University of Nottingham
UK | CHINA | MALAYSIA

A Novel Dynamic Maximum Demand Reduction Controller of Battery Energy Storage System for Educational Buildings in Malaysia

NG RONG WANG

Thesis submitted to the University of Nottingham for the degree of Doctor of Philosophy

June 2022

Abstract

Maximum Demand (MD) management is essential to help businesses and electricity companies saves on electricity bills and operation cost. Among different MD reduction techniques, demand response with battery energy storage systems (BESS) provides the most flexible peak reduction solution for various markets. One of the major challenges is the optimization of the demand threshold that controls the charging and discharging powers of BESS. To increase its tolerance to day-ahead prediction errors, state-of-art controllers utilize complex prediction models and rigid parameters that are determined from long-term historical data. However, long-term historical data may be unavailable at implementation, and rigid parameters cause them unable to adapt to evolving load patterns. Hence, this research work proposes a novel incremental DB-SOINN-R prediction model and a novel dynamic two-stage MD reduction controller. The incremental learning capability of the novel DB-SOINN-R allows the model to be deployed as soon as possible and improves its prediction accuracy as time progresses. The proposed DB-SOINN-R is compared with five models: feedforward neural network, deep neural network with long-short-term memory, support vector regression, ESOINN, and k-nearest neighbour (kNN) regression. They are tested on day-ahead and one-hour-ahead load predictions using two different datasets. The proposed DB-SOINN-R has the highest prediction accuracy among all models with incremental learning in both datasets. The novel dynamic two-stage maximum demand reduction controller of BESS incorporates one-hour-ahead load profiles to refine the threshold found based on day-ahead load profiles for preventing peak reduction failure, if necessary, with no rigid parameters required. Compared to the conventional fixed threshold, single-stage, and fuzzy controllers, the proposed two-stage controller achieves up to 6.82% and 306.23% higher in average maximum demand reduction and total maximum demand charge savings, respectively, on two different datasets. The proposed controller also achieves a 0% peak demand reduction failure rate in both datasets. The real-world performance of the proposed two-stage MD reduction controller that includes the proposed DB-SOINN-R models is validated in a scaled-down experiment setup. Results show negligible differences of 0.5% in daily PDRP and MAPE between experimental and simulation results. Therefore, it fulfilled the aim of this research work, which is to develop a controller that is easy to implement, requires minimal historical data to begin operation and has a reliable MD reduction performance.

Affirmation

The work presented in this thesis is, to the best of my knowledge, original, and has not been submitted for any other degree. This research was carried out in the Department of Electrical and Electronics Engineering, University of Nottingham Malaysia from October 2018 to December 2021. This thesis does not exceed 100,000 words. Important findings of this research work have been published in journals as follows.

Publications from the research work:

Journal Papers:

- **R. W. Ng**, K. M. Begam, R. K. Rajkumar, Y. W. Wong, and L. W. Chong, "An improved self-organizing incremental neural network model for short-term time-series load prediction," *Appl. Energy*, vol. 292, p. 116912, Jun. 2021, doi: 10.1016/j.apenergy.2021.116912. (Q1, Impact Factor: 9.746)
- **R. W. Ng**, K. M. Begam, R. K. Rajkumar, Y. W. Wong, and L. W. Chong, "A novel dynamic two-stage controller of battery energy storage system for maximum demand reductions," *Energy*, vol. 248, p. 123550, Jun. 2022, doi: 10.1016/j.energy.2022.123550. (Q1, Impact Factor: 7.147)

Other publications:

- L. W. Chong, **R. W. Ng**, Y. W. Wong, and R. K. Rajkumar, "One-Hour Ahead Prediction of Solar Irradiance Using Support Vector Machines," in 6th International Electrical Engineering Congress, 2018

Acknowledgment

First and foremost, I would like to express my sincere gratitude to my supervisors Dr. Mumtaj Begam, Dr. Wong Yee Wan, Dr. Chong Lee Wai, and Dr. Nafizah Khan for the continuous support of my PhD study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. Additionally, I would like to thank Dr Rajprasad Kumar for his encouragement and guidance on my applications of PhD and scholarship.

I gratefully acknowledge the University of Nottingham Malaysia for providing me with scholarship funding, and I thank them for providing various learning, enriching, networking, and professional and personal development experiences.

I would also like to thank all of my friends who supported me in writing and motivated me to strive towards my goal. Finally, I would like to thank my family, my parents and my brother and sister for supporting me spiritually throughout writing this thesis and my life in general.

Table of Contents

List of Figures

List of Tables

Abbreviations

DR	Demand Response
DSM	Demand Side Management
EV	Electric Vehicle
ESS	Energy Storage System
BESS	Battery Energy Storage System
AI	Artificial Intelligence
ANN	Artificial Neural Network
MLP	Multi-Layer Perceptron
FFNN	Feedforward Neural Network
RBF	Radial Basis Function Network
DNN	Deep Neural Network
LSTM	Deep Neural Network with Long-Short Term Memory
SVR	Support Vector Regression
SOINN	Self-organizing Incremental Neural Network
ESOINN	Enhanced Self-organizing Incremental Neural Network
kNN	K-nearest neighbour
kNN-IDW	K-nearest-neighbor with inverse distance weighting regression
DB-SOINN-R	Density-based Self-organizing Incremental Neural Network with kNN-IDW
ACF	Autocorrelation function
PACF	Partial autocorrelation function
ξ	New data
$\mathcal{N}$	Set of all nodes
N_a	The a-th node in $\mathcal{N}$
N_1	The 1st winner
N_2	The 2nd winner
T_a	Threshold of node a
$\mathcal{P}_{N_a}$	Set of nodes connected to node a
WT_a	Winning time of node a
LT_a	Lifetime of node a
$\mathcal{p}_a$	Number of edges for node a
d_a	Node density of node a
D_a	Local density of node a
λ	Denoise interval
age_{max}	Maximum age for edges
C_1	Denoise control parameter for $\mathcal{p}_a = 1$
C_2	Denoise control parameter for $\mathcal{p}_a = 2$
$k_{idw\_}$	Number of k used in IDW
$k_{denoise\_}$	Number of k used in denoising of DB-SOINN
$\mathcal{K}_a$	Set of $k_{denoise}$ nearest neighbor nodes to node a
MPC	Model Predictive Control

MD	monthly maximum demand [W / kW]
P_{Th}	demand threshold [W / kW]
P_D	BESS Discharge Power [W / kW]
P_C	BESS Charge Power [W / kW]
P_{grid}	grid power [W / kW]
P_{load}	load power [W / kW]
SOC	state-of-charge of BESS [%]
$BESS_{size}$	size of BESS [kWh]
E_{used}	BESS energy used in peak demand reduction [kWh]
$E_{projected}$	projected energy usage for the next hour [kWh]
t_{end}	projected peak demand reductions end time [minutes]
$SOC_{projected}$	projected SOC at t_{end} [%]
$DisRate_{projected}$	projected BESS discharge rate [kWh/hr]
y_p^i	i-th point of the interpolated next hour profile [kW]
k	number of 5-min intervals between the first demand reduction of the day and current time t [-]
MDRP	maximum demand reduction percentage [%]
PDRP	daily peak demand reduction percentage [%]
PDR_{FR}	peak demand reduction failure rate [%]
MAPE	mean absolute percentage error [%]
$MD_{recorded}$	current MD of the day [W / kW]
$MD_{original}$	MD before reduction [W / kW]
$MD_{reduced}$	MD after reduction [W / kW]
t	time [minutes]
d	day
$P_{BESS,DC}$	DC power of the BESS-side [W]
$P_{BESS,AC}$	AC power of the BESS-side [W]
$P_{Th,Charge}$	P_{Th} for controlling AC-DC charger [W]
$P_{Th,Discharge}$	P_{Th} for controlling DC-AC inverter [W]
V_{grid}	Grid Voltage [Vrms]
PF	Power Factor
PF_{load}	Load Power Factor
PF_{grid}	Grid Power Factor

Chapter 1 – Introduction

1.1 Overview

This chapter introduces the research work in this thesis. An overview of the thesis is presented. Following this, the problem statements of the research work is given. The aim and objectives of the research are listed. In the subsequent sections, the significant of research is highlighted, followed by the scope of research. The outline of this thesis is provided in the last section.

The electricity consumption in different sectors is increasing yearly due to urbanization and electrification. To cope with the increasing trend, the capacity of installed electricity generation and transmission systems may need to be expanded periodically, or significantly oversize to defer the next expansion period. However, this is not an economic approach. Electricity companies seek other methodologies such as demand-side management or demand response to defer the expansions, thus cutting down investment costs and greenhouse emissions substantially. One of the methods is a time-of-use tariff in the form of maximum demand (MD) charges. The MD charge introduces awareness of peak demand management among consumers. The higher the consumers' participation in peak demand reduction, the higher the effectiveness of the tariff at reducing the peak demand at the distribution level. To reduce the MD charge, consumers apply peak shaving or peak reduction to reduce their peak demand or MD. Although MD charges are typically calculated monthly, it is common to reduce the daily peak demand because estimating the peak demand of a day is always easier than predicting the peak demand of the month and on which day it occurs. Successfully reducing daily peak demand for a month is equivalent to successfully reducing the peak demand of a month. Therefore, this research work studies both daily peak demand reduction and monthly maximum demand controller, with maximum demand controllers prioritized on the amount of monthly MD reduction.

Battery energy storage systems (BESS) is an approach to achieve peak shaving. The BESS stores energy during low demand or off-peak hours and discharge its energy to the grid during high demand or peak hours. When it discharges to the grid, the power goes to the load, so the load draws less power from the utility, thus the utility company sees a lower demand usage. This approach has the highest potential due to its high flexibility and high compatibility with most buildings. Unlike load management

approaches that pre-determines a specific operating schedule for the controllable loads that may switch off some loads to reduce the demand usage, BESS needs no control of the load. Unlike diesel generators where its location must not be near common buildings and renewable energy sources such as photovoltaic where its installation location must be optimal, BESS can be installed virtually anywhere with no frequent maintenance required. One of the major challenges for peak shaving with BESS is the optimization of the demand threshold that controls the charging and discharging powers of BESS. A demand threshold that is too high may result in a very small peak reduction while too low may result in excessive BESS discharge and insufficient energy to reduce all anticipated demands, resulting in little or no peak reduction. There are two types of conventional peak reduction controllers for BESS. The first type is fixed threshold controllers that use a fixed threshold that is predetermined from historical data at the time of implementation. The second type is adaptive threshold controllers that adjusts the threshold in real-time based on the predicted day-ahead load profile.

Conventional fixed threshold controllers are easy to implement as the controller just maintain the power demand at the pre-determined fixed threshold during the peak reduction operations without any threshold adjustments. However, it usually requires a large amount of historical data to predetermine an optimal fixed threshold, and the rigid threshold makes it not able to cope with load profile variation in the future, thus a high chance of peak reduction failure for load profiles that differs considerably from the historical data used to determine the fixed threshold. Conventional adaptive threshold control adjusts the threshold in real-time based on predicted day-ahead load profiles. The goal of conventional adaptive threshold control is to minimize the peak demand by maximizing the BESS utilization for peak reduction. If the day-ahead load profile is under-forecasted, the threshold may be too low for the actual profile, causing unnecessary high BESS discharge. As a result, the BESS may be exhausted before successfully reducing all anticipated demands, resulting in little or no peak demand reduction. Hence, the peak reduction performance of conventional adaptive threshold controllers is highly dependent on the accuracy of day-ahead load profiles.

To tackle this issue of high dependency on accurate day-ahead load profiles for conventional adaptive threshold controller, researchers proposed different state-of-the-art controllers. Some controllers tackle this by improving the prediction accuracy, but cannot eliminate prediction errors completely, so there is still the possibility of poor

accuracy for some days. Additionally, the identification and validation of an appropriate prediction model for different sites and buildings are time-consuming. Alternatively, some controllers minimize peak demand reduction failure rate by considering the potential of inaccurate day-ahead load profiles when adjusting the threshold. Typically, they achieve this with the inclusive of rigid parameters that are predefined from at least a year of historical data at the time of implementation. Rigid parameters cause these systems unable to adapt to evolving profiles where the prediction errors may become larger and render the existing rigid parameters inappropriate, causing low or no peak reduction. Moreover, users may not necessarily have the historical data collected at the time of implementation, and it is unrealistic to wait for a year of data collection before the implementation of the peak reduction controller. There is limited research that considers the problem of lack of historical data at the time of implementation, which is important for actual implementation. For monthly maximum demand reduction, a day of peak reduction failure can cause little to no MD reduction for that month, which defeats the purpose of the maximum demand reduction controller.

