term load forecasting model for domestic consumers. The large amount of smart metering data are ideal for developing deep neural networks for load forecasting. The adoption of transfer learning on the deep models leverages the knowledge learned from one forecasting task for forecasting another, which not only reduces the required

computational power for training from scratch but also improves the forecasting accuracy.

- Proposing a boosting tree ensemble model for the inference of consumer's social-economic status from smart metering data. Meaningful features are designed and extracted from the smart metering data to reduce the dimensionality of the data. Extreme Gradient Boost, a tree-based ensemble model, is adopted to predict the social-economic status data. Comprehensive interpretation of the trained model by analyzing the individual base trees is given.
- Proposing a Deep Convolutional Neural Network (DCNN) based multi-task learning (MTL) method for the inference of consumer's smart metering features from social-economic data. It leverages the convolutional kernel and deep architecture in DCNN to overcome the hurdle brought by mixed types of social-economic data and infers multiple load characteristics simultaneously.
- Proposes to a novel spectral and saliency analysis (SSA) methodology to identify households' phases using their smart metering data. Specifically, the proposed method combines spectral and temporal domain feature extraction techniques on the smart metering data and do not require 100% smart meter penetration ratio in the network.

1.4 Thesis Layout

The rest of the thesis is organized as follows:

Chapter 2 provides an extensive and comprehensive literature review of current research on smart metering data analytics. The review will focus on three aspects: 1) load forecasting, 2) customer characterization with smart metering data and social-economic data, 3) connection and phase identification. The limitations of the existing methods are identified and discussed.

Chapter 3 proposes to leverage the knowledge learned from one forecasting task to achieve more efficient and accurate forecasting for another by utilizing transfer learning and deep learning. The proposed method is tested using real data and the transferability of different deep models and different layers in the deep models is also assessed.

Chapter 4 proposes an ensemble framework to infer the consumer's social-economic information from smart metering data. Meaningful features are firstly extracted from smart metering data and then fed to a gradient boosted ensemble tree structure for training. The features that significantly help the inference of the social-economic status are lastly identified.

Chapter 5 proposes a Deep Convolutional Neural Network (DCNN) based multi-task learning (MTL) method for the inference of consumer's smart metering features from social-economic data. It leverages the convolutional kernel and deep architecture in DCNN to overcome the hurdle brought by mixed types of social-economic data and infers multiple load characteristics simultaneously.

Chapter 6 proposes a novel spectral and saliency analysis (SSA) methodology to identify households' phases using their smart metering data. Specifically, the proposed method combines spectral and temporal domain feature extraction techniques on the smart metering data and do not require 100% smart meter penetration ratio in the network. The proposed method is then validated under different data conditions to demonstrate its robustness and accuracy.

Chapter 7 summarizes the key findings from the research and the major contribution of the work.

Chapter 8 presents some potential research topics are as future work.

Chapter 2

Literature Review

This chapter provides an extensive and comprehensive literature review of current research on smart meter data analytics. The limitations of the existing methods are identified and discussed.

2.1 Introduction

An unprecedented amount of fine-grained smart meter data have been generated over the past decade owing to the rapid and wide deployment of the smart meters. For example, in the UK 10.02 million smart meters have been installed in domestic households by the end of 2017 [27]. A wide range of data analytics has since been developed to utilize the smart meter data. This thesis focus on three aspects of the smart meter data analytic research:

1) Load forecasting for domestic customers;
2) Customer characterization with smart meter data and social-economic data;
3) Phase identification.

2.2 Load Forecasting

Load forecasting refers to the prediction of future load. It has been widely used and studied due to its significance in the planning and operation of the power system [28]. Based on the forecasting horizons, load forecasting can be divided into four categories [29] as depicted in Figure 2-1:

1) Very short-term load forecasting (VSTLF) [30-33];
2) Short-term load forecasting (STLF) [16, 23, 24, 34-37];
3) Medium-term load forecasting (MTLF) [38-40];
4) Long-term load forecasting (LTLF) [41-43].

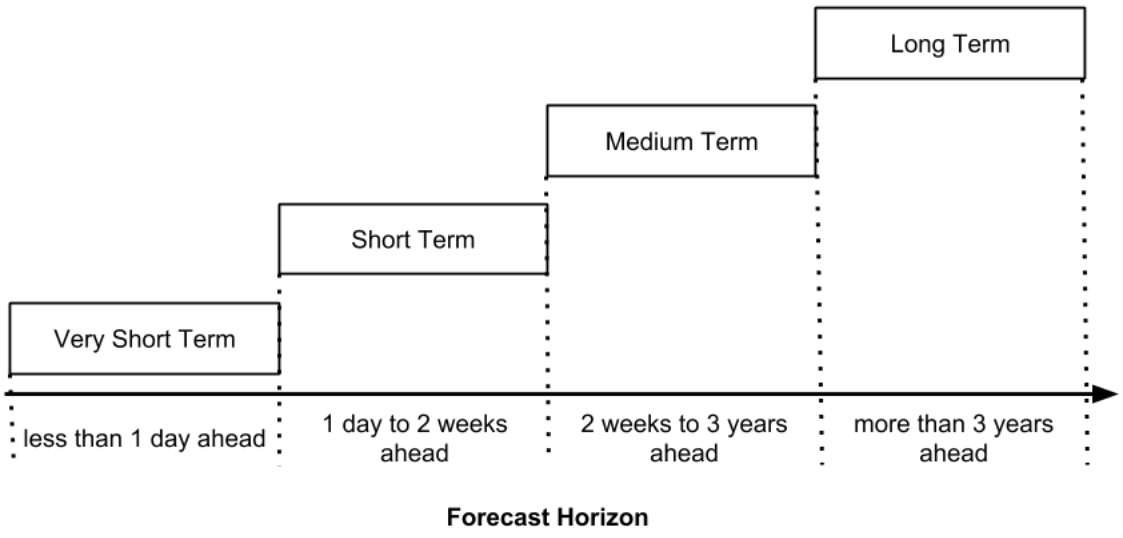

Figure 2-1 Load forecasting by forecast horizon

VSTLF is generally used for the real-time scheduling of generation [32]. STLF is gaining more attention due to the emergence of the smart grid [28]. Accurate STLF is the prerequisite for effective DSR and EMS [37], cost-effective electricity purchase [15], facilitating LCTs [44], etc. MTLF is normally for maintenance scheduling [38] and efficient operation [39]. LTLF generally takes the future energy policy and economic status into consideration and is used for the long-term power system planning such as the development of infrastructures [42].

Traditional load forecasting mainly focuses on the system level load, regional level load, and feeder level load. Unlike traditional aggregated load, the smart meter data present a new perspective to the way load forecasting can be performed. Smart meter data depict the consumer level load, which are much more volatile and uncertain. Figure 2-2 demonstrates how the aggregation affect the load.

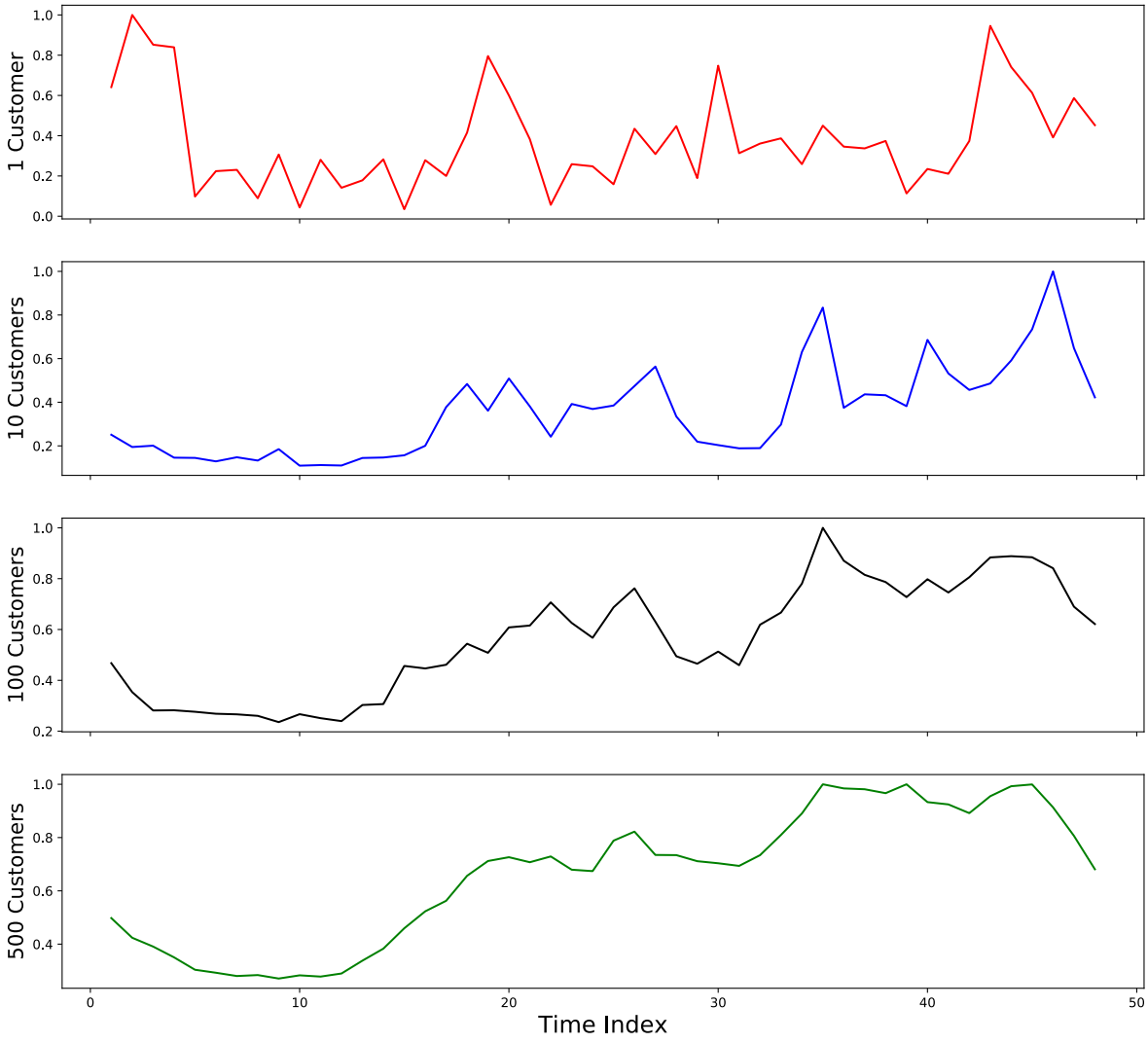

Figure 2-2 Normalized load with varying levels of aggregation

Current research on load forecasting with smart meter data can be classified into two groups, i.e., deterministic or point forecasting and probabilistic forecasting.

2.2.1 Deterministic Load Forecasting

In deterministic or point load forecasting, the numerical values of future load are directly predicted. Current research on point forecasting with smart meter data can be grouped into two classes: 1) direct use or modification of traditional load forecasting methodologies; 2) developing or adopting new algorithms.

In [17-20, 45-47], experiments have been conducted to benchmark the traditional and classical forecasting methods for STLF at the individual consumer level. The tested methods include both time-series techniques, e.g., ARIMA and exponential smoothing, and state-of-the-art machine learning approaches, i.e., SVM and Artificial Neural Network (ANN). As testified in [19, 45], linear regression and persistence forecasting outperformed all the tested approaches for domestic household STLF. In [48], linear regression, shallow neural networks, SVM and some variants of them were tested on both building level load and domestic household level load. Results have shown that accurate forecasting could only be made for the buildings, which coincides with the findings in [19, 45].

Very recently, with the advancement of deep learning, various forms of deep models have been adopted and applied for load forecasting. In [49], Factored Conditional Restricted Boltzmann Machine (FCRBM) and Conditional Restricted Boltzmann Machine (CRBM) were assessed on a benchmark dataset consisting of fine-grained data collected from an individual residential customer. The results show that FCRBM outperforms ANN, SVM, RNN and CRBM.

A pooling-based Deep Recurrent Neural Network (DRNN) was proposed in [16]. During the training of the DRNN, data from different consumers are pooled together, which drastically increases the training data volume and postpones the overfitting of the network. It was validated on residential load data and outperformed ARIMA, SVM, and classical deep RNN.

In [22], multiple Deep Belief Networks (DBN) are trained first on the intrinsic mode functions (IMFs). The IMFs are obtained by decomposing the original data by Empirical Mode Decomposition (EMD) algorithm. The individual forecasts of the DBNs

are then combined together to generate the final prediction. The proposed method was compared with 9 other machine learning based methods, including Support Vector Regression (SVR), Random Forest (RF), ANN, and some EMD based variants. It was shown that the proposed method achieved the best performance for the majority of the tests.

A Long Short-Term Memory (LSTM) based deep learning model is proposed in [23]. In addition, the appliance consumption sequential data are fed into the model as well. It is shown in the research that the forecasting accuracy is remarkably improved by incorporating appliance measurements into the training data. The proposed method is validated on a real-world dataset and has shown an improvement of accuracy compared to ANN and K-Nearest Neighbours (KNN).

In [24], a Deep Residual Network (ResNet) based STLF method is presented. With the adoption of ResNet as the building block for the deep neural network, the network could go deeper and potentially have stronger predictive power. Though the proposed method achieved the best result among of the benchmarking methods, it was not validated on individual household level load data.

2.2.2Probabilistic Load Forecasting

Domestic household load are highly volatile and uncertain. They are difficult to be accurately predicted by point forecasting. Therefore, predictive models with the ability to quantify the various uncertainties are desirable [50]. With the ability to represent the uncertainties, probabilistic forecasting is receiving increasing attention [29].

Based on the way the forecasts are made, probabilistic forecasting methods can be divided into three groups [29]: 1) generating multiple inputs and feeding them to a point forecasting model of choice; 2) utilizing probabilistic forecasting models that are inherently capable of quantifying uncertainties, such as quantile regression; and 3) post-processing of results such as residual simulation.

There is a wide range of research works on input scenario generation. Generating different scenarios for temperature has been empirically proven to be very effective. Commonly used and effective methods for generating temperature scenarios are: using the temperatures from previous years with the dates being fixed [43], shifting the

temperatures from previous years [51], and bootstrapping the temperatures from previous years [52].

As for the probabilistic models that are inherently capable of generating probabilistic forecasts, a variety of them has been developed and proposed such as quantile regression, Bayesian models, Gaussian process, and nonparametric density estimation. In [53], a method that combines the gradient boosting method and quantile regression was proposed.

Probabilistic forecasting can also be achieved through the post-processing of the point forecast results. As summarized in [29], residual simulation and forecast combination are two common techniques to convert point forecast to probabilistic forecast. In [54], different methods for residual simulation were assessed, showing that adding the simulated residuals under the normality assumption could the forecasting performance. In [55], eight individual forecasts are generated and combined using the quantile regression averaging (QRA) method.

Different methods could also be combined together to form an ensemble forecaster. In [56], a total number of 13 quantile regression models are generated and combined together. The combination was achieved by formulating the QRA as a linear programming problem with the objective to minimize the pinball loss.

In summary, the volatile and uncertain nature of smart meter data has arisen tremendous challenges for both point and probabilistic forecasting. Traditional methods cannot capture the underlying patterns and make prediction very well. The emergence of deep learning addresses the challenges for its strong predictive power. However, deep learning models are much more complex than traditional models. Building deep models requires not only specific expertise but also requires much more effort as they are extremely difficult and time-consuming to optimise.

2.3 Customer Characterization

Energy consumers are characterized by not only the smart meter data but also the social-economic data. The two streams of data depict consumers from different angles. The smart metering data of the consumers demonstrate the consumption patterns of the consumers directly and intuitively, whereas the social-economic or social-demographic status of consumers provides deeper insights into the consumers.

Revealing the correlation and linking the two sets of data are crucial for the consumers themselves to save energy and for the suppliers/retailers to develop customized services.

Based on the research objective, existing research on characterizing the customers using smart metering data and socio-economic data can be broadly categorised into two types:

1) Research on inferring social-economic/social-demographic status from smart metering data [57-60];
2) Research on inferring smart meter data/characteristics from social-economic/social-demographic data [61-64].

2.3.1 Inferring Social-Economic Status

The inference of social-economic status can be achieved through either classification or regression in machine learning. In [65], Discrete Fourier Transform (DFT) is first applied to convert the load data to into frequency domain, at the same time reducing dimensions of input data. Classification and Regression Tree (CART) is then used classify the consumers into different groups based on the coefficients of harmonics in the frequency domain. In [66], a different transformation technique on the load data is applied to extract usage patterns, which is Non-negative sparse coding. Then the extracted patterns are fed into an SVM to make a classification prediction.

Domain knowledge based manual feature extraction and selection have also been used. In [58], features, such as the average consumption and the ratios of consumptions over different time periods are extracted from the original smart meter data. Then, different classifiers or regressors are tested to predict the social-economic status. In [67], a classification method named CLASS is proposed. CLASS also takes in the manually extracted features as input and make predictions through different classifiers, including KNN, Linear Discriminant Analysis (LDA), Mahalanobis classifier, and SVM.

Recently, deep learning based method has been proposed to infer the social-economic status in [60]. Deep CNN is used in the method to automatically extract features from the raw smart meter data. Results show that the proposed method outperforms traditional methods.

In summary, the traditional machine learning model based methods have reasonable model interpretability, for example, Multiple Linear Regression (MLR). However, the prediction is not as accurate as the deep learning based method. On the contrary, deep learning models have much stronger predictive power, but its interpretability is quite poor.

2.3.2Inferring Load characteristics

Not all consumers' load data are available as many of them are still using the conventional meters. Rather than installing costly smart meters, some research works have been conducted to infer load characteristics from consumers' social-economical information.

In [64], the MLR model is used to map the load characteristics from the social-economic data. As MLR is a linear model, the significance of each input feature can be easily explained by its corresponding weight.

As proposed in [61], load profiles of different consumers are firstly allocated to different groups using x-means, which does not require to pre-set the cluster number k as compared to k-means clustering. Two novel indicators, Energy Behaviour Correlation Rate (EBCR) and Indicator Dominance Index (IGD) are derived to collectively quantify and identify the significance of a social-economic question in helping differentiate different clusters.

In summary, research works on inferring load characteristics from consumers' social-economic data are limited and mainly utilize traditional methods. The prediction accuracy is relatively low compared to inferring the social-economic status. New methods are needed to tackle the challenge and improve the performance.

2.4 Phase Identification

Broadly, current phase identification methods could be divided into three groups: phase identification through extra equipment, similarity analysis of voltage data and analysis of load data.

2.4.1Phase Identification through Extra Equipment

Apart from the time-consuming and inefficient manual checking on site, the easiest method one can think of is by introducing signal injection equipment. In the context of cable identification, two methods are widely used and could be adapted to identify end-user's phase: light test and tone generation [68]. In the first method, a DC power source is connected at one end of a known cable and a light is connected at another end of a tested cable. The cable is identified by the illuminating of the light. While in the second method, the DC power source is replaced by a tone generator and the cable is identified by receiving the tone. The main disadvantage of these two methods is that all the cables have to be de-energized prior to the test, which would be unacceptable considering the long time it may take. Though paper [68] proposed a device that can identify phase under energized cable condition, the time it may take and the expense it may cost make it still unpractical. Similarly, Caird in his patent [69] presents a signal injection and receiving system to identify phases. In this system, a signal has to be injected into the phase line from the substation and a signal discriminator is needed at the user end. [70] introduces a similar signal injection method. This system is originally used to measure impedance. With the signal processing tool it presents, the method could be applied to identify phase connectivity. However, due to the need for installation of signal generators and receiving equipment in the above methods, the capital and maintenance costs will be increased greatly. Also, there are methods making use of smart metering data, i.e. load data and voltage data. The literature on these methods will be reviewed in the following two sections.

2.4.2Phase Identification through Analysis of Voltage Data

Another set of methods to perform phase identification is through the analysis of voltage data. Phase could be detected by similarity analysis of end-user's voltage profile and the phase line's voltage profile. In [71], a correlation based method is proposed. The research is developed from a solar city program where the feeder is aerial and all the consumption, voltage, and current data is recorded by smart meters at a frequency of every 15 minutes. The method is based on the assumption that voltage data within the same phase should share a similar shape. In this method, cross-correlation analysis is performed due to its popularity in feature and signal detection [72, 73]. Cross-correlation analyses the similarity between two time series

taking into consideration the time lag of the series. In other words, it will not only quantify how similar the two series are but also show at what time lag the two signals match each other best. (2-1) shows how cross-correlation of $x(n)$ and $y(n)$ is calculated.

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l) \qquad l = 0, \pm 1, \pm 2, \ldots \tag{2-1}$$

The cross-correlation could be normalised by (2-2).

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \tag{2-2}$$

where $r_{xx}(0)$ and $r_{yy}(0)$ are the mean square values of the signals of x(n) and y(n).

In [71, 74], for single-phase load, there are three possible connections, i.e. the load could be connected to either of the three phases. After performing cross-correlation of the single-phase consumer's voltage data with voltage data of the three phases of the transform, there will be three sequences of values. Each sequence is the correlation results between the load and the corresponding phase taking time lag into consideration. Then, the maximum value of each sequence is selected as an indicator to show correlated the load is with the particular phase voltage data. Finally, the maximum value of the three indicators is picked to show which phase the load is connected to. For three-phase load, the identification process is similar. The main difference is that there are six possible phase connections.

Similarly, [75] presents a method in which voltage data from the consumer side is correlated with the voltage data from the SCADA system. The SCADA system is applied to monitor the substation's condition whose voltage data is treated as the reference data. The correlation analysis is not achieved by performing cross-correlation. Alternatively, the method seeks time instances where the voltage change is unique to only one phase, i.e. out of the three phases, the voltage of only one phase changes significantly. After that, the consumers' voltage data variation is compared with the voltage change of the phase. The phase connectivity can then be predicted. To perform the correlation analysis more accurately, Short's [76] presents a method where a linear regression model for voltage is built taking consumer's and substation's

meter voltage and power into consideration. The proposed method could not only identify phase but also reconstruct the network's topology.

Chen [77] introduces a phase identification system (PIS) to detect phase connectivity. Though [78] is published under a different first author's name, it can be seen that the proposed system is exactly the same to the one in [77]. Generally, a signal within an AC system could be represented by a phasor, which can be uniquely determined by its frequency, amplitude and phase angle [79]. The frequency within one system is the system. Instead of analysing the similarity of the amplitude of voltage data, the proposed system focuses on another aspect of phasors: phase angle. The PIS requires high accuracy measurement of phase angles which is achieved by using phasor measure units (PMUs). Traditionally, PMUs are used to evaluate the stability of the power system [80, 81]. In this system, with highly synchronised PMUs, the PIS takes the substation's phase angles as a reference. By comparing the phase angles from the load end at the corresponding time instances with the reference angles, phase connectivity of the load could then be identified.

As a combination of Chen's [77, 78] and Pezeshki's [71, 74] methods, Wen [82] proposes a method by performing cross-correlation of voltage data as well as comparing phase angles. Moreover, instead of using PMU the proposed approach adopts a more accurate micro-synchrophasor (μPMU) [83]. The μPMUs are able to measure phase angles at a precision of 0.01°.

All the methods mentioned above make use of voltage data which will be a problem in the UK power system. Though the UK has installed various types of smart meters and AMIs, the voltage data is not commonly available in most smart meters. Even voltage data could be obtained, the sample rate in the UK is every 30 minutes. Compared with the 15 minutes rate in [71, 74] and hundreds of Hertz rate provided by PMUs or μPMUs, half an hour rate voltage data has not been validated to be applicable in phase identification. Moreover, PMUs or μPMUs still have not justified their cost to install them.

2.4.3Phase Identification through Analysis of Load Data

As mentioned, most smart meters in the UK do not provide consumers' voltage data. In DECC's technical specifications for smart meters [84], only consumption data is

required to record by smart meters. Based on the analysis consumption data, which are load data, several methods have been developed.

Dilek [85] describes a phase prediction system where tabu search (TS) is used as an optimisation procedure to find the phases of laterals. Phase is identified on condition that the calculated power flow using predicted phase information could well match the measured power flow. However, the method is based on the assumption that the topology of the LV network is known. This means that even though the phasing of a lateral is missing, the position of lateral along with its load data must be known. This requirement arises from the need to perform power flow analysis.

Arya [86] introduces a method based on the fact that the sum of the load within the same phase should be approximately equal to the load monitored at the substation. The method formulates the identification process into a Mixed Integer Linear Programming (MILP) problem with several linear equations. The connectivity of phases could be solved by calling IBM's CPLEX MIP solver. Moreover, basing on whether the data is noiseless, the objective function of this optimization problem would change accordingly. The identification process is based purely on end-users' and substation's load data.

All the approaches mentioned above are derived from the rule of conservation of electric charge and all of them make use of load data which is available in UK's smart meters. The main drawback is that both of the methods require a 100% penetration rate of smart meters in the LV distribution network. While in the UK, the installation of smart meters is not mandatory. According to the Government's report released [27], an estimated total of 1,317,900 smart meters (including smart gas meters) have been installed nationwide representing less than 3% of all domestic meters operated by large suppliers. As a result, traditional phase identification methods making use of complete load data are not applicable in the UK or nations with a similar issue. A novel approach to identify phase using incomplete load data is in urgent need to be developed.

Chapter 3

Transfer Learning for Short-Term Residential Load Forecasting

This chapter proposes to leverage the knowledge learned from one forecasting task to achieve more efficient and accurate forecasting for another by utilizing transfer learning.

3.1 Introduction

Short-term load forecasting (STLF) for the individual residential consumer was traditionally unattainable until the rollout of smart meters. In recent years, it is receiving increasing attention due to the wide deployment of Low Carbon Technologies (LCTs) at domestic households [87].

To facilitate the penetration of domestic LCTs, Peer-to-Peer (P2P) energy trading [88, 89], Demand Side Response (DSR) [90, 91], Energy Management System (EMS) [92, 93] along with some other technologies or frameworks [94, 95] have been proposed and studied. The precursor to achieve all the above-mentioned tasks is to be able to predict each household's load accurately [16]. The inherent uncertainty and volatility underlying domestic load behaviours pose tremendous challenges to traditional load forecasting methods [17-20]. It was not until the recent adoption of deep learning that the prediction performance has been elevated to an acceptable level [16, 21-24]. However, the improved accuracy by implementing deep learning is often accompanied by other inconvenience. Some of the main challenges regarding the implementation of the deep models are summarized as follows:

1) Training deep neural networks is difficult. A number of tricks and pitfalls should be considered before training the network. For example, depending on the choice of the optimizer, the weights must be initialized accordingly. Otherwise, the gradients during backpropagation are very likely to vanish or explode hence the network will not converge [96];

2) Training deep neural networks is time-consuming. The power of deep models is their ability to approximate really complex functions and it lies in the complexity of the models. In other words, deep models are extremely complex and could easily have millions of parameters, for example, a VGGNet [97] have about 140 million parameters. Training deep models, given a large amount of data, means gradually updating all those millions of parameters up to the point where the model optimally approximates the projection from the input to the output. It is very common to take days or even weeks to train a deep model without consideration of hyperparameter optimization;

3) Building deep neural networks is arduous and requires a certain amount of experience. There are numerous forms or architectures to choose from, e.g., Recurrent Neural Networks (RNNs) [98], Convolutional Neural Networks (CNNs) [99], and Residual Networks (ResNets) [100]. In addition, each category of deep neural network has its own sub-categories and different variants as well. Building a deep neural network means to choose from tens of various architectures and even mix and combine them;

4) Hyperparameter optimization or tuning of the deep neural networks is challenging. Regardless of the architecture of the network, there are various hyperparameters to optimize. For example, learning rate, the decay of learning rate, momentum in certain optimizers, number of hidden layers, number of neurons in each layer, form and degree of regularization, dropout rate, etc. The possibilities of combinations of different hyperparameters grow exponentially with the increase in hyperparameter number. Traditional grid search or cross-validation are simply impractical in deep learning [101].

In addition to those issues about the models, one other major concern comes from the requirement of data for training the models:

1) Individual consumers' data are not large enough to train a deep model. According to the research reported in [16], deep models are prune to overfit using just one consumer's load data to be trained on;

2) Individual consumers' data are not equal in amount. Take the UK electric market as an example. Domestic consumers have the rights to decide whether to install the smart meters or not and to decide when to install smart meters. Consequently, different consumers would likely to have different length of available historical data. Some consumers with sufficient data would then benefit more accurate forecasting brought by deep learning, whereas those with little data would not;

3) New consumers have hardly any data. The predictive models for those consumers who have just have smart meters installed would perform unsatisfactorily as the amount of data are quite limited.

Considering the above-mentioned reasons, building a deep model for every one of the consumers is not only challenging but also unrealistic in certain situations. Similar

issues could be found in computer science domain as well, where transfer learning is widely applied to tackle the problems and has made great success especially in the fields of Computer Vision (CV) and Natural Language Processing (NLP) [102].

In this chapter, a transfer-learning-based framework is proposed to tackle the addressed issues by leveraging the learned knowledge from other consumers. The key contributions of the research are summarized as follows:

1) The applicability of transfer learning in STLF has been examined;

2) The transferability of various forms of deep models has been examined;

3) The transferability of features in different layers has been examined.

The remainder of this chapter is organized as follows: Section 3.2 introduces the concept of transfer learning and the rationale of applying it in STLF. Section 3.3 explains the setup of the experiments. Section 3.4 presents the experimental results and discussion. Section 3.5 draws the conclusions.

3.2 Transfer Learning

Machine learning, especially deep learning, has achieved significant success in many domains. However, the effort, computational power and amount of data that are needed to train a deep model are beyond what most researchers can obtain. In such a context, transfer learning is brought into play.

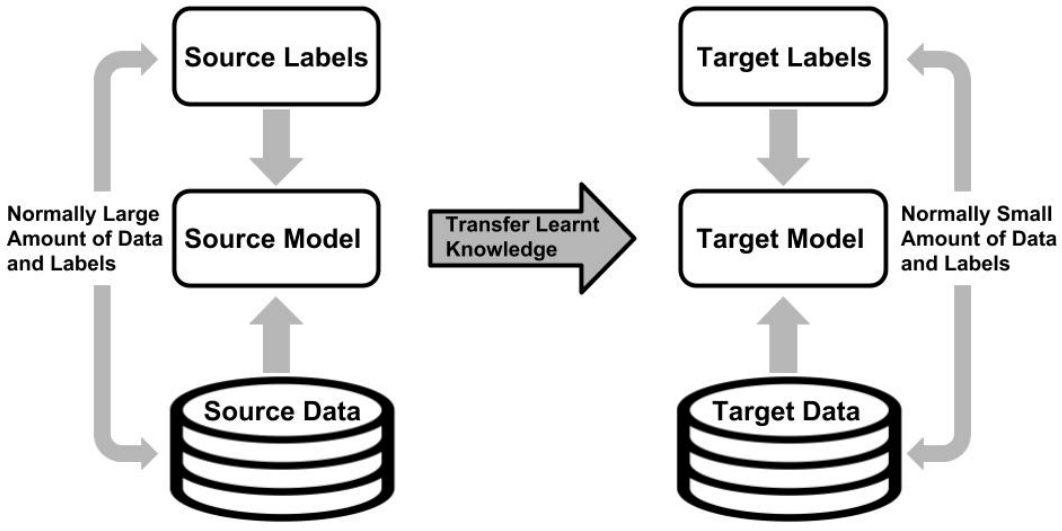

Figure 3-1 General form of transfer learning

3.2.1Introduction of Transfer Learning

Transfer learning is to utilize the knowledge learned from solving one task to solve a different task. Figure 3-1 depicts the general perception of (supervised) transfer learning. A more rigorous definition will be given after introducing two important concepts, i.e., domain and task.

A domain $\mathcal{D}$ is characterised by a feature space $\mathcal{X}$ and a marginal probability distribution $P(X)$, where $X = \{x_1, x_2, \ldots x_n,\} \in \mathcal{X}$ [103]. For instance, if the task is to predict the next day's load using seven-day-long load data, $\mathcal{X}$ is then the space of all available seven-day-long load data and X is a sample. Two domains are defined as different if they either have different feature space $\mathcal{X}$ or have different marginal probability distribution $P(X)$.

Given the domain $\mathcal{D} = \{\mathcal{X}, P(X)\}$, a task is denoted as $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$ and is composed of a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ [103]. Following the previous example, $\mathcal{Y}$ is then the space of all the daily load data to be predicted.

Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, transfer learning is to improve the learning of predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$ [103].

Transfer learning has been widely studied and successfully applied in many fields, including text mining [104], image processing [105], collaborative filtering [106], activity recognition [107] and etc. With different criterions of interest, transfer learning could be classified differently. For deep-learning-based transfer learning, it can be divided into three categories, i.e., supervised, semi-supervised, and unsupervised [108]. The task of STLF using transfer learning belongs to supervised transfer learning, where both the source and target domains provide the correct labels.

3.2.2The rationale of Applying Transfer Learning in STLF

In recent years, smart meters have been rolled out to a large number of domestic households across the world. The timely sampling ability of electric load by these smart meters is creating one of the largest datasets in the power industry. The large amount of data makes it ideal for them to be exploited by deep learning. However, the challenges and issues come with deep models which are discussed in Section I are limiting the deployment and utilization of them. Similar issues in computer science

domain exist as well and have been successfully solved by using transfer learning. The reasons why transfer learning is ideal to assist deep models to tackles these issues are summarised as below:

1) The reusability of trained models

One way of transfer learning in deep learning is to use the new data to train on a pre-trained model where only a few layers will be replaced and are trainable. Since the models have already been built and trained, those who want to use these models for transfer learning will not be bothered to design, build and train the deep model from scratch. Additionally, it is found that the first few layers in deep models are trying to learn low-level and general features and the last several layers transition to be task-specific, hence less transferable. So normally only the last or the last few layers are replaced and retrained using the new data in transfer learning. The computational burden will be dramatically reduced as there are much fewer parameters in the trainable layers. This is ideal for domestic STLF as new consumers are coming into the system every day and using transfer learning not only saves the trouble to develop deep models for all consumers but also greatly reduces the computational requirement to re-train the model.

2) The reusability of learned knowledge

The knowledge that deep models have learned from the data are embedded in the models themselves. Specifically speaking, the weights and biases within the deep models represent what the models have learned. Transferring the model to another domain or task is actually transferring the knowledge which will improve the performance on similar or related target domains or tasks.

The transferability of knowledge is extremely useful when there is insufficient data in the source domain or task. In such a situation, traditional machine learning models would break down or performs unsatisfactorily due to the lack of data. However, the model developed using transfer leaning will perform reasonably well by leveraging the knowledge learned from the source domain and task, where there are sufficient data. For example, there are a large number of labelled images on cats, cars, ships, etc. Many successful deep models have been developed on these data. Some of popular models include AlexNet [99], GoogLeNet [109], VGGNet [97] and ResNet

[100]. On the contrary, there is a limited amount of labelled medical images. Building and training a model upon these data would yield quite poor performance. Using the deep model and knowledge learned from other images for transfer learning would significantly improve the predictive performance on these medical images [110].

As discussed in Section I, domestic consumers would have a variable length of available load data. Some may be suitable for building deep models and some may not. The rationale for using transfer learning for STLF, apart from the reduced computational requirement, is the improvement of performance on tasks with relatively fewer data. Research in [111] has shown that even for tasks with similar size of labelled data, transfer learning can still help improve the performance on target task. By applying transfer learning, a well-designed and fine-tuned deep model could be built on consumers with sufficient data. The model can then be transferred to new consumers or consumers with relatively small data. And those consumers can enjoy a significant improvement on their load forecasting accuracy.

3.3 Experiment Setup and Procedure

The goal of the experiment is to explore the applicability of transfer learning in domestic STLF. In order to achieve that, the transferability of different layers in a deep model and the transferability of various forms of deep models should both be considered simultaneously. Specifically, in this experiment, two commonly used architectures for load forecasting, Multi-Layer Perceptron (MLP) and Long Short Term Memory (LSTM), are selected.

3.3.1 Data Description

The data used in this experiment are taken from the Smart Metering Electricity Customer Behaviour Trials (CBTs) initiated by the Commission for Energy Regulation (CER) in Ireland [112]. The trials spanned from July 2009 to December 2010 and contain over 5000 consumers. The full anonymized data sets are publicly available online and contain not only the half-hourly sampled electricity consumption (kWh) from each participant but also the customer type, tariff, and stimulus description, which specifies customer types, allocation of tariff scheme and Demand Side Management (DSM) stimuli.

The focus of the experiment is to validate whether transfer learning would help improve STLF performance for general domestic consumers. Hence, only the residential (type 1) consumers with the controlled stimulus (stimulus E) and controlled tariff (tariff E) are most appropriate for the experiment as the majorities of consumers outside the trial are of the type. However, it is not only impractical but also not necessary to test every one of the consumers for all the 920 1-E-E consumers. Therefore, a subset of 100 randomly selected 1-E-E consumers are used in this experiment which is reasonably sized and practical to be implemented.

3.3.2MLP Deep Model Transfer Learning

MLP or Feedforward Networks are a type of acyclic neural networks [113], which can be graphically presented as in Figure 3-2.

MLPs are one of the most fundamental and quintessential models in deep learning and have proven themselves to be effective in a wide range of tasks. It is reasonable and common practice to start a deep learning task with MLP. The experiment with MLP consists of several steps which are explained below.

1) Data Munging and Partitioning:

   For simplicity, an additional criterion is applied before randomly selecting the 100 consumers, i.e., the data of consumer should consecutively range from July 14th 2009 to December 31st 2010. Hence, all the selected consumers have the same amount of data. The list of the selected consumers can be found in Appendix A.

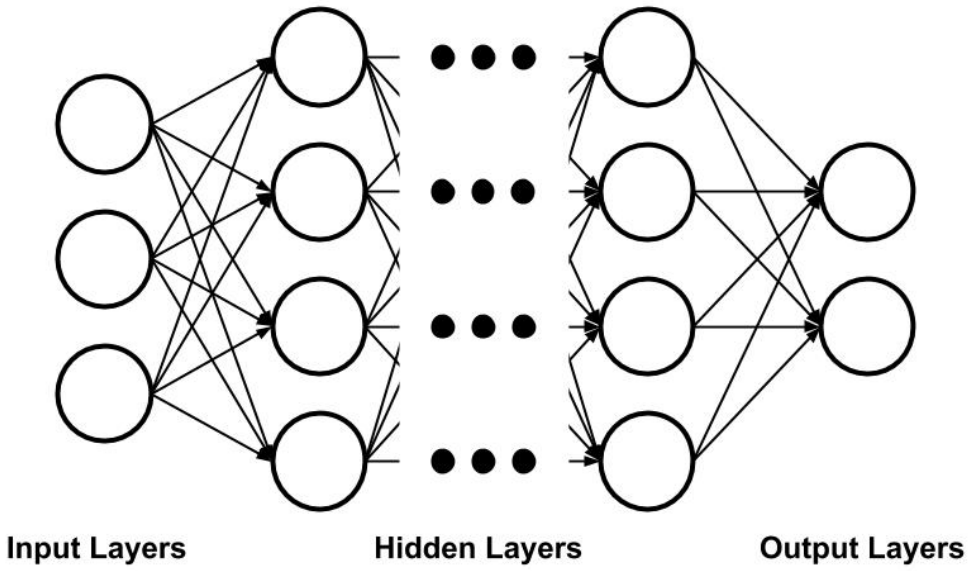

Figure 3-2 Typical structure of MLP

The data are then reshaped and standardised according to (3-1)

$$z = \frac{x - \mu}{\sigma} \tag{3-1}$$

where z is the rescaled value, x is the original value, μ is the mean of the original data, and σ is the standard deviation of the original data.

The obtained data for each consumer are further divided into three disjoint parts, i.e., training set, validation set, and test set. The proportions are 70%, 10%, and 20% respectively.

2) Hyperparameter Optimization:

As mentioned earlier, training deep neural networks involves a larger number of parameter settings. For building MLPs, this experiment takes the following hyperparameters into consideration.

- Initial learning rate
- Number of hidden layers
- Number of neurons per layer
- Activation function
- Weight initialization
- Dropout rate
- Batch size
- Optimizer
- Regularizer and its strength

The number of possible combinations of hyperparameters grows exponentially with added dimensions of the search space. To reduce the complexity, some of the

Table 3-1 Predefined hyperparameters for MLP

Hyperparameter	Setting
Activation function	ReLu
Weight initialization	Glorot
Regulazizer	L2
Regularization strength	0.1
Optimizer	Adam
Adam Beta_1	0.9
Adam Beta_1	0.999

hyperparameters are settled prior to the optimization, which can be found in Table 3-1. These predefined hyperparameters are either widely used in deep learning applications as default settings or have been preliminarily tested.

As for the optimisation strategy, random search is chosen over grid search. The reason is two-fold. Firstly, the dimension of the search space is so high that searching over all the possible combinations would be either too time-consuming or impossible. Secondly, some of the hyperparameters matter much more than others. So performing a random search rather than a grid search is more likely to find a good set of hyperparameters as repetitions of the search along the trivial hyperparameters are avoided. Comprehensive research on why random search is better than grid search in deep learning could be found in [111].

The loss function used in the experiment is the mean squared error (MSE), which takes the mean of the squares of the losses as given in (3-2).

$$L_i = \frac{1}{n}\sum_{t=1}^{n}(y_i^t - \hat{y}_i^t)^2 \tag{3-2}$$

where L_i is the MSE loss for consumer i, n is the number of time steps to be predicted, y_i^t is the actual standardized load value of consumer i at time t, and $\hat{y}_i^t$ is the predicted standardized load value of consumer i at time t.

One assumption that was made for the hyperparameter optimisation in this experiment is that despite the diversity and volatility of domestic consumers, the complexity of the required deep neural networks is the same. The assumption is

reasonably solid as the consumers selected for this experiment are of the same type and have exactly the same amount of data, which is assured during data munging. Consequently, the optimised hyperparameters for any one of the 100 consumers are also optimal for the rest and hence could be directly applied to them. In other words, only one consumer is needed for hyperparameter optimisation instead of all the 100 consumers.

In this experiment, the consumer with ID 1002 was selected for the hyperparameter optimisation. A total number of 10,000 random combinations of hyperparameters have been generated and evaluated. It was executed on 10 parallel High Performance Computing (HPC) clusters to accelerate the optimisation process. The 10,000 models for evaluation are trained on the training set specified earlier in this chapter and validated on the validation data set. The training and validation are performed in an alternative order so that early stopping strategy can be applied to avoid overfitting. Particularly, early stopping is set to 4 epochs during the optimisation, which means that if the validation accuracy has been decreasing for 4 epochs, the deep neural network will stop training. Lastly, the trained models are tested against the test dataset. The hyperparameters of the model with the best test performance will be selected.

3) Training Individual MLP Models:

After the optimal set of hyperparameters are found, all of the 100 consumers are then individually trained on them. Towards the end of the training, the learning rate is reduced to fine-tune the models. Subsequently, all the 100 models are saved for later experiments, i.e., all the weights, bias, and connections of the trained networks are stored.

4) Consumer Similarity Analysis:

The transferability of the trained model from one domain to another is closely related to the similarity of the source and target domains [103]. In the context of load forecasting, it is assumed that the transferability is correlated with the similarity of load profiles between the source consumer and the task consumer. In order to assess the assumption, the similarity between different consumers should be calculated first.

A number of metrics are available to quantify the similarity, e.g., Euclidean distance, Minkowski distance, KL divergence, cosine similarity, etc. Euclidean distance of the mean daily load profiles of the consumers is chosen as the similarity metric for its simplicity and versatility. Taking the mean, instead of all the load profile, is to reduce the dimensionality of the original data, which is quite high-dimensional and would lead to unreliable result for distance-based metrics. The pair-wise Euclidean distance of all the consumers is then calculated according to (3-3).

$$d_{Euclidean} = \sqrt{(x_i - x_j)^T (x_i - x_j)}, \qquad i \neq j \tag{3-3}$$

where x_i represents the mean load vector of consumer i, x_j represents the mean load vector of consumer j.

5) Transferring Base Models to Consumers With Varying Similarity:

In the last two steps, the base models have been trained on individual consumers and their similarity has been computed. The final step is then to transfer different models with a varying number of layers to different consumers.

Consumer 1 is taken as an example for concreteness. Out of the 99 remaining consumers in our data set, the most similar consumer to consumer 1, the most different consumer from consumer one, and the consumer with an intermediate degree of similarity with consumer 1 are selected. The trained models of the three consumers are then transferred to the base consumer, which is consumer 1 in the example. However, it would be naïve and non-optimal if all the layers in the model

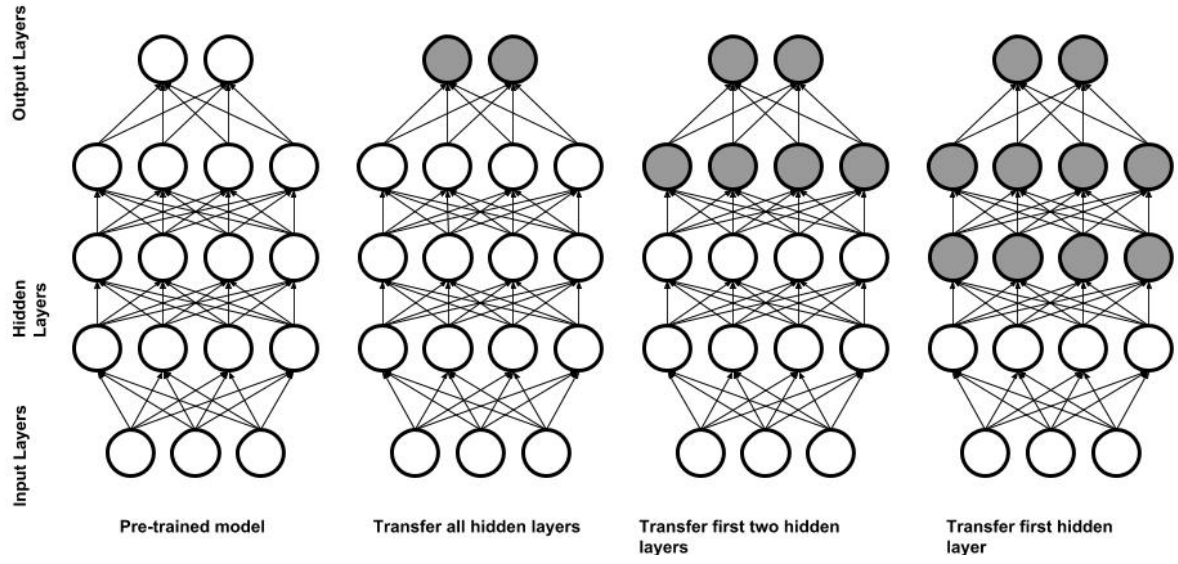

Figure 3-3 Transferring different layers in a MLP model

are transferred as different layers in deep neural networks have varying degrees of transferability. In order to evaluate the transferability of different layers and find out the optimal number of layers to transfer, an exhaustive transfer test would be conducted on each model. To put it into perspective, if the model to be transferred has n hidden layers, there will be a total number of n ways to transfer the model, i.e., start with transferring only the first hidden layer, all the way to transferring all the hidden layers. Figure 3-3 illustrates the transferring of different layers in a pre-trained model with three hidden layers as exan ample. The hollow circles in the figure represent the transferred neurons whereas the shaded circles represent the neurons (layers) that need to be re-trained. In other words, the transferred layers are held constant and the added new layers are re-trained on the same training data.

3.3.3LSTM Deep Model Transfer Learning

Currently, state-of-the-art deep learning models specifically designed for handling sequential data are Recurrent Neural Networks (RNNs). They are a family of networks that are specialized for processing a sequence of values. They attempt to model the time dependency of sequential data by feeding back the output of a neural layer at time (t) to the input of the same layer at time $(t+1)$. A typical RNN is illustrated in Figure 3-4.

The figure shows what a normal RNN cell would look like. Internally, there still is one unit. The unfoldment is more on the temporal domain. It means that, internally, there still is one RNN cell. The horizontal cascading of the cells only indicates that the output of the previous time $(t-1)$ step from the cell is fed to the same cell but at the current time step (t). Basically, it is the same cell iterating over the sequential data one at a

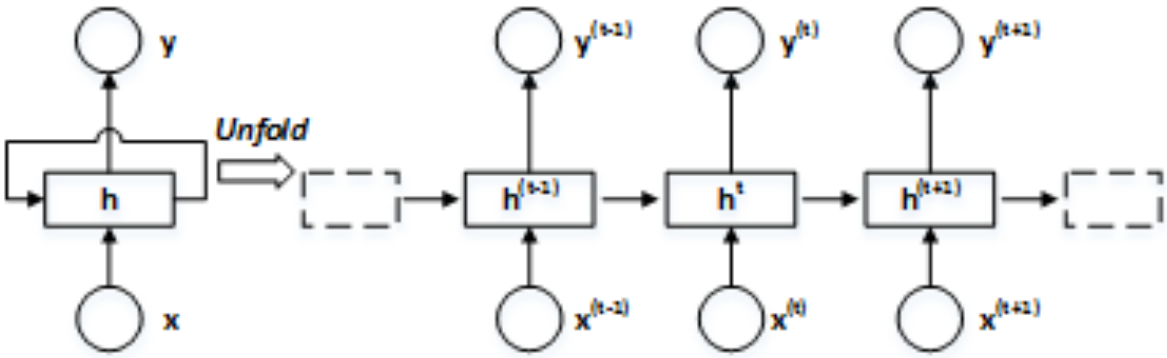

Figure 3-4 Unfolding a RNN cell

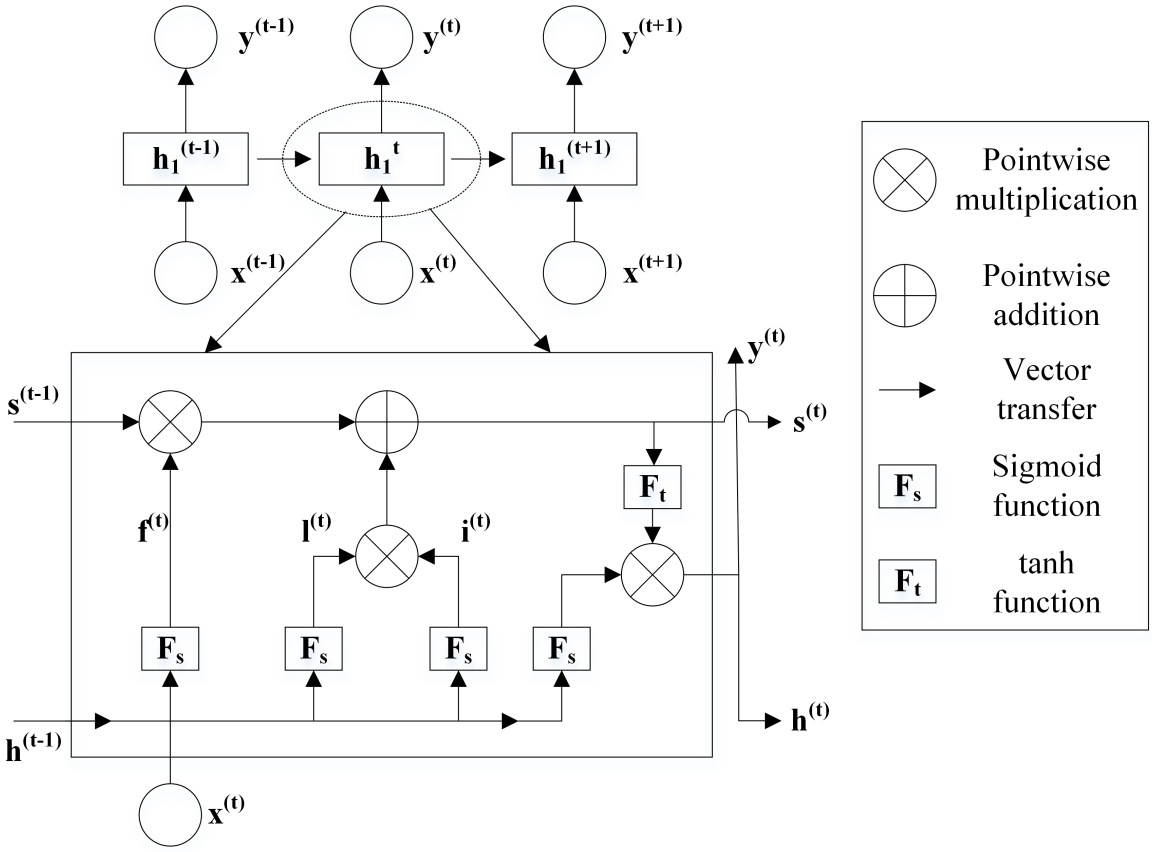

Figure 3-5 Unfolding a LSTM cell

time, taking the previous output (some states) into consideration. In such a way, the network prediction would take the time dependency into consideration. However, this typical RNN structure looks back at the data only one time step before the prediction. The ability to connect to previous data is very limited. Consequently, RNNs with this generic form of cells cannot memorize long-term information.

Long short term memory networks (LSTMs) are a special kind of RNNs that were developed to deal with the addressed long-term dependency problems. The LSTM block diagram is illustrated in Figure 3-5. It could remember information for a long period of time. Besides passing the current output to the next time step, some information named cell state is also passed.

Compared to other RNNs that only have the outer recurrence by connecting cells recurrently, LSTM cells also have an internal recurrence (a self-loop) [113]. Different from the typical RNN cells, there are four gates in an LSTM cell, each of which is fundamentally a single layer neural network. Three of the gates are sigmoid and the other is tanh. They work collaboratively to update the cell state and generate output.

LSTM networks have manifested themselves to learn long-term dependencies more efficiently than the simple RNN cell based networks on different tasks [113].

The experiment with LSTM is almost the same as the one for MLP. However, due to the innate different structures of the two models, some aspects regarding the hyperparameter optimisation and transferring of the layers are performed differently.

1) Data Munging and Partitioning:

The munging and partitioning of the raw data are exactly the same as for MLP.

2) Hyperparameter Optimization:

The structures of MLP and LSTM cell are fundamentally different. As a result, the hyperparameters that need to be optimised are different. For building LSTM networks, this experiment takes the following hyperparameters into consideration.

- Initial learning rate
- Number of hidden layers (LSTMS cells)
- Number of units in LSTM cell gates
- Activation function
- Weight initialization
- Dropout rate
- Batch size
- Optimizer
- Regularizer and its strength

The hyperparameters that are settled prior to the optimization can be found in Table 3-2. As for the optimisation strategy, random search is chosen over grid search for the same reasons that were mentioned earlier. Moreover, MSE is again used as the loss for the optimisation.

The assumption that was made in MLP case holds for LSTM networks for the same reasons as well. The assumption is that despite the diversity and volatility of domestic consumers, the complexity of the required deep neural networks is the same. Therefore, only one consumer is needed for LSTM hyperparameter optimisation instead of all the 100 consumers.

In this LSTM experiment, the consumer with ID 1002 was selected for the hyperparameter optimisation again. A total number of 10,000 random combinations

Table 3-2 Predefined hyperparameters for LSTM

Hyperparameter	Setting
Activation function	ReLu
Weight initialization	Glorot
Regulazizer	L2
Regularization strength	0.1
Optimizer	Adam
Adam Beta_1	0.9
Adam Beta_1	0.999

of hyperparameters have been generated and evaluated. The 10,000 models for evaluation are trained on the training set specified earlier in this chapter and validated on the validation data set, which is the same as for MLP. In addition, the early stopping is set to 4 epochs during the optimisation as well. Lastly, the trained models are tested against the test dataset. The hyperparameters of the model with the best test performance will be selected.

3) Training Individual LSTM Models:

The training process of all the individually LSTM models is the same as for MLP.

4) Consumer Similarity Analysis:

The similarity consumer analysis is the exactly the same as for the MLP. Hence, there is no need to repeat this step. The results gained in the previous MLP experiment can be directly applied in this LSTM experiment.

5) Transferring Base Models to Consumers With Varying Similarity:

Again, the final step is the same as for MLP which is to transfer different models with a varying number of layers to different consumers. The difference is that in MLP networks, the layers to be transferred are fully connected or dense layers, whereas in LSTM networks, the transferred layers are LSTM cells.

The full and detailed results are given in the following next section, complemented by the author's analysis.

3.4 Results and Discussion

The results of the experiments for MLP and LSTM are presented in this section.

3.4.1Transfer Learning with MLPs

As mentioned in Section 3.3.2, 10,000 random searches on the hyperparameters have been conducted. The optimal hyperparameter set was given in Table 3-3.

Table 3-3 Optimal hyperparameters for MLP

Hyperparameter	Setting
Learning rate	0.00696
Number of hidden layers	4
Number of neurons for each layer	40
Dropout	0.120592
Batch size	23

These hyperparameters, combined with the predefined ones in Table 3-1, are then used to create deep MLPs for every consumer of the selected 100 consumers.

To assess the performance of the models, 4 widely used metrics in deterministic load forecasting area are chosen, including mean absolute percentage error (MAPE), mean absolute error (MAE), root-mean-square error (RMSE), and normalised root-mean-square error (NRMSE), which are given in respectively.

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y^t - \hat{y}^t}{y^t}\right| \tag{3-4}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y^t - \hat{y}^t| \tag{3-5}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y^t - \hat{y}^t)^2} \tag{3-6}$$

$$NRMSE = \frac{RMSE}{y_{max} - y_{min}} \tag{3-7}$$

where n is the number of time steps to be predicted, y^t is the actual standardized load value of at time t, and $\hat{y}^t$ is the predicted standardized load value at time t, y_{max} and y_{min} are the maximum and minimum value in the test set.

Note that the four metrics are measuring the forecasting performance from four different perspectives, whereas the loss function that is used to train the deep network is only taking MSE into consideration. In other words, it is very difficult to achieve optimal for all the four metrics simultaneously.

As proposed earlier in this chapter, the transferring performance may be affected by the number of layers to be transferred and the similarity of the source and target domains. Due to the high-dimensional nature of the experiment data, it is impossible to demonstrate all the suspected relations of the data in one figure or table. Hence multiple figures are generated for a clearer demonstration purpose.

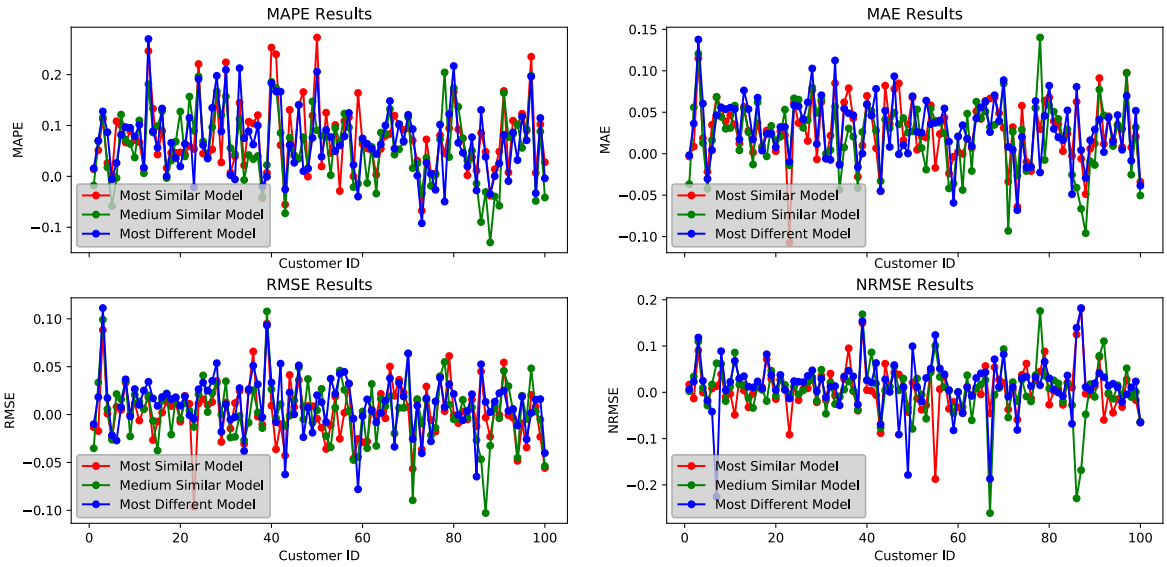

Figure 3-7 Performance improved by transferring the first hidden layer in MLP

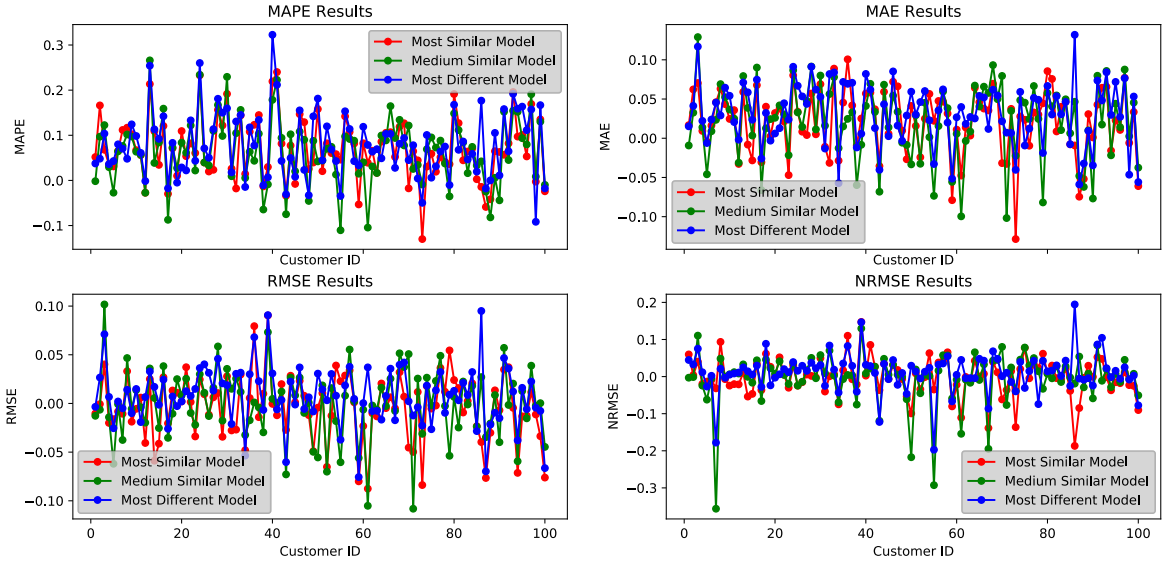

Figure 3-6 Performance improved by transferring the first two hidden layers in MLP

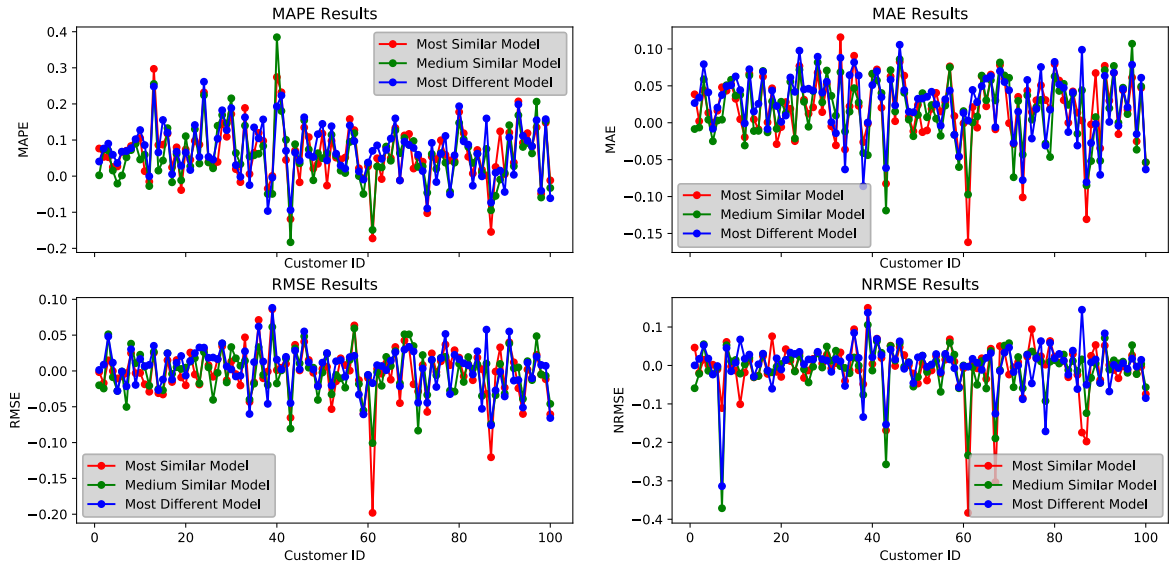

Figure 3-9 Performance improved by transferring the first three hidden layers in MLP

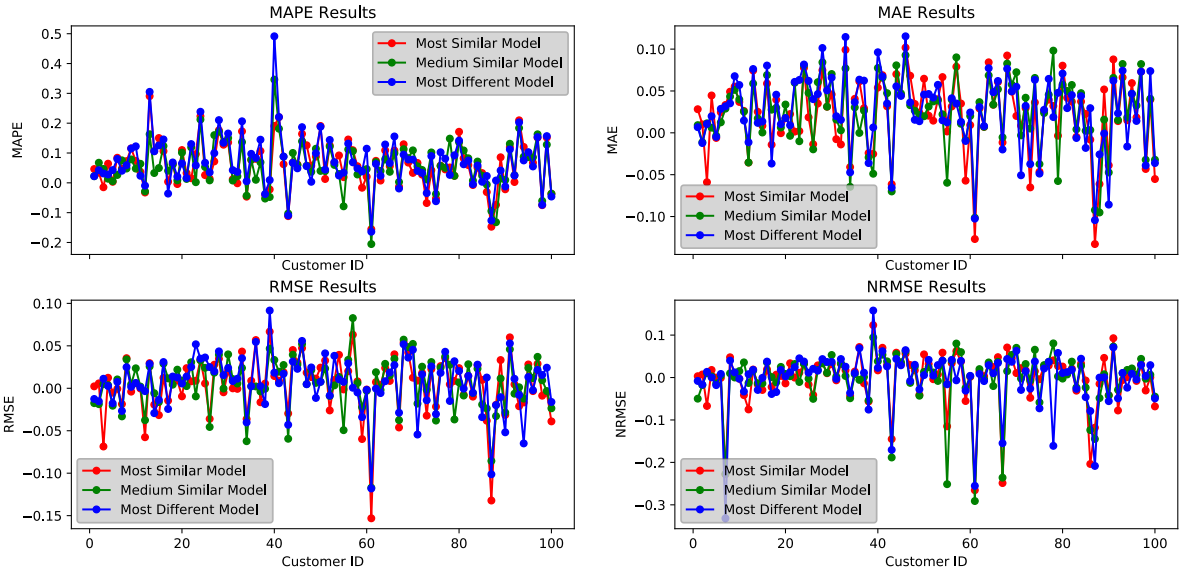

Figure 3-8 Performance improved by transferring all four hidden layers in MLP

Figure 3-7, Figure 3-6, Figure 3-9, and Figure 3-8 demonstrate the performance improved by transferring only the first hidden layer, the first two hidden layers, the first three hidden layers, and all the four hidden layers respectively. The horizontal axis represents the customers. Each subplot in the figure counts for one metric out of the four. Moreover, different colours indicate that the models are transferred from customers with different similarity levels with respect to the base model customer.

The values presented in the figures are the relative improvement/decline with respect to the base model forecasting performance given by (3-8).

$$Error_{relative} = \frac{Error_{base} - Error_{transfer}}{Error_{base}} \quad (3\text{-}8)$$

where $Error_{base}$ is the error of the base model and $Error_{transfer}$ is the error after transfer learning.

The reason for using the relative error with respect to the base model error is that the errors of the 100 consumers are not on the same scale. Plotting all the original errors in the same figure will mask a lot of the information for the models who already present small error values. Instead, rescaling all the error values to relative values will make the values more intuitive and comparable.

As can be seen for the figures, regardless of which layers are transferred, though there are a few cases where the performance decline has been seen, for the majority of consumers, the forecasting performance has been improved. This in the figures is explained by that most of the lines are above zero. Nevertheless, some metrics have been improved more significantly than the others. Specifically, the forecasting performance in terms of MAPE and MAE are significantly improved than RMSE and NRMSE. Additionally, it is observed that the most downwards spikes that are below zero in the figures are mostly in red and green, which indicates that transferring from a non-similar consumer seems to improve performance more significantly. That is quite the opposite to the previous hypothesis. The potential reason is that though transferring a model trained on one consumer to another is transferring from one domain to a different one. However, the target domain and the source domain are not drastically different. They are very similar in many aspects, unlike other widely acknowledged transfer learning examples. Consequently, being mostly similar, transferring relatively dissimilar consumers would actually provide more information and insights. The extreme example would be that if the data of the target consumer and the source consumer in our forecasting task are identical, the performance would not be improved at all.