This research work aims to develop a maximum demand reduction controller that can be implemented with a little amount of historical data and can be easily implemented for different buildings. Therefore, this research work proposes a dynamic two-stage maximum demand reduction controller that consists of a novel incremental model named DB-SOINN-R for time-series load prediction and a novel two-stage control strategy. The novel controller adjusts the demand threshold in real-time based on two load predictions of different lead times, which are day-ahead and one-hour-ahead. The novel DB-SOINN-R incremental prediction model can be in operation after its initial training with simple data of as low as 30 days, and it can update its model with new data as time progress. Results show it has better prediction accuracies and is faster compared to conventional supervised models. Despite that, it does not eliminate prediction errors, and there may still have days with high prediction errors. The goal is of the DB-SOINN-R is to make sure the prediction model can adapt to future profiles through incremental learning, so the prediction errors are within the correctable range of threshold control.

Hence, there is still a need for the novel two-stage control that can achieve a considerable amount of peak reduction when the day-ahead load prediction has poor accuracy for that day. Unlike other state-of-the-art controllers, the novel two-stage

controller achieves it without relying on rigid parameters that can deteriorate the adaptability of the controller. The novel two-stage controller achieves this by incorporating one-hour-ahead load prediction in its threshold real-time optimization control. The novel two-stage control consists of two stages. The first stage determines the threshold based on the predicted day-ahead load profile to achieve the highest possible peak reduction with the remaining BESS energy. The second stage refines the threshold based on the predicted one-hour-ahead load profile to prevent peak demand reduction failure when necessary. This ensures successful peak demand reduction regardless of the accuracy of day-ahead load profiles. To the author's knowledge, no article utilizes one-hour-ahead for daily peak demand reduction. This research work presents the proposed two-stage control as an example that incorporates one-hour-ahead for daily peak demand reduction.

Combining both the novel DB-SOINN-R prediction model and the novel two-stage control forms the two-stage maximum demand reduction controller proposed in this research work. Both the DB-SOINN-R prediction model and novel two-stage control need no large amount of historical data at the time of implementation. This allows the controller adapts to future profiles automatically and prevent peak demand reduction failure without requiring rigid parameters that need to be determined from long-term historical data at the time of implementation. Therefore, the controller is easy to implement for different buildings, and can potentially boost consumer's demand response participation.

In this research work, the DB-SOINN-R and proposed two-stage control strategy are evaluated in MATLAB. The DB-SOINN-R is compared with six conventional models: feedforward neural network (FFNN), support vector regression (SVR), deep LSTM neural network (LSTM), enhanced self-organizing incremental neural network (ESOINN), and k-nearest neighbour (kNN) regression. The proposed two-stage control strategy is compared with a conventional fixed threshold controller, a conventional adaptive threshold controller, and a state-of-the-art fuzzy controller. The proposed two-stage controller is also tested in a single-phase AC 240V experiment setup with a peak power of 1.8kW and a 48V lead-acid battery system to validate its performance in real-world. The real-world results are also compared with simulation results for validation and identification of potential rectification necessary for real-world implementation.

1.2 Problem Statement

Some of the research issues which may regarded as problem statements are as follows:

- Conventional adaptive threshold controllers are highly dependent on accurate day-ahead load predictions. They may fail to reduce the peak demand if the accuracy of load prediction is too low, and the controller is unable to cope with the low accuracy. Some studies proposed controllers that use day-ahead load prediction with better accuracy. However, it is difficult to produce a consistent prediction accuracy that also works for future profiles, especially for predictions of buildings' load profiles. Studies have shown buildings' load profiles have lower predictability due to the distinct occupant behaviours and activities. Moreover, conventional threshold optimization control lacks the handling of unexpected high prediction errors and result in poor peak demand reduction performance.
- There are state-of-the-art controllers that rely on rigid parameters to prevent peak demand reduction failure when the day-ahead load profile is under-forecasted. The use of rigid parameters may deteriorate the adaptability of the controller, where the existing parameters may not necessarily be suitable for future profiles, thus increasing the possibility of peak reduction failure. Studies have shown that the load variations are evolving due to climate change and increased sector coupling. This makes the adaptability of the controller an important factor for reliable long-term MD reductions.
- Although the incremental learning of ESOINN has the potential to tackle the issue of lack of historical data and is adaptive to future trends, it is originally designed for clustering applications. For prediction tasks, it has three problems that need to be solved. The first problem is the original denoising of ESOINN may incorrectly remove important nodes when the data has an uneven density distribution. The second problem with the original ESOINN is the use of Euclidean distance often causes the selection of incorrect nodes that has a significantly different timestamp or minute of the day, resulting in the predicted profile lagging the actual profile. The third problem is the ESOINN cannot produce a unique output for each input. The predicted output of the original ESOINN is based on the nearest node. Hence, two different inputs may result in the same predicted output.

- To combat adaptability, conventional prediction models and controllers often use a large amount of historical data, which is typically longer than one year, to train the initial model and to pre-define the rigid parameters used for preventing peak reduction failures. The assumption is that the longer the data, the more trends can be captured. However, the user may not necessarily have a large amount of historical data at the time of implementation, and it is impractical to wait for the collection of data. Insufficient historical data can worsen the prediction accuracy and rigid parameters determined from it may be inappropriate for future operations and cause the peak reduction performance to decline significantly. There are limited studies that consider this issue, which is an important factor for actual system implementation.
- The implementation complexity and compatibility are also factor that user should consider. Most studies need their proposed controller to be specifically designed per site. The rigid parameters used by these controllers often need to be determined from collected historical data of the target building and cannot be shared with other buildings directly. To improve load prediction accuracy, studies often suggest the use of complex and difficult-to-acquire input features, such as buildings' internal environment readings, weather information, and occupants' behaviour. However, the correlations of these features may not necessarily be applicable to other buildings. It is time-consuming to identify and evaluate the feasibility of the prediction model for each specific building based on its collectable data. A high complexity system increases the time required before implementation, while users always want to reduce the monthly maximum demand charge as soon as possible.
- Although studies have shown the feasibility of their system through actual implementation, the results presented are often in the period of days. However, long-term monthly results should be presented to validate its consistency for monthly maximum demand reduction. Also, no study considers and studies the effect of using a small amount of historical data at the time of implementation.

1.3 Aims and Objectives

The aim of this research work is to develop an MD reduction controller of BESS that is easy to implement, requires minimal historical data to begin operation, and has a reliable MD reduction. To fulfil the aim of the research work, the following objectives need to be accomplished:

- To design an incremental load prediction model that uses basic input features and performs better or is comparable to conventional models.
- To design a new prediction-based threshold control strategy that use dynamic values and can prevent peak demand reduction failure if the predicted day-ahead load profile has low accuracy.
- To validate the MD reduction performance based on the integration of the incremental load prediction model and two-stage control strategy into a controller.

1.4 Significance of Research

This research work proposes a two-stage controller that consists of a novel incremental unsupervised DB-SOINN-R prediction model and a novel dynamic two-stage control strategy. The significant contributions that have been made in this research work are listed below:

1. Unlike other controllers that rely on improved prediction models or rigid parameters to handle under-forecasted prediction error, the proposed controller needs no rigid parameters that need to be obtained from historical data analysis at the time of implementation. The proposed controller uses two load predictions of different lead times, which are day-ahead and one-hour-ahead, to optimize the threshold in real-time. With the integration of one-hour-ahead load prediction, the controllers adjust the threshold dynamically, not using any pre-determined rigid parameters or constants but only values obtained in real-time. To the author's knowledge, there is no article found using one-hour-ahead load prediction in daily peak demand reduction. This research work demonstrates the usage of one-hour-ahead load prediction in daily peak demand reduction. Results show the proposed controllers have zero peak demand reduction failure up to ten months of operations and outperform the conventional fixed threshold

controller, conventional adaptive threshold controller and a state-of-the-art fuzzy controller.

2. Unlike other load prediction models that require a large amount of training data, the incremental DB-SOINN-R prediction model can be trained with a small amount of data and updates its model daily. Its incremental learning capability allows it to update its model instead of retraining the model, significantly reducing the training time spent on updating the model. The proposed DB-SOINN-R has the highest prediction accuracy compared to other models. Moreover, the DB-SOINN-R is an unsupervised model, so it has both faster prediction time and training time compared to other supervised models.
3. Combining both the incremental DB-SOINN-R model and the dynamic two-stage control, it forms the novel dynamic two-stage maximum demand reduction controller. The controller can begin peak reduction operations with 30 days of historical data only. The DB-SOINN-R uses only load profile data, reducing the complexity of the controller. The dynamic two-stage control strategy automatically optimizes the threshold based on the installed BESS capacity and predicted profiles. To demonstrate the adaptability of the proposed controller, the long-term evaluation (up to ten months of MD reduction) of the controller is evaluated on two different datasets. The settings of the controller remain the same for all tests. The proposed controller is also tested on a different BESS size to demonstrate its flexibility. The controller dynamically finds the optimal threshold based on the installed BESS capacity, so users can allocate BESS capacity based on their affordability. The higher the BESS capacity, the higher the MD reduction. Then, the controller is evaluated for up to ten months of MD reduction on two different datasets, which such long-term evaluation is not commonly found in demand reduction articles.

1.5 Scope of Research

This research work focuses on developments of a new prediction model and a new threshold control strategy that can be implemented with limited historical data at the time of implementation, capable of peak demand reduction under various circumstances, adaptive to evolving profiles, and easy to implement for different target buildings or sites. The optimal sizing of BESS and advanced optimization algorithm are not in the scope of this research work. Instead, this research work focuses on the

development of an easy-to-implement controller that can automatically adjust and adapt to the installed BESS capacity. This allows users to scale the BESS capacity depending on the affordability of the user.

1.6 Thesis Outline

This thesis consists of 7 chapters. Chapter 1 discusses the overview, problem statements, research aim, research objectives, significance of research, and scope of research of this research work.

Chapter 2 presents the background information on demand reduction, different demand reduction methodology, the definition of maximum demand, the conventional demand reduction with BESS, and different load prediction models. It also includes a comprehensive literature review on state-of-the-art peak demand and maximum demand reduction controllers of BESS and state-of-the-art unsupervised self-organizing incremental neural networks. Lastly, the research gaps found based on the literature review are presented.

Chapter 3 presents the structure of maximum demand reduction using BESS. Two load profiles are used in this research work, with one profile collected specifically for this research work while another one is online data. The collection of the data is explained in this chapter. The characteristics of the two load profiles are also discussed in this chapter.

Chapter 4 proposes the first novelty of this research work, which is an improved unsupervised self-organizing incremental neural network, namely DB-SOINN-R. The prediction model is compared with five conventional prediction models. Unlike conventional evaluation methods that have more training data than testing data, this chapter evaluates their incremental prediction by pretraining the models with a small amount of historical data and testing on a long period of testing data, with the models updating incrementally daily. The execution times are also evaluated by measuring the time taken to train the initial model, predict and daily incremental training.

Chapter 5 proposes the second novelty of this research work, which is a novel dynamic two-stage maximum demand reduction controller. This controller uses the proposed DB-SOINN-R in Chapter 4 as the prediction model and incorporates a new two-stage control strategy at finding the optimal threshold that prevents peak demand reduction failure when the accuracy of the day-ahead load profile is inadequate. It is

compared with two conventional controllers and one state-of-the-art fuzzy controller on the two different datasets. To demonstrate the flexibility of the proposed controller, it is also tested on a different BESS size.

Chapter 6 demonstrates the real implementation of the proposed dynamic two-stage maximum demand reduction controller that includes the proposed DB-SOINN-R prediction models. The controller is tested on a single-phase AC 240V setup with a 48V battery bank. The detailed setup of the whole experiment, including the specification of the sensors and equipment used, is presented. The differences between actual implementation and MATLAB simulations are presented. Lastly, it presents the effect of the controller on the grid quality.

Chapter 7 draws the conclusion of the research work and recommends potential future works to further enhancement of the research work.

Chapter 2 – Literature Review

2.1 Overview

The chapter presents the literature review for this research work. Section 2.2 explains peak demand and why peak demand reduction is required. Section 2.3 discusses the different methodologies for peak reduction, specifically demand-side management (DSM) and demand response (DR). Following in Section 2.4 provides the definition of maximum demand, its implication for specific consumers in the form of MD charges, and the calculation of MD charges. Then, the different maximum demand reduction techniques for demand-side are discussed in Section 2.5. A comprehensive literature review on state-of-the-art peak demand reduction controllers of BESS is presented in Section 2.6. One of the crucial components for many peak demand reduction controllers is the load prediction model. Section 2.7 shows different conventional and state-of-the-art time-series load prediction models. Section 2.8 provides a comprehensive literature review on self-organizing incremental neural networks (SOINN), including improved versions of SOINN. Section 2.9 discusses the different optimization algorithms that are suitable for demand threshold optimization. The research gaps found based on the literature reviews are discussed in Section 2.10. Lastly, a summary is provided in Section 2.11.

2.2 Peak demand

Figure 2.1 shows the notably increasing trend in electricity consumption in different sectors in Malaysia due to urbanization and electrification. As the electricity consumption increases, it is safe to assume the peak demand increases as well.

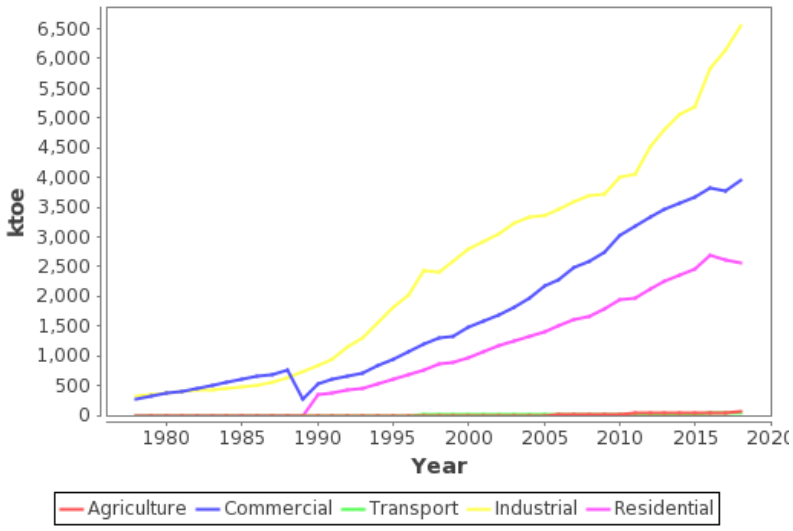


Figure 2.1: Retrieved from [1], it shows the electricity consumption per sector in Malaysia between years 1978 – 2018

Electricity companies need to ensure the energy generated met all consumers' demands with proper sizing of generators and equipment in distribution stations. Since energy is the total amount of power used over a period of time, electricity companies also need to ensure their generators and distributions can supply the required power demand at any moment. For example, a 10W light bulb that operates for ten hours has the same energy consumption of 100Wh as a 100W light bulb that operates for one hour. However, the 100W light bulb draws a peak power of 100W. Hence, the generators need to be able to provide that peak power of 100W for an hour. This is known as peak demand. Hence, electrical companies need to ensure the capacities of generation, transmission, and distribution fulfil both the energy and peak power demands. However, the pure expansion of the existing system is not economical to cope with the increasing trend, in which the costs will be passed to customers through tariffs [2]. Therefore, they seek other methodologies, such as demand-side management or demand response (DR) [3], to defer the expansions, thus cutting down investment costs and greenhouse emissions substantially [4].

2.3 Peak reduction methodology

Peak reduction, which is also referred to as peak shaving [3], is a process of flattening the load profile by shifting workloads during high demands to periods with lower demand. From the perspective of electricity companies, they are looking forward to peak reduction on the demand-side instead of expanding the generation and

distribution capacities. There are two types of demand-side management (DSM) [3]. The first type involves retrofitting existing equipment and devices with more energy-efficient alternatives. For example, replacing conventional fluorescent lights with LED lights and replacing conventional air-conditioners with inverter air-conditioners. However, it is better for long-term energy consumption (kWh) reduction instead of peak demand (kW) reduction. The second type of DSM is demand response (DR), which targets peak demand reductions. Instead of expansion of generation and distribution capacities for short-duration peak demands, DR helps defer these expansions by regulating the peak demands.

There are two types of DR mechanisms: price-based and incentive-based [5]. Price-based DR uses different dynamic pricing mechanisms, which include Time of use (TOU), Real-Time pricing (RTP), Critical Peak Pricing (CPP), Variable Peak pricing (VPP), Extreme Day Pricing (EDP), and Peak Time Rebate (PTR). A simple definition of each pricing is summarized in Table 2.1. In response to the changes in electricity prices, users change their usage patterns, consuming energy during periods with lower tariffs to reduce the electricity bills, thus reducing the peak demand on the grid. The purpose of this DR mechanism is to introduce awareness of peak demand management among consumers. Users will try to reduce peak demand to reduce the operation cost. Incentive-based DR is programs initiated by the utilities, such as direct load control and interruptible load [5], where users allow the electricity companies to alter their load usage in response to electricity prices, incentives, or overall stability [6], [7]. Table 2.2 shows the definition of the different incentive-based DR programs.

Table 2.1: Definition of different pricing-based DR mechanisms

Pricing mechanism	**Definition**
TOU	Different pricing based on time. (Eg. On-peak, off-peak, seasons.)
RTP	Electricity price fluctuates every interval (Eg. 15-min, 60-min) with the changes reported ahead to customers.
CPP	A high rise in pricing for a particular period in case of critical events.
VPP	High electricity rate for a specified period.
EDP	Similar to CPP but applied for the whole day of a 24-hour period.
PTR	In events of high wholesale price anticipation or power system emergency, the customers are offered rebates if their load activities are below a baseline load.

Table 2.2: Definition of different incentive-based DR programs

DR program	**Definition**
Direct load control	Utility controls the customer's loads, and the customer is incentivized.
Interruptible load	Customers agree to meet their demands with their power generation.
Emergency demand response program	Customers are incentivized for curtailing loads during a utility-declared emergency when there is a reserve shortfall.
Capacity market program	In this program, the customers agree to reduce their loads to a pre-specified level at the time of system contingencies. The customers receive guaranteed payments for such obligations and are penalized if load reduction is not observed.
Demand bidding program	The customers—usually large consumers—are allowed to bid for curtailing loads in wholesale and retail electric markets.
Ancillary services market program	Customers can bid for load curtailment in ISO/RTO markets. If a bid is accepted, the customers are incentivized at market price for their obligation to be on standby. When a load reduction is required, these customers are asked for curtailment and are paid according to the market energy price.

Both price-based and incentive-based DR are common in their effect on the consumers' electricity bills. Ultimately, the consumers are penalized for higher peak demand usage with a higher electricity bill while rewarded with a lower electricity bill if they can reduce peak demands successfully.

For DR to thrive at peak reduction and be relevant for electricity companies, it requires customers' participation and supporting infrastructures [5]. The effect of DR and peak reduction seen by electricity companies will be minimal if the participation rate is low. Instead, the companies only receive payment for continuous capacity expansion and potentially electricity rate inflation to cope with the upward growth of electricity usage. Various DR programs, such as the incentive-based DR programs in Table 2.2, are introduced to increase customers' participation. However, not all countries have different DR programs. In Malaysia, there is no incentive-based DR program but relies on the TOU pricing only, in which consumers are charged a maximum demand (MD) charge on top of the typical energy consumption.

2.4 Maximum demand (MD)

On top of the typical charge based on energy consumption of the billing month, the applicable commercial and industrial businesses in Malaysia are also charged based on their maximum demand (MD) of the billing month. Unlike energy consumption, which has a unit of kilowatt-hours (kWh), the unit of MD is kilowatts (kW). As shown in Figure 2.1, these two sectors have significantly higher load consumption compared to residential. They should contribute a significant portion of the peak demand. Moreover, they show a steeper growth rate than other sectors, especially in the past decade, as observed by the increasing difference between the lines.

According to Tenaga National Berhad (the only electricity distribution company in Peninsular Malaysia), the MD is twice the largest energy consumption recorded during any consecutive 30-minutes interval per billing month [8]. Since energy is power multiplied by time, with *E* referred to as the energy consumption of any consecutive 30-min interval, the MD is effectively the conversion between energy and power, as shown in equation (1).

$$E = MD \times 0.5h$$
$$MD = 2E \quad (1)$$

Since MD is twice the largest energy consumption recorded during any consecutive 30-minutes interval, the energy consumption per 30-minutes can be converted to kW using equation (1). The converted energy consumption per 30-minutes is referred to as MD_{block} in this research work. There are 48 MD_{block} per day, and the highest MD_{block} of the month will be the MD for the month. For example, if the highest MD_{block} in June happens at 15:00 – 15:30 of 15th June with a recorded energy consumption of 40kWh for this 30-minute interval, the MD for June would be 80kW.

Other than the doubling energy consumption approach, as shown in equation (1), the MD_{block} can be calculated by averaging the power. Assuming a load profile with an interval of m minute and the power (P_t) remains the same within the m interval, the energy per m minute interval (E_t) can be obtained using equation (2). Then, equation (3) is used to calculate the total energy consumption of the 30-min (E_{total}). Finally, the E_{total} is converted to MD_{block} using equation (4).

$$E_t = P_t \times \frac{m}{60} \tag{2}$$

$$E_{total} = \sum_{t=1}^{\frac{30}{m}} E_t = \frac{m}{60} \sum_{t=1}^{\frac{30}{m}} P_t \tag{3}$$

$$MD_{block} = E_{total} \div \frac{30}{60} = \frac{m}{30} \sum_{t=1}^{\frac{30}{m}} P_t \tag{4}$$

If the load profile has an interval of 1-min (m = 1), the MD_{block} will be the sum of thirty P_t and then divided by 30. If the load profile has an interval of 5-min (m = 5), the MD_{block} will be the sum of six P_t and then divided by 6. In other words, the MD block is the average power during the 30-min interval.

Table 2.3 shows the different rates for the different tariffs by Tenaga National Berhad Malaysia. Tariffs B and D are low voltage and do not have the MD charge. For tariffs C2, E2, and E3, their MD charge only considers within the peak period, which is from 08:00 to 22:00, according to TNB's Tariff book [9].

Table 2.3: MD Rates for Different Tariffs by Tenaga National Berhad Malaysia [10]

Tariff	**Rates (RM/kW)**
B - Low Voltage Commercial Tariff	-
C1 - Medium Voltage General Commercial Tariff	30.30
C2 - Medium Voltage Peak/Off-Peak Commercial Tariff	45.10
D - Low Voltage Industrial Tariff	-
E1 - Medium Voltage General Industrial Tariff	29.60
E2 - Medium Voltage Peak/Off-Peak Industrial Tariff	37.00
E3 - High Voltage Peak/Off-Peak Industrial Tariff	35.50

Maximum demand (MD) charges have always been a concern for commercial and industrial sectors, in which the charges can be as high as 50% of the electricity bill [11]. The high percentage means the customer is not performing peak reduction with most loads operating at the same short period of time, resulting in a higher MD charge when compared to the energy consumption charge. Hence, it urges different MD reduction systems to reduce the MD charges.

Theoretically, MD reduction can be achieved by simply reducing the peak demand of the month, which could be just one or a few days. However, it is very difficult to accurately predict which days have the peak demand of the month. Instead, it would be easier to predict the next day's profile and reduce the peak demand daily. With daily peak demand reduced, the MD of the month will also be reduced. Therefore, the following section discusses not only monthly MD reduction methods but also covers daily peak reduction methods.

2.5 Demand-side peak reduction

Unlike the direct load control under the incentive-based DR programs that operate at the demand-side but are controlled by utility, this section focuses on methodologies that are implemented, controlled, and maintained at the demand-side by consumers. Generally, demand-side peak reduction can be achieved with direct load control and distributed energy resources.

One of the direct load management systems reduces or shifts specific controllable loads. A simple method is turning off pre-allocated and pre-prioritized loads to prevent power demands from exceeding a predetermined demand threshold [12]. Another type of direct load management system turns off certain loads to reduce the power drawn from the grid [13] or shift the operation of controllable loads to a low

demand period [14], [15], with [14] also use a small wind turbine, solar panels, and batteries. The most common controllable and shiftable loads are air conditioning, water heater, and electrical vehicle (EV) charging. Other than load management, the integration of distributed generators can also greatly improve flexibility compared to BESS-only systems. A multi-level demand-side management algorithm is proposed in [16] that achieves peak reduction with a generator and BESS. Stochastic optimization is used in [17] to optimize operations of multiple generators and BESS for operation cost reduction. The integration of BESS can provide more flexibility to load variations [18]. However, the BESS in [7], [14]–[16] is mostly for storing renewable generation surplus, while the peak reductions are mainly achieved by generators and direct load control systems.

One of the limitations of the implementation of direct load control systems is the incapability and unwillingness of users to shift the operation of loads to a different time. Another problem is the requirement of sophisticated planning with detailed knowledge of loads, such as the specification, priority, operation time, expected usage, and more. With these data, users can decide which loads can be shifted to a different time for peak demand reduction without significantly disturbing the occupants' comfort and daily operations. The shifting of loads' operation time decreases operational flexibility since the usage of the loads is predetermined based on the schedule. Although different topologies have been proposed to regain schedule flexibility [19]–[21], they may not satisfy certain businesses and users.

Generation units such as diesel generators, renewable energy sources, and energy storage systems (ESS), such as battery energy storage systems (BESS), are categorized under distributed energy resources [22]. Peak reduction with on-site generators supplies extra power to the load if necessary. It reduces the power drawn from the grid, and the electricity company sees a smaller load demand. Diesel generators are typically used [23] as the capital cost of these small generators is low, but the operating and maintenance costs are high [3]. Moreover, they have high carbon emissions [24]. Many countries have pledged towards net-zero emissions by 2050, in which Malaysia has also increased its commitment to carbon reduction [25]. The installation of diesel generators may require a permit or license from the government due to their noise and carbon emissions, as stated in the Environmental Quality Act

1974 under the laws of Malaysia [26]. The potential of the imposition of carbon tax [27] is also further reducing the interest of generators among consumers.