To better demonstrate the impacts of transferred layers in transfer learning, the results are then replotted in Figure 3-10, Figure 3-11, and Figure 3-12. These figures are plotted using the same results for plotting Figure 3-7, Figure 3-6, Figure 3-9, and Figure 3-8. The difference is that this time for each figure, the similarity level is held constant so that the relation between transferred layers and the performance change

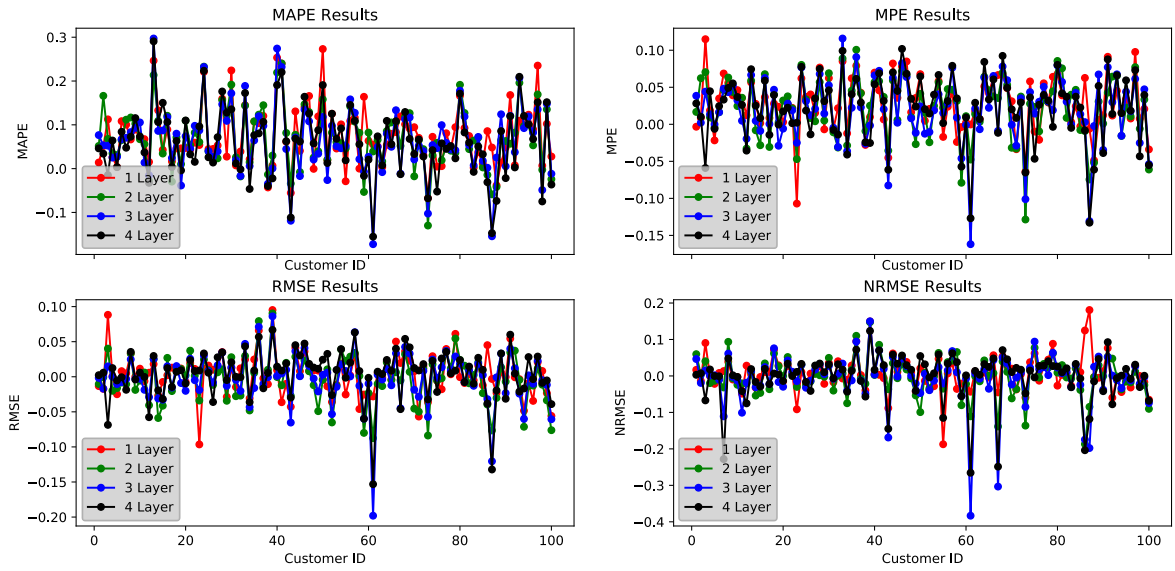

Figure 3-10 Performance improved by transferring from the most similar consumer for MLP

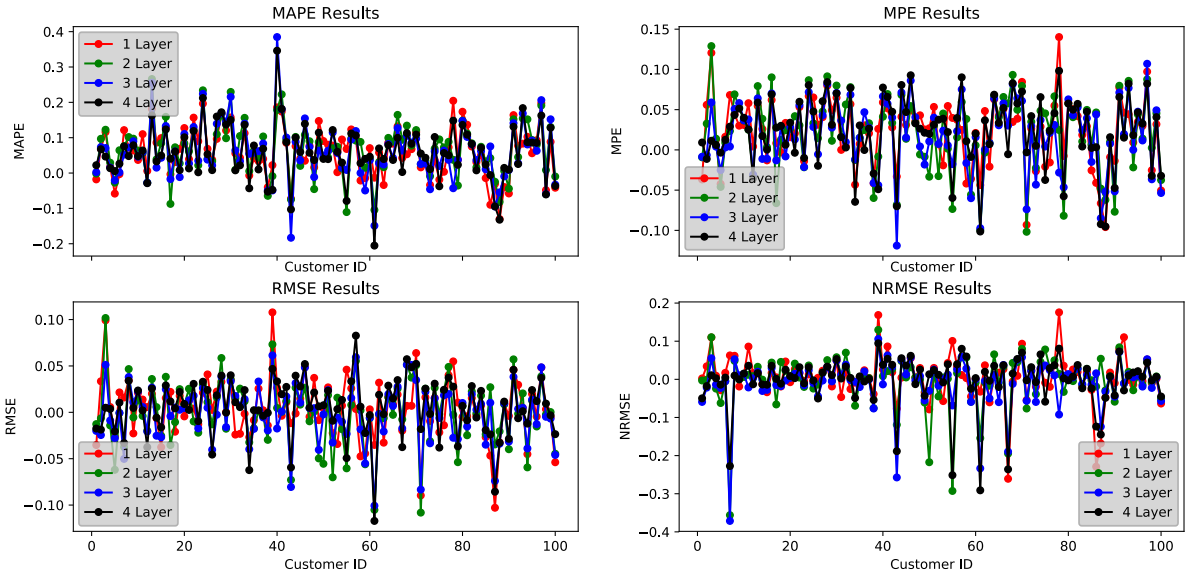

Figure 3-11 Performance improved by transferring from consumer with medium similarity level for MLP

can be illustrated more clearly.

As depicted in those figures, no matter which layers are transferred, the performance of most models have been improved. Yet, transferring all 4 layers seems to crease the performance for some consumers in terms of RMSE and NRMSE. In order to more precisely and clearly understand how various factors are impacting the performance, the mean values of all the metrics are calculated and listed in Table 3-4, Table 3-5, Table 3-6, and Table 3-7.

Evidently, MAPE and MAE performance is improved most significantly. And

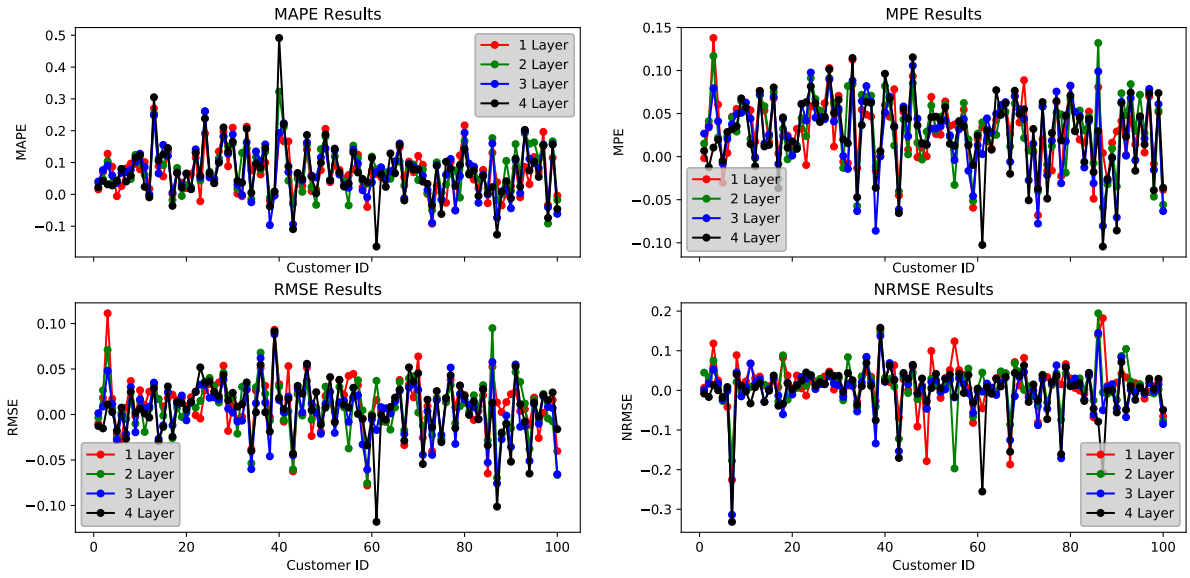

Figure 3-12 Performance improved by transferring from the most different consumer for MLP

transferring the first layer and the first two layers improve the performance the most. In addition, as can be seen in the four tables, transferring from a dissimilar consumer tend to improve the performance the most. However, the exception in the table is transferring only the first layer and measure the performance in MAPE, where transferring the most similar consumers brings the most significant improvement. After observing Figure 3-7 again, it can be found that for only several consumers, the most similar transferring strategy outperforms the most different transferring strategy. In addition to that, for the above-mentioned consumers, the most similar transferring strategy outperforms the most different transferring strategy by a relatively large degree. It can be deduced that it is those cases that list the mean MAPE improvement for transferring similar consumers. Hence, the general trend still holds, which is that transferring dissimilar consumers rather than similar consumers to the base models is more beneficial for load forecasting task.

Table 3-4 MLP MAPE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	7.50%	5.93%	6.87%
2 Layers	7.02%	6.54%	7.80%
3 Layers	6.31%	5.86%	7.16%
4 Layers	5.99%	5.71%	7.10%

Table 3-5 MLP MAE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	2.68%	2.22%	3.02%
2 Layers	2.12%	2.11%	3.16%
3 Layers	1.89%	1.75%	2.68%
4 Layers	2.17%	2.30%	2.67%

Table 3-6 MLP RMSE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	0.25%	0.36%	1.13%
2 Layers	- 0.38%	- 0.09%	0.77%
3 Layers	- 0.28%	- 0.17%	0.38%
4 Layers	0.39%	0.55%	0.61%

Table 3-7 MLP NRMSE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	0.60%	0.56%	1.52%
2 Layers	-0.21%	-0.46%	1.07%
3 Layers	-0.77%	-0.98%	0.10%
4 Layers	-0.34%	-0.36%	-0.26%

3.4.2Transfer Learning with LSTMs

The optimal hyperparameter set for building LSTM is listed in Table 3-8. These hyperparameters, combined with the predefined ones in Table 3-2, are then used to create deep LSTMs for every consumer of the selected 100 consumers.

Table 3-8 Optimal hyperparameters for LSTM

Hyperparameter	Setting
Learning rate	0.00286048
Number of hidden layers	4
Number of neurons for each layer	40
Dropout	0.327461
Batch size	128

The average improvements on MAPE, MAE, RMSE, and NRMSE are given in Table 3-9, Table 3-10, Table 3-11, and Table 3-12 respectively. Detailed plots for individual consumers can be found in Appendix A.

Table 3-9 LSTM MAPE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	1.31%	1.64%	1.76%
2 Layers	0.14%	1.04%	0.63%
3 Layers	-0.27%	0.68%	-0.16%
4 Layers	-1.07%	0.26%	-1.53%

Table 3-10 LSTM MAE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	-0.13%	-0.16%	0.15%
2 Layers	-0.33%	-0.14%	-0.07%
3 Layers	-0.45%	-0.13%	-0.34%
4 Layers	-0.74%	-0.31%	-0.74%

Table 3-11 LSTM RMSE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	-0.43%	-0.54%	-0.06%
2 Layers	-0.24%	-0.14%	0.02%
3 Layers	-0.23%	-0.05%	-0.16%
4 Layers	-0.38%	-0.12%	-0.29%

Table 3-12 LSTM NRMSE performance improvement results

Transferred Layers	Most Similar	Medium Similarity	Most Different
1 Layer	0.03%	0.09%	0.44%
2 Layers	-0.25%	0.76%	0.17%
3 Layers	-0.38%	0.66%	-0.44%
4 Layers	-0.63%	0.31%	-0.86%

As can be seen from the tables, MAPE and NRMSE can be improved by transferring the first two hidden layers in the network, whereas MAE and RMSE drop almost trivially. The reason is that the LSTM network already has a strong predictive power. Using the LSTM network on its own without transfer learning has already achieved an accurate prediction. Hence, the prediction may not be further improved by introducing transfer learning to the LSTM network. Additionally, the target domain and source domain have the same amount of data, which may not provide extra information to improve the prediction.

In this chapter, a transfer-learning-based framework is proposed to tackle the addressed issues by leveraging the learned knowledge from other consumers. The key contributions of the research are summarized as follows:

1) The applicability of transfer learning in STLF has been examined;

2) The transferability of various forms of deep models has been examined;

3) The transferability of features in different layers has been examined.

3.5 Chapter Summary

In this chapter, a novel transfer-learning-based framework is proposed to perform load forecasting. Two powerful and popular deep models that are commonly used for sequential data prediction are evaluated. The key findings are:

1) Transfer learning is suitable for load forecasting, in the sense that it could significantly simplify the deployment of deep models and improve the prediction performance ;

2) The LSTM network is more capable of forecasting load than MLP network;

3) Even when the target consumer and the source consumer have the same amount of smart meter data, transfer learning could still improve the prediction accuracy of MLP and LSTM.

4) For the MLP networks, on average, MAPE, MAE, RMSE, and NRMSE could be improved by 7.80%. 3.16%, 1.13%, and 1.53% respectively;

5) For the MLP networks, the first several hidden layers have higher transferability than the last several layers;

6) For the MLP network, transferring from a dissimilar consumer is generally better than transferring from a similar consumer;

7) For the LSTM networks, as using it along could already achieve a good prediction, the introduction could only improve MAPE and NRMSE by 1.76% and 0.76% respectively;

8) For the LSTM networks, the transferability (measured by improvement of MAPE and NRMSE) of first several hidden LSTM units is higher than the last several ones.

Chapter 4

Inferring Social-Economic Status from Smart Metering Data

This chapter proposes an ensemble framework to infer the consumer's social-economic information from smart metering data. Meaningful features are firstly extracted from smart metering data and then fed to a gradient boosted ensemble tree structure for training. The features that significantly help the inference of the social-economic status are lastly identified.

4.1 Introduction

Over the past decade, smart meters have been widely deployed at domestic households across the world. The last chapter has demonstrated how deep learning can be utilized to forecast load by using the massive amount of smart meter data. However, at the same time, owing to the rapid and ongoing digitalization of the society, a variety of information about the consumers which was traditionally unattainable has also streamed in and becomes available, such as the consumer's age, education, and other social-economic information.

The two streams of data depict consumers from different angles. The smart metering data of the consumers demonstrate the consumption patterns of the consumers directly and intuitively, whereas the social-economic or social-demographic status of consumers provides deeper insights into the consumers. Provided with the social-economic information, the retailers and the distribution network operators (DNOs) could then utilize it and provide tailored tariffs or services, and implement more efficient demand side responses (DSRs).

Based on the research objective, existing research on the relationship between smart metering data and socio-economic data can be broadly categorised into two types:

3) Research focusing on inferring social-economic/social-demographic status from smart metering data [57-60];
4) Research focusing on inferring smart metering data/characteristics from social-economic/social-demographic data [61-64].

The proposed research in this chapter falls within the first category. And the challenges for the existing methods in this category are two-fold.

1) The proposed methods in some literature focus more on the interpretability of the models. Hence most of them are developed upon traditional or classical machine/statistical learning methods, such as k-Nearest Neighbours (KNN), Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), etc. These models are easy to interpret, which is desirable since deeper insights could be gained from it. However, these models rely heavily on feature engineering [114] and the prediction is not as accurate as state-of-the-art methodologies, such as deep learning.

2) Methods based on deep learning inarguably give the best inference performance and do not require feature engineering. The downside is the lack of interpretability of the model and the features. Though some feature visualization techniques [115-121] have been developed for deep learning, it is still intuitive and hard to understand what really happens inside the deep models.

This chapter proposes to leverage the power of gradient boosted ensemble trees to maintain the interpretability of the model while achieving a decent predicting performance. The rest of the chapter is organised as follows: Section 5.2 introduces the concept of ensemble learning, especially gradient boosted tree ensembles. Section 5.3 explains the details of the proposed framework. Section 5.4 demonstrates the results of the proposed methods on the validation data set. And Section 5.5 summarises this chapter.

4.2 Tree Based Ensemble Learning

Tree based learning algorithms empower predictive models with high accuracy, stability and interpretability. Methods like decision trees, random forest, and gradient boosted trees are widely used to solve a variety of problems. Specifically, extreme gradient boost (XGBoost) models [122], a variant of gradient boosted trees, have won a large number of machine learning competitions and have been applied in practice by a wide range of companies like Tencent and Alibaba. This section will briefly introduce ensemble learning with a special focus on gradient boosted trees. In addition, the rationale of choosing gradient boosted trees over other machine learning algorithms will be explained.

4.2.1 Ensemble Learning

Ensemble learning in machine learning is a paradigm where multiple learners are trained and combined to make a prediction [123]. The constituent learners of the ensemble are called base learners or weak learners. The predictions of all the base learners are then unutilized and combined together to form an ensemble model where the resultant ensemble learner usually gives a stronger prediction and generalization ability empirically.

The base learners could be of any type, e.g., decision tree, support vector machine (SVM), neural network, etc. If the base leaners are of the same type in an ensemble model, they are called homogeneous base learners. In contrast, if multiple types of learners are combined, they are usually called heterogeneous learners.

Decision tree based ensemble models constitute a vital category of ensemble learning. To combine all the base tree learners, two techniques are widely used, i.e., Bagging [124] and Boosting [125, 126]. In Bagging, a number of base learners are generated and trained on data from bootstrap samples in parallel. Boosting, on the other hand, builds each base learners sequentially and attempt to reduce predicting errors progressively. In recent years, tree boosting models, especially XGBoost, have achieved great success in not only competitions but also in real applications.

4.2.2Extreme Gradient Boosting Trees

In 2014, XGBoost demonstrated its significant scalability and predictive performance by winning the Higgs Challenge [127] hosted by CERN, whose goal was to use a machine learning algorithm to classify signals from the Larger Hadron Collider. Since then, XGBoost has been widely used for all kinds of tasks.

In contrast to Random Forest [128], which uses a modified Bagging technique to construct tree ensembles, the XGBoost builds one simple tree at a time and trains the new tree to reduce the errors from the previous tree.

Suppose the objective function of the ensemble model is (4-1).

$$obj = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4-1}$$

where L is the loss function, Ω is the regularization term, n is the number of training instances, K is the number of trees, y_i is the real value for instance i, $\hat{y}_i$ is the predicted value for instance i, and f_k represents the function of k-th tree.

The loss function measures the predictive performance of the model on the training data and the regularization term is added as a penalty term to control the complexity of the model. Hence, the model is trained with complexity as constraints to be less prone to overfitting.

The model is trained one step at a time. During each step, what the model has learnt previously is fixed and a new tree is added. Let the prediction of the ensemble model at step t is $\hat{y}_i^{(t)}$, then the following equations can be developed.

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{4-2}$$

where x_i is the input.

The objective function at step t then becomes (4-3).

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) \\ &= \sum_{i=1}^{n} L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + constant \end{aligned} \tag{4-3}$$

If mean squares error (MSE) is selected as the loss function, the objective function becomes

$$\begin{aligned} obj^{(t)} &= \sum_{i=1}^{n} \left(y_i - (\hat{y}_i^{(t-1)} + f_t(x_i))\right)^2 + \Omega(f_t) + constant \\ &= \sum_{i=1}^{n} \left[2\left(\hat{y}_i^{(t-1)} - y_i\right) f_t(x_i) + f_t(x_i)^2\right] + \Omega(f_t) + constant \end{aligned} \tag{4-4}$$

To get the general form of the objective function for other losses, the Taylor expansion of (4-3) is taken to the second order. (4-5) is then obtained.

$$obj^{(t)} \approx \sum_{i=1}^{n} \left[L\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) + constant \tag{4-5}$$

where g_i and h_i are defined as

$$g_i = \partial_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \tag{4-6}$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} L(y_i, \hat{y}_i^{(t-1)}) \tag{4-7}$$

After all the constants in the equation are removed, the objective function at step t then becomes (4-8).

$$\sum_{i=1}^{n}\left[g_i f_t(x_i)+\frac{1}{2}h_i f_t^2(x_i)\right]+\Omega(f_t) \tag{4-8}$$

The objective function is still unclear due to the lack of specification of the regularization term.

Prior to the definition of the complexity of the tree, the tree itself is defined as below.

$$f_t(x)=w_{q(x)}, w\in\mathbb{R}^T, q:\mathbb{R}^d\rightarrow\{1,2,3,\cdots,T\} \tag{4-9}$$

where w is the vector containing all the scores on different leaves of the tree, q is a leaf index mapping function that maps an input data point to its corresponding leaf, T is the total number of leaves in the tree. Hence, the complexity for the XGBoost model is then expressed as (4-10).

$$\Omega(f)=\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \tag{4-10}$$

As can be seen, the complexity is defined as the weighted sum of the number of leaves and the L2 norm of leaf scores.

Taking (4-9) and (4-10) into (4-8), the equation is then reformulated as

$$\begin{aligned} obj^{(t)} &\approx \sum_{i=1}^{n}\left[g_i w_{q(x_i)}+\frac{1}{2}h_i w_{q(x_i)}^2\right]+\gamma T+\frac{1}{2}\lambda\sum_{j=1}^{T}w_j^2 \\ &=\sum_{j=1}^{T}\left[(\sum_{i\in I_j}g_i)w_j+\frac{1}{2}(\sum_{i\in I_j}h_i+\lambda)w_j^2\right]+\gamma T \end{aligned} \tag{4-11}$$

where $I_j=\{i|q(x_i)=j\}$ is the set of indices of data points who are allocated to the j-th leaf.

The objective function could be further reformed as below.

$$obj^{(t)}=\sum_{j=1}^{T}\left[G_j w_j+\frac{1}{2}(H_j+\lambda)w_j^2\right]+\gamma T \tag{4-12}$$

where $G_j=\sum_{i\in I_j}g_i$ and $H_j=\sum_{i\in I_j}h_i$.

Given a tree structure $q(x)$, the optimal w_j and the best objective function reduction are:

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{4-13}$$

$$obj^* = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{4-14}$$

4.2.3The rationale of Applying Gradient Boosted Trees

Current research on inferring social-economic status generally utilize either classical machine learning algorithm in isolation or state-of-the-art deep neural networks. Classical models, such as linear models and nearest neighbours, are easy to train and interpret. However, the drawback is the predictive power of them is relatively weak. Though deep neural networks are powerful, their disadvantages are obvious and difficult to overcome:

1) Training deep models requires a large amount of data. For the addressed task in this chapter, the available data are likely to be insufficient for the effective training of deep models;
2) Deep models are difficult to train due to the large number of required hyperparameters. In addition, training a deep model is time-consuming, which makes it even harder to find the optimal set of hyperparameters;
3) Interpreting deep models is difficult. Deep neural networks are widely recognised as black box models, which means that no insights can be gained on how the model makes the decisions.

Gradient boosting trees, especially XGBoost, have proven to be effective and efficient in solving similar problems. The rationale for applying XGBoost is four-fold:

1) XGBoost has strong predictive power due to its ensemble nature which has been proven in many competitions and applications;
2) XGBoost works well on both big data set and a relatively small dataset. The flexibility of changing base tree number and constraining base tree complexity enables the XGBoost model to be fairly adaptable to the data set;

3) XGBoost model is easy and fast to train. XGBoost has much fewer hyperparameters to tune compared to deep models. In addition, XGBoost could utilize parallel processing. Hence, it is very fast to train;
4) XGBoost models are easy to interpret. XGBoost models are constructed by combining multiple decision trees. The base trees are relatively simple due to the regularization term in the objective function and are hence easy to be interpreted. In addition, as given in (4-2), XGBoost takes the linear combination of all the individual base trees to make the predictions, which is also relatively easy to explain.

4.3 Proposed Interfering Framework

The proposed framework addresses the need for ease of interpretation and decent prediction accuracy. It consists of three steps, which is illustrated in Figure 4-1.

4.3.1 Data Pre-Processing

The data used here are taken from the Smart Metering Electricity Customer Behaviour Trials (CBTs) initiated by the Commission for Energy Regulation (CER) in Ireland [112]. The trials spanned from July 2009 to December 2010 and contain over 5000 consumers. The full anonymized data sets are publicly available online and contain not only the half-hourly sampled electricity consumption (kWh) from each participant but also the customer type, tariff and stimulus description, which specifies customer types, allocation of tariff scheme and Demand Side Management (DSM) stimuli. Additionally, one pre-trial trial and one post-trial surveys were conducted on the participated consumers in CBTs. The surveys are in the form of questionnaire and the participated consumers in CBTs. The surveys are in the form of questionnaires and the feedback from the participants are provided as well. These surveys contain the social-economic information about the consumers. As the answers (social-economic data) of the surveys are in the forms of text, such as ‘Yes’ and ‘No’, they have to be converted to numerical values first, so that they can be properly used as labels or target data in the machine learning workflow.

The raw smart metering data are unstructured, i.e., they are neither ordered by customer ID nor time index. Additionally, the data are split into multiple text format files. The raw data are then pre-processed so that they are ordered by customer ID and

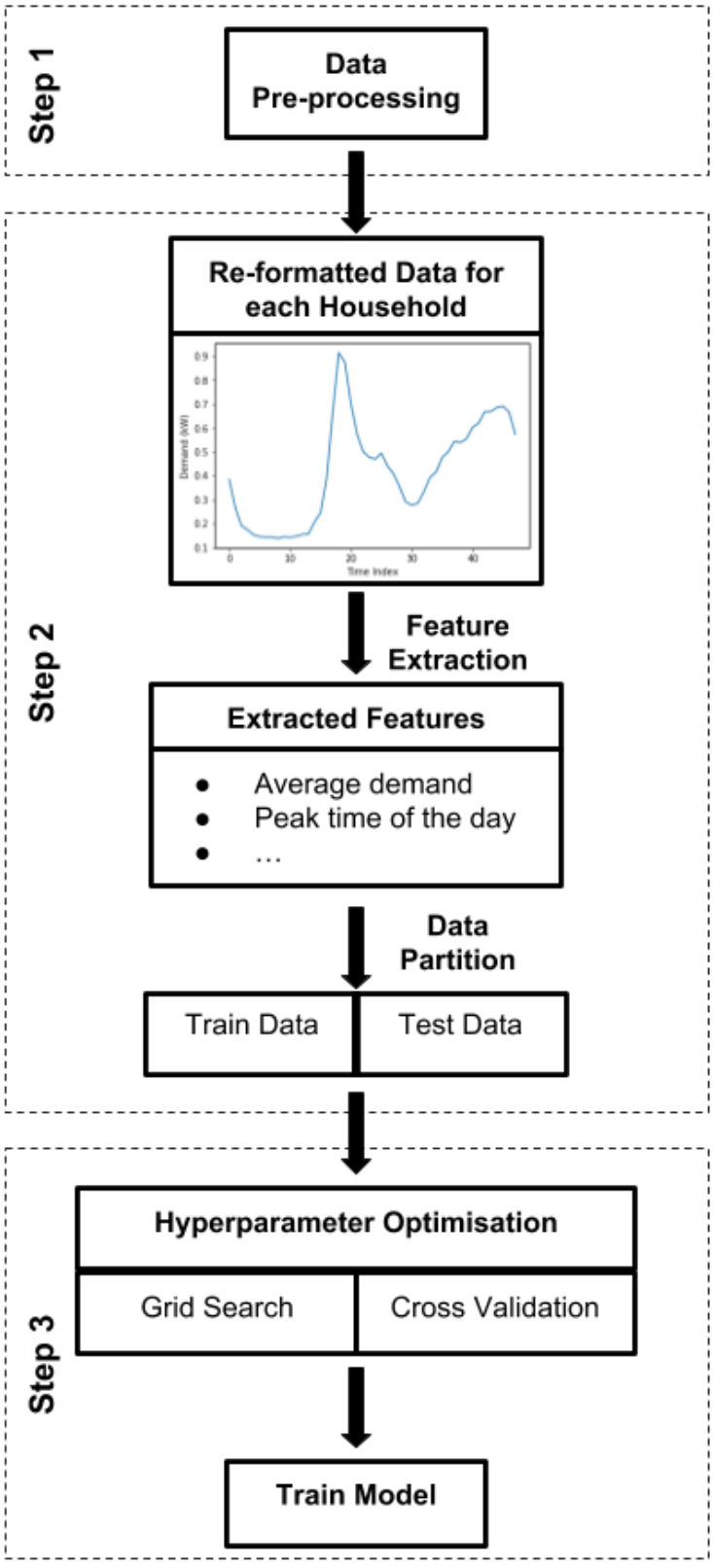

Figure 4-1 Proposed inferring framework

time index. Though data missing issue is not encountered in the dataset. It is found that there are continuous days when demand is constantly zero. Such data pattern

should be very uncommon in real life, as various appliances have their standby power consumption. As a result, periods with such pattern are treated as data missing period and are deleted from the data set.

Despite the smart metering data and the surveys are sampled from the same population, it is found that the consumer IDs of the smart metering data are slightly different from the IDs of the survey data. Consequently, the intersection of the two groups of data (have the same customer IDs) have to be found.

4.3.2Load Feature Extraction and Data Partition

The smart metering data are sampled half-hourly which would add up to 17,520 features over a year. The high dimensionality of the smart metering data will lead to the over-fitting of the XGBoost model as the number of training samples is much smaller than the number of features (no more than 5,000).

In order to reduce the dimensionality of the input data, while maintaining the necessary information for classification, a variety of techniques are available, such as clustering [129, 130], Principal Component Analysis (PCA) [131], feature selection [132, 133], etc. In this proposed framework, domain knowledge based feature extraction is used to extract meaningful and interpretable features, such as daily mean demand and the time peak demand. Though the other above-mentioned techniques could possibly capture more distinctive characteristics of the smart metering data, none of them is able to provide interpretable physical meanings to the features that they extracted. Conversely, human-defined features are obviously interpretable, which is crucial for post-training analysis.

Smart metering data features are extracted across different time horizons which are depicted in Figure 4-2. Specifically, the features are extracted from annual data, winter data, spring data, summer data and autumn data. For each season or the whole year, the extraction is performed on finer time horizons, i.e., weekday, weekend and holidays. For each specific time interval, four types of features are extracted:

1) Consumption figure related features, such as daily average demand, evening average demand, the average of daily peak demand, etc;
2) Ratio features. For example, the ratio of mean demand over peak demand;
3) Occurrence/time-related features. For instance, the time of peak demand;

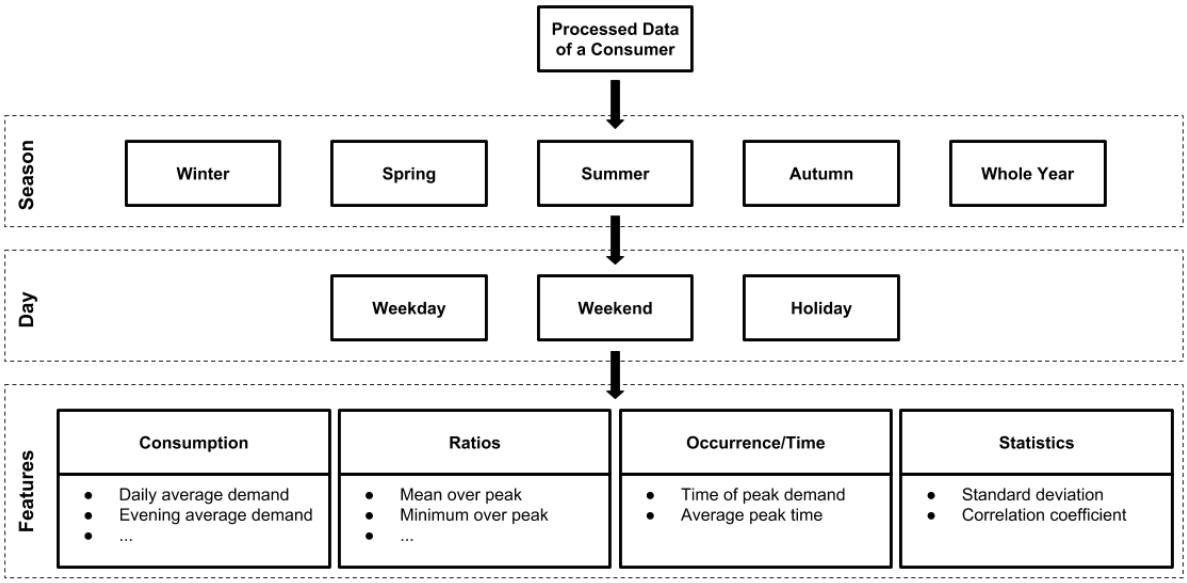

Figure 4-2 Smart metering data feature extraction

4) Statistical features, such as the average of daily standard deviation.

Detailed list of extracted features can be found in Appendix B.

The data are then partitioned into training data set (80%) and test data set (20%). The validation data set are not specified as cross-validation (CV) would be used later.

4.3.3Hyperparameter Optimisation and Training of XGBoost

The performance of the XGBoost model depends on the setting of its hyperparameters, i.e., maximum depth of the tree, number of base learners, learning rate, etc. In order to achieve the optimal performance, the hyperparameters of all the XGBoost models are optimised using grid search and cross-validation (CV). To put it into perspective, all the possible combinations of the hyperparameters (hence grid search) of an XGBoost model are examined using cross-validation. Specifically, in this experiment, 5-fold CV is used. That means that the original training dataset is randomly split into 5 datasets of the same size. The model is trained using 4 sets of the data and validated using the remaining one. In total, the model could be trained and validated differently 5 times. The average performance of the model over the 5 different validation set is calculated and used as the performance indicator of the model under the specific hyperparameter setting. The settings of hyperparameters that give the best CV results are recorded. And the final XGBoost model is trained using the corresponding hyperparameters.

4.4 Demonstration and Results

The proposed method is demonstrated against real data from Ireland, which was introduced in the previous section. After pre-processing, a total number of 3977 valid consumers are identified, out of which 3181 consumers are used for training. As for the social-economic data from the surveys, there are more than 100 questions (and answers). Furthermore, some of the social-economic data are very difficult to predict or may be unable to predict. For demonstration purpose, the proposed method is validated on only 10 of the social-economic questions, which are listed in Table 4-1.

Table 4-1 Selected social-economic questions

No.	Question Index	Questions
1	300	Age
2	310	Employment status of the chief income earner
3	408	Are there other people in the household that use the intern et regularly
4	4333	I/we am/are interested in changing the way I/we use electr icity if it helps the environment
5	410	What best describes the people you live with?
6	430	And how many adults are typically in the house during the day
7	49004	How often do you use dish washer?
8	4905	How many stand-alone freezers do you have?
9	4900009	How often do you use stand-alone freezers?
10	490002	How many desktop computers do you have?

4.4.1 Performance Evaluation

The proposed method is validated on the ten questions selected in the table. In addition, the proposed model is compared with three commonly used methods, i.e., KNN, SVM and Random Forest.

Classification accuracy is most widely used for evaluating the performance of a classification model. However, it is observed that the data in the experiment are imbalanced, which means that each class in the data do not have the same number of samples. Hence classification accuracy alone is not adequate enough to quantify

the classification performance. F1 score is proposed to complement the quantification of the model performance. In a binary case, the classification result can be illustrated by Figure 4-3.

F1 score is defined as:

$$F_1 = 2\frac{Pre \cdot Rec}{Pre + Rec} \tag{4-15}$$

where Pre and Rec are the precision and recall respectively and are calculated by the following equations.

$$Pre = \frac{TP}{TP + FP} \tag{4-16}$$

$$Rec = \frac{TP}{TP + FN} \tag{4-17}$$

In the multi-class case, the average of the F1 score of each class is taken. To account for the imbalance of different classes, the micro average of F1 is used, where the TP, TN, FP, and FN of all the classes are all taken into consideration.

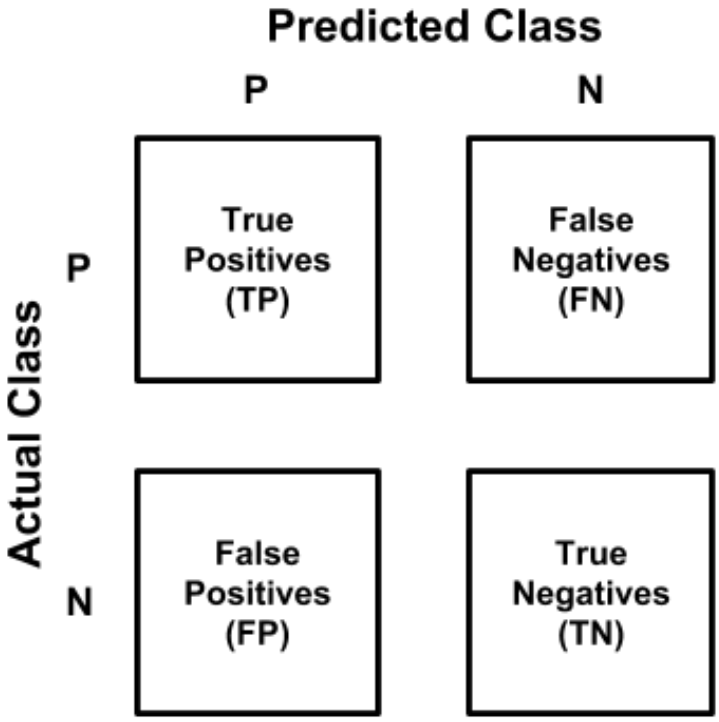

Figure 4-3 Binary classification

The prediction accuracy and F1 scores of the proposed method along with KNN, SVM, and Random Forest (RF) are listed in Table 4-2 and Table 4-3 respectively.

Table 4-2 Comparison of prediction accuracies of different methods

	1	2	3	4	5	6	7	8	9	10
KNN	0.358	0.587	0.702	0.623	0.578	0.420	0.515	0.593	0.595	0.604
SVM	0.378	0.590	0.697	**0.654**	0.620	0.477	0.533	0.612	0.623	0.625
RF	0.360	0.562	0.711	0.643	0.598	0.476	0.533	0.633	0.633	**0.637**
Proposed	**0.386**	**0.591**	**0.725**	0.646	**0.630**	**0.486**	**0.558**	**0.634**	**0.636**	0.632

Table 4-3 Comparison of F1 scores of different methods

	1	2	3	4	5	6	7	8	9	10
KNN	0.358	0.587	0.702	0.623	0.578	0.420	0.515	0.593	0.595	0.604
SVM	0.378	0.590	0.697	**0.654**	0.620	0.477	0.533	0.612	0.623	0.625
RF	0.360	0.562	0.711	0.643	0.598	0.476	0.533	0.633	0.633	**0.637**
Proposed	**0.386**	**0.591**	**0.725**	0.646	**0.630**	**0.486**	**0.558**	**0.634**	**0.636**	0.632

As can be observed from the two tables, the proposed method achieved the best accuracies and F1 scores for almost all the 10 social-economic questions except for the fifth and the tenth ones.

4.4.2Model Interpretation

While the proposed framework using XGBoost has proven its accurate predictive power in the previous subsection, it also comes with sound interpretability, which could help the understanding of the model.

The XGBoost models are built upon a number of base decision trees. To build a tree, the training data are divided recursively several times. At the end, a group of observations are obtained, which, in our case, are the answers to the selected social-economic question. Each division operation is called a split. Each group at each division level is called a branch and the deepest level is called a leaf. Different features in the training data are selected to assist the split. Basically, attaining the best split very much depends on the selection of features. Intuitively, the number of the feature that appears in the trees, which is termed as weight in XGBoost model, is a reasonable indicator of the importance of the features. The feature importance of the first five of the social-economic questions in Table 4-1 is plotted in the figure below. The plots of feature importance of the rest of the question in the table can be found in Appendix B.

The names of the features in the plots normally consists of three parts:1) the first part specifies the seasonality, i.e., whether it is one of the four seasons or the whole year;

2) the second part specifies the type of the days, meaning that it is workday, weekend or holiday; 3) the rest in the name describes the nature of the feature, with ‘ave’ being the average, ‘o’ being over (take division) and ‘corr’ being the correlation coefficient.

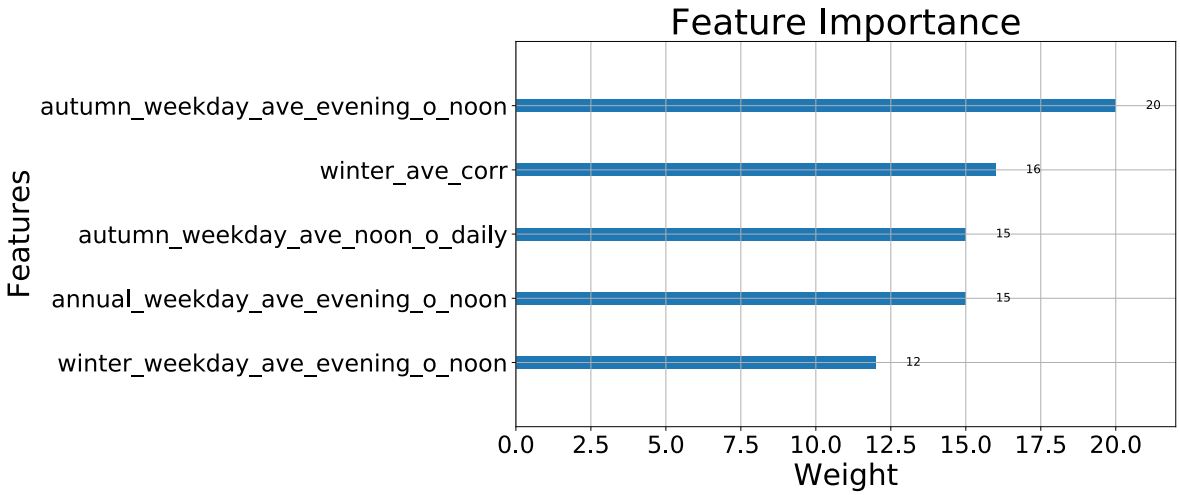

Figure 4-4 Feature importance for social-economic question 1

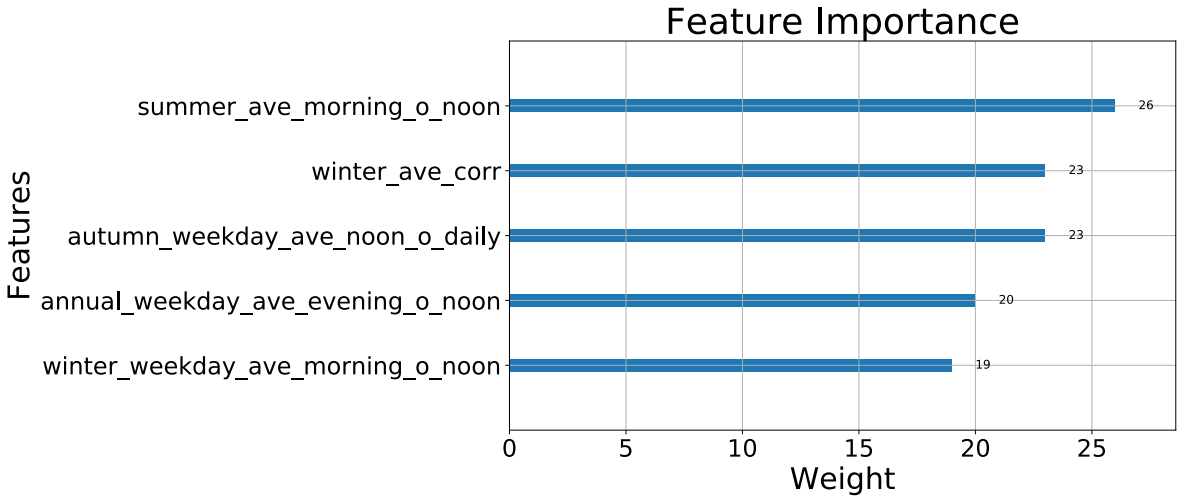

Figure 4-5 Feature importance for social-economic question 2

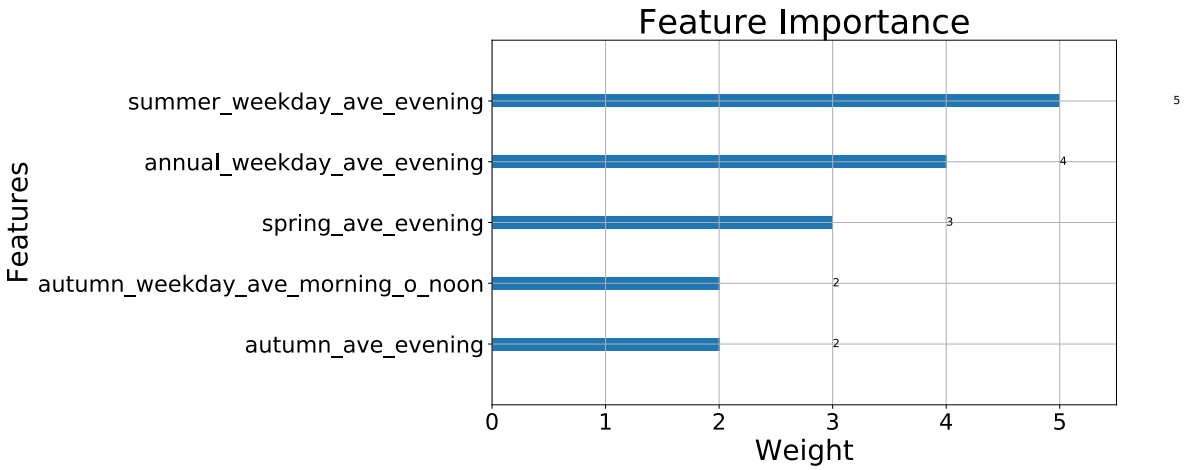

Figure 4-6 Feature importance for social-economic question 3

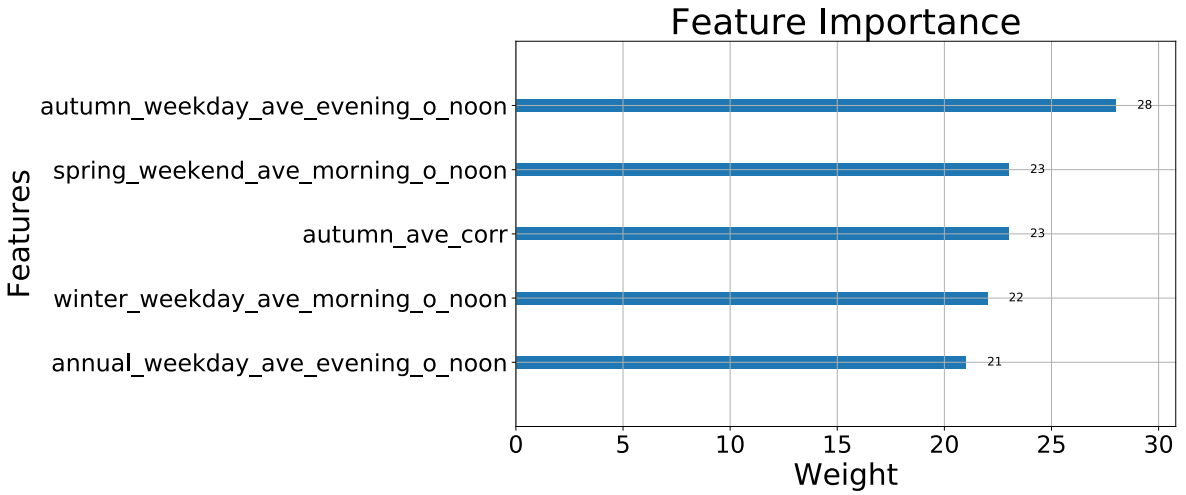

Figure 4-7 Feature importance for social-economic question 4

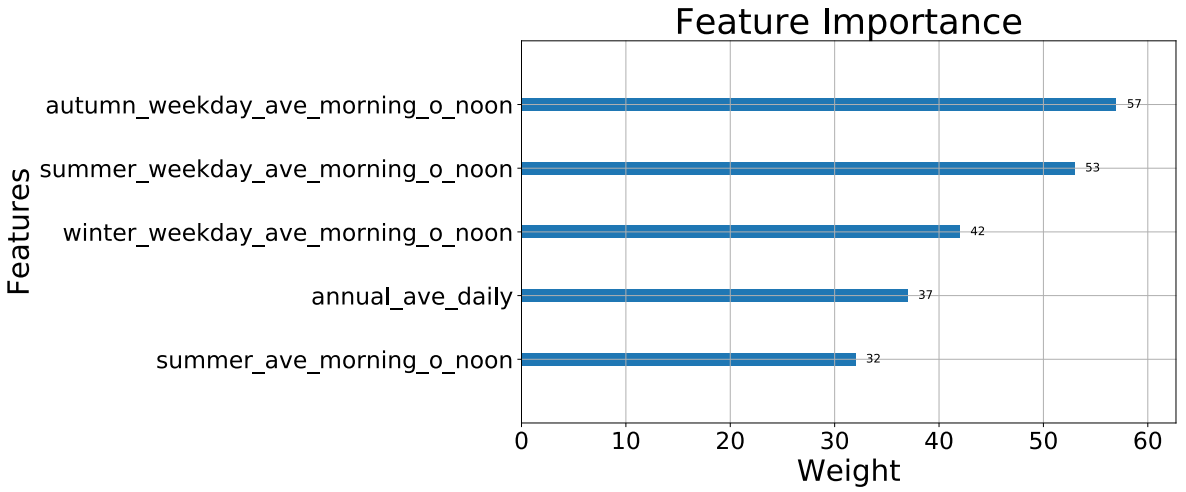

Figure 4-8 Feature importance for social-economic question 5

As can be observed from the plots, different social-economic features are affected by a different range of smart metering features.

For the age of the consumer, which is the first question, it is predominantly affected by the cold season (autumn, winter) weekday features, especially the consumption ratio between evening and noon. This could be explained as that different people of different ages have varying degrees of endurance capacity for coldness. Also, the second most significant feature is the correlation coefficient during winter time. This reflects the consumption pattern consistency over the selected time period. It is reasonable considering that people of different ages do have different levels of consumption consistency. With the base load in winter being slightly higher, any irregular patterns during winter may even be exaggerated. However, these are all assumptions and need further investigation to be verified.

For the employment status of the consumer, there is no obvious strong correlation between it and the seasonality. However, the consumption ratio relating to the noon during weekdays are quite important. This could be caused by the fact that self-employed or unemployed people may spend more time during daytime (hence consume more electricity) than those who are employed and work in offices.

The third question is about whether there are other regular internet users in the house. As shown in Figure 4-6, it is mostly affected by the evening consumption ratio during weekdays. It is quite reasonable considering the fact that most regular internet users, especially children or students, would use the internet during the evening.

As for the fourth question, which is the attitude towards saving energy and protecting the environment, no evident correlation with the seasonality can be observed neither. The key factors are related to the consumption ratios between noon and morning and between noon and evening during weekdays.

The fifth and the last question that was investigated is regarding the people with whom the consumer live with. The answers are 1) live alone, 2) with adults, and 3) with adults and children. As can be observed from Figure 4-8, it is primarily affected by the consumption ratio between morning and noon during the weekend. This could be attributed to the fact that families with child/children would consume electricity quite differently. In addition, the annual average demand is also a key factor. It is obviously

true that the consumer who lives alone would have a lower average consumption level than those who live with others.

4.5 Chapter Summary

This chapter proposes a prediction framework that infers the social-economic status of consumers from their smart metering data. It utilises a tree based ensemble model, XGBoost, and consist of three major steps: data pre-processing, feature extraction and model training and hyperparameter optimisation. It has been validated on real data from Ireland and has been compared with three state-of-the-art methods, KNN, SVM and Random Forest.

Results have shown that the proposed method outperforms the state-of-the-art method for almost all features. In addition, due to the tree ensemble nature, the developed models could be interpreted by observing the splits of each individual trees. Specifically, the importance of the smart metering data features in helping the models make their predictions can be explored and ranked.

The method has proven that it is not only able to predict the social-economic status accurately, but also helpful for gaining insights from the data. DNOs and retailers could leverage the insights and develop tailored DSR schemes and tariffs.

Chapter 5

Inferring Household Load Characteristics from Social-Eco Data

This chapter proposes a Deep Convolutional Neural Network (DCNN) based multi-task learning (MTL) method for the inference of consumer's smart metering features from social-economic data. It leverages the convolutional kernel and deep architecture in DCNN to overcome the hurdle brought by mixed types of social-economic data and infers multiple load characteristics simultaneously.

5.1 Introduction

Household load characteristics, e.g., peak demand and average demand, are extracted from smart metering data. They are traditionally unavailable until the recent roll-out of smart meters. The attained visibility on household's load characteristics would help consumers save energy usage and choose better tariff, provide energy suppliers insights into their customers, and facilitate more efficient Demand Side Response (DSR) schemes in the network. However, a large proportion of consumers, even in some developed countries, are still using the conventional meters [27], which are unable to monitor consumers' timely energy consumption. Consequently, none of the consumers, suppliers nor Distribution Network Operators (DNOs) understands how electricity is consumed behind the meters.

The research problem in the last chapter will be reversed in this chapter. Rather than installing costly smart meters, consumers' social-economic information, which can be easily collected by means of surveys, could be used to infer the load characteristics. The inferring could be formulated as a classification or regression problem. The main challenge for an accurate prediction arises from the fact that social-economic data normally consist of a mixture of numerical data, categorical data and ordinal data, whereas traditional prediction methods are mostly designed for handling only one type of the data. In addition, the prediction of each load characteristic requires the development and training of a separate model. Consequently, a tremendous amount of effort and time would be devoted to getting a reasonably sized set of useful load characteristics.

This chapter proposes a one-dimensional deep convolutional neural network (DCNN) based multi-task learning (MTL) method. It leverages the convolutional kernel and deep architecture in DCNN to overcome the hurdle brought by mixed types of data and infers multiple load characteristics simultaneously. The rest of the chapter is organised as follows: Section 6.2 introduces CNN and MTL. Section 6.3 explains the details of the proposed framework. Section 6.4 demonstrates the results of the proposed method on the validation data set. And Section 6.5 summarises this chapter.

5.2 Preliminaries

5.2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a special form of neural networks. It was first proposed as Neocognitron by Kunihiko Fukushima in the 1980s [134]. This early form of CNN was further developed by Yann LeCun in 1998 [135] for hand-written ZIP code recognition, which is already quite close to the modern CNN architecture. The major breakthrough in CNN was in 2012. AlexNet [99], an architecture of CNN, was designed by Alex Krizhevsky, Ilya Sutskever and Geoff Hinton, and won the first place in the 2012 ImageNet ILSVRC challenge. It outperformed the second runner-up significantly and has popularized CNN and Deep Learning (DL). Ever since then, great advancement has been achieved in CNNs and DL. Nowadays, CNNs and other forms of DL have been widely applied in all kinds of fields for various tasks, such as image and video recognition, Natural Language Processing (NLP), and recommender systems.

This section will provide an architectural overview of the CNNs.

1) Convolutional Layer:

The convolutional layer is the essential building block of CNNs. Each convolutional layer consists of a set of filters. The filters are normally small in size and consist of some parameters that are to be learned through training. The term convolution actually refers to the operation of taking the dot product of the input data of the filters with the parameters within the filters. For example, if the input of the network is a 2-dimensional matrix, each filter will slide along the width and the height of the input matrix, taking the dot product of the input data at every position and producing a new 2-dimensional matrix called the activation map (the size of the new matrix depends how the filters are sided and if the input matrix is padded).

The filter in convolutional layers is generally much smaller than the input volume. It could only receive data from a restricted area from its input. This input area is called the receptive field, and it is of the same size as the filers. As each filter slides over the input matrix, the data it would receive would change. However, for each filter, the parameters are shared during the sliding. To put it into perspective, the same set of parameters are used for each filter to the take the dot product during the sliding. (5-1) formulates the convolution operation for discrete input data. Figure 5-1 demonstrates how the convolution operation is performed intuitively in a simple case.

Note that the size of the activation map in the plot is made up just for demonstration purpose.

$$f_{conv}(\boldsymbol{x}_s, \boldsymbol{W}^k) = \boldsymbol{W}^{\boldsymbol{k}} \cdot \boldsymbol{x}_s + b_k \quad (5\text{-}1)$$

where $\boldsymbol{x}_s$ is the vectorized received data for filter k, $\boldsymbol{W}^k$ is the weight vector for filter k, and b_k is the bias for filter k.

2) Pooling Layer:

Convolutional layers are usually stacked together so that the features can be

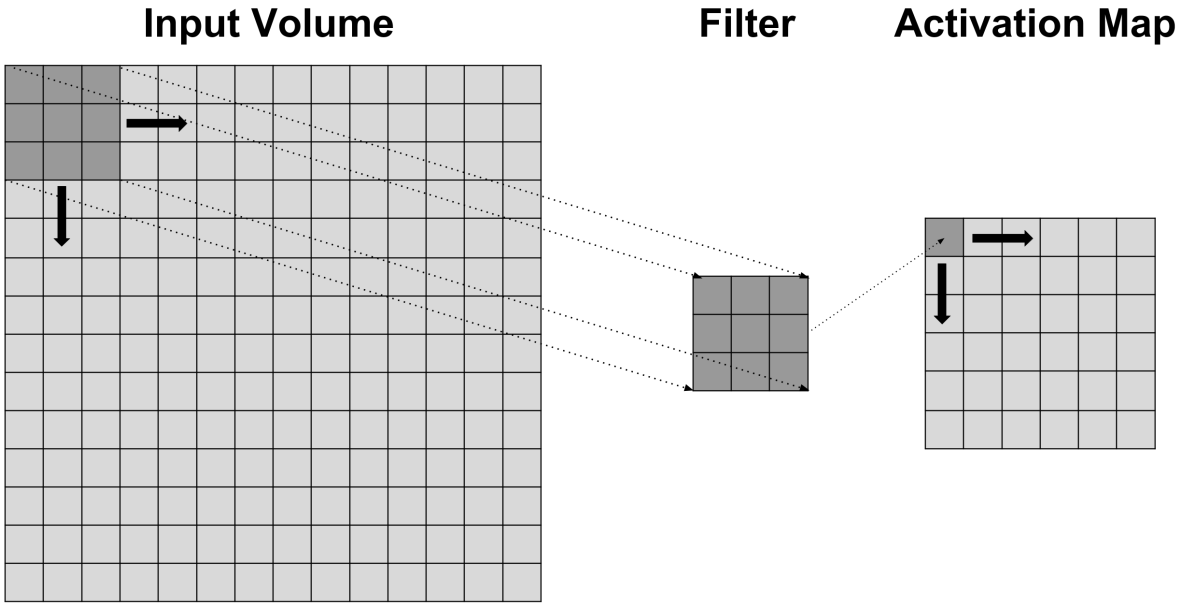

Figure 5-1 Convolutional layer demonstration

extracted hierarchically. However, most convolutions are performed to generate multiple activation maps with unchanged size, which would progressively increase the depths of intermediate features and parameters. A pooling layer is commonly added in-between successive convolutional layers. The insertion of a pooling layer is not compulsory. Normally, it is added periodically. The main functionality of it is to reduce the size of the activation maps progressively. As a result, fewer parameters are required to be trained. Like the convolutional layers, the pooling layers also have small filters, normally of size 2x2 in CV applications. The most common operation in pooling layers is to perform max operation which only returns the maximum value of the input data:

$$f_{\max\_pool}(\boldsymbol{x}_s) = \max(\boldsymbol{x}_s) \quad (5\text{-}2)$$

3) Fully-Connected Layer:

As the name suggests, all the neurons in a fully-connected (FC) layer are connected to all the neurons in the upper layer. In other words, unlike the convolutional layers, fully-connected (FC) layers can receive all the data from the input data. It can be viewed as an extreme case of a non-sliding convolutional layer with a receptive filed larger than or equal to the size of the input. It is the same as the layers in MLP. However, in a CNN architecture, it is usually used in the last or last few layers. The activation maps of different filters are concatenated and connected to an FC layer.

The rationale for applying CNN in the inferring of load characteristics is two-fold:

1) Learning features automatically and hierarchically: CNNs have sparse interactions as the receptive field is smaller than the input. Interactions of local features are learned first. As more convolutional layers are stacked together, the latter convolutional layers take in the output of the previous convolutional layers and would have an extended view on the input data. Consequently, the learned features are becoming more abstract as the layer number increases. This means that the CNN models can learn to extract small and meaning features themselves and are more robust and powerful [113], which is ideal for dealing with the complex social-economic data;
2) Sharing parameters: as mentioned earlier, each filter in the CNNs shares the same set of parameters/weights, regardless of where the filter is. As a result, the number of parameters in the model can be drastically reduced and less prone to overfit the data.

5.2.2Multi-Task Learning

Multi-Task Learning (MTL) is a learning paradigm in machine learning with the aim to leverage useful information contained in multiple related tasks to elevate the generalization performance on all the tasks [103]. MTL has been widely studied and successfully applied in a variety of applications, such as NLP [136], CV [137], and drug discovery [138].

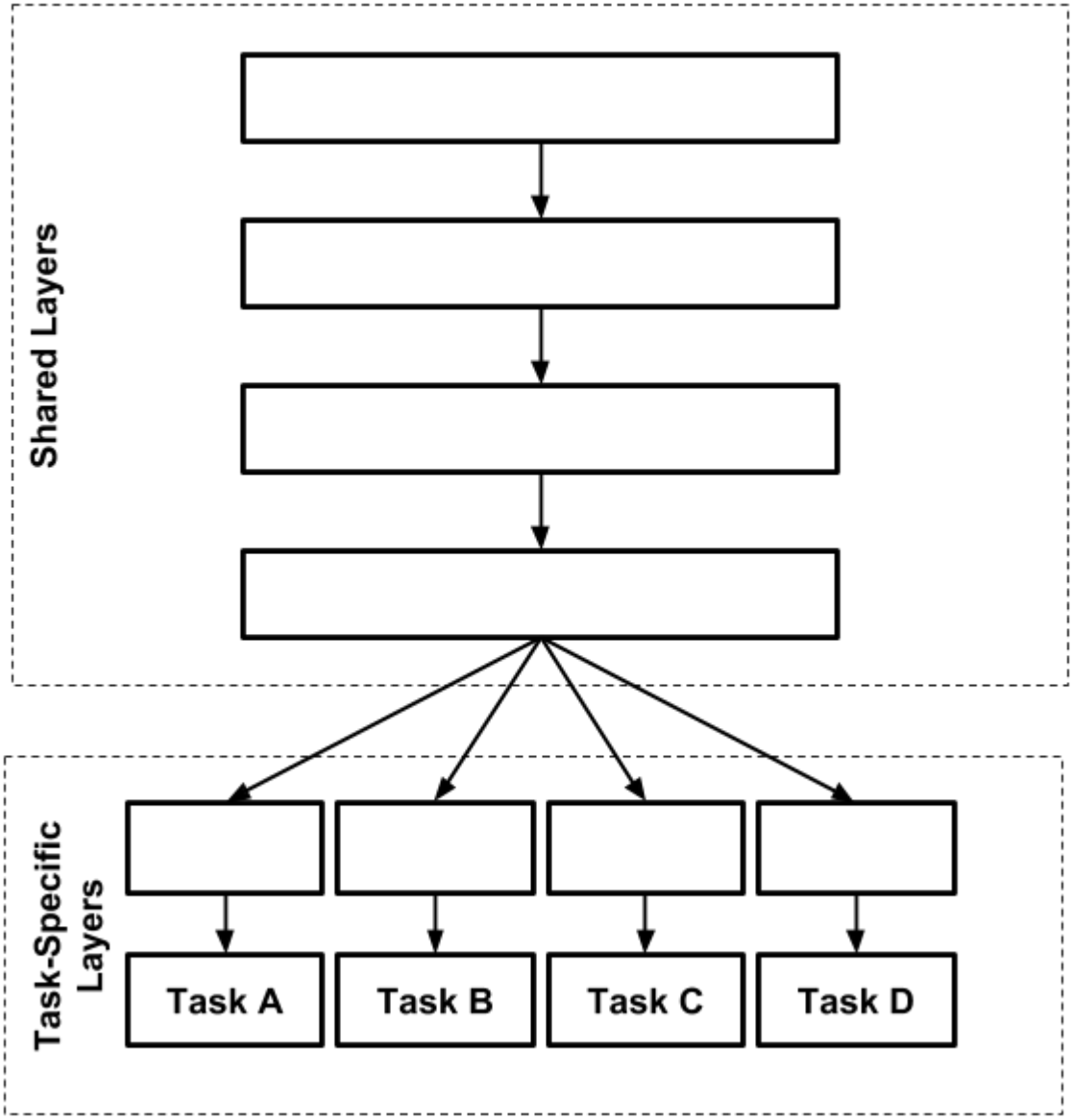

Figure 5-2 Hard parameter sharing in deep learning

MTL is a learning paradigm, which means it can be achieved on different machine learning models. In the context of deep learning, MTL can be achieved through either hard or soft parameter sharing of the hidden layers.

1) Hard Parameter Sharing:

Hard parameter sharing is demonstrated in Figure 5-2. It is commonly used on neural network models to perform MTL. It was first proposed by [139] in 1993. With the recent popularity of deep learning, it begins to be more widely used.

The power of hard parameter sharing lies in its ability to reduce overfitting. As proven in [140], by using hard parameter sharing for MLT, the chance of overfitting the shared parameters is an order N, where N is the number of tasks, which is smaller

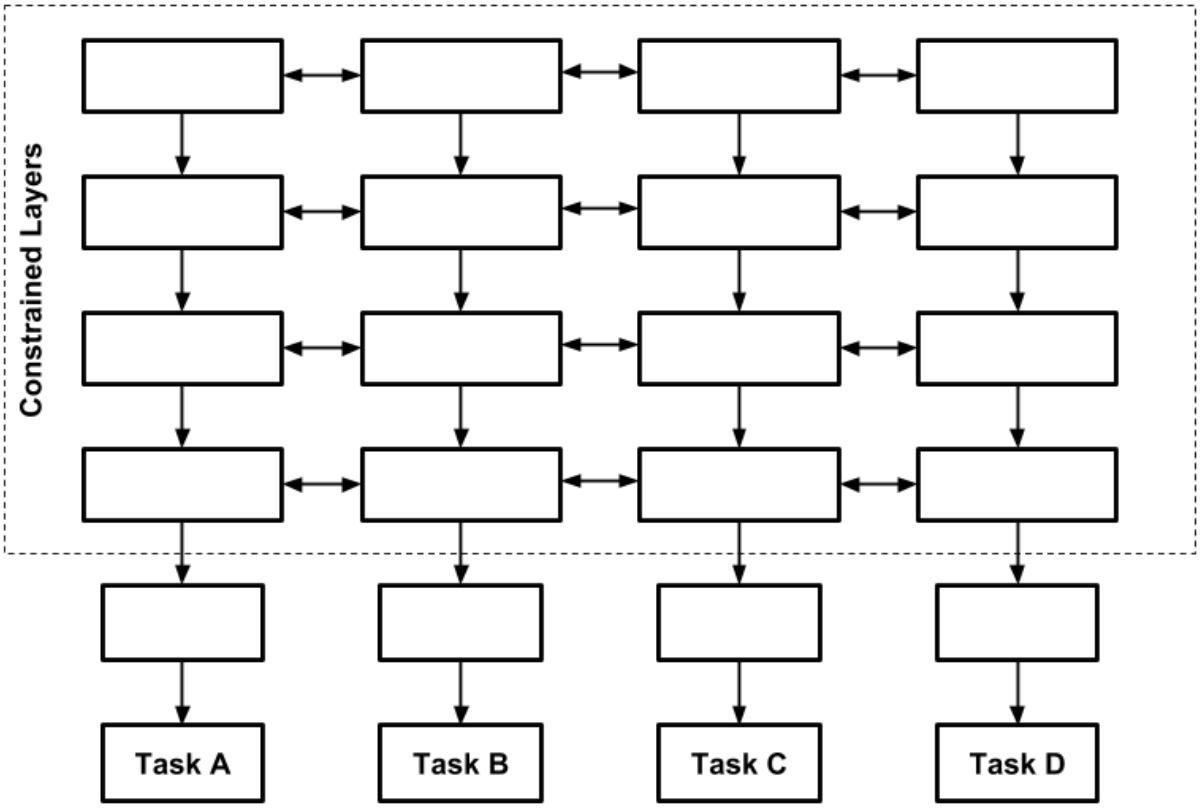

Figure 5-3 Soft parameter sharing in deep learning

than overfitting the task-specific parameters.

2) Soft Parameter Sharing:

In contrast to hard parameter sharing, one model is developed for each task in soft parameter sharing. The sharing of the parameters refers to the fact that during the training the parameters of each model is regularized so that the parameters of different models become very similar. The regularization is achieved by constraining the distances of the parameters of different models. Different distance metrics can be used, such as the L2 norm [141] and the trace norm [142]. Figure 5-3 illustrates how soft parameter sharing is achieved in deep learning.