Intermittent renewable energy sources alone, such as photovoltaic and wind have limited potential for peak reduction due to their non-dispatchable output power [28] that is unpredictable in nature [29]. They are recommended to pair with BESS for reliable peak reduction [4]. Peak demand reduction can also be achieved using BESS alone, and it is the most flexible method since it is compatible with the implementation in most buildings and sites. With BESS, it does not need to alter the operation of loads, thus not affecting daily operations and occupants' comfort. One of the key challenges for peak reduction using BESS is finding the optimal demand threshold that determines the charging and discharging powers of BESS [3]. The BESS recharges from the grid when the load demand is lower than the threshold and discharges energy to the load when the load demand is higher than the threshold to reduce power drawn from the utility grid.

Another challenge for peak shaving with BESS is finding the optimal BESS size [3]. The size of the BESS determines the amount of peak reduction. The higher the BESS capacity, the higher the possible peak reduction. A demand threshold that is too low can cause the BESS to be exhausted before reducing all anticipated demands. Once the BESS is exhausted, the peak demand reduction is halted until the BESS is recharged. As a result, the day may have low or no peak demand reduction. The BESS size is typically fixed after implementation, and the BESS size may vary depending on users' affordability and expected peak demand reduction. On the other hand, the demand threshold can be adjusted by the controller in real-time, depending on the type of controller. EVs parked in the parking lot can also be utilized for peak reduction [30]. The EVs replace a dedicated BESS system for peak shaving. It utilizes vehicle-to-grid technology to allow bidirectional power flows between the EVs and the grid. In such a setup, the available BESS capacity varies depending on the number of EVs in the parking lot and the SOC of EVs. The available BESS energy varies over time and is different every day. The management system not only needs to use the EVs for peak reduction but also needs to ensure the EVs are charged when the EVs are leaving. Another alternative to conventional BESS is using hydrogen fuel cells paired with a solar-wind hybrid hydrogen generator [31]. Although it provides better efficiency and higher energy density, it requires more complex controls and maintenance of the

generator. This research work focuses on the development of a controller that performs real-time threshold adjustment automatically based on the installed BESS size.

2.6 Peak reduction controllers with BESS

There are two types of conventional controllers for peak demand reduction. The first type is the fixed threshold controller, which uses a fixed threshold that must be predetermined from historical load profiles before implementation [32], [33]. The fixed controller is simple and easy to implement, but its threshold does not adjust in real-time, so it cannot cope with daily variations in load patterns [32]. Unexpected high demand may result in peak demand reduction failure for fixed threshold controllers, as shown in Figure 2.2(a), where the BESS is emptied before reducing all anticipated demands. The blue-shaded region is the energy supplied by the ESS. Since the fixed threshold controller uses a fixed threshold that remains unchanged, the ESS discharges completely at time *t*, and demand after *t* cannot be further reduced, so the grid and load powers are the same.

The second type of conventional controller utilizes day-ahead predicted load profiles and linear optimization or model predictive control (MPC) to adjust the threshold in real-time [34]–[38]. This second type of MPC controller is referred to as single-stage control in the remaining of this thesis. For maximizing the BESS utilization for peak demand reduction, the threshold is optimal when the remaining BESS energy matches the total anticipated BEES discharge energy for the rest of the day based on the predicted day-ahead load profile. MPC has been used for peak demand reduction in a university building in Malaysia [35], Korea [34], and an office building with a PV microgrid [37]. Although MPC shows a higher peak reduction than fixed threshold control [38], it needs a relatively good day-ahead load prediction accuracy. If the day-ahead load profile is significantly under-forecasted, peak demand reduction failure can occur [37], as shown in the example in Figure 2.2(b). Although the controller did have a consistent threshold increment over time to compensate for the day-ahead prediction error, it underestimates the under-forecast error, and the threshold increment is insufficient to compensate for the errors. As a result, the BESS is emptied at time *t* with no demand reduced after *t*, indicating peak demand reduction failure. Therefore, peak demand reduction failure can occur if the error exceeds the prediction error tolerance of the conventional single-stage controller.

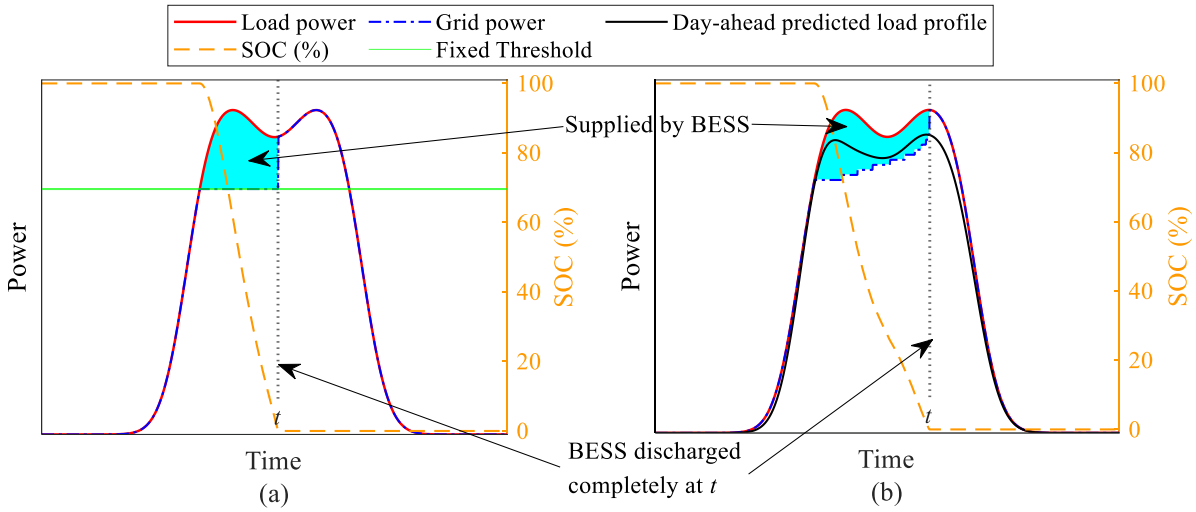


Figure 2.2: An example of peak demand reduction failure for (a) fixed threshold and (b) single-stage controllers

It is inevitable for a prediction model to have prediction error and fail to predict some peak demands [39] and some days of inaccurate day-ahead load profiles throughout a period, especially for buildings that have low predictability due to their distinct occupant activities and facilities [40]. A day with peak demand reduction failure may contribute to no MD charge reduction for the month [41]. Hence, it is important to prevent peak demand reduction failure for reducing the monthly MD. Table 2.4 lists the state-of-the-art peak demand reduction controllers using BESS. Different approaches are applied to prevent peak demand reduction failure that is caused by inaccurate day-ahead load profiles.

One of the approaches is using an improved prediction model such as the ensemble prediction method [42], complex-valued neural network [43], probabilistic forecast using scenario tree [44], and improved regression approach [45]. There are also controllers that incorporate predicted profile manipulations for reducing the prediction errors, such as linear regression using historical prediction errors [46], interpolating the next hour data of the day-ahead predicted profile [47], and applying an offset to the day-ahead predicted profile [4], [48]. Despite the accuracy improvement, it does not guarantee zero uncorrectable error, especially for buildings. Compared to aggregated load profiles of multiple buildings or houses at distribution or generation stations, individual buildings' load profiles have less predictability [49] due to the distinct occupant activities and facilities [40]. Additionally, the identification and validation of an appropriate prediction model are time-consuming.

Alternatively, peak demand reduction failure rate can be minimized by adopting controllers that consider potentially inaccurate day-ahead load profiles when adjusting the threshold. Y. Hida et. al. [48] adds a margin to the optimized threshold. It is simple to implement but difficult to find one margin that fits all scenarios. Instead of demand thresholds, K. H. Chua et. al. [36] presents a novel fuzzy controller for peak demand reduction. Instead of demand thresholds, it determines the discharge power for the next interval based on the time of the day and SOC. The fuzzy rules are optimized daily at the beginning of each day using the predicted day-ahead load profile. However, its membership functions need to be specifically customized using historical data to suit the load characteristics of each building. It includes the predetermination of the range for the discharge power, time of operation, and SOC for the membership function. These membership functions are rigid and are obtained from historical data. The fuzzy controller is tested on two educational buildings in Malaysia. However, the methodology of the prediction model and the accuracy of predicted day-ahead load profiles are not showed or discussed. Moreover, the results demonstrate that the conventional single-stage controller performs better in three of the four cases in the second building. There are only results for four days of peak reduction for two buildings. The SOC of the battery is also not shown in the results. Load prediction and SOC are important to determine whether a controller misbehaves by adjusting the threshold unintentionally.

L. C. Hau et. al. [4] presents a self-adjusting controller for MD reduction for an educational building in Malaysia. It consists of anticipated action that adds a negative 20% offset to the predicted day-ahead load profile and preparatory actions that reserve 30% of the battery energy for demand reduction after 13:00. The offset should be determined through the historical net demand analysis of the building and is constrained to the range of -20% to 15%. Both the 30% reserve amount and 13:00 time used in the preparatory action are rigid parameters that are user-defined from historical demand analysis. The amount of historical data used to predetermine these rigid parameters is not specified, but it mentioned the usage of data from the previous year in its next-day load profile prediction. Hence, it should have at least one year of historical data to predetermine the rigid parameters at the time of implementation.

Table 2.4: The state-of-the-art peak demand reduction controllers using BESS

Ref	Description	Limitation
[44]	• Using scenario tree to determine the different possible scenarios.	• There are days with peak demand reduction failure.
[42]	• Using complex combination forecast method that is a weighted combination of five least-squares linear regression models. • Using linear optimization to adjust the threshold.	• Requires accurate day-ahead load profile. • Using long one-year data to train the prediction model. • Does not present accuracy of load forecast.
[43]	• Using day-ahead load forecast for peak reduction and 20mins-ahead for load smoothing. • Using complex-valued neural networks for load forecasts.	• Requires accurate day-ahead load profile. • Tested on one day of profile only.
[35]	• Using scenario tree to determine the different possible scenarios.	• There are days with peak demand reduction failure.
[38]	• Using linear optimization to adjust the threshold.	• Requires accurate day-ahead load profile. • Tested on one day of profile only.
[36]	• Using fuzzy control to determine the BESS discharge power.	• Does not present day-ahead load prediction method and accuracy. • Tested on five days of profile only.
[41]	• Using an MPC-based nonlinear optimization model to minimize the daily electricity usage cost while regulating the peak. • Peak regulation with rule-based control to handle day-ahead load forecast uncertainty.	• Needs to predefine a fixed threshold from one year of historical data. • No real-time threshold adjustment. • Uses a simple rule-based for peak reduction if violated the pre-defined peak limit or threshold.
[33]	• Fixed threshold controller	• Needs to predefine a fixed threshold. • Tested on one day of similar profile only.
[50]	• A decision-tree based algorithm to mitigate peak demand complications in an islanded microgrid.	• Not reducing peak gird power but reducing peak generator power for optimal cost-benefits.
[4]	• A spontaneous self-adjusting controller that employs MPC and dynamic programming for monthly maximum demand reduction.	• Needs to predefine multiple rigid parameters (minimum threshold, offset, reserve amount, reserve time) from historical data analysis.

2.7 Short-term Load Prediction

Load predictions can be categorized based on their predict horizon. Predict horizons of 30-min ahead and up to the next two weeks are considered short-term [23]. This research work uses one-hour-ahead and day-ahead load predictions. Therefore, the literature focuses on short-term load prediction and pays extra attention to articles discussing the two abovementioned predict horizons.

Building load prediction is one of the crucial components in many different tasks, such as energy management and optimization. Compared to aggregated load profiles at distribution or generation stations, individual buildings have significantly different load patterns [51]. The approaches for load prediction can be categorized into three: white-box or statistical, black-box or data-driven, and grey-box or ensemble [52]. White-box approach performs prediction based on equations and known relations. There are software programs designed to predict the load profile based on the inputs entered by the user. The more detailed the inputs are, the better the accuracy. Black-box approaches are data-driven. Historical data is used to construct a model that maps the relationship between input and output. It is the simplest method since it requires only historical data to train the model.

The most common prediction models are artificial neural network (ANN), which includes feedforward neural network (FFNN) [53], reinforcement neural network [54], generalized neural network [55], wavelet neural network [56], regression neural network [57], recurrent neural network [58], [59], convolutional neural network [60], and long-short-term-memory (LSTM) [61], [62], with recurrent neural network and LSTM only applicable to deep neural networks. In general, they can be classified into two groups based on their layer depth. Networks with a three layers configuration (input-hidden-output) are known as shallow networks, while networks with more than one hidden layer are known as deep neural networks (DNN). LSTM is a type of recurrent neural network that is specifically designed for learning time-series data and long-term dependencies [52]. It has memory nodes that feedback on the outputs from the previous timestep for new prediction output.

There are also regression-based models, such as autoregressive integrated moving average (ARIMA) and support vector regression (SVR). SVR is a regression model of support vector machines and has been studied [63]–[66]. An advantage of

SVR is its capability to find global solutions [63] since conventional ANN networks such as FFNN often suffer from overfitting and the possibility of being stuck in a local minima solution. K-nearest-neighbor (kNN) regression [67], [68] that works by averaging the target data of the *k* number of nearest neighbors are also used in prediction. Its improved version uses inverse distance weighting (IDW) instead of simple averaging [69] and is known as kNN-IDW regression in this thesis. Training data are used directly for searching the *k* number of nearest neighbors. The disadvantage of kNN regression is the storage and handling of a large amount of data as more historical data is collected and its incapability to deal with noise data in the collected historical data.

Other than the single prediction model, studies have also proposed ensemble models that fuse the outputs of multiple different prediction models [67]. The downside of such ensemble models is the increased complexity and difficulty. For example, the parameters of each model need to be optimized separately. Ensemble models based on decision trees are gaining popularity, such as gradient boosting (GB) and random forest (RF) [52], [67], [68], [70]. These tree-based models have fewer hyperparameters to optimize but require more computing resources to train the trees, especially with high numbers of trees.

Most research works aimed to improve prediction accuracy and did not consider practicality, such as implementation and long-term performance. Data collection is important for load prediction, both before and after implementation. Although there is increasing usage of smart meters and internet-of-things (IoT) sensors, it does not apply to all countries and existing buildings. Some of the inputs are difficult to measure. This paper [71] uses an IoT-based big data platform for the day-ahead prediction of building heating and cooling demands. K-means clustering is used to cluster the data then each cluster is trained with its own ANN model. Various types of readings are measured, which include (i) inside building surface temperatures, (ii) outside building surface temperatures, (iii) indoor air temperature, (iv) recorded schedules of occupants, equipment, and lighting, (v) recorded weather profile, and (vi) historical building heating and cooling demands. It studied the correlation coefficient between the various types of data, and the measure of correlation used in the article is shown in Figure 2.3. Results indicate that the temperature of inside building surfaces, the temperature of outside building surfaces, indoor air temperature, outdoor dry air bulb temperature, and global radiation have weak or no relation with the heating and cooling demands. The

operating schedules of lightning and equipment have weak or general relation with the heating and cooling demands. Generally, the schedule of occupants has high relation with the cooling and heating demand. Since the cooling and heating demand often holds up to 40% of the buildings' demand [72], the building's cooling and heating demand should be correlated to the building's total demand.

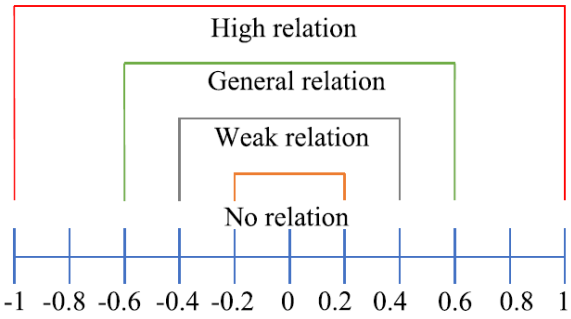


Figure 2.3: Measure of Correlation [73]

However, the building studied in this thesis is not equipped with these sensors. It is difficult to incorporate all the necessary sensors into existing buildings in terms of execution and cost. The collection, storage, and processing of data are complex processes. The difficulty of implementing sensors for existing buildings has been studied in [74]. Instead, only historical load demands are used since it is easy to obtain, and only the installation of one sensor at one location is required.

Another concern of implementation is the lack of historical data. Every model needs to be pre-trained with a certain amount of historical data, but the historical data is not always available when consumers may want the prediction model to be deployed as soon as possible. One of the potential solutions to this problem is transfer learning [75], [76]. It sources a trained model from a different site, with weights tunable for better prediction accuracy at the target site [77]. The problem with transfer learning is the availability of the model. J. Moon et. al. proposed a transfer learning-based model to solve the cold-start problem, which is the lack of historical data [76]. Its predicted output is the combination of outputs of two models. The first transfer learning model is trained using profiles of the most similar buildings in 15 different source buildings. The second time-based model is trained using data from the target building. As given in the results of the study, there is no source building data that is similar to the target building. The transfer-learning model has a much worse MAPE than the time-based model, resulting in worse accuracy after combination. Hence, profiles of a source building that

has a similar pattern to the target building are recommended, but such profiles cannot be easily acquired.

2.8 Self-Organizing Incremental Neural Network

Another potential solution to the lack of historical data is incremental learning. It updates the trained model with newly collected data, thus adapting to any new trend in the profile. However, incremental learning is rarely studied, which is probably due to the assumption of the trained model can handle new trends and the lack of algorithms that has native incremental learning. Native incremental learning models update their trained model with newly collected data only, instead of retraining a new model with all collected data, including data used to train the previous model. Hence, the self-organizing incremental neural network (SOINN) is an appealing option as it is designed for life-long learning through incremental learning. Unlike the previously mentioned supervised learning models, SOINN is an unsupervised learning model. The advantages of SOINN are as follows: 1) it does not require a pre-defined network architecture, 2) it learns faster than supervised learning, and 3) its incremental learning capability allows adaptation to evolving patterns. There are many different versions of SOINN since the original version was published by S. Furao et. al. [78] in 2006. In 2007, an enhanced SOINN (ESOINN) was proposed by S. Furao et. al. [79], that improves the usability and stability of SOINN. It replaces the two-layer of SOINN with a single-layer network and requires fewer hyperparameters. It also has a subclass update process that better separates classes that overlap at high-density regions. In 2008, S. Furao et. al. [80] proposed the adjusted SOINN (A-SOINN), which is part of its adjusted SOINN classifier that consists of k-mean-clustering, noise-reduction, and center-cleaning. The A-SOINN is almost identical to ESOINN but without the subclass update process and a simple denoising stage. The denoising of A-SOINN is simplified because it is proposed alongside external denoising after the learning process, forming the adjusted SOINN classifier. In 2010, S. Furao et. al. [81] proposed a semi-supervised version of SOINN (S-SOINN) with a simplified subclass update process and added the capability to learn unlabeled and labeled data. In 2014, H. Zhang et. al. [82] proposed a load-balancing SOINN (LB-SOINN) with improvements over ESOINN. It incorporates a load balancing process after the denoising process, a new subclass update process that is better at separating high-density overlaps than ESOINN, and a new distance metric to tackle the unsuitability of Euclidean distance with high-dimension data. The new

distance metric is a combination of Euclidean and cosine distances that works by having the Euclidean distance as the majority with low-dimension data and the cosine distance as the majority with high-dimension data. Y. Nakamura et. al. [83] proposed kernel density estimation SOINN (KDE-SOINN) in 2017, which is an improvement of A-SOINN. It incorporates a new condition at the competition stage that determines whether the new data should be inserted as a new node or merged with the existing node. Instead of the typical hypersphere threshold region, the threshold region is defined using the Mahalanobis distance with local network covariance matrices. The new threshold region makes the nodes less likely to merge, thus less edge connection. For compensation, the denoising process removes only nodes with no edge instead of nodes with no or one edge, and new edges are created based on k-NN graph after the denoising process. C. Wiwatcharakoses et. al. [84] proposes SOINN+ in 2020 that differs at the creation and deletion of edges, and the deletion of nodes, to achieve a more graceful forgetting for the incremental learning of SOINN.

The abovementioned studies use SOINN for clustering, but it can also be applied for time-series prediction. T. Avdeenko et. al. [85] used ESOINN to predict taxi service pricing. It achieved good accuracy and fast speed. K. W. Kow et. al. [86] proposed memory SOINN (M-SOINN) to predict output powers for power fluctuation event detection. The M-SOINN is a simplified version of SOINN. It only finds the first winner and does not form edge connections, and the denoising process is removed. Instead of age for edges, it counts the age for nodes, and the nodes are removed if their age exceeds a user-defined threshold. B. K. Puah et. al. [87] proposes a regression-enhanced SOINN (RE-SOINN) for one-hour-ahead solar irradiance time-series prediction. Since the original ESOINN only outputs discrete output, it adds a regression method to the ESOINN to obtain continuous outputs. Table 2.5 summarized the different SOINN models proposed over the years, with their highlights and application.

Table 2.5: Summary of different SOINN models

Model	Year	Highlight	Application	Source
SOINN	2006	• Incremental learning.	Clustering	[78]
ESOINN	2007	• Simpler structure, fewer hyperparameters, and more stable than SOINN. • New subclass update to better separation of high-density overlapping classes.	Clustering	[79]
A-SOINN	2008	• Simpler subclass update and denoising than SOINN.	Clustering	[80]
S-SOINN	2010	• A semi-supervised version of SOINN that can learn both unlabelled and labelled data.	Clustering	[81]
LB-SOINN	2014	• Introduce load balancing and a new subclass update for better separation of high-density overlapping classes. • Use a new distance metric to cater for both high and low dimensions data.	Clustering	[82]
KDE-SOINN	2017	• Use a new threshold region to reduce the merging of low similarity data.	Clustering	[83]
M-SOINN	2018	• A simplified version of SOINN. • Removed the subclass update and denoising	Output power time-series prediction	[85]
SOINN+	2020	• Different creation, deletion of edges, and the deletion of nodes to achieve a more graceful forgetting for the incremental learning.	Clustering	[84]
RE-SOINN	2020	• Uses the original ESOINN and added regression to the output to obtain continuous outputs.	Solar Irradiance time-series prediction.	[86]

2.8.1 Enhanced Self-Organizing Incremental Neural Network (ESOINN)

The ESOINN is a single-layer network with nodes that are self-expandable during training and edges that connect nodes. Each edge is associated with age. ESOINN has four parameters: max age for edge (age_{max}), denoise interval (λ), and two denoise controls (C_1 and C_2). Figure 2.4 illustrates the flowchart for the ESOINN, and Algorithm 3 in Appendix 1 shows the pseudocode for the ESOINN. Table 2.6 shows the definitions of variables used in this section.

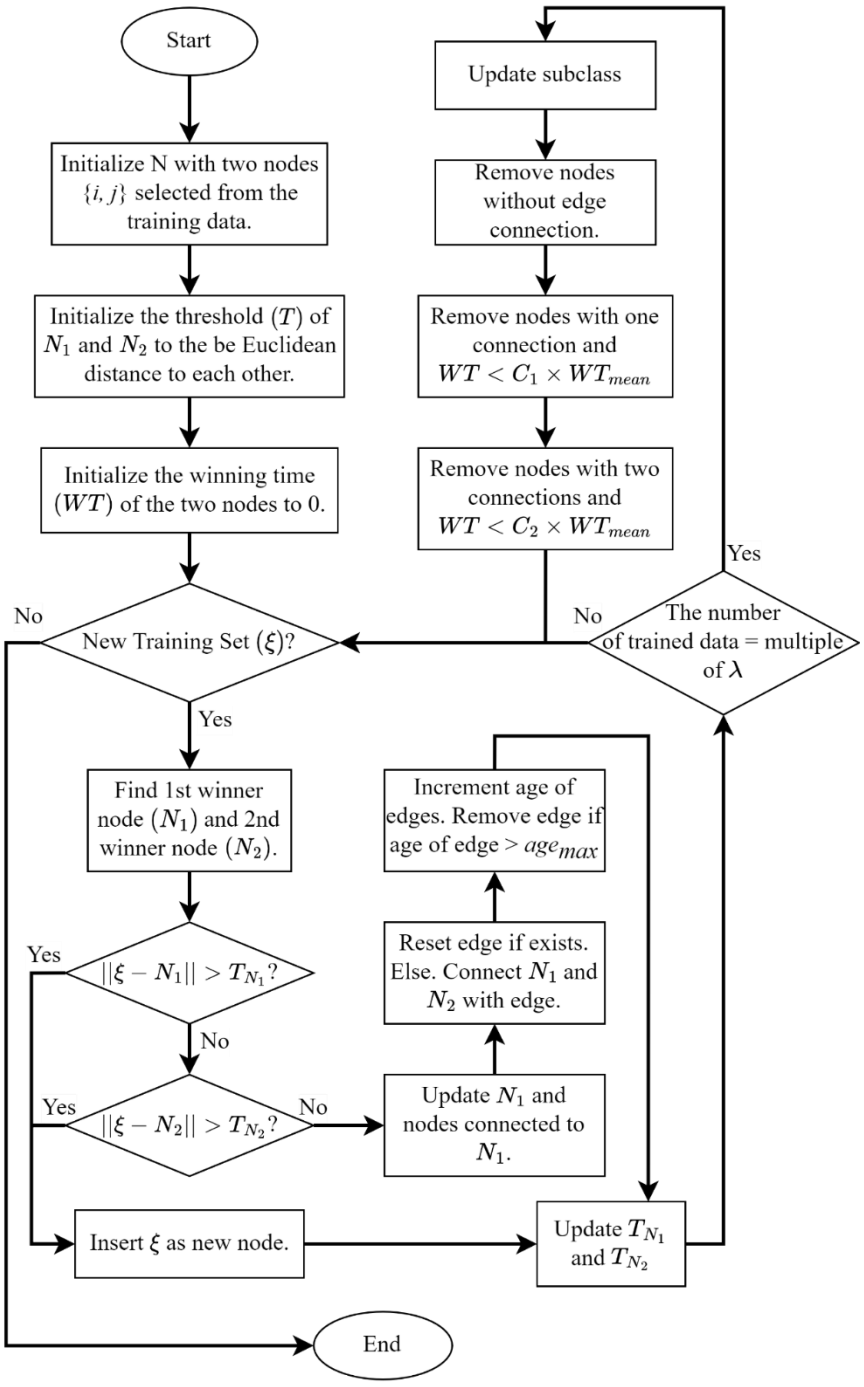


Figure 2.4: Flowchart of the ESOINN Algorithm

Table 2.6: Definitions of variables for ESOINN and DB-SOINN

Variable	**Definition**
ξ	New data
$\mathcal{N}$	Set of all nodes
N_a	The a-th node in $\mathcal{N}$
N_1	The 1st winner
N_2	The 2nd winner
T_a	Threshold of node a
$\mathcal{P}_{N_a}$	Set of nodes connected to node a
WT_a	Winning time of node a
LT_a	Lifetime of node a
$\wp_a$	Number of edges for node a
d_a	Node density of node a
D_a	Local density of node a
λ	Denoise interval
age_{max}	Maximum age for edges
C_1	Denoise control parameter for $\wp_a = 1$
C_2	Denoise control parameter for $\wp_a = 2$
$k_{idw\_}$	Number of k used in IDW
$k_{denoise\_}$	Number of k used in denoising of DB-SOINN
$\mathcal{K}_a$	Set of $k_{denoise}$ nearest neighbor nodes to node a