5.2.3The Rationale for Combining Convolutional Neural Network and Multi-Task Learning

As discussed in the earlier section, CNNs are ideal for processing the complex social-economic data due to their ability to learn features automatically and hierarchically and share weights. However, CNNs are very difficult to build and train. In order to achieve a good prediction performance for just one load characteristic (there are many), the

hyperparameters of the CNN model should be optimised and fine-tuned which would not require a lot of computational power but also many times of trials.

The load characteristics are all extracted from the smart meter data and are inherently related. By extending the CNN model and sharing the low-level hidden layers (low-level features, not task-specific) between different load characteristics, the new MTL model may not only generalize better on the individual models but also saves a lot of effort to fine-tune each model.

5.3 Proposed Methodology

The proposed method consists of three key steps, 1) data pre-processing, 2) feature and label generation, and 3) training the model. Figure 5-4 gives a schematic overview of the proposed method.

5.3.1 Data Pre-Processing

The data used in this chapter are the same as the data used in Chapter 4, which are taken from the Smart Metering Electricity Customer Behaviour Trials (CBTs) initiated by the Commission for Energy Regulation (CER) in Ireland [112]. The trials spanned from July 2009 to December 2010 and contain over 5000 consumers. The full anonymized data sets are publicly available online and contain not only the half-hourly sampled electricity consumption (kWh) from each participant but also the customer type, tariff and stimulus description, which specifies customer types, allocation of tariff scheme and Demand Side Management (DSM) stimuli. Additionally, one pre-trial trial and one post-trial surveys were conducted on the participated consumers in CBTs. The surveys are in the form of questionnaire and the participated consumers in CBTs. The surveys are in the form of questionnaires and the feedback from the participants are provided as well. These surveys contain the social-economic information about the consumers. As the answers (social-economic data) of the surveys are in the forms of text, such as ‘Yes’ and ‘No’, they have to be converted to numerical values first.

The raw smart metering data are unstructured, i.e., they are neither ordered by customer ID nor time index. Additionally, the data are split into multiple text format files. The raw data are then pre-processed so that they are ordered by customer ID and time index. Though data missing issue is not encountered in the dataset. It is found that there are continuous days when demand is constantly zero. Such data pattern

should be very uncommon in real life, as various appliances have their standby power consumption. As a result, periods with such pattern are treated as data missing period and are deleted from the data set.

Despite the smart metering data and the surveys are sampled from the same population, it is found that the consumer IDs of the smart metering data are slightly different from the IDs of the survey data. Consequently, the intersection of the two groups of data (have the same customer IDs) have to be found.

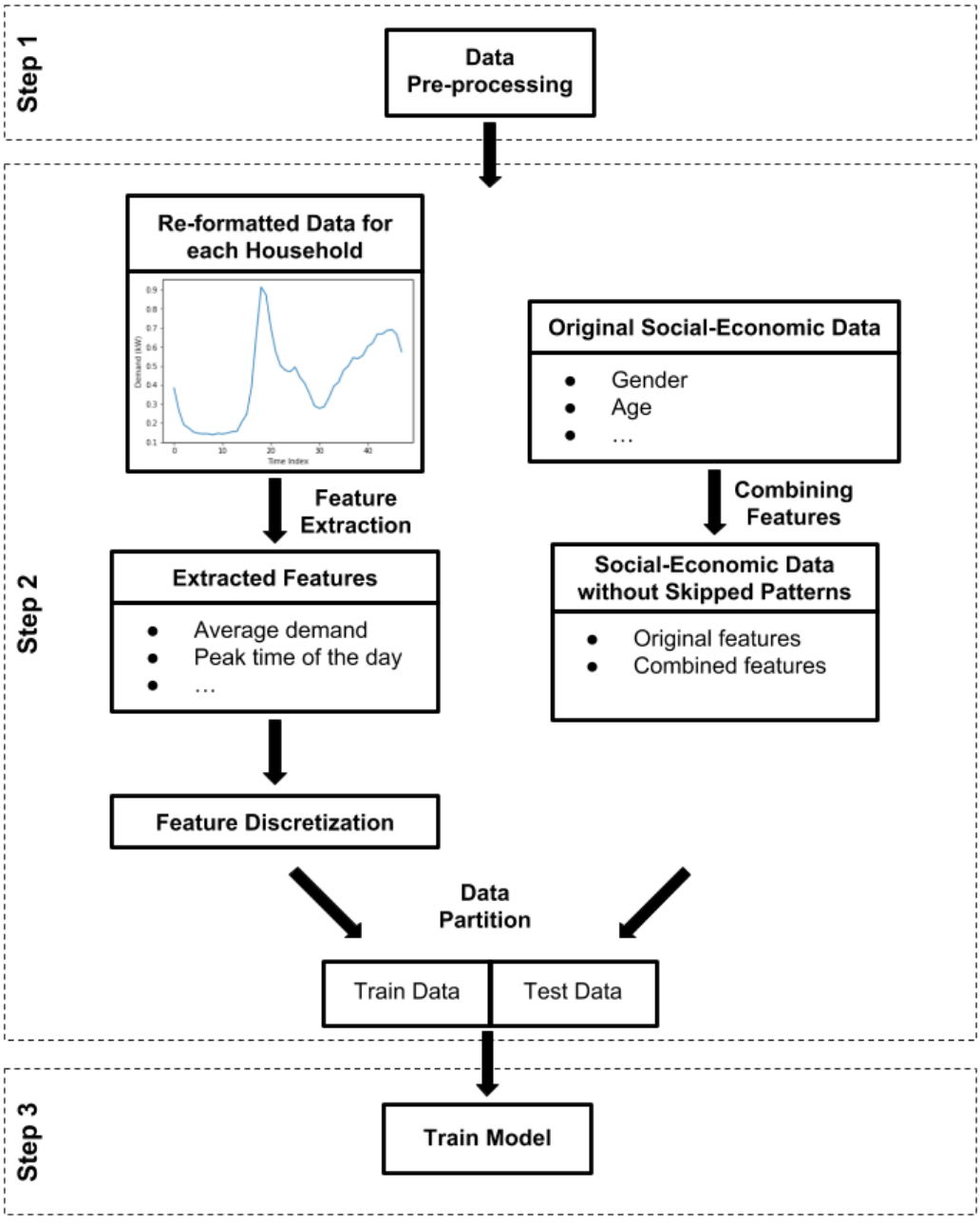

Figure 5-4 Schematic overview of the proposed DCNN based MTL

5.3.2Label Generation and Feature Combination

As can be seen from Figure 5-4, the second step consists of multiple sub-steps. Details of these sub-steps are given below:

1) Load Characteristics Extraction:

Smart meter data in their original form are hard to predict using just social-economic data and more importantly, do not provide direct and intuitive information about the consumer's consumption characteristics.

Instead, features of interest could be designed manual and extracted from the smart meter data, such as peak demand and average demand. As described in Chapter 5, the same set of load features/characteristics are extracted here. The full set of the features can be found in Appendix B. One problem with the full set of the extracted features is the inter-correlation between different features. For example, for most people, the average annual peak demand and the average peak demand during summer or spring are highly correlated. The prediction of either of the features could be directly used as the prediction for the other correlated features. Hence, the full set of the data are decorrelated, i.e., only keep one feature for a group of highly correlated features. Then a subset of the original feature set can be obtained and is used as the dataset of load characteristics.

2) Load Characteristics Discretization:

The obtained load characteristics are extracted from the smart meter data and are continuous. Instead of forecasting the exact value of the characteristics, the class (high, medium, and low) of the characteristic is proposed to be predicted. In order to get the classes of different characteristics, the characteristics should be discretized. There are many techniques that could convert numerical values to discrete counterparts. As the prediction problem in this chapter is formulated as a classification problem, a balanced distribution of the different classes are desirable. Hence, the equal-frequency discretization strategy is applied on the continuous, i.e., the tertiles of each set of load characteristic are used to divide each set of characteristics into three equally sized parts (high, medium, and low). Then the discretized load characteristics can be used as the labels for the subsequent training of the model. For better demonstration purpose, a set of 10 features are used for testing the performance of the proposed method.

3) Social-Economic Feature Combination:

The missing data issue has been solved for smart metering data. As for the social-economic data, the same issue is encountered as well. The difference is the missing data in smart meter data is mostly unintentional, for example could be due to communication failure and database failure, whereas the missing data in the social-economic data set are designed so intentionally.

The social-economic data are essentially converted from the answers from the questionnaire. For this particular questionnaire, multiple skipped patterns are identified, leading to a situation where all the consumers have missing data.

In a questionnaire, the skip pattern is a series of questions associated with a conditional response. It is designed so to pertain only to certain respondents of a question. For instance, in this data set, a question inquiries whether a respondent has a dishwasher at home. The skip pattern for this question is asking 'if you have a dishwasher at home, how often do you use it?'. The questions skipped by those who do not have dishwashers might refer to issues that are relevant only to those who have. Multiple skipped patterns have been identified in the questionnaire and are listed in Table 5-1.

Questions in each skipped pattern are then combined and encoded to one numerical value.

Table 5-1 Skipped patterns in the questionnaire

No.	Question Indexes
1	405,406
2	410,420,430,43111,4312
3	4332,4321,4352,433,434
4	453,4531
5	4701,4801
6	471,472
7	473,474
8	49002,49001,49004,490004,4900004,4900005,4900006,4900007,4900008, 4900009
9	490002,4902,49022
10	5414,5145,54155

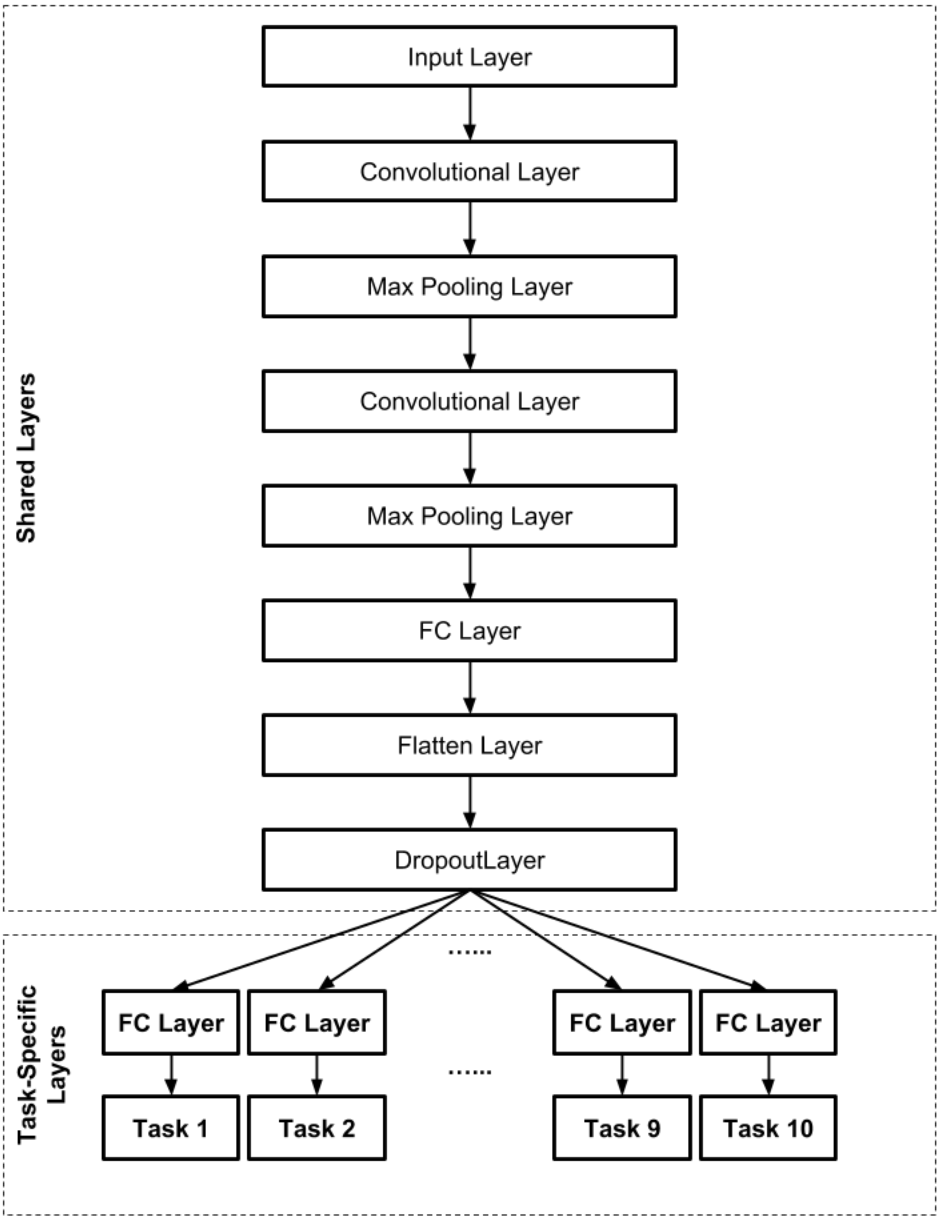

Figure 5-5 Proposed structure of DCNN based MTL model

The converted social-economic data and the social-economic data that do not have skipped patterns are then standardised, which is given by (3-1). They these standardised social-economic data (112) and discretized load characteristics are then partitioned into training data set (80%) and test data set (20%).

5.3.3CNN based MTL Model Building and Training

Figure 5-5 demonstrates the structure of the proposed DCNN based MTL model. It uses the hard parameter sharing technique, with 7 hidden layers being shared. The filter/kernel size in the two convolutional layers is the same, which is 1×3. In addition,

5 filters are used in the first convolutional layer and 10 filters are used in the second convolutional layer. For the filters in the max pooling layers, the size is set to 1×2.

As for the fully-connected layers, 20 neurons are used for the shared FC layer, and 10 neurons are used in each of the FC layers for the task-specific layers.

For the training of the network, Adam optimiser is adopted, with the loss being the sum of the ten tasks' softmax cross entropy.

In order to prevent the overfitting of the network, the following techniques have been used.

1) Data Argumentation:
To prevent overfitting of the network, one solution is to increase the size of the training dataset. However, in this case, no similar dataset could be found. Instead, the social-economic data are augmented by adding Gaussian noise to them;
2) Dropout:
A dropout layer is inserted in the shared layers. The function of the dropout layer is to randomly set the output a fraction of the neurons to 0 with a preferred probability. Dropout has been widely used to prevent overfitting, as it can prevent the network from memorising certain instance-specific patterns.
3) Early Stopping:
Early stopping comes in many forms. In the training of the proposed model, if 5 consecutive epochs have passed and no loss improvement has been seen on the validation dataset (20% of the training dataset), the training will stop.
4) Regularization:
L2 regularization is used to penalize the squared magnitudes of the network parameters. Intuitively, this would help reduce the occurrence of spiky parameters in the model, unless those weights greatly improve the prediction performance. The weight matrix of the model is more likely to be more diffused. Hence, the network will become less likely to heavily utilize only a few of the input features and overfit.

5.4 Results and Discussion

A set of 10 load characteristics are selected to test the proposed method. They are listed in Table 5-2. The details of how the load characteristics are extracted are given in Appendix B.

Table 5-3 List of selected load features

No.	Load Features
1	annual_ave_daily
2	autumn_ave_peak
3	annual_holiday_ave_std
4	summer_weekend_ave_evening
5	autumn_weekend_ave_noon
6	summer_weekday_ave_noon
7	annual_holiday_ave_noon
8	annual_holiday_ave_valley
9	autumn_weekday_ave_morning
10	summer_weekday_ave_night

The proposed method is tested using the test dataset which was defined in Section 5.3. The performance of the proposed method is compared with three state-of-the-art methods, k-Nearest Neighbour (KNN), XGBoost (XGB), and Support Vector Machine (SVM).

Table 5-2 Prediction accuracies of different methods

No.	KNN	SVM	XGB	Proposed	Improvement
1	52.79%	57.36%	58.36%	**64.17%**	9.95%
2	54.30%	58.62%	**59.37%**	58.77%	-1.02%
3	50.02%	54.47%	55.35%	**56.70%**	2.44%
4	52.79%	56.10%	56.35%	**58.88%**	4.48%
5	50.40%	54.09%	55.97%	**56.70%**	1.29%
6	48.51%	53.46%	53.33%	**57.94%**	8.39%
7	49.90%	53.21%	53.96%	**56.70%**	5.07%
8	46.00%	50.82%	52.33%	**53.58%**	2.40%
9	48.51%	49.43%	51.70%	**54.21%**	4.85%
10	45.50%	49.94%	52.83%	**53.48%**	1.22%

The labels in the dataset are balanced due to the way they were discretised. Hence, the prediction accuracy alone is capable of quantifying the prediction performance.

The prediction accuracies of the proposed method and other benchmarking methods are listed in Table 5-3.

As can be observed from the table, the proposed method outperformed the competing methods for all the load characteristics, except for the second load characteristic. For the second characteristic, the proposed method is the second best. Additionally, apart from the proposed method, XGB gives the most accurate prediction, possibly due to that its tree nature is better at handling mixed types of data.

For the ten selected load characteristics, the proposed method could improve the prediction accuracy of the second best model by 3.91% on average. The prediction of the first load characteristic could be even improved by 9.95%. Besides the performance improvement, only one model was built and trained.

5.5 Chapter Summary

The research problem in the previous chapter has been extended to a reverse context in this chapter.

A Deep Convolutional Neural Network (DCNN) based multi-task learning (MTL) method is proposed to infer the consumer's smart metering features from social-economic data. It leverages the convolutional kernel and deep architecture in DCNN to overcome the hurdle brought by mixed types of social-economic data. The MTL framework added to the DCNN enables to infer multiple load characteristics simultaneously and more accurately.

Case studies have been conducted to validate the performance of the proposed method. Results show that on average, the proposed method outperforms the previously best model by 3.91%. For some specific characteristics, the improvement could even reach 9.95% and 8.39%.

The proposed method has provided the necessary analytic for energy consumers, retailers, and DNOs to understand the energy behaviours without the need to install costly smart meters.

Chapter 6

Inferring Phase Connectivity from Smart Metering Data

This chapter proposes to a novel spectral and saliency analysis (SSA) methodology to identify households' phases using their smart metering data. Specifically, the proposed method combines spectral and temporal domain feature extraction techniques on the smart metering data and do not require 100% smart meter penetration ratio in the network.

6.1 Introduction

The last two chapters have demonstrated how smart metering data and social-economic data can be linked together. This chapter will discuss how smart metering data could be used for inferring phases, i.e., to identify the physical connectivity of a household in an LV distribution network.

Traditional methods require either inefficient filed check from house to house or installation of costly signal injecting equipment. Recently, with the roll-out of smart meters, methods based on the analysis of smart metering data have been developed: 1) Approaches measure similarities of individual consumer's voltage data. 2) Methods treat phase identification as a subset sum problem using load data. The major challenges are: voltage data is not commonly available from smart meters and solving the subset sum problem to get phase connectivity requires complicated optimization algorithm with heavy computation burdens such as Integer Programming (IP) and Quadratic Programming (QP). Additionally, the subset sum method requires distribution networks with 100% penetration rates of smart meters which are not the cases in the UK.

The major contribution of this chapter is to fill the research gap by developing phase identification approaches under incomplete consumer data situation, i.e., to perform phase identification with only a proportion of consumers having smart meters in the network.

The content of this chapter is cited from the author's published article [143] in IEEE Transactions on Smart Grid. The structure of this chapter is organised in an alternative-based format. The rest of this chapter could be summarised as follows. Section 7.2 presents the published paper, which introduces the problem of phase identification, the formulation of the problem, the proposed SAS algorithm, experiment results, and discussion. Section 7.3 will demonstrate the validity of the proposed method with further experiment and results. Conclusions are then drawn in Section 7.4.

6.2 Phase Identification with Incomplete Data

Phase Identification with Incomplete Data

Minghao Xu, Ran Li, and Furong Li, *Senior Member, IEEE*

***Abstract*— Phase identification is a process to determine which of the three phases a particular house is connected to. The state-of–the-art identification methods usually exploit smart metering data. However, the data sets are not always available and the major challenge is hence to identify phases with incomplete data set. This paper proposes a novel spectral and saliency analysis (SSA) identification method to overcome this hurdle. Spectral analysis is firstly performed to extract the high-frequency features from the incomplete data. Saliency analysis is then adopted to extract salient features from the variations of high-frequency loads in the time domain. Correlation analysis between customer features and the phase features is used to determine customers' phase connectivity. The method is executed iteratively until all customers with smart meters have been allocated to a specific phase or no salient features can be found. It is validated against real data from over 6000 smart meters in Ireland and achieves an accuracy of over 93% with only 10% smart meter penetration ratio in a 100 household network.**

***Index Terms*—Phase identification, spectral analysis, LV distribution network, smart metering data, incomplete data set.**

I. Introduction

Traditional research and design on low voltage (LV) distribution level rarely take the phase connectivity of individual consumption into consideration [1]. This leads to an urgent problem that the existing networks are poorly 3-phase balanced [2]. Such unbalanced loads will lead to extra power loss and reduced lifespans of assets. Recently in the UK, the 3-phase imbalance issue in the existing LV networks have been further aggravated due to the wide deployment of Low Carbon Technologies (LCTs) at household level. In order to accommodate the fast growing LCTs meanwhile considering the phase balance of LV networks, a vital problem that the Distribution Network Operators (DNOs) are facing is to identify which phase a particular house is connected to.

Traditionally, the DNOs would send electricians to check the phase connectivity manually in the field which is inherently inefficient. Installing advanced signal injecting and receiving equipment on both ends of the networks [3], [4] is another option for the DNOs. These devices

are accurate and fast but are at the cost of increased capital and maintenance fees. The introduction of other high-precision metering devices [5]-[9] provides another opportunity to identify phases in an indirect way. The inevitable high cost of these devices become the main obstacle for them to be widely deployed.

Recently, with the roll-out of smart meters, data-driven methods based on the analysis of smarting data have been developed. By the data type they require, these methods could be categorized into two sets:

1) Voltage data [7], [8], [10], [11]: measuring shape similarities between household voltage and phase voltage measured at substation through correlation analysis, regression or clustering techniques. It assumes that consumers share similar voltage patterns within the same phase;

2) Load data [12]-[15]: based on the law of conservation of energy, finding the optimal combination of households to provide similar aggregation load as the phase load.

However, the limitation of the first category of methods is that voltage data are not commonly provided by most smart meters [16]. In the second category, the methods are designed for handling data with small degrees of loss or error, requiring that the distribution networks to be analyzed should have 100% or nearly 100% penetration ratios of smart meters. Whereas smart meters are not widely deployed in most places. In the UK consumers could even opt not to install smart meters [17]. Therefore, there is a critical need to develop phase identification methods with incomplete consumer data i.e., to perform phase identification with only a proportion of consumers having smart meters in the network.

In this paper, a novel approach based on spectral and saliency analysis (SSA) of consumers' load has been developed. SSA aims to extract customer's load features from both time and frequency domains. Hence it could effectively identify phase connection from limited data compared with traditional methods which directly operate on the raw data. Firstly, spectral analysis using Fourier Transform on both consumer's load data and phase load data is performed to filter out the low-frequency components. Then the variations of each consumer's remaining high-frequency load are extracted as their features. Following that, the saliency of these features are assessed to form the salient feature vectors for consumers. Lastly, the salient feature vectors of each consumer are correlated with the corresponding high-frequency variations on each of the three phases. Given the load variations in salient feature vector are significant, the corresponding phase variations should present similar variation pattern and the phase can therefore be identified. To the best of the users' knowledge, this is the first time phase identification has been achieved under incomplete load data condition. It is validated using real smart metering data form Smart Meter Trial in Ireland [18]. This paper has evaluated the accuracies of proposed phase identification method under different data conditions. It was

carried out by gradually changing the smart meter penetration ratio and time length of the available data.

The reminder of this paper is organized as follows: Section II shows how the problem is mathematically formulated and Section III presents the proposed method. Section IV validates the proposed approach and compares it with other technique. Section V draws the conclusions.

II. Problem Formulation

The mathematical model for the problem is developed as follows.

Suppose there are n consumers in this network and for each consumer, m measurements of load data are taken by smart meters over the time. Since the network is three-phase, the set of indices of phases, J, consists only three elements. (1), (2) and (3) represent sets of the indices for phases, consumers, and measurements respectively.

$$\boldsymbol{J} = \{\mathbf{1}, \mathbf{2}, \mathbf{3}\} \tag{1}$$

$$C = \{1, 2, \dots, n\} \tag{2}$$

$$M = \{1, 2, \dots, m\} \tag{3}$$

Let h_{ki} represents the measured load at time k for consumer i. p_{kp} denotes the substation's load for phase p at time k. Consumer load matrix H and phase load matrix L are expressed in (4) (5) respectively.

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h_{11}} & \cdots & \boldsymbol{h_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{h_{m1}} & \cdots & \boldsymbol{h_{mn}} \end{bmatrix} \quad \forall \boldsymbol{n} \in \boldsymbol{C} \quad \forall \boldsymbol{m} \in \boldsymbol{M} \tag{4}$$

$$L = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & p_{k3} \\ \vdots & \vdots & \vdots \\ p_{m1} & p_{m2} & p_{m3} \end{bmatrix} \quad \forall k \in M \tag{5}$$

Let x_{ij} be the phase indicator of consumer i to phase j. 1 means true and 0 indicates false. The connectivity matrix would then be as follows.

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x_{11}} & \boldsymbol{x_{12}} & \boldsymbol{x_{13}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{x_{i1}} & \boldsymbol{x_{i2}} & \boldsymbol{x_{i1}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{x_{n1}} & \boldsymbol{x_{n2}} & \boldsymbol{x_{n3}} \end{bmatrix} \tag{6}$$

Traditionally, the problem is formulated as (7) and various optimization techniques could be applied to get the optimal mathematical solution of X.

$$\boldsymbol{HX} = \boldsymbol{L} \tag{7}$$

However, due to the above formulation, all of the existing methods are fundamentally derived from the law of conservation of energy, i.e., the sum of the individual's loads within the same phase is equal to the corresponding phase load monitored at the substation. The inevitable problem caused by this is that under incomplete data condition, the accuracy of X would be fairly poor.

III. Spectral and Saliency Analysis Algorithm

To tackle the limitation under incomplete data condition, this paper proposes to extract distinct features from individual load profiles and correlate it with phase load to estimate the connectivity. Fig 1 is the overall flowchart for this algorithm. It is achieved by executing the following several steps.

Firstly, it filters out the low-frequency load from each household's and phase load by performing Fourier Transform (FT) and Inverse Fourier Transform (IFT). Then it takes the variations of the remaining load between two arbitrary time intervals as the features. After that, the algorithm analyses the saliency of all the features and extracts the salient variations of each consumer. Lastly, depending on the number of salient variations of the consumer, the method identifies the phase by either correlation analysis or the contribution factor analysis proposed

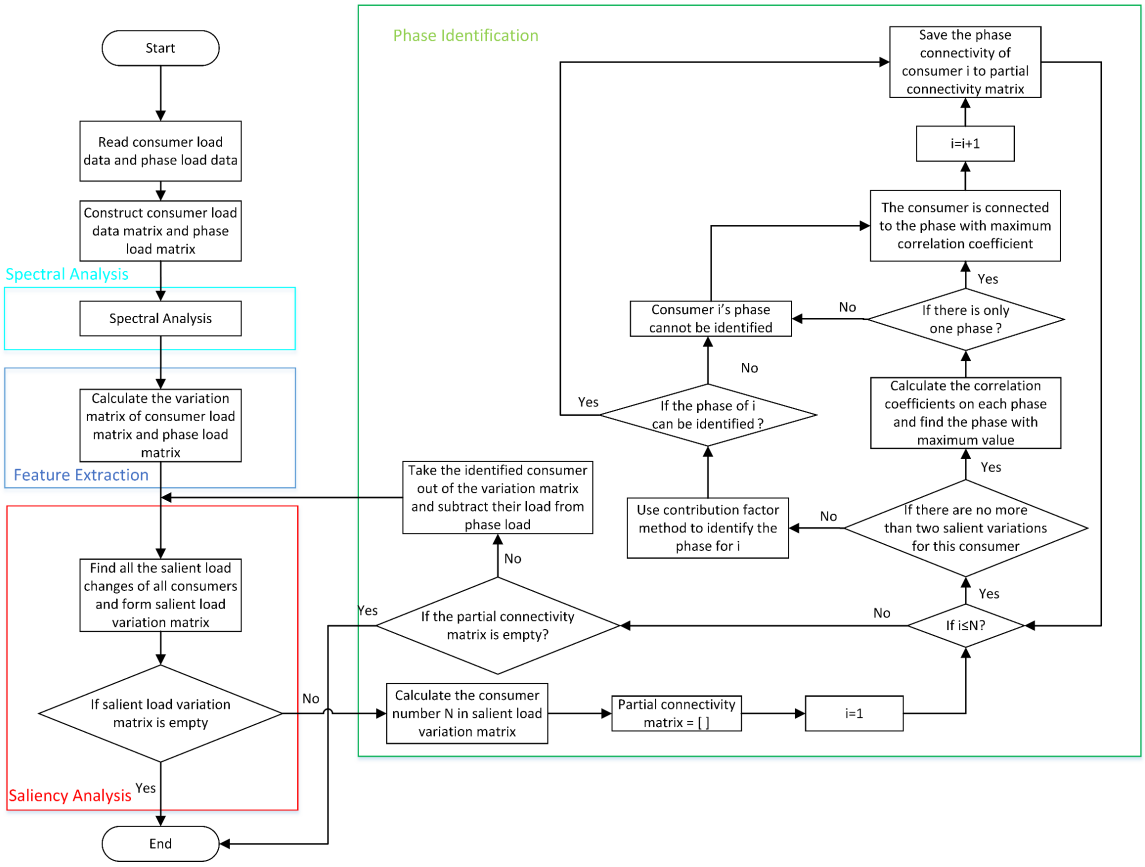

Fig. 1. Flowchart for phase identification using saliency analysis algorithm

in this paper. The algorithm then removes the identified consumers from the overall data set and the above steps will be repeated until: 1) there are no salient variations can be found; 2) no consumer's phase can be identified by analyzing the selected salient variations. Detailed explanations are presented in the following subsections.

A. Spectral Analysis

The first step is to perform spectral analysis on both the consumers' load data and the phase load data. Discrete Fourier Transform (DFT) is applied in the paper to get the spectrum of the data. Suppose b is a time series load profile, which in this paper represents an arbitrary column in the consumer load matrix H or an arbitrary column in the phase load matrix L. The DFT of b is formulated below.

$$\boldsymbol{B_k = \sum_{n=0}^{m-1} b_n e^{-j\frac{2\pi kn}{m}}, k = 0, \dots, m-1} \tag{8}$$

where m is the number of measurement and $B_k = \beta_k e^{j\theta_k}$ is the frequency spectrum with magnitude β_k and phase angle θ_k.

The low-frequency components of load data mostly follow a regular pattern while high-frequency components are different from house to house, representing the unique energy usage habit of the customer. After setting a cut-off frequency, f_c, the magnitudes of the low-frequency harmonics, whose frequencies are below f_c, are all set to zeros. Due to the symmetry property of DFT, the harmonics that are symmetrical with the low-frequency harmonics about the Nyquist frequency should be set to zeros as well to completely filter out the low-frequency harmonics. Then the high-frequency load profile in time domain can be obtained by applying Inverse Discrete Fourier Transform (IDFT) with the remaining frequency spectrum.

$$\boldsymbol{b_n^r = \frac{1}{m} \sum_{k=0}^{m-1} B_k e^{j\frac{2\pi kn}{m}}, n = 0, \dots, m-1} \tag{9}$$

where b^r is the reconstructed time series load profile.

After performing DFT and IDFT on the consumer load matrix H and phase load matrix L, the high-frequency parts of the consumer load H_{high} and phase load L_{high} are obtained.

$$\boldsymbol{H_{high} = \begin{bmatrix} h_{high11} & \cdots & h_{high1n} \\ \vdots & \ddots & \vdots \\ h_{highm1} & \cdots & h_{highmn} \end{bmatrix} \quad \forall n \in C \quad \forall m \in M} \tag{10}$$

$$L_{high} = \begin{bmatrix} p_{high11} & p_{high12} & p_{high13} \\ \vdots & \vdots & \vdots \\ p_{highk1} & p_{highk2} & p_{highk3} \\ \vdots & \vdots & \vdots \\ p_{highm1} & p_{highm2} & p_{highm3} \end{bmatrix} \quad \forall k \in M \tag{11}$$

B. Feature Extraction

The second step is to extract the features from the remaining high-frequency load profiles. As mentioned in the introduction section, there exist identification methods using signal injecting equipment. The injector poses a unique electric signal from the demand side. At the substation, there are three receivers waiting to detect the signal. The phase at which the receiver captures the injected signal is the phase which the household is connected to. This is in nature to detect the external turbulence in the network. Similarly, the saliency analysis method is proposed in this paper. Instead of injecting external signals into the network, the proposed method seeks salient high-frequency load variations of consumers. For example, during a period, all the households are consuming electricity at a steady level, except for one household. The residents living in this house may return home late and turn on the light, kettle, etc. As a result, the demand or consumed energy within this period for this particular household would increase significantly. On the substation side, the corresponding phase load would increase accordingly. Since other households in the network are consuming energy almost constantly during this period, the load increase at the corresponding phase should be quite noticeable. The phase of the household could then be identified. In this paper, the high-frequency loads are obtained to reveal the unique energy usage habit of the households, and the variations of the high-frequency loads are extracted as features.

The variation of consumer i between two adjacent time intervals k and $(k+1)$ are calculated by (12). They can reflect the change of consumer's energy behavior.

$$\begin{gathered} \boldsymbol{Vh_{hki} = h_{high(k+1)i} - h_{highki}} \\ \boldsymbol{\forall k \in M \quad k \neq 1 \quad \forall i \in C} \end{gathered} \tag{12}$$

The variation of phase j between periods k and $(k+1)$ is expressed as (13).

$$\begin{gathered} \boldsymbol{Vp_{kj} = p_{high(k+1)j} - p_{highkj}} \\ \boldsymbol{\forall k \in M \quad k \neq 1 \quad \forall j \in J} \end{gathered} \tag{13}$$

Hence, the variation matrices of consumer load and phase load are shown in (14) and (15) respectively.

$$\boldsymbol{VH_1} = \begin{bmatrix} \boldsymbol{Vh_{11}} & \cdots & \boldsymbol{Vh_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Vh_{(m-1)1}} & \cdots & \boldsymbol{Vh_{(m-1)n}} \end{bmatrix} \tag{14}$$

$$VL_1 = \begin{bmatrix} Vp_{11} & Vp_{12} & Vp_{13} \\ \vdots & \vdots & \vdots \\ Vp_{k1} & Vp_{k2} & Vp_{k3} \\ \vdots & \vdots & \vdots \\ Vp_{(m-1)1} & Vp_{(m-1)2} & Vp_{(m-1)3} \end{bmatrix} \quad (15)$$

The above variation matrices only represent the variations between two adjacent time intervals. The variation matrices could be further expanded by introducing variations between any two time intervals. Fig 2 demonstrates all the possible load variations of consumer i. There are $m-1$ pairs of columns and each column represents the consumer's load data with m measurements. The right-hand-side column in each pair slides downwards gradually. By subtracting the right-hand-side data from the corresponding left-hand-side data, all the variations can be calculated. For consumer with m measured loads, the total number of variations, N_{va}, is given by (16).

$$\begin{aligned} \boldsymbol{N_{va}} &= \boldsymbol{(m-1)+(m-2)+(m-3)+\cdots+1} \\ &= \boldsymbol{(m-1)m/2} \qquad\qquad \boldsymbol{m>1} \end{aligned} \quad (16)$$

The variation of consumer i's load between time interval k and $(k+\mathrm{t})$ can then be expressed as (17).

$$\begin{aligned} \boldsymbol{Vh_{kit}} &= \boldsymbol{h_{high(k+t)i} - h_{high(ki)}} \\ \boldsymbol{\forall t \in M \quad t \neq m \quad \forall k \in M \quad k \neq 1 \quad \forall i \in C} & \end{aligned} \quad (17)$$

The load variation of phase j between periods k and $(k+\mathrm{t})$

$$\begin{aligned} \boldsymbol{Vp_{kjt}} &= \boldsymbol{p_{high(k+t)j} - p_{high(kj)}} \\ \boldsymbol{\forall t \in M \quad t \neq m \quad \forall k \in M \quad k \neq 1 \quad \forall j \in J} & \end{aligned} \quad (18)$$

The variation matrices of consumer load then become (19).

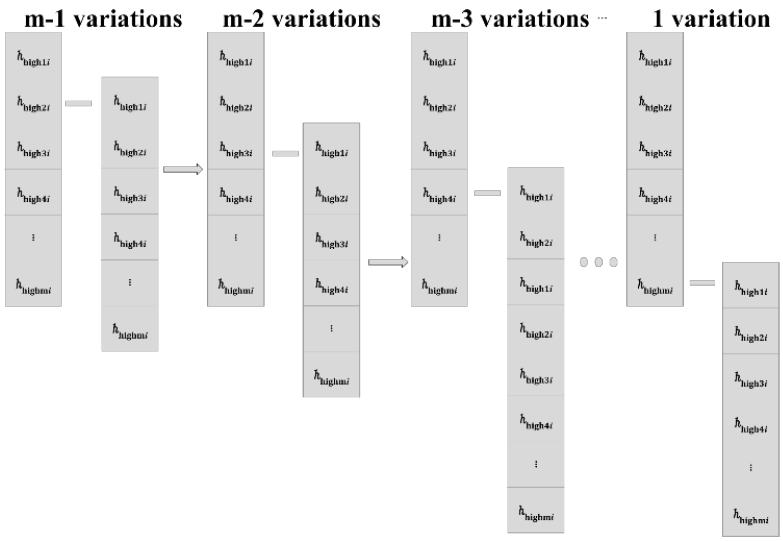

Fig. 2. Load variations between two arbitrary time intervals

$$\boldsymbol{VH} = \begin{bmatrix} \boldsymbol{VH_1} \\ \vdots \\ \boldsymbol{VH_t} \\ \vdots \\ \boldsymbol{VH_{m-1}} \end{bmatrix} \quad \forall \boldsymbol{t} \in \boldsymbol{M} \quad \boldsymbol{t} \neq \boldsymbol{m} \tag{19}$$

where VH_t represents

$$\boldsymbol{VH_t} = \begin{bmatrix} \boldsymbol{Vh_{11t}} & \cdots & \boldsymbol{Vh_{1nt}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Vh_{(m-t)1t}} & \cdots & \boldsymbol{Vh_{(m-t)nt}} \end{bmatrix} \tag{20}$$

The variation matrix of phase load is given in (21).

$$\boldsymbol{VL} = \begin{bmatrix} \boldsymbol{VL_1} \\ \vdots \\ \boldsymbol{VL_t} \\ \vdots \\ \boldsymbol{VL_{m-1}} \end{bmatrix} \quad \forall \boldsymbol{t} \in \boldsymbol{M} \quad \boldsymbol{t} \neq \boldsymbol{m} \tag{21}$$

where VL_t represents

$$\boldsymbol{VL_t} = \begin{bmatrix} \boldsymbol{Vp_{11t}} & \boldsymbol{Vp_{12t}} & \boldsymbol{Vp_{13t}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{Vp_{k1t}} & \boldsymbol{Vp_{k2t}} & \boldsymbol{Vp_{k3t}} \\ \vdots & \vdots & \vdots \\ \boldsymbol{Vp_{(m-t)1t}} & \boldsymbol{Vp_{(m-t)2t}} & \boldsymbol{Vp_{(m-t)3t}} \end{bmatrix} \tag{22}$$
$$\forall \boldsymbol{t} \in \boldsymbol{M} \quad \boldsymbol{t} \neq \boldsymbol{m} \quad \boldsymbol{k} \in \boldsymbol{M} \quad \boldsymbol{k} \leq \boldsymbol{m} - \boldsymbol{t}$$

C. Saliency Analysis

After extracting the variations as the feature, the next step is to analyze the features, i.e., to identify the salient variations. Within the same time step, if a consumer's load variation is significantly higher than the sum of other loads' variation, this load variation is defined as a salient variation. Mathematically, the salient changes are defined as the changes which satisfies the following condition.

$$|\boldsymbol{Vh_{kit}}| \geq \boldsymbol{TH} \times \sum_{\boldsymbol{c=1, c \neq k}}^{\boldsymbol{n}} \boldsymbol{Vh_{kct}} \tag{23}$$

where:

$\mathbf{Vh_{kit}}$	The load change of consumer $\boldsymbol{i}$ during time interval $\boldsymbol{k}+\boldsymbol{t}$ and $\boldsymbol{k}$;
$\boldsymbol{n}$	The consumer number in the network;
$\boldsymbol{TH}$	The threshold value to adjust how salient the changes are.

The reason for selecting the salient variations is that they are more likely to be observed from phase load variation. In other words, since each consumer is connected to only one phase, the salient variation of one consumer is more likely to cause the load variation of the corresponding phase. TH can be changed so that the saliency is adjustable according to various data conditions.

D. Phase Identification

For each consumer with m measurements, there are in total $(m-1)m/2$ variations could be calculated. Suppose there are g salient variations for consumer i. There salient variations form a row vector SVh_i.

$$\boldsymbol{SVh_i} = [\boldsymbol{SVh_{i1}} \quad \boldsymbol{SVh_{i2}} \quad \cdots \quad \boldsymbol{SVh_{ig}}] \tag{24}$$

Accordingly, l phase load variations can be found for each phase, SVl_{i1}, SVl_{i2}, SVl_{i3}.

$$\boldsymbol{SVl_{i1}} = [\boldsymbol{SVl_{i11}} \quad \boldsymbol{SVl_{i12}} \quad \cdots \quad \boldsymbol{SVl_{i1g}}] \tag{25}$$

$$SVl_{i2} = [SVl_{i21} \quad SVl_{i12} \quad \cdots \quad SVl_{i1g}] \tag{26}$$

$$SVl_{i3} = [SVl_{i31} \quad SVl_{i32} \quad \cdots \quad SVl_{i3g}] \tag{27}$$

The consumer's salient variations are correlated with the corresponding phase variations on each of the three phases. Three correlation coefficients are then obtained for each consumer. By selecting the phase which is tightly coupled with the consumer's load variations, the phase connectivity of the consumer can then be identified. Pearson's correlation coefficient is adopted to indicate how strong the consumer's load variation and the phase load variation are correlated with each other. It is formulated below.

$$\begin{aligned} \boldsymbol{\rho}(\boldsymbol{SVh_i}, \boldsymbol{SVl_{ij}}) &= \frac{\boldsymbol{cov}(\boldsymbol{SVh_i}, \boldsymbol{SVl_{ij}})}{\boldsymbol{\sigma_{SVh_i}}\, \boldsymbol{\sigma_{SVl_{ij}}}} \\ &= \frac{\mathbf{1}}{\boldsymbol{g}-\mathbf{1}} \sum_{\boldsymbol{ind}=\mathbf{1}}^{\boldsymbol{g}} \frac{(\boldsymbol{SVh_{i\,ind}} - \boldsymbol{\mu_{SVh_i}})(\boldsymbol{SVl_{ij\,ind}} - \boldsymbol{\mu_{SVl_{ij}}})}{\boldsymbol{\sigma_{SVh_i}}\boldsymbol{\sigma_{SVl_{ij}}}} \end{aligned} \tag{28}$$

where:

$\boldsymbol{SVh_i}$	Salient variations of consumer $\boldsymbol{i}$;
$\boldsymbol{SVl_{ij}}$	Corresponding salient variations of consumer i on phase j;
$\boldsymbol{cov}(\boldsymbol{SVh_i}, \boldsymbol{SVl_{ij}})$	Covariance of SVh_i and SVl_{ij};
$\boldsymbol{\sigma_{SVh_i}}$	Standard deviation of SVh_i;
$\boldsymbol{\sigma_{SVl_{ij}}}$	Standard deviation of SVl_{ij}.
$\boldsymbol{g}$	Number of salient variations for consumer i, $g > 1$;
$\boldsymbol{ind}$	Index of salient variation in the row vector;
$\boldsymbol{\mu_{SVh_i}}$	The mean value of SVh_i;
$\boldsymbol{\mu_{SVl_{ij}}}$	The mean value of SVl_{ij}.

ρ is a quantitative measure of the correlation between two series. After computing the three coefficients for each consumer, phase of the consumer is identified by selecting the phase with maximum correlation coefficient.

However, when the load data are not measured for a long period (m is small), the number of salient variations (g) of the consumer is likely to be small. When there is only one salient variation, ($g-1$) and the two standard deviations equal to zero. As a result, (28) cannot be used. As for consumers with two salient variations, the correlation coefficients could only reflect the changing trend of the two series. In other words, whatever the two variations are in the two series, ρ will be either 1 or -1. This is proved as follows.

Suppose the two salient variations for consumer i are as (29).

$$\boldsymbol{SVh_i} = [\boldsymbol{SVh_{i1}} \quad \boldsymbol{SVh_{i2}}] \tag{29}$$

The corresponding load variation on phase j is as follows.

$$\boldsymbol{SVl_{ij}} = [\boldsymbol{SVl_{ij1}} \quad \boldsymbol{SVl_{ij2}}] \tag{30}$$

Substitute the variables in (29), (31) is obtained.

$$\boldsymbol{\rho}(\boldsymbol{SVh_i}, \boldsymbol{SVl_{ij}}) = \frac{(\boldsymbol{SVh_{i1}} - \boldsymbol{SVh_{i2}})(\boldsymbol{SVl_{ij1}} - \boldsymbol{SVl_{ij2}})}{|\boldsymbol{SVh_{i1}} - \boldsymbol{SVh_{i2}}||\boldsymbol{SVl_{ij1}} - \boldsymbol{SVl_{ij2}}|} \tag{21}$$

For each consumer with only two salient variations, three ρ values can be obtained. While as can be seen from the above equation, the ρ value will be only 1 or -1. This does not provide enough information for phase identification. To tackle the problem caused by consumers with no more than two salient variations, the contribution factor is defined as follows.

$$\boldsymbol{CF_{i\,j\,ind}} = \frac{\boldsymbol{SVh_{i\,ind}}}{\boldsymbol{SVl_{ij\,ind}}} \tag{32}$$

where:

$\boldsymbol{ind}$	Index of salient variations of consumer $\boldsymbol{i}$ in the row vector;
$\boldsymbol{j}$	Phase index, $\forall j \in J$
$\boldsymbol{CF_{i\,j\,ind}}$	The contribution factor of the ind's salient variation of consumer i to the corresponding load variation on phase j;

The ratio, $CF_{i\ j\ ind}$, is a measure of how much the phase variation is contributed by the consumer's load variation. With larger value of $CF_{i\ j\ ind}$, there is more confidence in estimating that the consumer is connected to this phase. The contribution factor could be used to help identify phase connectivity for consumer with no than two salient variations. Fig 3 shows how phase is identified for a consumer with no more than two salient variations. This is part of the overall algorithm shown in Fig 2.

IV. Validation and Results

The proposed approach is validated using real smart metering data from the Smart Metering

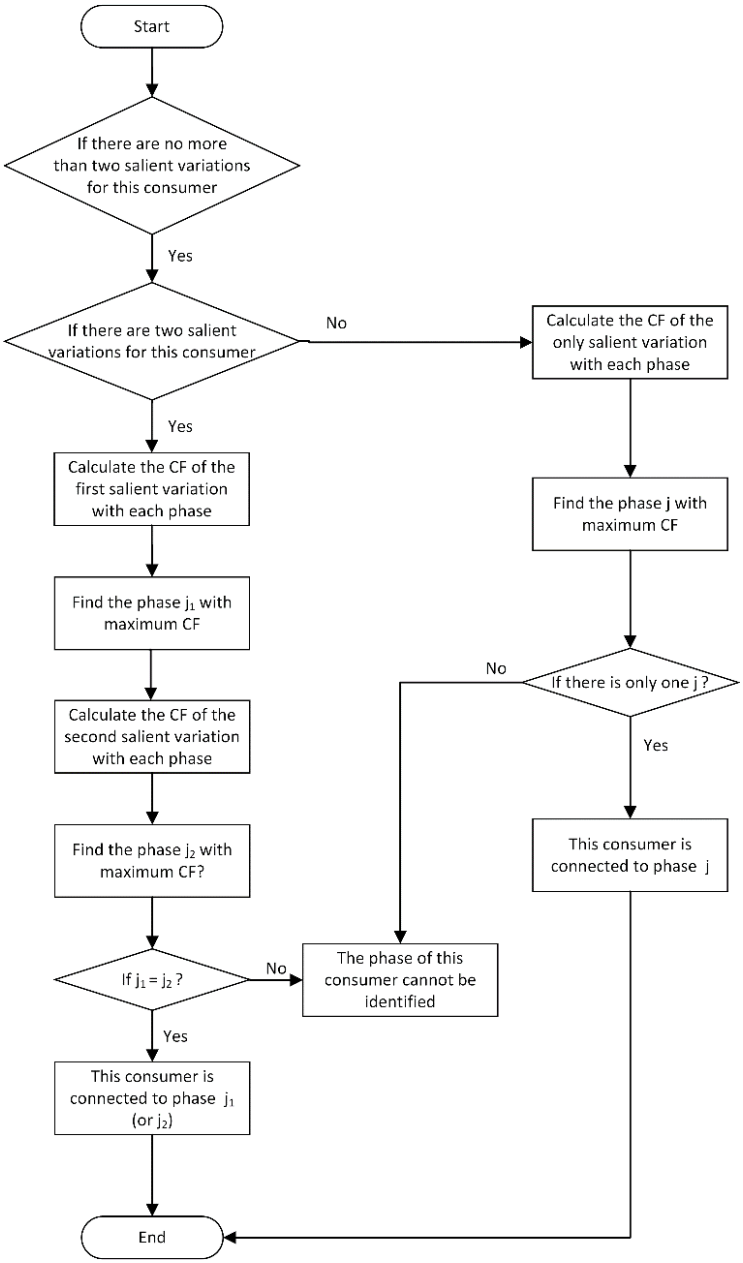

Fig. 3. Phase identification flow chart for consumers with no more two salient variations

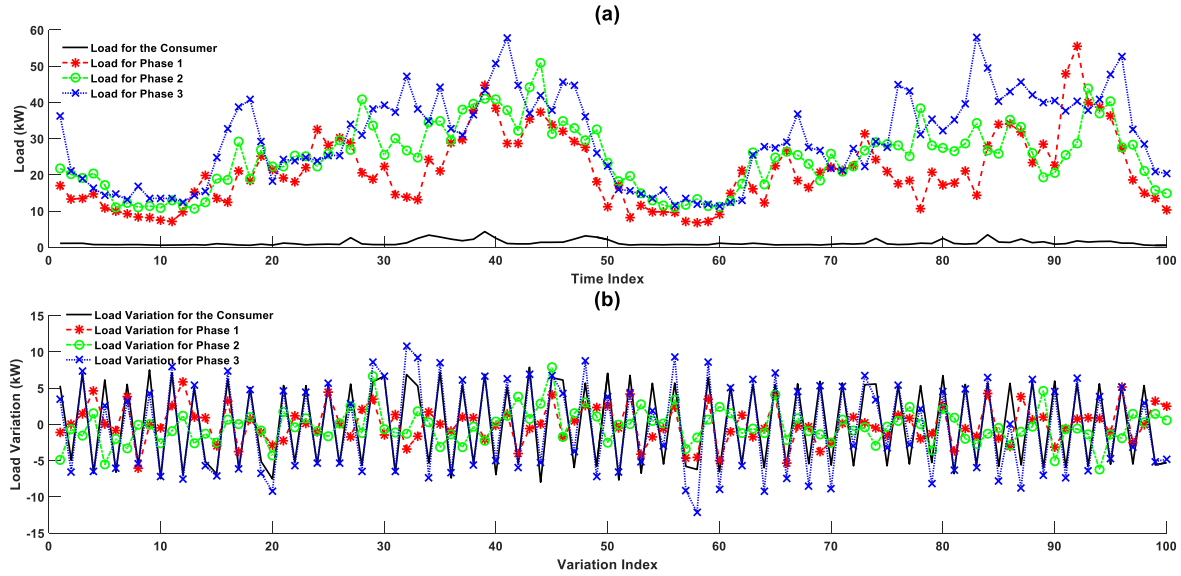

Fig. 4. Correlation improvement: (a) Original load for the consumer and the three phases; (b) Salient variations of the consumer and the three phase variations during the corresponding intervals.

Electricity Customer Behaviour Trials (CBTs) initiated by Commission for Energy Regulation (CER) in Ireland [18]. The data were taken on a half-hourly basis from 1st July 2009 to 31st December 2010.

A. Improvement of Correlation

Unlike voltage data, load data are highly dependent on consumers' behaviours and they do not share any similar patterns within the same phase. In this part, the test shows how the individual consumption correlates with its phase load and how the proposed SSA method improves the degree of correlation.

In this preliminary test, 90 consumers out of 100 in the network have installed smart meters. A total length of two-month data were used to perform phase identification. The black line in Fig 4 (a) shows the load variations over 100 consecutive intervals of a randomly selected consumer connected to phase 3 in the network. As for the red, green, and blue curves, they represent the load on phase 1, phase 2 and phase 3 respectively. As can be observed, both the amplitude and the shape of the curves vary a lot from each other. No noticeable correlated relationship can be observed.

Fig 4 (b) takes the same consumer as in Fig 4 (a) but extracts the consumer's salient variations after spectral analysis as the salient feature vector, represented by the black curve. The red, green, and blue lines demonstrate the load variations during the same periods as the salient load variations on phase 1, phase 2 and phase 3 respectively. Phase 3, which is in blue, is obviously correlated with the consumer's salient load variations. This intuitively indicates that the consumer is connected to phase 3 which matches the real case.

Table I gives the Pearson correlation coefficients between the consumer and the three phases.

TABLE I
CORRELATION COEFFICIENTS BETWEEN CONSUMER AND PHASES

Type	Phase Number	Correlation Coefficients
Correlation analysis of original data	Phase 1	0. 1864
	Phase 2	0. 1662
	Phase 3	0. 1773
Correlation analysis after SSA	Phase 1	-0.1020
	Phase 2	-0.4662
	Phase 3	0.9301

The original load of selected consumer and phase 3 are only correlated by a coefficient of 0.1773, which is even lower than the coefficient with phase 1. However, after SSA, the correlation coefficient is significantly improved to 0.9301, which is much higher than the other two phases.

B. Performance Evaluation of the Proposed Method

This section takes the same distribution network as in Section A which consists of 100 domestic consumers. A comprehensive evaluation is performed by gradually adjusting the smart meter penetration ratio and length of data to be used.

The results of the evaluation are presented in Table II. As shown in the table, the penetration ratio of smart meters in the network increases from 10% to 100% along the horizontal direction and the data length increase from 1 month to 12 months along the vertical direction. Under each data condition, phase identification is performed for multiple times to get an average accuracy. It can be observed in the table that all the identification accuracies are satisfactory and are all above 95% except the most upper-left condition which is identification with one-month data and with 10% smart meter penetration ratio. With fixed data length, the general trend is that the identification accuracy would increase gradually with the increase in smart meter penetration ratio. The reason is that during saliency analysis, each household's high-frequency loads has to be compared with all others' high-frequency loads. With higher smart meter penetration ratio, the saliency analysis becomes more thorough and will return results with more precision and confidence.

C. Comparison with Other Published Method

To our best knowledge, the most comparable method to the proposed algorithm is [12], where the identification process is formulated as a Mixed Integer Quadratic Programming (MIQP) problem. However, this method is designed for handling data with small degrees of loss or error. Additionally, the MIQP method was validated using artificially generated load data and has never been tested under real incomplete data condition. In the paper, the authors replicated the work with small modifications by considering the missing data as noise. Due to license issue, the optimization solver used in this paper is Gurobi instead of CPLEX which was used in the original paper.

To demonstrate the significance of the proposed method over the MIQP method in larger networks, the comparison are performed in 200, 400, 600, 800, and 1000 households networks under various data condition. The results are in Fig. 5. In the figure, there are in total 10 rectangles in two rows wherein the upper row represents the results using the proposed SSA method and the lower row represents the results using MIQP method. Each rectangle consists of 120 (12 × 10) colour coded sub-rectangles and is a graphical representation of the identification accuracy. Within each rectangle, the vertical direction indicates the data length used (1-12 months) and the horizontal direction shows the penetration ratio of smart meters ranging (from 10% to 100%). All of the ten rectangles share a unified colour bar as shown on the right-hand side in Fig. 5. The colour varies from deep blue to deep red with deep red representing 100% identification accuracy.

As illustrated in the figure, several findings are:

1) Both of the methods provides 100% accuracies under complete data condition, i.e., the smart meter penetration ratio is 100%;
2) The performance of both of the methods decrease as the smart meter penetration ratio

TABLE II
OVERALL IDENTIFICATION ACCURACY

Time length (month)	Smart meter penetration ratios in the network									
	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
1	93.29%	96.33%	96.00%	96.84%	98.16%	99.04%	99.96%	99.63%	100.00%	100.00%
2	95.32%	97.35%	97.35%	99.88%	100.00%	99.71%	98.80%	100.00%	100.00%	100.00%
3	99.38%	97.35%	99.38%	98.87%	99.78%	98.70%	100.00%	100.00%	100.00%	100.00%
4	97.35%	96.33%	97.35%	98.36%	98.57%	98.02%	100.00%	100.00%	100.00%	100.00%
5	100.00%	100.00%	96.00%	97.86%	100.00%	99.04%	99.38%	100.00%	100.00%	100.00%
6	97.35%	96.33%	96.67%	99.38%	98.16%	98.70%	100.00%	100.00%	99.60%	100.00%
7	99.38%	100.00%	96.67%	99.38%	98.97%	98.70%	100.00%	100.00%	100.00%	100.00%
8	100.00%	95.32%	98.70%	98.87%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
9	97.35%	96.33%	96.00%	99.38%	96.94%	98.70%	100.00%	99.88%	100.00%	100.00%
10	100.00%	98.36%	100.00%	99.38%	100.00%	100.00%	99.96%	100.00%	100.00%	100.00%
11	95.32%	97.35%	97.35%	100.00%	98.16%	99.71%	99.38%	100.00%	100.00%	100.00%
12	100.00%	99.38%	98.02%	99.88%	99.38%	99.04%	99.67%	100.00%	100.00%	100.00%

TABLE III
COMPARISON OF IDENTIFICATION ACCURACIES WITH 10% SMART METER PENETRATION RATIO

Households Number	Average Accuracy		Accuracy Improvement Percentage
	MIQP Method	Proposed Method	
200	75.87%	91.86%	21.08%
400	75.27%	87.29%	15.97%
600	74.74%	84.97%	13.69%
800	73.46%	81.38%	10.78%
1000	72.79%	80.04%	9.96%

decreases or the data length decreases;

3) With the increase in network size, both of the methods tend to become less accurate slightly. For SSA, the reason is that, with larger network size, the salient features of individual households are more likely to cancel out with each other through aggregation on the phase load and hence it becomes more difficult to find these salient features. For MIQP, with increased network size, the number of possible combinations of households to provide similar aggregation load would increase significantly and hence the accuracy would drop accordingly;
4) The proposed SSA method outperforms the MIQP method under almost every scenario, especially when the data is incomplete (low smart meter penetration and short data length). This can be observed in the top left of every rectangle in Fig. 5 as the upper row are much 'redder' than the ones in the lower row.

To give a clear comparison, the identification accuracy with 10% smart meter penetration ratios in each rectangle are calculated are given in Table III. Compared with the MIQP method, the proposed method lifts the accuracy for networks with 200, 400, 600, 800, and 1000

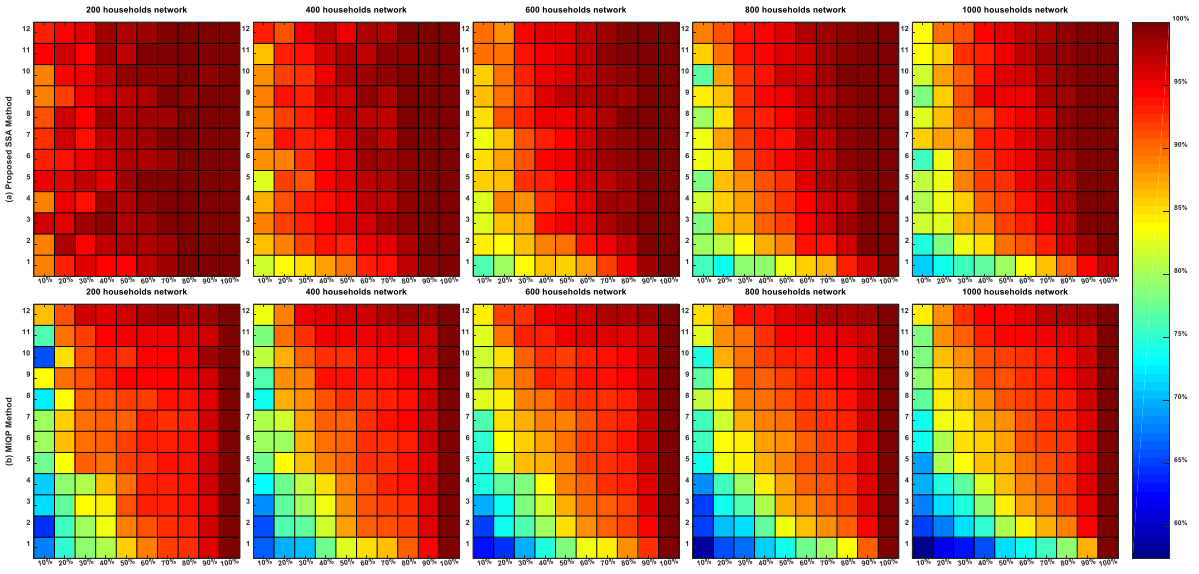

Fig. 5. Comparison of the Performance of the proposed SSA method and MIQP Method: (a) Overall identified accuracies vary with used time and penetration ratios of smart meters using the proposed SSA method in 200, 400, 600, 800, and 1000 households networks; (b) Overall identified accuracies vary with used time and penetration ratios of smart meters using MIQP method in 200, 400, 600, 800, and 1000 households networks.

households by 21.08%, 15.97%, 13.69%, 10.78%, and 9.96% respectively.

V. Conclusion

This paper proposes a novel phase identification method based on spectral and saliency analysis which can be used under incomplete data conditions. It essentially changes the rule on which most of the current identification methods are based. Also, it successfully develops a contribution factor and introduces correlation coefficient to help identify phases after SSA.

To demonstrate the significance of the method, an LV distribution network of 100 domestic consumers has been constructed using real smart metering data from Ireland to perform a preliminary test. Subsequent to the test, a comprehensive and thorough evaluation of the proposed method has been undertaken. Additionally, the performance of the proposed method has been compared with the available optimization method. Results have shown that:

- The SSA could help reveal the correlation between the consumer and the corresponding phase;
- The identification performance of the proposed method would grow with the increase in smart meter penetration ratio and load data length;
- The identification performance of the proposed method would decrease slightly with the increase in network size, i.e., the increase in the total number of household in the network;
- The proposed method outperforms the available method in almost every aspect. Under the extreme data condition where the smart meter penetration ratio is 10% in a 200-household network, the proposed method achieved an average identification accuracy of 91.86 % which is 21.08% higher than the available MIQP method.

The proposed method cannot only reduce uncertainties during the development of smart grid, but also provide necessary tool for DNOs to balance current networks.

Future work will focus on dynamic salience analysis, i.e., saliency criteria will adjust according to different data conditions. Additionally, the proposed method will be validated using networks with multiple consumer types.

References

[1] H. L. Willis, "Characteristics of Distribution Loads," in Electrical Transmission & Distribution Reference Book, ed: ABB Power T&D Company Inc., 1997, pp. 784-808.

[2] K. Ma, R. Li, and F. Li, "Quantification of Additional Asset Reinforcement Cost From 3-Phase Imbalance," IEEE Transactions on Power Systems, vol. PP, pp. 1-7, 2015.

[3] K. J. Caird, "Meter phase identification," ed: US Patents, 2010.

[4] S. Zhiyu, M. Jaksic, P. Mattavelli, D. Boroyevich, J. Verhulst, and M. Belkhayat, "Three-phase AC system impedance measurement unit (IMU) using chirp signal injection," in Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE, 2013, pp. 2666-2673.

[5] C. Chao-Shun, K. Te-Tien, and L. Chia-Hung, "Design of Phase Identification System to Support Three-Phase Loading Balance of Distribution Feeders," Industry Applications, IEEE Transactions on, vol. 48, pp. 191-198, 2012.

[6] K. Te-Tien, C. Chao-Shun, L. Chia-Hung, and H. Chin-Ying, "Design of phase identification system for phase measurement of distribution transformer," in Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on, 2012, pp. 1146-1149.

[7] H. Pezeshki and P. J. Wolfs, "Consumer phase identification in a three phase unbalanced LV distribution network," in Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on, 2012, pp. 1-7.

[8] H. Pezeshki and P. Wolfs, "Correlation based method for phase identification in a three phase LV distribution network," in Universities Power Engineering Conference (AUPEC), 2012 22nd Australasian, 2012, pp. 1-7.

[9] M. H. F. Wen, R. Arghandeh, A. v. Meier, K. Poolla, and V. O. K. Li, "Phase Identification in Distribution Networks with Micro-Synchrophasors," presented at the IEEE Power and Energy Society General Meeting, Denver, 2015.

[10] B. K. Seal and M. F. McGranaghan, "Automatic identification of service phase for electric utility customers," in Power and Energy Society General Meeting, 2011 IEEE, 2011, pp. 1-3.

[11] T. A. Short, "Advanced Metering for Phase Identification, Transformer Identification, and Secondary Modeling," Smart Grid, IEEE Transactions on, vol. 4, pp. 651-658, 2013.

[12] V. Arya, D. Seetharam, S. Kalyanaraman, K. Dontas, C. Pavlovski, S. Hoy, et al., "Phase identification in smart grids," in Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on, 2011, pp. 25-30.

[13] M. Dilek, R. P. Broadwater, and R. Sequin, "Phase prediction in distribution systems," in Power Engineering Society Winter Meeting, 2002. IEEE, 2002, pp. 985-990 vol.2.

[14] P. Satya Jayadev, A. Rajeswaran, N. P. Bhatt, and R. Pasumarthy, "A novel approach for phase identification in smart grids using Graph Theory and Principal Component Analysis," in 2016 American Control Conference (ACC), 2016, pp. 5026-5031.

[15] V. Arya, V. T. Chakaravarthy, K. J. Dontas, S. T. Hoy, J. R. Kalagnanam, S. Kalyanaraman, et al., "Systems and methods for phase identification," ed: Google Patents, 2014.

[16] "Smart Metering Implementation Programme Smart Metering Equipment Technical Specifications Version 1.58," Department of Energy and Climate Change, Ed., ed, 2014.

[17] Ofgem. (20/12/2015). Transition to smart meters. Available: https://www.ofgem.gov.uk/electricity/retail-market/metering/transition-smart-meters

[18] SEAI. (2012, 20/12/2015). Press Release – Full Data from National Smart Meter Trial Published. Available: http://www.seai.ie/News_Events/Press_Releases/2012/National_Smart_Meter_Trial_Data_Release.pdf

6.3 Results and Discussion

The previous section has presented the SAS algorithm to identify phases. However, the rationale and results for combining both the spectral and saliency analysis were not given. To demonstrate the assessment, a network where 10 out of 100 consumers do not have smart meters has been created as an example. As plotted in Figure 6-1, the black line represents a random consumer's load and the red, green, and blue represent the loads on phase 1, phase 2, and phase 3 respectively. It is almost impossible to identify the phase connection from the raw data (correlation to three phases: 0.19, 0.21, and 0.20).