The ESOINN is first initialized with two training data randomly selected from the training data, with other necessary variables such as threshold (T) and winning time (WT). For each new training dataset (ξ), the first winner (N_1) and the second winner (N_2) are found (lines 5-6 of Algorithm 3). The ESOINN uses Euclidean distance (Ed) in equation (5) as the distance metric.

$$\|\xi - N\| = Ed(\xi, N) = \sqrt{\sum_{i=1}^{n}(\xi^i - N^i)^2} \qquad (5)$$

where n denotes the number of dimensions of ξ or N. If the distances to the winners exceed the threshold of either winner, the ξ is added to $\mathcal{N}$ as a new node (lines 7-8 of Algorithm 3). Otherwise, the ξ is merged with N_1, the weights of its connected nodes are updated, the N_1 and N_2 are connected with an edge, edges connected to N_1 have their age increment by 1, and edges that exceed the user-defined parameter age_{max} are removed (lines 9-15 of Algorithm 3). Then, the threshold of the N_1 (T_1) and threshold of N_2 (T_2) are updated using equations (6) and (7), respectively (line 17 of Algorithm 3).

$$T_{N_1} = \begin{cases} \|N_1 - N_2\|, & \mathscr{p}_{N_1} = 0 \\ \max_{a \in \mathcal{P}_{N_1}} \|N_1 - N_a\|, & \mathscr{p}_{N_1} > 0 \end{cases} \quad (6)$$

$$T_{N_2} = \begin{cases} \|N_2 - N_1\|, & \mathscr{p}_{N_2} = 0 \\ \max_{a \in \mathcal{P}_{N_2}} \|N_2 - N_a\|, & \mathscr{p}_{N_2} > 0 \end{cases} \quad (7)$$

If the number of trained data is the multiple of the user-defined parameter λ, it performs the subclass update (line 19 of Algorithm 3), and denoising (lines 20-23 of Algorithm 3). The purpose of the subclass update is to separate subclasses at high-density overlapped regions, which is not used in prediction, so is not discussed in this research work. During denoising, nodes with $\mathscr{p} = 0$ are removed, and for nodes with $\mathscr{p} = 1 \wedge 2$, their *WT* are checked and removal if eligible (lines 22-23 of Algorithm 3).

2.9 Optimization algorithm

Optimization algorithms are used to solve optimization problems, such as optimizing the parameters of a prediction model for better accuracy. as reported in [57], [88]. Generally, the optimization algorithms are classified under two groups: swarm intelligence, such as particle swarm optimization (PSO), and evolutionary computation, such as genetic algorithm (GA) [89]. There are also classical optimization methods, such as dynamic programming and mixed integer programming. Similar to ANN, dynamic programming also encounters the "curse of dimensionality" problem [45]. PSO and GA are the common optimization techniques applied for control optimization [90].

Studies have shown that PSO converges faster than GA with comparable results [91]–[93]. Both GA and PSO are applied for tuning the PID controller for power level control in nuclear power plants in [93]. Results show GA and PSO have negligible differences in the final output, while PSO converges faster than GA. F. D. Wihartiko et. al. [91] compares the performance of GA and PSO for model integer programming bus timetabling problem and stated PSO is easier to implement when compared to GA. Y. Ding et. al. [92] demonstrated that PSO has a quicker and closer convergence solution than GA based on the three-component parallel reaction mechanism of biomass pyrolysis. As a result of the closer solutions, the PSO achieves a 30% outcome better than GA.

There are modified PSO aimed at achieving better peak reduction. A new algorithm named Multi-pass Iteration Particle Swarm Optimization (MIPSO) is

proposed in [94] to find the optimal BESS operating schedule for industrial Time-Of-Use (TOU) rate users with a wind turbine generator. The MIPSO is an improved version of PSO modified based on dynamic programming. Differing from the MD calculation used by the power company in Malaysia, the Taiwan power company has a different calculation method for MD charges. Each industrial customer has a contract capacity where the demand charge is two times the exceeded amount if the over-contract load is less than or equals to 10% of the contract capacity and is three times the exceeded amount if the over-contract load is higher than 10% of the contract capacity. It means the rate is higher if exceeds 110% of the contract capacity. Therefore, the goal of the controller is to reduce the power below 110% of the contract capacity to prevent getting charged at a higher rate, and thus 110% of the contract capacity is set as the threshold. This approach makes it similar to a fixed threshold controller. Instead, the MIPSO is targeted to improve computation efficiency and solution quality.

A novel SOM-PSO is also proposed in [95] to find the charge and discharge threshold for performing peak shaving on a battery with a supercapacitor. It achieves up to 68.97% faster optimization time compared to conventional PSO. With this faster optimization time, it is able to perform optimization per minute and outperforms filtration-based controller and PSO-optimized fuzzy logic controller by up to 7.33 times higher supercapacitor utilization, up to 91.94% reduction in the mean absolute rate of change of battery power, and up to 61.36% reduction in battery peak demand.

Since using improved state-of-the-art optimization to achieve faster convergence speed and better convergence is not within the scope of this research work, PSO is selected to be the optimization algorithm used in this research work due to its simplicity, reliable performance, and ease to adapt to different applications [96]. As previously discussed, PSO has been well applied and proven to be functional for the applications in this research work. When compared to other swarm algorithms, PSO may not necessarily have the best performance, but it can achieve similar results while having an average convergence time [97]. Comparing GA and PSO, they have negligible differences in results, but the PSO converges earlier than GA [93]. Therefore, PSO is better for optimizing the threshold in the MD reduction controller.

2.9.1 Stopping condition

PSO and GA work by evaluating the inputs with an objective function. Both require a stopping condition to stop evaluating the inputs and return the results. Generally, there are five stopping conditions [98]:

- **Maximal time budget** – This algorithm returns result after a predefined absolute time.
- **Maximal number of generations –** The algorithm returns result after running for a given maximum number of generations.
- **Maximal number of objective function evaluations** – The algorithm returns result after running for a given maximum number of objective function evaluations.
- **Hitting a bound** – The algorithm returns results if the best cost of the objective function is obtained.
- **K-iterations** - The algorithm returns results if the cost of the objective function remains unchanged for *K* iterations.

This research work uses a combination of **Hitting a bound** and **K-iterations**. The objective function in this work is designed to be optimal when the cost is zero. Therefore, the evaluation stops if it is **Hitting a bound**, which is when the cost is zero for this research work. However, it is possible that the cost stays very close to zero but stays relatively static and does not equal zero even after many iterations. Thus, the **K-iterations** condition is used as the second stopping criteria. If the cost remains unchanged for *K* iterations, the optimization stops and returns the result.

2.10 Research gap

Below lists the research gaps found based on the literature review:

- There are limited studies that consider the availability of historical data at the time of implementation. Historical data is crucial for both load prediction and peak reduction. Prediction models use historical data to pre-train the prediction models. For peak reduction with BESS, historical data are used for BESS sizing. Conventional fixed threshold controller uses historical data to pre-determine the fixed threshold. Some state-of-the-art controllers, such as [4], [33], [36], [41], require historical data for pre-determining some important rigid parameters that are used in the demand threshold finding process. They often require historical data of one year or longer [41].
- Models with incremental learning can update their trained model with new data without retraining a new model. It helps adaptability and prediction accuracy by learning the recent data. However, incremental learning is rarely studied. There are not many available models that support native incremental learning. To achieve incremental learning with conventional models, a new model needs to be retrained with accumulated data.
- For a model with native incremental learning, the existing model is updated with new data only. ESOINN is one of the models designed for incremental learning. ESOINN is an unsupervised learning algorithm that is not commonly used in time-series load prediction.
- There is a research gap where no peak reduction controller can be implemented with limited historical data and is capable of continuously adapting to future load profiles. Conventionally, state-of-the-art peak reduction controllers use day-ahead load prediction for optimizing the demand threshold. However, it is inevitable for day-ahead load prediction to have prediction errors. To prevent peak reduction failure due to poor day-ahead load prediction accuracy, most state-of-the-art peak reduction controller uses rigid parameters [4], [33], [36], [41]. These rigid parameters are impacting the adaptiveness of the controller. Pattern variations are evolving over time due to climate change and increased sector coupling. As a result, controllers that cannot adapt may lead to undesired outcomes [99].

- Prediction models with a shorter lead time, such as one-hour-ahead, are typically more accurate than day-ahead prediction models [52], [70], [100], [101]. Despite that, to the authors' knowledge, no study is found incorporating load prediction with lead times shorter than day-ahead for daily peak demand reduction of buildings. E. Reihani [100] does use two load predictions of different lead times but for different purposes, with day-ahead for peak demand reduction and 20-min ahead for load demand smoothing.

2.11 Summary

This chapter presents the literature review about the peak reduction methodology, maximum demand, demand-side peak reduction approaches, optimization algorithms, peak reduction controllers with BESS, and short-term load prediction models. Lastly, this chapter listed the research gaps found based on the literature review.

The literature review highlighted that the peak reduction with BESS has the highest potential among other approaches due to its compatibility and flexibility. Generally, studies do not consider the implementation practicality and adaptiveness of a controller. This includes the availability of historical data at the time of implementation, the complexity of the data acquisition, the determination of parameters, the time to implement, and its ability to deal with pattern variations that are evolving over time. These are applied to both prediction models and control strategy, which are the two main components of a modern peak demand reduction controller. This chapter has identified ESOINN as the potential prediction model that can solve the problems and fill the discovered research gaps. Since one-hour-ahead load prediction has better and more consistent accuracy than day-ahead load prediction, it should be able to incorporate into the peak reduction controller to assist the threshold optimization process instead of relying on rigid parameters that deteriorate the adaptability of the controller to evolving patterns.

Chapter 3 – Maximum Demand Reduction System with BESS

3.1 Overview

The overall structure of the maximum demand reduction systems with BESS is presented in this chapter. Following this, the two load profile datasets that are used to evaluate the performance are provided. Both datasets are analysed with autocorrelation function (ACF) and partial autocorrelation function (PACF), which are useful to determine the length of time lag for the time-series load prediction in Chapter 4.

3.2 Structure of Maximum Demand Reduction System

Figure 3.1 shows the system structure of the MD reduction system with BESS for implementation in a building. The BESS recoups energy during off-peak periods and discharges to the building's AC bus during peak hours. Thus, the building's loads draw less power from the grid, reducing the grid power (P_{grid}) that is seen by the electricity company. The MD reduction controller consists of two components: load prediction and threshold control. The load prediction collects the building's load power (P_{load}) and uses it to predict the future load profile. The threshold control determines the demand threshold (P_{Th}) for the next interval based on the predicted profile and state-of-charge (SOC) of the BESS. In this research work, the controller re-optimizes the P_{Th} every 5 minutes.