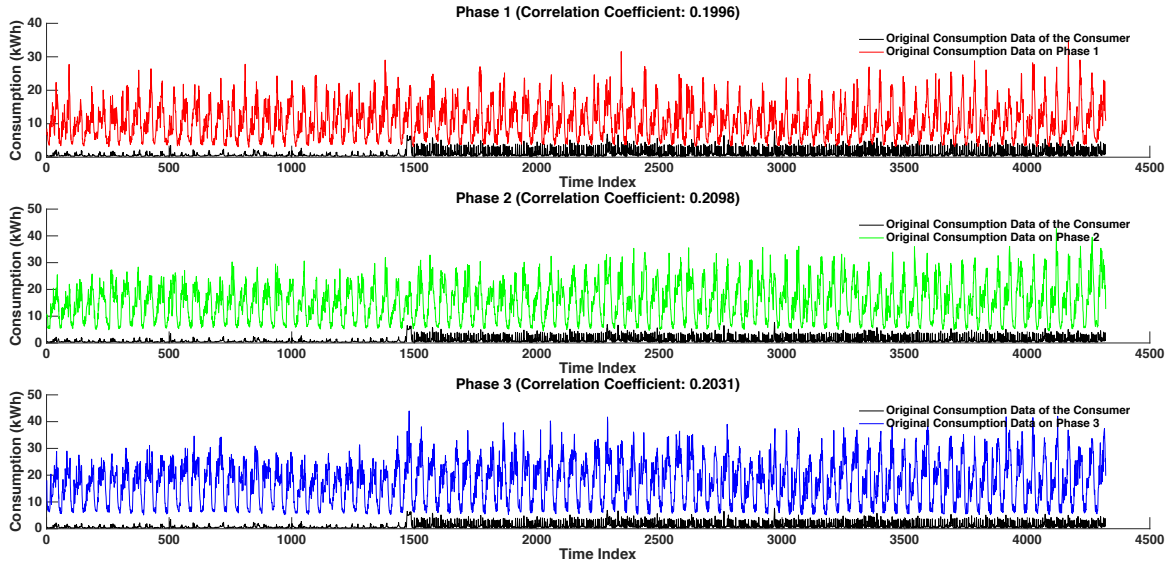

Figure 6-1 The selected consumer's original load over three months VS three phases load

Firstly, the spectral analysis method is performed. In Figure 6-2, the black line represents the high-frequency components of the same consumer and the red, green, and blue lines are for the high-frequency components of phase 1, phase 2 and phase 3 respectively. As can be seen, the correlation to phase 3 increases, but the confidence is still low (0.18, 0.18, 0.20).

Secondly, saliency analysis on the same consumer was performed. As plotted in Figure 6-3, the black line represents the consumer's salient load variations and the red, green, and blue lines are for the corresponding load variations at each phase.

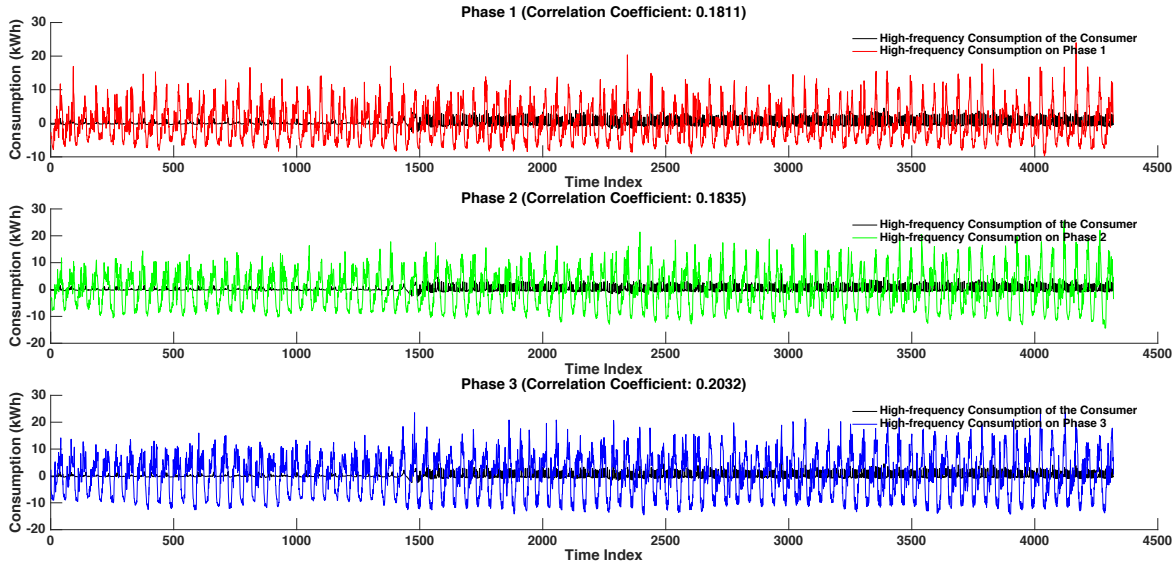

Figure 6-2 Customer's high-frequency load VS three phases high-frequency load

This time, it is quite obvious that the consumer is highly correlated with phase 3 and it turns out that the consumer is indeed connected to phase 3.

Thirdly, the combined Spectral and Saliency Analysis was conducted. Spectral analysis is performed through Discrete Fourier Transform (DFT). After filtering out the low-frequency harmonics, Inverse Discrete Fourier Transform (IDFT) is performed to get the high-frequency load. Then saliency analysis is performed on the high-frequency load. The results were depicted in Figure 6-4. The black line represents the

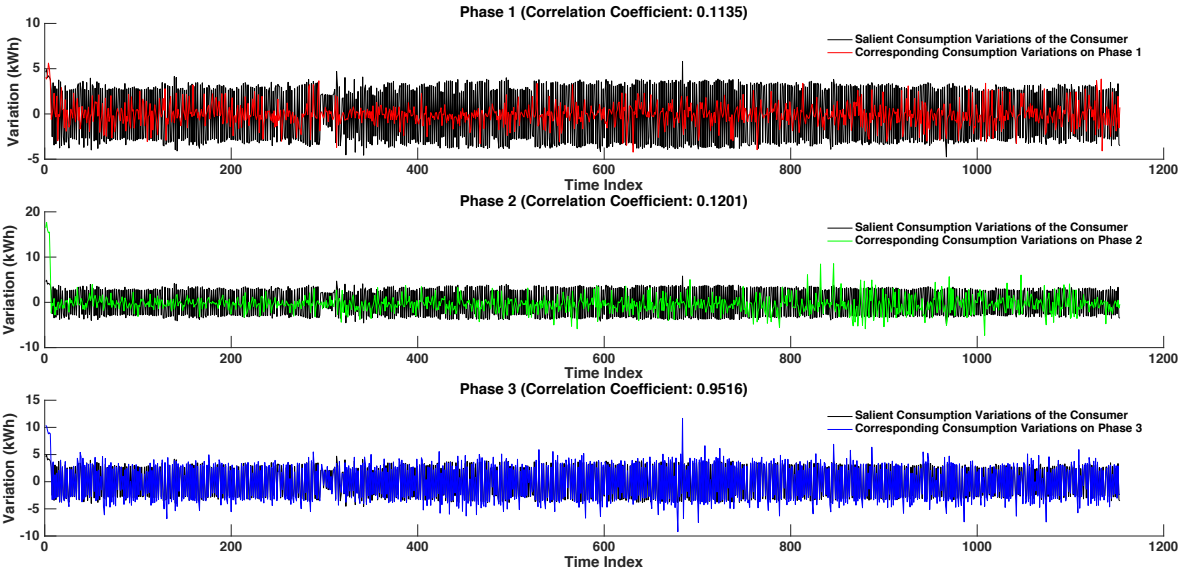

Figure 6-3 Customer's salient variations VS three phases salient variations

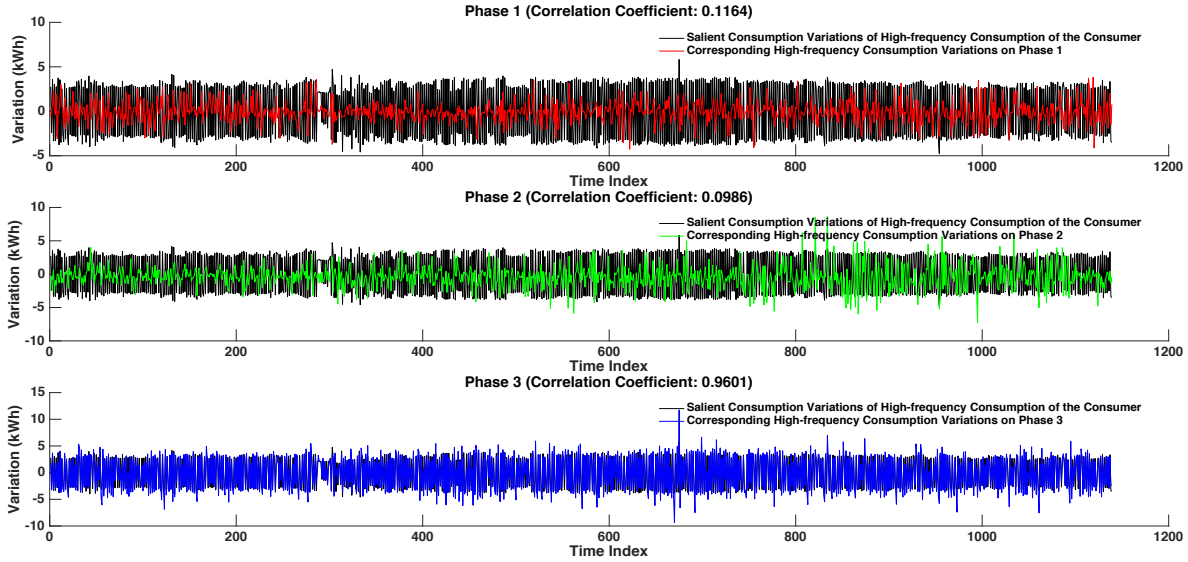

Figure 6-4 The high-frequency salient variations VS three phases high-frequency salient variations

Table 6-1 Overall identification accuracy

Method	Time length (month)											
	1	2	3	4	5	6	7	8	9	10	11	12
Saliency Analysis (%)	88.2	91.0	90.8	89.6	91.0	89.2	89.0	91.0	90.9	91.0	90.2	91.4
Spectral Analysis (%)	91.5	91.4	92.7	93.3	91.1	91.4	90.2	90.8	88.6	92.1	90.5	90.8
SSA (%)	96.9	98.2	98.2	98.2	98.6	99.0	99.4	100.0	100.0	99.8	100.0	100.0

consumer's high-frequency salient load variations and the red, green, and blue lines are for the corresponding high-frequency load variations at each phase. It can be confidently concluded that the customer is connected to phase 3.

Finally, a comprehensive test was performed in networks with a different number of customers and smart meter penetration rates. The average identification accuracies of the three methods are given in Table 6-1. As can be seen, the combined Spectral and Saliency Analysis (SSA) method significantly improves the identification accuracy compared with both methods on their own.

The underlying rationale for the combination is: the saliency analysis is able to extract customers' features in the time domain while the high-frequency components (by spectral analysis) are equivalent to the features in the frequency domain. The

combined method could effectively identify phases from limited data by extracting features from both time and frequency domain.

6.4 Chapter Summary

This chapter proposes a combined Spectral and Saliency Analysis (SSA) method to identify phase, which is more practical and fundamentally different from the existing methods found in the literature. It is validated using real data from Ireland.

Results have shown that the proposed method can identify the phases for networks where smart meters are not 100% penetrated. It outperforms the state-of-the-art method in almost all scenarios, especially when the smart meter penetration ratio is low. Performance of SAS follows the rule that the more measurements the network take and the fewer non-smart-meters the network has, the higher accuracy would be.

The method has proven that smart metering data can not only be used for load forecasting but also can be effectively used to infer physical property of households. The proposed SAS method not only reduced uncertainties in distribution network at an affordable cost but also provided necessary techniques and tools for distribution network operators and designers to better accommodate LCTs in the network and make the distribution network 3-phase power flow in a more balanced manner.

Chapter 7

Conclusions

This chapter draws the conclusion to the thesis by outlining the major contributions and key findings.

Traditional data analytics in the power systems were mainly developed for transmission networks or higher-level systems due to the limited visibility on domestic consumers and distribution networks. The emergence of smart meters not only brings a compelling opportunity to increase the distribution network's visibility and accessibility but also poses tremendous challenges to traditional data analytical tools.

This thesis aims to develop a range of novel analytical methods to handle the smart meter data, which are highly volatile, extremely large in volume, and are constantly generating at a fast speed. Specifically, the research works in the thesis are developed to address two major challenges:

1) The incapability of traditional methods to uncover the underlying patterns of the smart metering data efficiently and effectively, specifically in the area of load forecasting;
2) The incapability of traditional methods to reveal the interconnection between smart meter data and consumer's data from other aspects.

Two main contributions that correspond to the arisen challenges are made in the thesis and are summarized as follows:

1) Development of transfer learning based short-term load forecasting model for domestic consumers. The large amount of smart metering data are ideal for developing deep neural networks for load forecasting. The adoption of transfer learning for the deep models leverages the knowledge learned from one forecasting task for forecasting another, which not only reduces the required computational power compared to training from scratch but also improves the forecasting accuracy;
2) Development of predictive models for i) the two-way inference between smart meter data and consumers' social-economic/demographic status, and ii) the inference of consumer's phase connectivity in the network from smart meter data.

In summary, the research in the thesis demonstrates how to mine useful information about the consumers from smart meter data and other sources of data.

In detail, the research was carried out on the following five perspectives:

7.1 Load Forecasting for Individual Domestic Consumers

Short-term load forecasting (STLF) for the individual residential consumer was traditionally unattainable until the rollout of smart meters. Research has shown that direct application of traditional forecasting methods on smart meter data cannot reach a good prediction.

A novel transfer-learning-based framework is proposed to perform load forecasting. It utilizes the state-of-art deep learning models and applies transfer learning on top of that. By doing so, the strong predictive power of deep learning models can be exploited, at the same time reducing a lot of computational power and efforts to fine-tune each model from scratch.

Two powerful and popular deep models, MLP and LSTM, that are commonly used for sequential data prediction, are evaluated. The following observations are reached:

1) Transfer learning is suitable for point load forecasting, in the sense that it could significantly simplify the deployment of deep models and improve the prediction performance ;

2) The LSTM network is more capable of forecasting load than MLP network;

3) Even when the target consumer and the source consumer have the same amount of smart meter data, transfer learning could still improve the prediction accuracy of MLP and LSTM.

4) For the MLP networks, on average, MAPE, MAE, RMSE, and NRMSE could be improved by 7.80%. 3.16%, 1.13%, and 1.53% respectively;

5) For the MLP networks, the first several hidden layers have higher transferability than the last several layers;

6) For the MLP network, transferring from a dissimilar consumer is generally better than transferring from a similar consumer;

7) For the LSTM networks, as using it along could already achieve a good prediction, the introduction could only improve MAPE and NRMSE by 1.76% and 0.76% respectively;

8) For the LSTM networks, the transferability (measured by improvement of MAPE and NRMSE) of first several hidden LSTM units is higher than the last several ones.

7.2 Inferring Social-Economic Status from Smart Meter Data

Owing to the rapid and ongoing digitalization of the society, a variety of traditionally unavailable information about the consumers has streamed in and becomes available, such as the consumer's age, education, and other social-economic information. Such social-economic information about the consumers is of great significance for not only the retailers but also and DNOs to gain deeper insights on their consumers. However, only a fraction the consumers in the power systems have the social-economic data currently.

A prediction framework that infers the social-economic status of consumers from their smart meter data is proposed to tackle the above-mentioned challenge. It utilises a tree based ensemble model, XGBoost, and consist of three major steps: data pre-processing, feature extraction and model training and hyperparameter optimisation. It has been validated on real data from Ireland and has been compared with three state-of-the-art methods, KNN, SVM and Random Forest.

Results have shown that the proposed method outperforms the state-of-the-art method for almost all features. In addition, due to the tree ensemble nature, the developed models could be interpreted by observing the splits of each individual trees. Specifically, the importance of the smart metering data features in helping the models make their predictions can be explored and ranked. For example, it can be observed that knowing the ratio of consumer's average consumption during noon and evening in cold seasons (Autumn and Winter) would help identify the consumer's age.

The method has proven that it is not only able to predict the social-economic status accurately, but also helpful for gaining insights from the data. DNOs and retailers could leverage the insights and develop tailored DSR schemes and tariffs.

7.3 Inferring Load Characteristics from Social-Economic Data

Following the research on inferring the social-economic status of consumers, the problem could also be extended to a reverse context. Rather than installing costly smart meters, consumers' social-economic information, which can be easily collected by means of surveys, could be used to infer the load characteristics.

The main challenge for performing an accurate prediction in a reverse way arises from the fact that social-economic data normally consist of a mixture of numerical data, categorical data and ordinal data, whereas traditional prediction methods are mostly designed for handling only one type of the data.

A Deep Convolutional Neural Network (DCNN) based multi-task learning (MTL) method is proposed to tackle the problem. The convolutional kernel in DCNN is able to extract features automatically and hierarchically, which is ideal for processing mixed types of data. In addition, the MTL framework added to the DCNN enables to infer multiple load characteristics simultaneously and more accurately, which is much more efficient than predicting each characteristic individually.

Case studies have been conducted and show that on average, the proposed method outperforms the previously best model by 3.91%. For some specific characteristics, the improvement could even reach over 8%.

The proposed method has provided the necessary analytic tool for energy consumers, retailers, and DNOs to understand the energy behaviours without the need to install costly smart meters.

7.4 Inferring Phase Connectivity from Smart Meter Data

Lastly, the smart meter data are used to infer the consumer's phase connectivity, i.e., to identify which of the three phases the consumer's house is connected to in a distribution network.

The state-of-the-art identification methods usually exploit smart meter data. However, the datasets are not always complete and the major challenge is hence to identify phases with incomplete data set.

A novel spectral and saliency analysis (SSA) identification method is proposed to tackle the challenge. Spectral analysis is firstly performed to extract the high-frequency features from the incomplete data. Saliency analysis is subsequently adopted to extract salient features from the variations of high-frequency loads in the temporal domain. Correlation analysis between customer features and the phase features is used to determine customers' phase connectivity.

The proposed method has been validated against real data from Ireland and achieves an accuracy of over 93% with only 10% smart meter penetration ratio in a 100 household network.

The method has proven that smart metering data can not only be used for inferring the social-economic status of a consumer but also can be effectively used to infer the physical property of a household. The proposed SAS method not only reduced uncertainties in distribution networks at an affordable cost but also provided necessary techniques and tools for distribution network operators and designers to better accommodate LCTs in the network and make the distribution network 3-phase power flow in a more balanced manner.

Chapter 8

Future Works

This presents some potential research topics as future work.

8.1 Applications of Transfer Learning Based Load Forecasting

The research in the thesis has already proven the applicability of the proposed transfer learning based load forecasting method for domestic STLF, which is widely acknowledged as one of the most challenging forecasting tasks in the area of load forecasting. There is a wide range of other forecasting paradigms as well.

The proposed method could be adapted for:

1) Network level or system level load forecasting: Load profiles at higher aggregation levels are generally smoother and less volatile. In addition, the accompanied information at higher aggregation level is usually more complete, for example, the temperatures and sunshine conditions are available most of the time. These data are useful for improving the forecasting performance, However, the relationship between the load and other information are highly non-linear and complex, hence is very difficult to learn and model. By applying the proposed method, the complex relationship learned from one forecasting task can be passed onto the next forecasting task, which would save a lot of computation power and efforts to learn from the start.
2) Load forecasting with multiple datasets: The potential of transfer learning has not been fully exploited yet as it is especially powerful when the source domain has more data than the target domain. In the load forecasting context, it means that the new consumer whose load profiles are to be forecasted should have less data than what has already been available and used for training the model. That actually reflects the real situation. By combining multiple datasets and training all the consumers together, a single well-tuned powerful model can be obtained. The model could then be effectively transferred to new consumers with much less data. A significant improvement on the forecasting performance can then be expected.

8.2 Unsupervised Load Feature Extraction

In the proposed methods, the inference between social-economic data and smart meter utilizes the extracted features from smart meter data. The features are extracted

manually, meaning they are designed by the researcher using domain knowledge. The benefit of doing so is that the features are highly meaningful and interpretable. However, manual extraction of the features is not only inefficient but also leads to significant information loss.

Recent research on image processing and natural language processing has shown that, with enough data, features extracted from unsupervised learning could also be applied for supervised tasks, which is classification in our case. As the features are extracted in an unsupervised way, they are more general than the features extracted from task-specific supervised learning. Hence, they could be potentially applied to a wide variety of tasks.

In addition, the load feature extraction is, in essence, a way of compressing the original smart meter data. Hence, by applying unsupervised feature extraction, deeper insights into smart meter data compression could also be gained.

8.3 Interpretations of the Deep Learning Model for Inferring Load Characteristics

The proposed DCNN based MTL model is capable of predicting multiple load characteristics simultaneously. However, it is unclear how the DCNN model makes its predictions.

Though deep learning models have been widely acknowledged as ‘black box‘ models, there have emerged some techniques that are trying to understand what features the deep models are looking for. For example, gradient ascent could be used to inversely construct a set of input features that will minimise the loss. Alternatively, each feature in the input could be masked to assess its impact on the model’s performance and eventually get a saliency map of the input feature.

These techniques could also be applied in the proposed DCNN model. Though applying these techniques would not enable us to fully understand how the DCNN works or makes predictions, it would potentially help us recognize which of the input features (social-economic data in our case) are important for making the decision. Such knowledge would provide deeper insights into the relationship between consumer’s energy behaviours and consumer’s social-economic status.

8.4 Development of Tariff and DSR Recommender System

Recommender systems are widely used by online shopping websites or medium-service (videos and music) providers. The recommender system analyses the historical behaviours of the consumers and recommends contents that the consumers may be interested in.

As identified earlier in the thesis, smart meter data and social-economic data are crucial for consumers in the sense that they can help consumers save energy (through DSR or EMS) and choose better tariffs. However, it is very complicated and time-consuming for the consumers to do so. Moreover, for consumers without any domain knowledge in the field, they are very likely to under-utilize the data and make non-optimal decisions. Building a tariff and DSR scheme recommender system is hence useful to make decisions or provide suggestions for the consumers.

8.5 Interconnecting Datasets from Other Domains

The smart meter data have been linked with consumers' social-economic data and the phase connectivity in this thesis. With the rapid development of the internet and IoT, there will be an increasing amount of consumer-related data from various domains that can be linked with the smart meter data. Such as consumer's driving patterns monitored by the vehicles and consumer's availability at home monitored by smart appliances.

Datasets from different domains or sectors depict the consumers' different characteristics. These datasets can then be linked together to infer one from the others. Alternatively, the datasets can be used collectively to construct a recommender system, which would provide much better suggestions.

Publications

Journal Publications

H. Shi, M. Xu and R. Li, "Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN," in IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5271-5280, Sept. 2018.

M. Xu, R. Li and F. Li, "Phase Identification With Incomplete Data," in IEEE Transactions on Smart Grid, vol. 9, no. 4, pp. 2777-2785, July 2018.

Conference Publications

H. Wyman-Pain, Y. Bian, S. Williams, M. Xu and F. Li, "An assessment of energy deficits in the future electricity system of the United Kingdom with a significant penetration of intermittent renewable generators," 2017 IEEE Power & Energy Society General Meeting, Chicago, IL, 2017, pp. 1-5.

Y. Bian, H. Wang, H. Wyman-Pain, M. Xu and F. Li, "Availability of CHPs to provide primary frequency response in the great Britain power system," 2017 IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, 2017, pp. 1-6.

H. Shi, M. Xu, Q. Ma, C. Zhang, R. Li, and F. Li, "A Whole System Assessment of Novel Deep Learning Approach on Short-Term Load Forecasting," Energy Procedia, vol. 142, pp. 2791-2796, 2017.

Q. Ma, M. Xu, R. Li and F. Li, "Quantitative benefit assessment of electricity settlement using smart meters," 2016 13th International Conference on the European Energy Market (EEM), Porto, 2016, pp. 1-4.

Reference

[1] (2010). *Coping with the Energy Challenge The IEC's role from 2010 to 2030 Smart electrification – The key to energy efficiency White Paper*. Available: https://www.iec.ch/smartenergy/pdf/white_paper_lres.pdf

[2] K. Protocol, "United Nations framework convention on climate change," *Kyoto Protocol, Kyoto,* vol. 19, 1997.

[3] Parliament of the United Kingdom. (2008). *Climate Change Act 2008*. Available: http://www.legislation.gov.uk/ukpga/2008/27/pdfs/ukpga_20080027_en.pdf

[4] (2018). *2017 UK GREENHOUSE GAS EMISSIONS, PROVISIONAL FIGURES*. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/695930/2017_Provisional_Emissions_statistics_2.pdf

[5] (2010). *Creating Britain's low carbon future. Today*. Available: https://www.ofgem.gov.uk/ofgem-publications/84840/2013innovationcompetitionbrochure-pdf

[6] N. Pearsall, K. Hynes, C. Martin, and M. Munzinger, "Analysis of performance parameters for UK domestic PV systems," in *Photovoltaic Energy Conversion, Conference Record of the 2006 IEEE 4th World Conference on*, 2006, vol. 2, pp. 2300-2303: IEEE.

[7] U. DECC, "Renewable energy roadmap," *Crown Copyright, Department of Energy & Climate Change, London,* 2011.

[8] H. Gharavi and R. Ghafurian, "Smart grid: The electric energy system of the future," 2011.

[9] H. Farhangi, "The path of the smart grid," *IEEE power and energy magazine,* vol. 8, no. 1, 2010.

[10] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid—The new and improved power grid: A survey," *IEEE communications surveys & tutorials,* vol. 14, no. 4, pp. 944-980, 2012.

[11] C. Greer *et al.*, "NIST framework and roadmap for smart grid interoperability standards, release 3.0," 2014.

[12] N. Hadjsaïd and J.-C. Sabonnadière, *Electrical distribution networks*. John Wiley & Sons, 2013.

[13] Ofgem. *Transition to smart meters*. Available: https://www.ofgem.gov.uk/electricity/retail-market/metering/transition-smart-meters

[14] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future generation computer systems,* vol. 29, no. 7, pp. 1645-1660, 2013.

[15] D. W. Bunn, "Forecasting loads and prices in competitive power markets," *Proceedings of the IEEE,* vol. 88, no. 2, pp. 163-169, 2000.

[16] H. Shi, M. Xu, and R. Li, "Deep learning for household load forecasting—A novel pooling deep RNN," *IEEE Transactions on Smart Grid,* vol. 9, no. 5, pp. 5271-5280, 2018.

[17] B. Stephen, X. Tang, P. R. Harvey, S. Galloway, and K. I. Jennett, "Incorporating practice theory in sub-profile models for short term aggregated residential load forecasting," *IEEE Transactions on Smart Grid,* vol. 8, no. 4, pp. 1591-1598, 2017.

[18] M. Chaouch, "Clustering-based improvement of nonparametric functional time series forecasting: Application to intra-day household-level load curves," *IEEE Trans. Smart Grid,* vol. 5, no. 1, pp. 411-419, 2014.

[19] A. Marinescu, C. Harris, I. Dusparic, S. Clarke, and V. Cahill, "Residential electrical demand forecasting in very small scale: An evaluation of forecasting methods," in *Software Engineering Challenges for the Smart Grid (SE4SG), 2013 2nd International Workshop on*, 2013, pp. 25-32: IEEE.

[20] A. Veit, C. Goebel, R. Tidke, C. Doblander, and H.-A. Jacobsen, "Household electricity demand forecasting: benchmarking state-of-the-art methods," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 233-234: ACM.

[21] H.-z. Wang, G.-q. Li, G.-b. Wang, J.-c. Peng, H. Jiang, and Y.-t. Liu, "Deep learning based ensemble approach for probabilistic wind power forecasting," *Applied energy,* vol. 188, pp. 56-70, 2017.

[22] X. Qiu, Y. Ren, P. N. Suganthan, and G. A. Amaratunga, "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," *Applied Soft Computing,* vol. 54, pp. 246-255, 2017.

[23] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu, "Short-term residential load forecasting based on resident behaviour learning," *IEEE Transactions on Power Systems,* vol. 33, no. 1, pp. 1087-1088, 2018.

[24] 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,* 2018.

[25] C. Keerthisinghe, G. Verbič, and A. C. Chapman, "A fast technique for smart home management: ADP with temporal difference learning," *IEEE Transactions on smart grid,* vol. 9, no. 4, pp. 3291-3303, 2018.

[26] S. Sun, Q. Yang, and W. Yan, "Optimal temporal-spatial PEV charging scheduling in active power distribution networks," *Protection and Control of Modern Power Systems,* vol. 2, no. 1, p. 34, 2017.

[27] (2018). *Smart Meters Quarterly Report to end December 2017*. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/694355/2017_Q4_Smart_Meters_Report.pdf

[28] A. R. Khan, A. Mahmood, A. Safdar, Z. A. Khan, and N. A. Khan, "Load forecasting, dynamic pricing and DSM in smart grid: A review," *Renewable and Sustainable Energy Reviews,* vol. 54, pp. 1311-1322, 2016.

[29] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting,* vol. 32, no. 3, pp. 914-938, 2016.

[30] C. Xia, J. Wang, and K. McMenemy, "Short, medium and long term load forecasting model and virtual load forecaster based on radial basis function neural networks," *International Journal of Electrical Power & Energy Systems,* vol. 32, no. 7, pp. 743-750, 2010.

[31] W. Charytoniuk and M.-S. Chen, "Very short-term load forecasting using artificial neural networks," *IEEE transactions on Power Systems,* vol. 15, no. 1, pp. 263-268, 2000.

[32] J. W. Taylor, "An evaluation of methods for very short-term load forecasting using minute-by-minute British data," *International Journal of Forecasting,* vol. 24, no. 4, pp. 645-658, 2008.

[33] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, "Very short-term load forecasting: wavelet neural networks with data pre-filtering," *IEEE Transactions on Power Systems,* vol. 28, no. 1, pp. 30-41, 2013.

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

[35] A. Kavousi-Fard, H. Samet, and F. Marzbani, "A new hybrid modified firefly algorithm and support vector regression model for accurate short term load forecasting," *Expert systems with applications,* vol. 41, no. 13, pp. 6047-6056, 2014.

[36] S. Li, P. Wang, and L. Goel, "Short-term load forecasting by wavelet transform and evolutionary extreme learning machine," *Electric Power Systems Research,* vol. 122, pp. 96-103, 2015.

[37] F. Javed, N. Arshad, F. Wallin, I. Vassileva, and E. Dahlquist, "Forecasting for demand response in smart grids: An analysis on use of anthropologic and structural data and short term multiple loads forecasting," *Applied Energy,* vol. 96, pp. 150-160, 2012.

[38] N. Abu-Shikhah and F. Elkarmi, "Medium-term electric load forecasting using singular value decomposition," *Energy,* vol. 36, no. 7, pp. 4259-4271, 2011.
[39] N. Amjady and F. Keynia, "Mid-term load forecasting of power systems by a new prediction method," *Energy Conversion and Management,* vol. 49, no. 10, pp. 2678-2687, 2008.
[40] M. Ghiassi, D. K. Zimbra, and H. Saidane, "Medium term system load forecasting with a dynamic artificial neural network model," *Electric power systems research,* vol. 76, no. 5, pp. 302-316, 2006.
[41] M. Kandil, S. M. El-Debeiky, and N. Hasanien, "Long-term load forecasting for fast developing utility using a knowledge-based expert system," *IEEE transactions on Power Systems,* vol. 17, no. 2, pp. 491-496, 2002.
[42] M. Askari and A. Fetanat, "Long-term load forecasting in power system: Grey system prediction-based models," *Journal of Applied Sciences,* vol. 11, no. 16, pp. 3034-3038, 2011.
[43] T. Hong, J. Wilson, and J. Xie, "Long term probabilistic load forecasting and normalization with hourly information," *IEEE Transactions on Smart Grid,* vol. 5, no. 1, pp. 456-462, 2014.
[44] L. Hernández, C. Baladrón, J. M. Aguiar, B. Carro, A. Sánchez-Esguevillas, and J. Lloret, "Artificial neural networks for short-term load forecasting in microgrids environment," *Energy,* vol. 75, pp. 252-264, 2014.
[45] S. Humeau, T. K. Wijaya, M. Vasirani, and K. Aberer, "Electricity load forecasting for residential customers: Exploiting aggregation and correlation between households," in *Sustainable Internet and ICT for Sustainability (SustainIT), 2013*, 2013, pp. 1-6: IEEE.
[46] S. Haben, J. Ward, D. V. Greetham, C. Singleton, and P. Grindrod, "A new error measure for forecasts of household-level, high resolution electrical energy consumption," *International Journal of Forecasting,* vol. 30, no. 2, pp. 246-256, 2014.
[47] Y.-H. Hsiao, "Household Electricity Demand Forecast Based on Context Information and User Daily Schedule Analysis From Meter Data," *IEEE Trans. Industrial Informatics,* vol. 11, no. 1, pp. 33-43, 2015.
[48] R. E. Edwards, J. New, and L. E. Parker, "Predicting future hourly residential electrical consumption: A machine learning case study," *Energy and Buildings,* vol. 49, pp. 591-603, 2012.
[49] E. Mocanu, P. H. Nguyen, M. Gibescu, and W. L. Kling, "Deep learning for estimating building energy consumption," *Sustainable Energy, Grids and Networks,* vol. 6, pp. 91-99, 2016.
[50] Y. He, R. Liu, H. Li, S. Wang, and X. Lu, "Short-term power load probability density forecasting method using kernel-based support vector quantile regression and Copula theory," *Applied energy,* vol. 185, pp. 254-266, 2017.
[51] P. R. A. P. Department, "PJM Load Forecast Report," January 2015 2014, Available: https://www.pjm.com/-/media/library/reports-notices/load-forecast/2015-load-forecast-report.ashx?la=en.
[52] R. J. Hyndman and S. Fan, "Density forecasting for long-term peak electricity demand," *IEEE Transactions on Power Systems,* vol. 25, no. 2, pp. 1142-1153, 2010.
[53] S. B. Taieb, R. Huser, R. J. Hyndman, and M. G. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid,* vol. 7, no. 5, pp. 2448-2455, 2016.
[54] J. Xie, T. Hong, T. Laing, and C. Kang, "On normality assumption in residual simulation for probabilistic load forecasting," *IEEE Transactions on Smart Grid,* vol. 8, no. 3, pp. 1046-1053, 2017.
[55] P. Wang, B. Liu, and T. Hong, "Electric load forecasting with recency effect: A big data approach," *International Journal of Forecasting,* vol. 32, no. 3, pp. 585-597, 2016.
[56] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirschen, and C. Kang, "Combining Probabilistic Load Forecasts," *IEEE Transactions on Smart Grid,* 2018.

[57] F. McLoughlin, A. Duffy, and M. Conlon, "A clustering approach to domestic electricity load profile characterisation using smart metering data," *Applied Energy,* vol. 141, pp. 190-199, 2015.
[58] C. Beckel, L. Sadamori, T. Staake, and S. Santini, "Revealing household characteristics from smart meter data," *Energy,* vol. 78, pp. 397-410, 2014.
[59] K. Hopf, M. Sodenkamp, I. Kozlovkiy, and T. Staake, "Feature extraction and filtering for household classification based on smart electricity meter data," *Computer Science-Research and Development,* vol. 31, no. 3, pp. 141-148, 2016.
[60] Y. Wang, Q. Chen, D. Gan, J. Yang, D. S. Kirschen, and C. Kang, "Deep Learning-Based Socio-demographic Information Identification from Smart Meter Data," *IEEE Transactions on Smart Grid,* 2018.
[61] X. Tong, R. Li, F. Li, and C. Kang, "Cross-domain feature selection and coding for household energy behavior," *Energy,* vol. 107, pp. 9-16, 2016.
[62] D. Vercamer, B. Steurtewagen, D. Van den Poel, and F. Vermeulen, "Predicting consumer load profiles using commercial and open data," *IEEE Transactions on Power Systems,* vol. 31, no. 5, pp. 3693-3701, 2016.
[63] A. Kavousian, R. Rajagopal, and M. Fischer, "Determinants of residential electricity consumption: Using smart meter data to examine the effect of climate, building characteristics, appliance stock, and occupants' behavior," *Energy,* vol. 55, pp. 184-194, 2013.
[64] F. McLoughlin, A. Duffy, and M. Conlon, "Characterising domestic electricity consumption patterns by dwelling and occupant socio-economic variables: An Irish case study," *Energy and Buildings,* vol. 48, pp. 240-248, 2012.
[65] S. Zhong and K.-S. Tam, "Hierarchical classification of load profiles based on their characteristic attributes in frequency domain," *IEEE Transactions on Power Systems,* vol. 30, no. 5, pp. 2434-2441, 2015.
[66] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Transactions on Power Systems,* vol. 32, no. 3, pp. 2142-2151, 2017.
[67] C. Beckel, L. Sadamori, and S. Santini, "Automatic socio-economic classification of households using electricity consumption data," in *Proceedings of the fourth international conference on Future energy systems*, 2013, pp. 75-86: ACM.
[68] T. Dunmore *et al.*, "Experimental studies of a phase identification system for distribution systems," in *Transmission and Distribution Conference and Exposition, 2010 IEEE PES*, 2010, pp. 1-4.
[69] K. J. Caird, "Meter phase identification," ed: US Patents, 2010.
[70] S. Zhiyu, M. Jaksic, P. Mattavelli, D. Boroyevich, J. Verhulst, and M. Belkhayat, "Three-phase AC system impedance measurement unit (IMU) using chirp signal injection," in *Applied Power Electronics Conference and Exposition (APEC), 2013 Twenty-Eighth Annual IEEE*, 2013, pp. 2666-2673.
[71] H. Pezeshki and P. J. Wolfs, "Consumer phase identification in a three phase unbalanced LV distribution network," in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*, 2012, pp. 1-7.
[72] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 15, no. 10, pp. 1042-1052, 1993.
[73] Y. Chien, "Pattern classification and scene analysis," *Automatic Control, IEEE Transactions on,* vol. 19, no. 4, pp. 462-463, 1974.
[74] H. Pezeshki and P. Wolfs, "Correlation based method for phase identification in a three phase LV distribution network," in *Universities Power Engineering Conference (AUPEC), 2012 22nd Australasian*, 2012, pp. 1-7.

[75] B. K. Seal and M. F. McGranaghan, "Automatic identification of service phase for electric utility customers," in *Power and Energy Society General Meeting, 2011 IEEE*, 2011, pp. 1-3.
[76] T. A. Short, "Advanced Metering for Phase Identification, Transformer Identification, and Secondary Modeling," *Smart Grid, IEEE Transactions on,* vol. 4, no. 2, pp. 651-658, 2013.
[77] C. Chao-Shun, K. Te-Tien, and L. Chia-Hung, "Design of Phase Identification System to Support Three-Phase Loading Balance of Distribution Feeders," *Industry Applications, IEEE Transactions on,* vol. 48, no. 1, pp. 191-198, 2012.
[78] K. Te-Tien, C. Chao-Shun, L. Chia-Hung, and H. Chin-Ying, "Design of phase identification system for phase measurement of distribution transformer," in *Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on*, 2012, pp. 1146-1149.
[79] J. C. Rawlins, "The Sine Wave and Phase," in *Basic AC Circuits*, 2 ed: Newnes, 2000.
[80] A. G. Phadke, "Synchronized phasor measurements in power systems," *Computer Applications in Power, IEEE,* vol. 6, no. 2, pp. 10-15, 1993.
[81] C. W. Liu and J. Thorp, "Application of synchronised phasor measurements to real-time transient stability prediction," *Generation, Transmission and Distribution, IEE Proceedings-,* vol. 142, no. 4, pp. 355-360, 1995.
[82] M. H. F. Wen, R. Arghandeh, A. v. Meier, K. Poolla, and V. O. K. Li, "Phase Identification in Distribution Networks with Micro-Synchrophasors," presented at the IEEE Power and Energy Society General Meeting, Denver, July 2015, 2015.
[83] A. von Meier, D. Culler, A. McEachern, and R. Arghandeh, "Micro-synchrophasors for distribution systems," in *Innovative Smart Grid Technologies Conference (ISGT), 2014 IEEE PES*, 2014, pp. 1-5.
[84] (2014). *Smart Metering Implementation Programme Smart Metering Equipment Technical Specifications Version 1.58*.
[85] M. Dilek, R. P. Broadwater, and R. Sequin, "Phase prediction in distribution systems," in *Power Engineering Society Winter Meeting, 2002. IEEE*, 2002, vol. 2, pp. 985-990 vol.2.
[86] V. Arya *et al.*, "Phase identification in smart grids," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, 2011, pp. 25-30.
[87] A. Navarro-Espinosa and L. F. Ochoa, "Probabilistic impact assessment of low carbon technologies in LV distribution systems," *IEEE Transactions on Power Systems,* vol. 31, no. 3, pp. 2192-2203, 2016.
[88] F. Li, R. Li, Z. Zhang, M. Dale, D. Tolley, and P. Ahokangas, "Big Data Analytics for Flexible Energy Sharing: Accelerating a Low-Carbon Future," *IEEE power and energy magazine,* vol. 16, no. 3, pp. 35-42, 2018.
[89] T. Morstyn, A. Teytelboym, and M. D. McCulloch, "Bilateral contract networks for peer-to-peer energy trading," *IEEE Transactions on Smart Grid,* 2018.
[90] J. Aghaei and M.-I. Alizadeh, "Demand response in smart electricity grids equipped with renewable energy sources: A review," *Renewable and Sustainable Energy Reviews,* vol. 18, pp. 64-72, 2013.
[91] V. M. Balijepalli, V. Pradhan, S. Khaparde, and R. Shereef, "Review of demand response under smart grid paradigm," in *Innovative Smart Grid Technologies-India (ISGT India), 2011 IEEE PES*, 2011, pp. 236-243: IEEE.
[92] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *IEEE Communications Magazine,* vol. 49, no. 4, 2011.
[93] Y.-S. Son, T. Pulkkinen, K.-D. Moon, and C. Kim, "Home energy management system based on power line communication," *IEEE Transactions on Consumer Electronics,* vol. 56, no. 3, 2010.
[94] S. Perry, J. Klemeš, and I. Bulatov, "Integrating waste and renewable energy to reduce the carbon footprint of locally integrated energy sectors," *Energy,* vol. 33, no. 10, pp. 1489-1497, 2008.

[95] P. Varbanov and J. Klemeš, "Analysis and integration of fuel cell combined cycles for development of low-carbon energy technologies," *Energy,* vol. 33, no. 10, pp. 1508-1517, 2008.
[96] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," presented at the Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, 2010. Available: http://proceedings.mlr.press
[97] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556,* 2014.
[98] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *arXiv preprint arXiv:1506.00019,* 2015.
[99] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097-1105.
[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[101] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research,* vol. 13, no. Feb, pp. 281-305, 2012.
[102] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *ieee Computational intelligenCe magazine,* vol. 13, no. 3, pp. 55-75, 2018.
[103] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering,* vol. 22, no. 10, pp. 1345-1359, 2010.
[104] C. C. Aggarwal and H. Wang, "Text mining in social networks," in *Social network data analytics*: Springer, 2011, pp. 353-378.
[105] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717-1724.
[106] W. Pan, E. W. Xiang, N. N. Liu, and Q. Yang, "Transfer Learning in Collaborative Filtering for Sparsity Reduction," in *AAAI*, 2010, vol. 10, pp. 230-235.
[107] D. Cook, K. D. Feuz, and N. C. Krishnan, "Transfer learning for activity recognition: A survey," *Knowledge and information systems,* vol. 36, no. 3, pp. 537-556, 2013.
[108] Y. Sawada, Y. Sato, T. Nakada, K. Ujimoto, and N. Hayashi, "All-transfer learning for deep neural networks and its application to sepsis classification," *arXiv preprint arXiv:1711.04450,* 2017.
[109] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.
[110] G. Litjens *et al.*, "A survey on deep learning in medical image analysis," *Medical image analysis,* vol. 42, pp. 60-88, 2017.
[111] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, 2014, pp. 3320-3328.
[112] SEAI. (2012, 20/12/2015). *Press Release – Full Data from National Smart Meter Trial Published*. Available: http://www.seai.ie/News_Events/Press_Releases/2012/National_Smart_Meter_Trial_Data_Release.pdf
[113] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016.
[114] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon, 2016*, 2016, pp. 1-6: IEEE.

[115] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034,* 2013.
[116] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818-833: Springer.
[117] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806,* 2014.
[118] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5188-5196.
[119] J. L. Long, N. Zhang, and T. Darrell, "Do convnets learn correspondence?," in *Advances in Neural Information Processing Systems*, 2014, pp. 1601-1609.
[120] O. Russakovsky *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision,* vol. 115, no. 3, pp. 211-252, 2015.
[121] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples (2014)," *arXiv preprint arXiv:1412.6572*.
[122] !!! INVALID CITATION !!! {}.
[123] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.
[124] L. Breiman, "Bagging predictors," *Machine learning,* vol. 24, no. 2, pp. 123-140, 1996.
[125] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences,* vol. 55, no. 1, pp. 119-139, 1997.
[126] R. E. Schapire, "The strength of weak learnability," *Machine learning,* vol. 5, no. 2, pp. 197-227, 1990.
[127] C. Adam-Bourdarios, G. Cowan, C. Germain, I. Guyon, B. Kégl, and D. Rousseau, "The Higgs boson machine learning challenge," in *NIPS 2014 Workshop on High-energy Physics and Machine Learning*, 2015, pp. 19-55.
[128] T. K. Ho, "Random decision forests," in *Document analysis and recognition, 1995., proceedings of the third international conference on*, 1995, vol. 1, pp. 278-282: IEEE.
[129] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR),* vol. 31, no. 3, pp. 264-323, 1999.
[130] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science,* vol. 2, no. 2, pp. 165-193, 2015.
[131] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*: Springer, 2011, pp. 1094-1096.
[132] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research,* vol. 3, no. Mar, pp. 1157-1182, 2003.
[133] M. Dash and H. Liu, "Feature selection for classification," *Intelligent data analysis,* vol. 1, no. 3, pp. 131-156, 1997.
[134] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*: Springer, 1982, pp. 267-285.
[135] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.
[136] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, 2008, pp. 160-167: ACM.
[137] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440-1448.
[138] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," *arXiv preprint arXiv:1502.02072,* 2015.

[139] R. Caruna, "Multitask learning: A knowledge-based source of inductive bias," in *Machine Learning: Proceedings of the Tenth International Conference*, 1993, pp. 41-48.

[140] J. Baxter, "A Bayesian/information theoretic model of learning to learn via multiple task sampling," *Machine learning,* vol. 28, no. 1, pp. 7-39, 1997.

[141] L. Duong, T. Cohn, S. Bird, and P. Cook, "Low resource dependency parsing: Cross-lingual parameter sharing in a neural network parser," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, vol. 2, pp. 845-850.

[142] Y. Yang and T. M. Hospedales, "Trace norm regularised deep multi-task learning," *arXiv preprint arXiv:1606.04038,* 2016.

[143] M. Xu, R. Li, and F. Li, "Phase identification with incomplete data," *IEEE Transactions on Smart Grid,* vol. 9, no. 4, pp. 2777-2785, 2018.

Appendix. A

Randomly Selected Consumers

The randomly selected consumers are differentiated by their unique smart meter IDs. The selected consumers are given in Table A-1.

Table A-1 IDs of selected consumers

ID	ID	ID	ID
1002	3600	4755	6272
1060	3617	4785	6281
1507	3661	4825	6312
1530	3686	4835	6321
1548	3718	4891	6383
1657	3908	5053	6433
1664	3923	5078	6460
2029	4027	5218	6591
2041	4052	5254	6661
2099	4059	5385	6678
2404	4120	5396	6720
2424	4241	5443	6751
2532	4244	5499	6770
2562	4284	5573	6811
2635	4300	5799	6854
2684	4311	5835	6969
2749	4358	5843	6978
2926	4365	5848	7040
2956	4373	5951	7225
2968	4469	6026	7251
2988	4477	6048	7289
3281	4493	6107	7365
3293	4575	6120	7396
3495	4640	6145	7436
3589	4644	6214	7437

Performance Improvement by Transfer Learning for LSTM Networks

There are 4 hidden layers (LSTM cells) in the LSTM network. The improvement on MAPE, MAE, RMSE, and NRMSE by transferring different number of hidden layers from consumers with different degrees of similarities are demonstrated in the following figures.

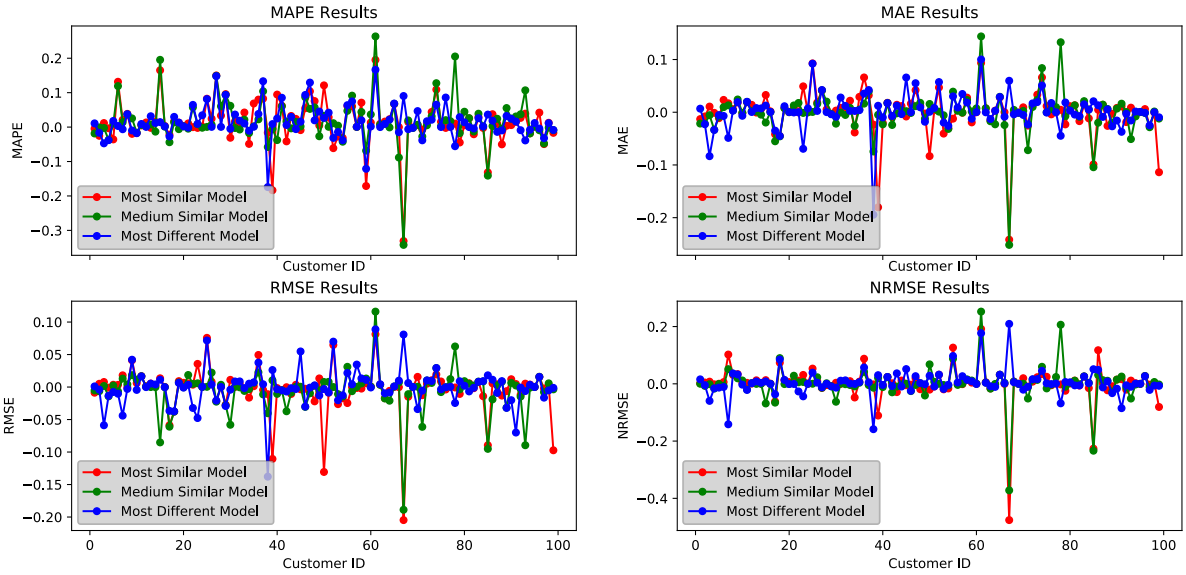

Figure A-1 Performance improved by transferring the first hidden layer in LSTM

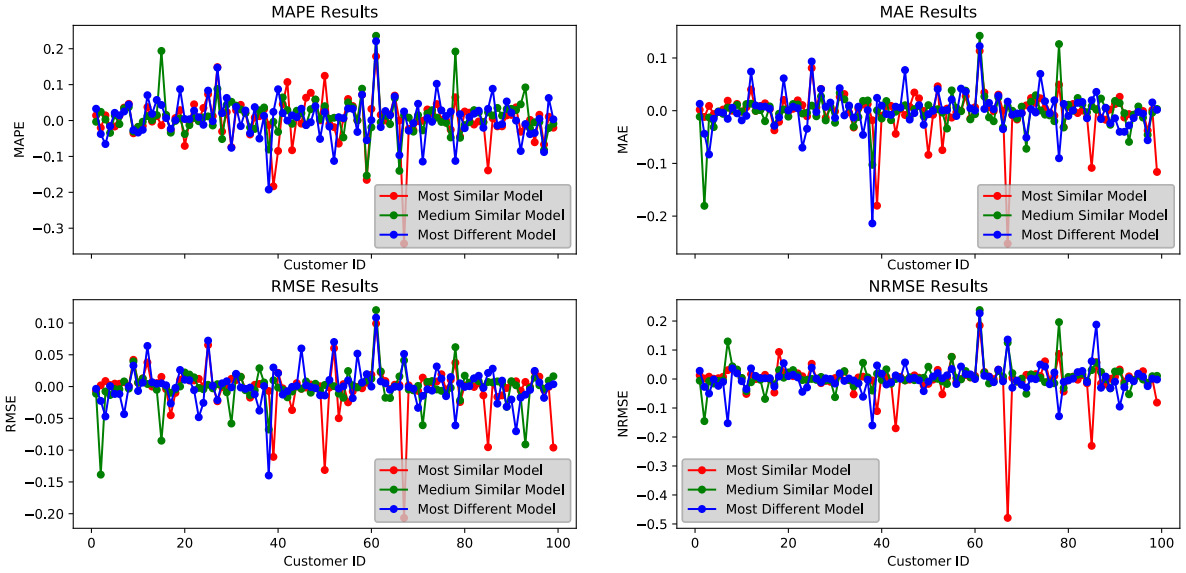

Figure A-2 Performance improved by transferring the first two hidden layers in LSTM

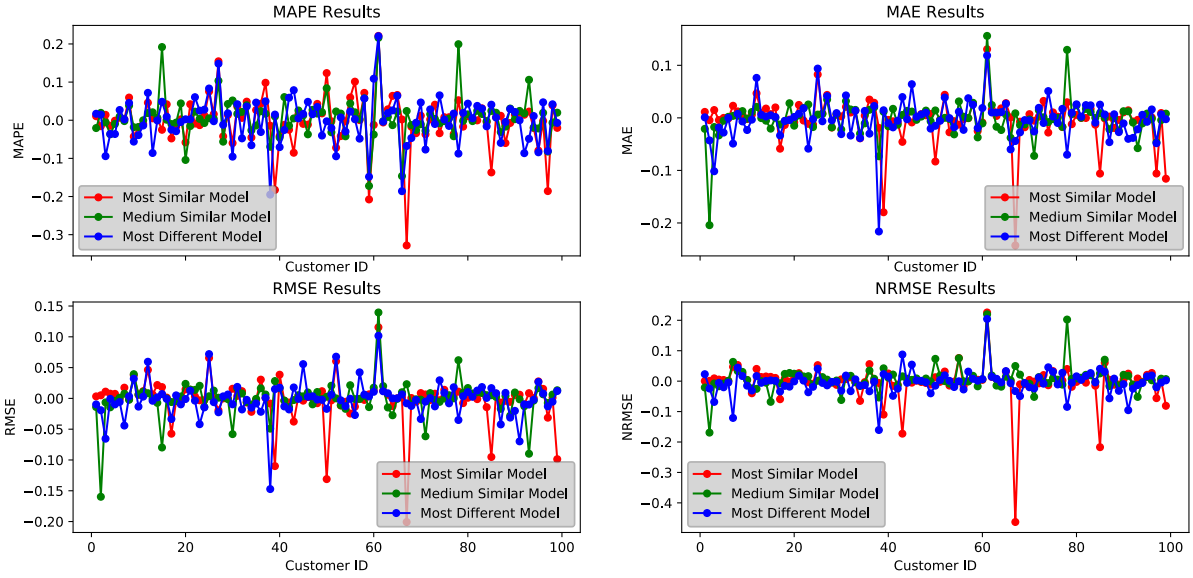

Figure A-3 Performance improved by transferring the first three hidden layers in LSTM

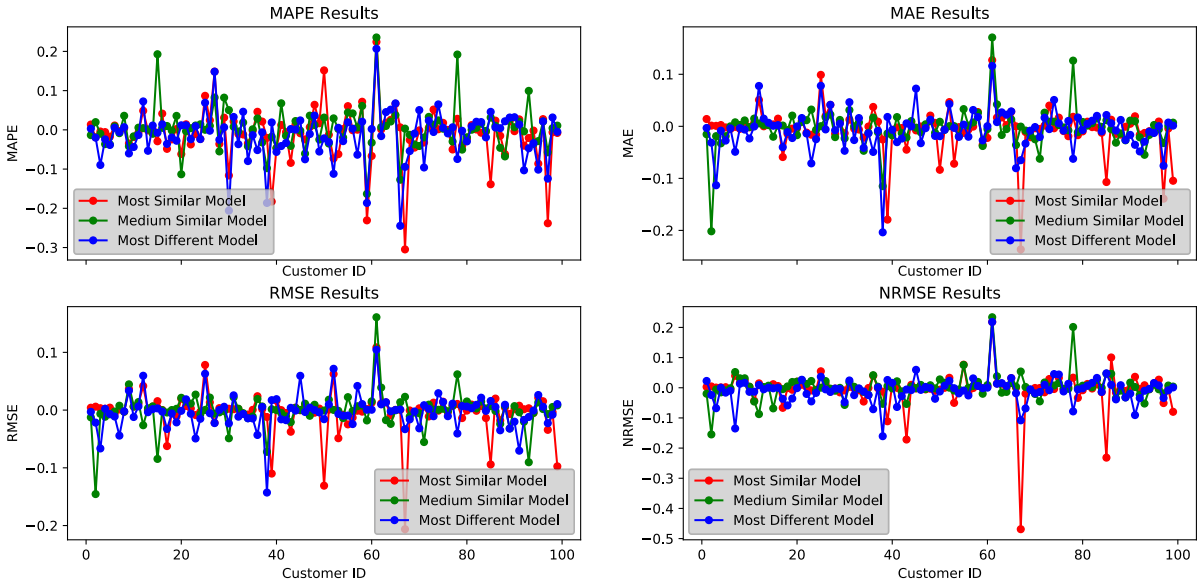

Figure A-4 Performance improved by transferring all four hidden layers in LSTM

Appendix. B

Smart Metering Data Feature Extraction

Features are extracted from consumers' smart metering data. They are extracted across different time horizons. Specifically, the features are extracted from annual data, winter data, spring data, summer data and autumn data. For each season or the whole year, the extraction is performed on finer time horizons, i.e., weekday, weekend and holidays. For each specific time interval, four types of features are extracted whose detailed descriptions are given in Table B-1.

Table B-1 Extracted features of smart metering data

Category	Descriptions
Consumption	Average of the whole day
	Average of daytime (6am-10pm)
	Average of evening (6pm-10pm)
	Average of morning (6am-10am)
	Average of night (1am-5am)
	Average of noon (10am-2pm)
	Average of daily peak demand
	Average of daily valley demand
Ratios	Average of mean over max
	Average of minimum over mean
	Average of minimum over max
	Average of morning over noon
	Average of evening over noon
	Average of noon over daily mean
	Average of night over daily mean
Occurrence/Time	Time when the demand is above mean
	Average peak demand time
Statistics	Standard deviation
	Average of correlation coefficient of current day and previous day

Feature Importance of Selected Social-Economic Questions

The features importance of the last five of the questions in Table 4-1 are plotted below.

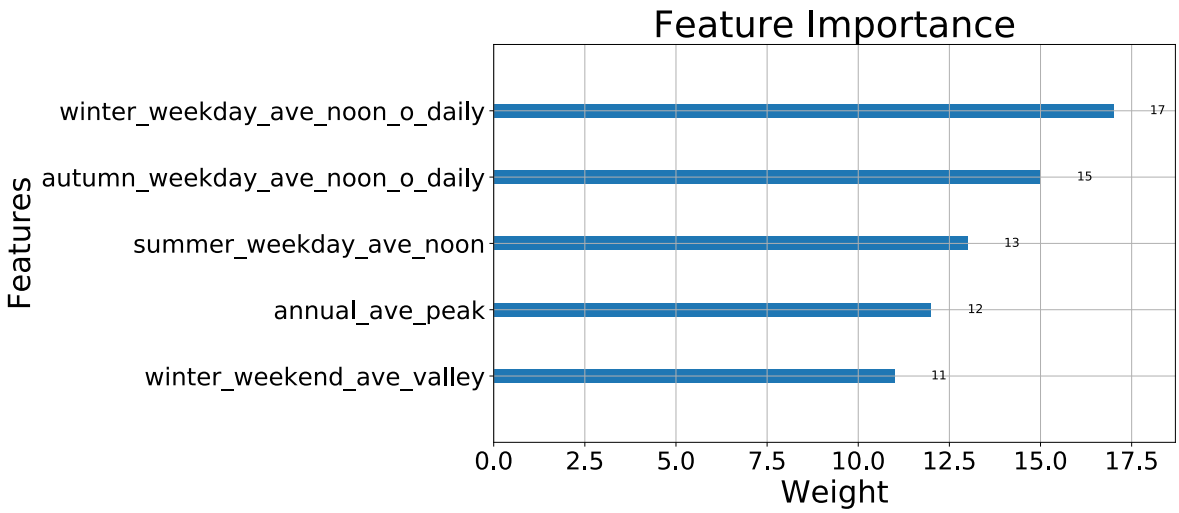

Figure B-1 Feature importance for social-economic question 6

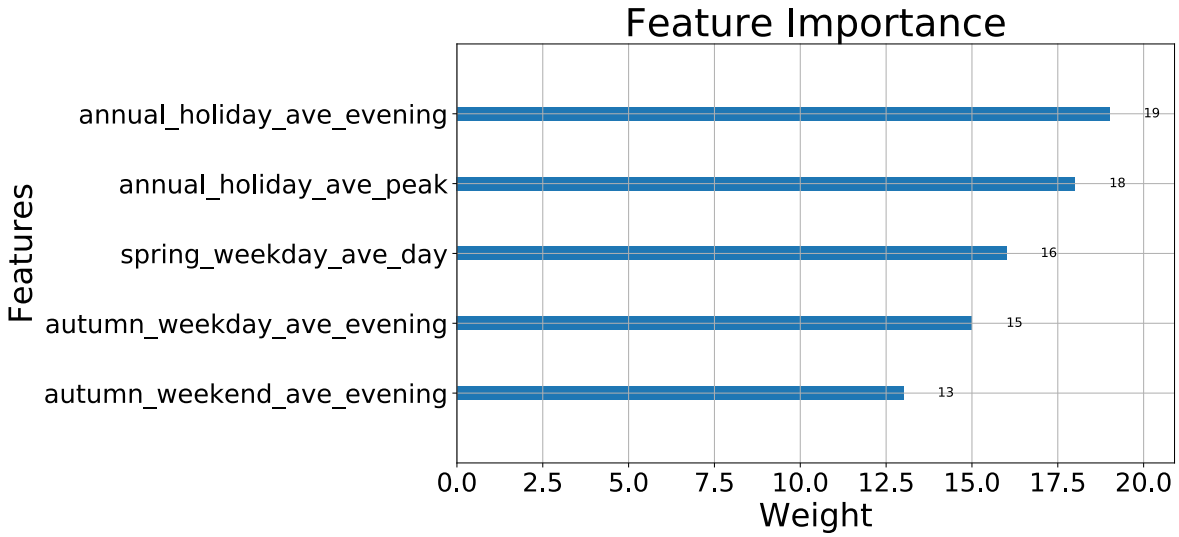

Figure B-2 Feature importance for social-economic question 7

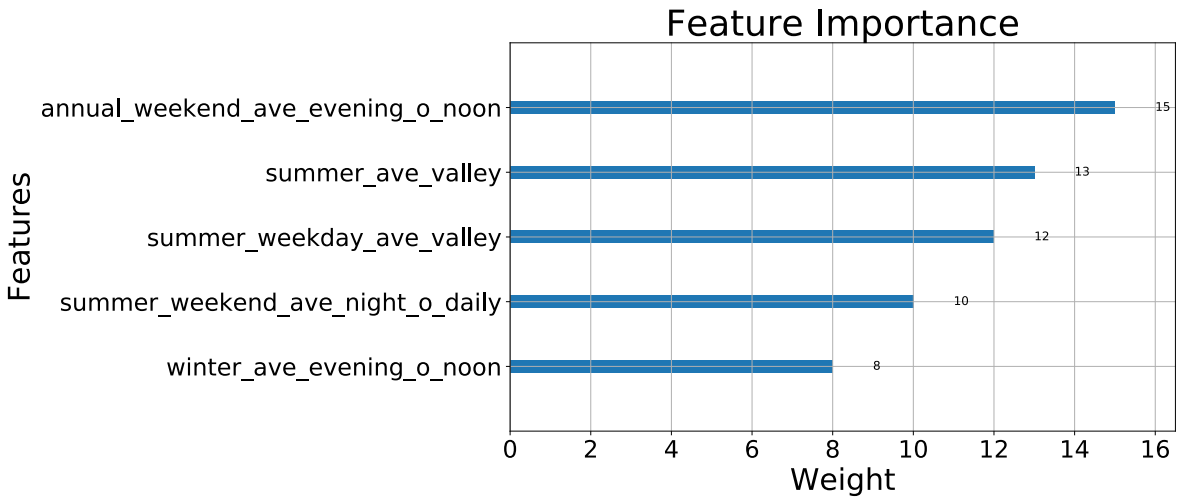

Figure B-3 Feature importance for social-economic question 8

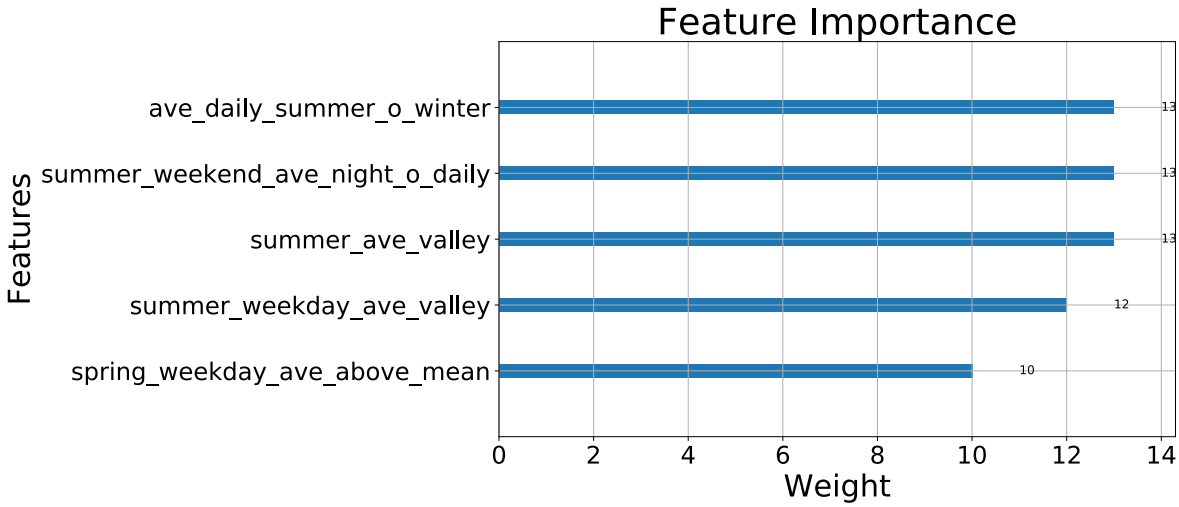

Figure B-4 Feature importance for social-economic question 9

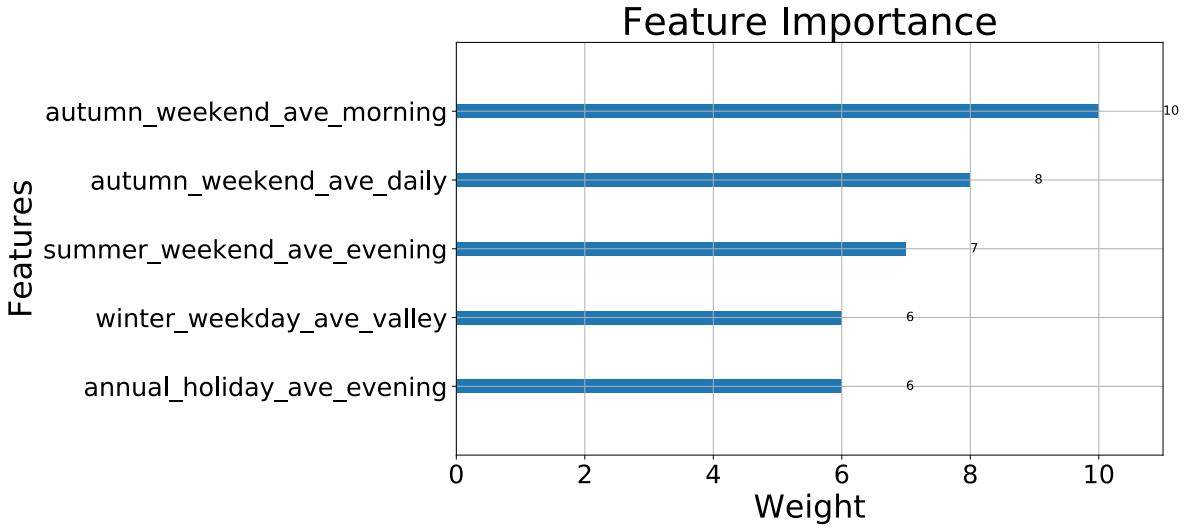

Figure B-5 Feature importance for social-economic question 10