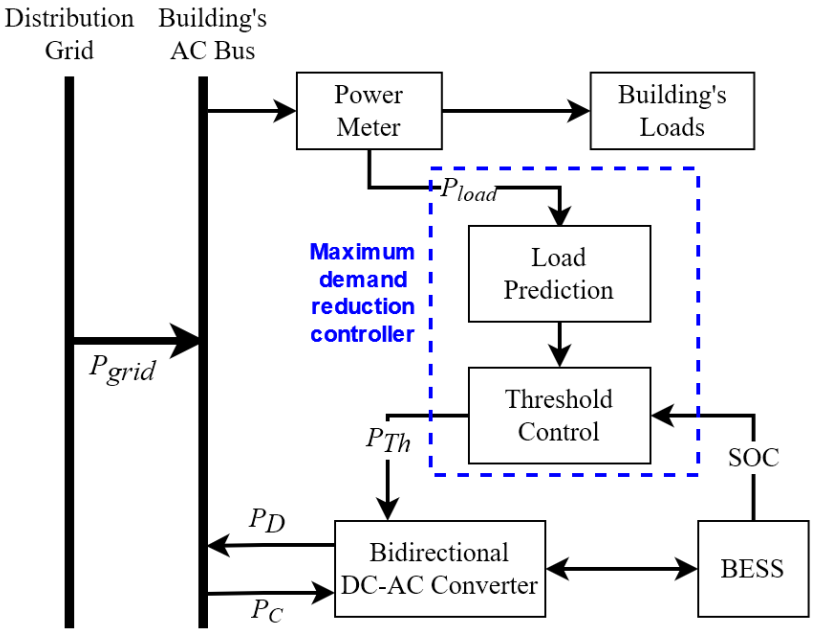


Figure 3.1: System structure of a MD reduction system with BESS [102]

The discharge power (P_D) and charge power (P_C) of the BESS are calculated using equations (8) and (9), respectively. When P_{load} is higher than P_{Th}, the difference in power is supplied by the BESS. When P_{load} is lower than P_{Th}, the difference in power is recouped to the BESS. The relationship between P_{grid} and P_{load} is shown in equation (10). The P_D and P_C affect the P_{grid} only, and the P_{load} remains unchanged.

$$P_D = \begin{cases} P_{load} - P_{Th} & ,if\ P_{Th} < P_{load} \\ 0 & ,otherwise \end{cases} \quad (8)$$

$$P_C = \begin{cases} P_{Th} - P_{load} & ,if\ P_{Th} > P_{load} \\ 0 & ,otherwise \end{cases} \quad (9)$$

$$P_{grid} = \begin{cases} P_{load} - P_D & ,\ if\ P_D > 0, P_C = 0 \\ P_{load} + P_C & ,if\ P_C > 0, P_D = 0 \end{cases} \quad (10)$$

This research work proposed a novel incremental prediction model for time-series load prediction and a novel dynamic two-stage controller that will be covered in the succeeding chapters.

3.3 Load Profile Datasets

The aim of this research work is to reduce MD for buildings in Malaysia, and the MD in Malaysia is calculated using energy consumption per 30-minute. Therefore, the demand threshold for MD reduction is allowed to adjust within this 30-minutes interval. This research work decided to have adjustments every 5-minutes, which allows up to 6 adjustments per 30-minute interval. Since MD is obtained monthly, it needs at least one month of complete data. Moreover, it also needs data for training the prediction model. Hence, the load profiles used in this research work are required to meet the three criteria below:

1. Has a data interval of 5-minutes or smaller,
2. Has at least three months of data for evaluating monthly MD reduction, and
3. Load profile collected in Malaysia.

After searching for online data intensively, no load profiles that met the criteria can be found. Hence, a dataset is collected within the premise of the University of Nottingham Malaysia Campus. The location is determined to be at the building of the faculty of electrical and electronics (Block D) at the University of Nottingham Malaysia. It is the only site with meter installation approved by the university and requires replacing two analogue meters with digital power meters only to keep the cost within

the available research budget. The collection of data is discussed in detail in Section 3.3.3. This dataset is referred to as *Dataset A* in the remaining of this thesis.

To demonstrate the scalability and adaptability of the algorithm, the system should be tested on more than one dataset. As previously mentioned, there is no online data that met the three requirements. Therefore, one online profile that met the 2nd and 3rd criteria is selected as the second dataset. The publicly available online load profile also allows this research work to be compared with other research works using the same dataset. This dataset is referred to as *Dataset B* in the remaining of this thesis.

3.3.1 Dataset A

Figure 3.2 depicts the *Dataset A* that is collected from 28th June 2019 to 16th December 2019. The data set has a time interval of 1-min. Figure 3.3 shows the layout of Block D. The building consists of two floors. The ground floor has different teaching facilities, while the first floor comprises of lecturer offices and meeting rooms. The building has cooling air conditioning that operates during working hours ranging from 09:00 to 18:00 on weekdays, excluding public holidays. The facility in the building has different operation type. Some facilities operate on-demand while some facilities operate as per teaching schedule. The operations of the facilities are summarized in Table 3.1.

Both Figure 3.2 and Table 3.2 show an increasing trend for the MD. The MD for July, August, and September are lower than other months as the facilities in the building are under-utilized due to the summer break of the campus. As the semester begins in October, the MD increases by 9.34%, from 51.69kW in September to 56.52kW in October. The relatively low increment of 9.34% is due to the low utilization of most lab facilities despite the semester starting in the mid of September. As the utilization rate rises and more equipment are operating at the same time, the MD begins to increase. The usage increases gradually as the usage of equipment increases as practical lectures begin in October. Additionally, the building does not shut down during the semester break since there are open for minority students, such as foundation and research students. The facilities that are always open include lights, air conditioning, the teaching computer lab, and lecturers' offices. When comparing the lowest and the highest MD within the period of collected data, which are the lowest MD of 47.195kW in July 2019 and the highest MD of 56.521kW in October 2019, the difference is at 17.76%, which is considerably higher.

The MD for November is similar to the MD for October. In December, the MD is slightly lower than the previous two months due to a lower utilization rate as it progresses towards the end of the semester. This research work considers only the load profiles on working days since the low power demand on non-working days does not contribute to the MD.

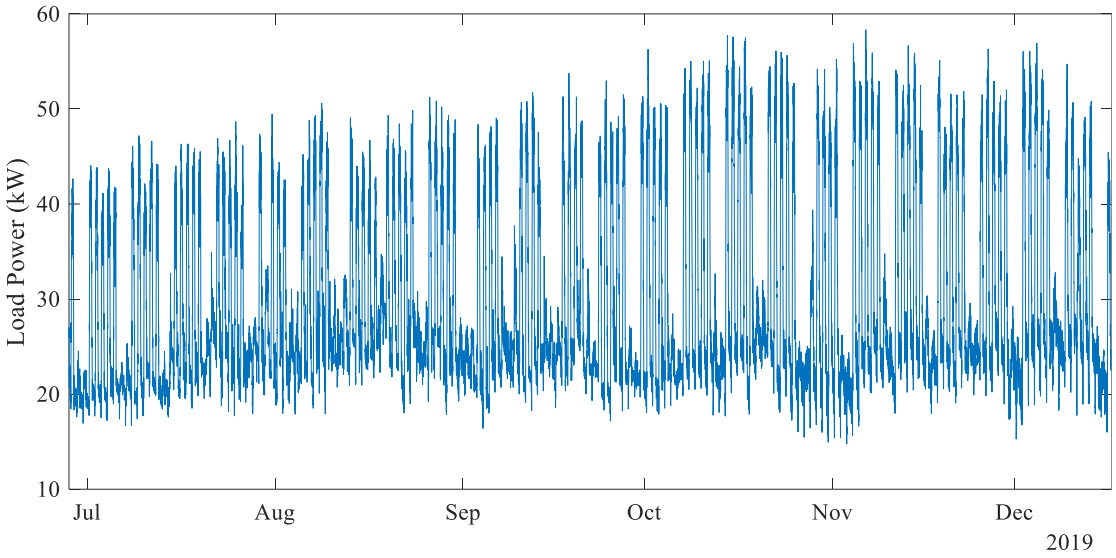


Figure 3.2: Dataset A

Table 3.1: Facilities in the educational building of Dataset A

Facility	**Operation type**
Teaching computer lab	• Operates as per teaching schedule. • Opens for all students outside teaching schedule.
Microprocessor lab	• Operates as per teaching schedule. • Opens for all students outside teaching schedule.
Technicians' and lecturers' office	• Operates during working hours.
Studio lab	• Operates as per teaching schedule.
Undergraduate research lab	• Operates on-demand.
High computing lab	• Operates on-demand.
High power motor lab	• Operates on-demand.
Postgraduate research lab	• Operates on-demand.
Server room	• Operates 24 hours a day.

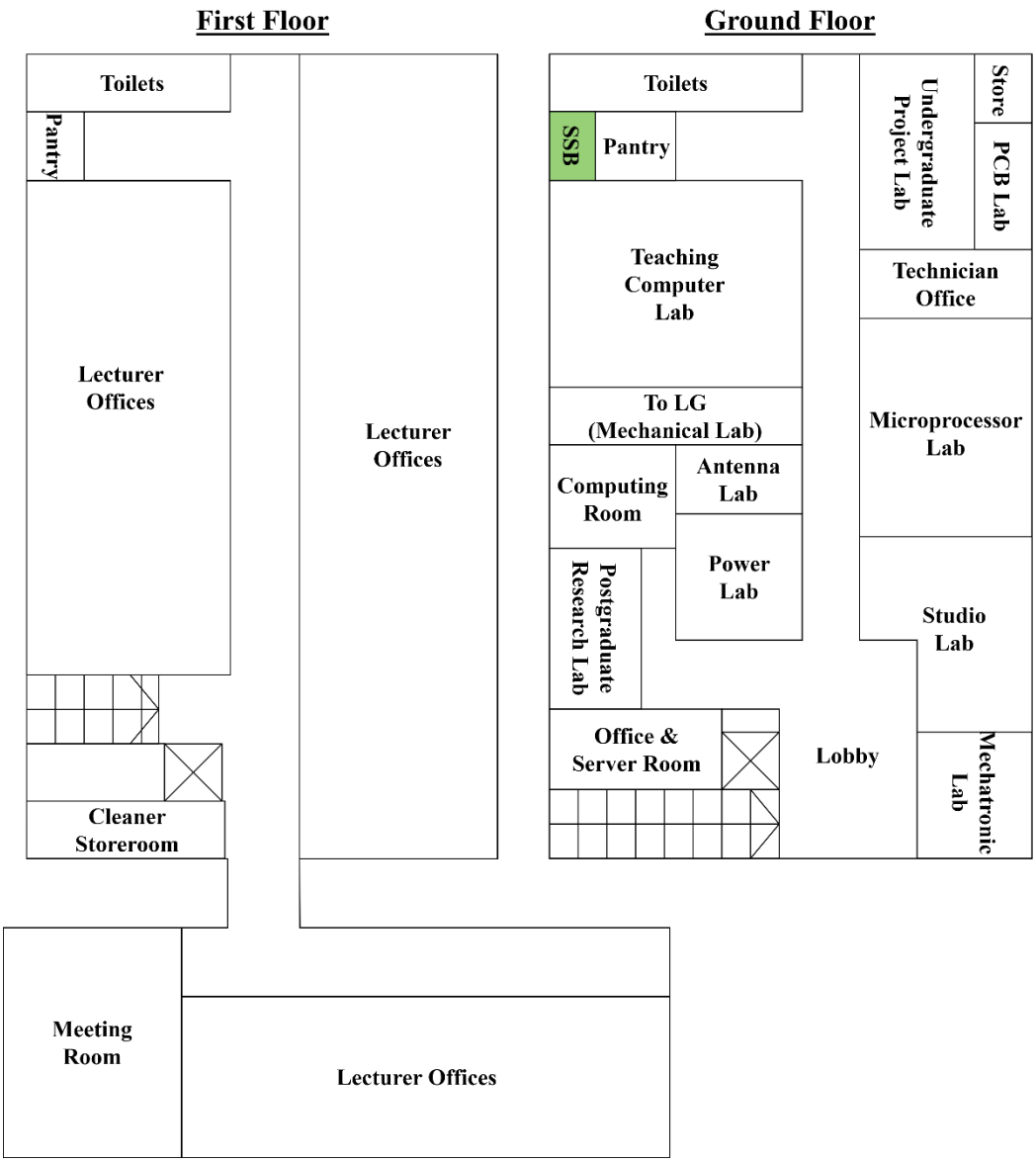


Figure 3.3: Layout of Block D in University of Nottingham Malaysia

Table 3.2: MD per month before MD reduction for Dataset A

Month	**MD before MD reduction (kW)**
Jul	47.195
Aug	49.864
Sep	51.587
Oct	56.521
Nov	56.036
Dec	54.879

3.3.2 Dataset B

Figure 3.4 shows the second dataset is hourly load data of the power supply company of the city of Johor in Malaysia generated in 2009, which is publicly available online at [103]. Since a publicly available load profile with a 1-min interval cannot be found, this hourly data is up-sampled to 1-min interval using spline interpolation to maintain the same 1-min resolution as the *Dataset A*. This dataset is referred to as *Dataset B* in this research work.

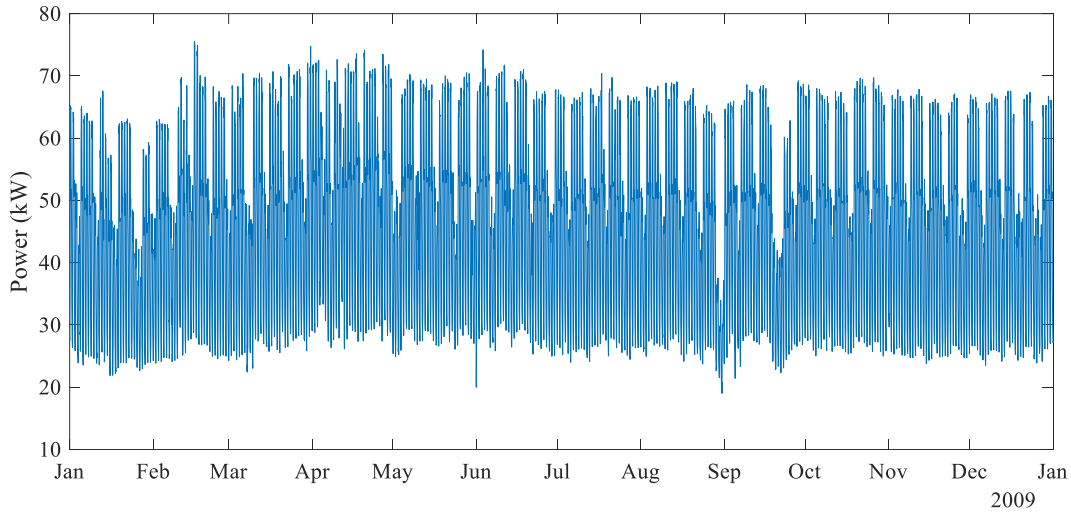


Figure 3.4: Dataset B

There is no detailed description of the building provided. Table 3.3 shows the MD before reduction per month for *Dataset B*. This dataset has a MD of 70kW between February to June that is higher than the MD for other months. The MD for February is the highest at 75.5kW.

Table 3.3: MD per month before MD reduction for Dataset B

Month	**MD before MD reduction (kW)**
Jan	67.485
Feb	75.473
Mar	74.699
Apr	74.054
May	70.494
Jun	73.814
Jul	69.958
Aug	69.028
Sep	69.172
Oct	69.554
Nov	67.792
Dec	67.555

3.3.3 *Dataset A* Data Collection

Originally, the sub-switch board (SSB) room in block D installed two analogue power meters to measure the two power lines connected to the ground and first floors of Block D. The connected loads per line are unknown, but the combined profile of these two meters is the load profile of the building. Since the analogue power meters do not support data logging or any communication protocol for data extraction, they are replaced with ION7330 digital power meters. Figure 3.5 presents the two ION7330 power meters and the wireless router in the Sub-switch board (SSB) room.

Figure 3.6 depicts the block diagram of the load profile data collection. The power meters are connected to a wireless router via ethernet. The wireless router is configured to have a local network. Other devices can connect to this local network via its password-protected Wi-Fi network to access and configure the meters. A raspberry pi 3B is used as the data logger to log the data from both meters every 5s through TCP Modbus protocol. Then, the data is then averaged to an interval of 1-min. With the 5 seconds sampling interval, it allows up to 11 data retrieve failures with a minimum of one success read. The ION7330 does have logging functionality but it is limited in terms of the type of data and the length of logging due to its limited storage capacity. Therefore, it is only used for redundant logging where it can log up to 2 weeks of 1-min important values in the event the raspberry pi 3 stopped working unexpectedly.

The raspberry pi 3 is the primary logger responsible for logging all values the meter measure. The raspberry pi 3 can store years of data with its 16GB microSD card.

Figure 3.5: ION 7330 meters and the wireless router in SSB room

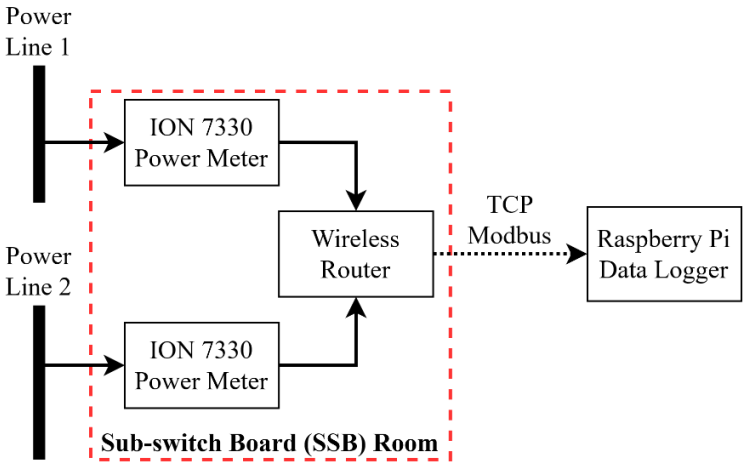


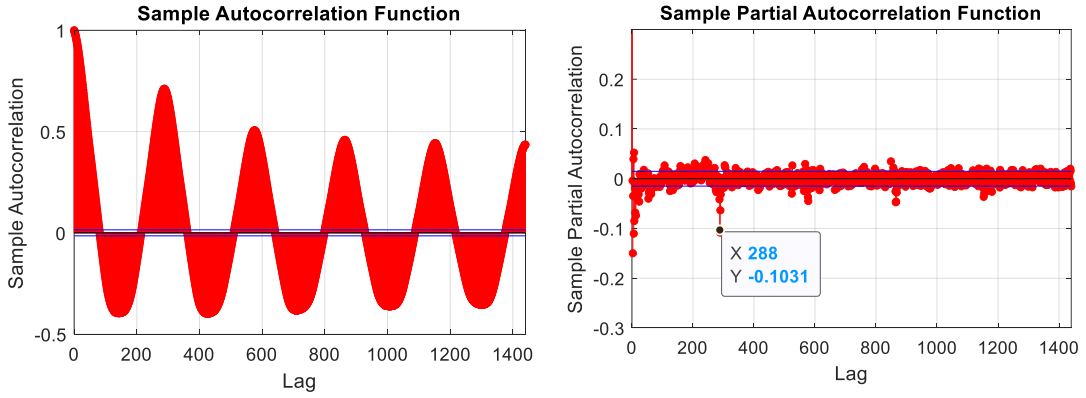
Figure 3.6: Block Diagram of Load Profile Data Collection

3.3.4 Load Data Analysis

Dataset A and *Dataset B* are from different business sectors. The educational load profile of *Dataset A* can be easily influenced by the semesters, while *Dataset B* depends on the number of people working or visiting the building, which may vary daily. Both datasets have observed variations in the monthly MD, which are suitable to test the adaptability of the maximum demand reduction system. The purpose of the second

profile is also to make sure the proposed MD reduction system not only works on a single load profile. Moreover, it is to test the adaptability and ease of implementation of the controller. The controller should require no changes in the algorithm when testing on the two datasets.

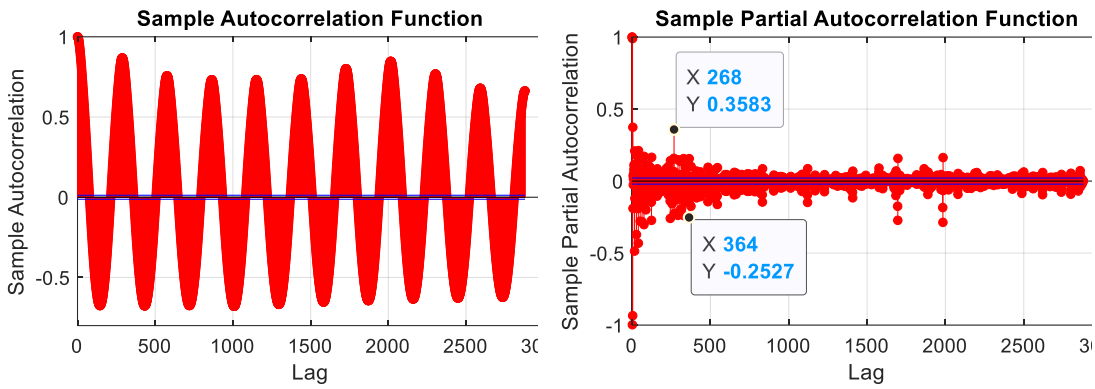
Autocorrelation function (ACF) and partial autocorrelation function (PACF) are applied to identify the suitable historical data included as input features for the load prediction models. As indicated in Figure 3.7(ii), it shows significance at the time lag of 288 (There are 288 data points for a day of profile with 5-min interval). Hence, the historical data for the day-ahead models are selected as the previous 288 historical data points. For the one-hour-ahead models, the previous 12 data points are selected and should be sufficient for the shorter one-hour lead time.



(i) (ii)

Figure 3.7: ACF and PACF plots of Dataset A

Figure 3.8 depicts the ACF and PACF plots of *Dataset B*. It shows high significance at time lag of 268. Although there is also relatively high significance time lag up to 364, the historical data for day-ahead is selected as previous 288 historical datapoints. This makes it uses the same amount of historical data points as *Dataset A*. The historical data points for one-hour-ahead models are selected as previous 12 data points, which is the same as *Dataset A*.



(i) (ii)

Figure 3.8: ACF and PACF plots of Dataset B

3.4 Summary

This chapter presents the overall structure of the maximum demand reduction system with BESS and the two load profile datasets used in this research work. An MD reduction system with BESS consists of four main components: power meters that measure loads of the building, BESS that stores and discharges energy, a bidirectional converter that control the charging and discharging powers of the BESS, and an MD reduction controller that is responsible for finding the optimal demand threshold. A modern MD reduction controller, including the proposed controller in this research work, consists of two parts: load prediction and threshold control. The load prediction predicts the load profile. The threshold control optimizes the demand threshold on the SOC, current load power, and the predicted load profile.

Two datasets, namely *Dataset A* and *Dataset B,* are used in this research work. *Dataset A* is collected from Block D in the University of Nottingham Malaysia, while *Dataset B* is a publicly available dataset. The purpose of using two datasets is to ensure the proposed system does not only work on a specific profile and does work on different profiles. The use of publicly available datasets allows cross-studies comparison with the same dataset. The collection of *Dataset A* is also explained in this chapter.

ACP and PACF are applied to both datasets to find the significance of time lag, which is used to determine the number of historical data points used for training the model. *Dataset A* shows the significance of time lag at up to 288 data points or one day of data, while *Dataset B* shows the significance of time lag at up to 364 data points or 1.26 days of data. Hence, the previous 288 historical data points or one day of historical data points are selected as the inputs for the day-ahead load prediction models, while

the previous 12 data points or one hour of data are selected as the inputs for the one-hour-ahead load prediction model. The selection of the shorter lead time for the one-hour-ahead load prediction model is better suited for the shorter lead time. These selections are applied to the prediction models of both *Dataset A* and *Dataset* B, so it is a common configuration for different profiles.

Chapter 4 – Improved Self-Organizing Incremental Neural Network Model for Short-term Time-series Load Prediction

4.1 Overview

To achieve a controller that is adaptive and can be implemented with limited historical data, the prediction model used by the controller needs to be adaptive and can be implemented with limited historical data. This chapter proposes a new incremental DB-SOINN-R prediction model for time-series day-ahead and one-hour-ahead load predictions. The DB-SOINN-R is first presented, followed by the elaboration of the simulation setup used to evaluate the proposed model. Then, the results and discussion are presented. In the last section, the findings of this chapter are summarized.

In this chapter, the proposed DB-SOINN-R is compared with six conventional models: feedforward neural network (FFNN), support vector regression (SVR), deep neural network with long short-term memory (LSTM), enhanced self-organizing incremental neural network (ESOINN), and kNN regression. All models are tested on lead times of day-ahead and one-hour-ahead, which are common lead times for building energy managements that have been used in other researches [52], [70], [104]. For example, one-hour-ahead is commonly used for reducing load fluctuation [95], [105], while day-ahead prediction is used for demand-side management peak reduction, distributed power network operation, and the integration of renewable energy sources [68]. Other than evaluating the prediction accuracy of models, the effect of incremental learning is also investigated by comparing the accuracy of the models that are trained with the training data only with models that are trained with the training data and updated or re-trained daily if native incremental learning is not supported.

4.2 Density-based Self-Organizing Incremental Neural Network with regression (DB-SOINN-R)

Based on the literature review of prediction models in Chapter 2, ESOINN is a great candidate. Its incremental learning allows learning of new data as time progress, instead of waiting to collect a long period of historical data. However, there are three problems with ESOINN. The first problem is the simple winning time (*WT*) based denoising process of the ESOINN may incorrectly remove important nodes when the data has uneven density distribution. The *WT*-based denoising process works by comparing the

WT of nodes with the mean *WT* of all the nodes in the model. It does a simple assumption that the higher the *WT*, the less likely it is a noise node. But not all nodes have similar *WT*. It is possible to have clusters that tend to have more nodes with low *WT* instead of fewer nodes but high *WT*. They are similar but not close enough to be merged. Moreover, the original denoising stage does not consider how many iterations have executed since the node is inserted, so a recently inserted node may be removed. The second problem is the inappropriate use of Euclidean distance as the distance metric when the input dataset consists of different type of data with each having different dimensions. The use of Euclidean distance often causes the selection of incorrect nodes that has a significantly different timestamp or minute of the day, causing the predicted profile to lag the actual profile. This imbalance in dimensions causes the minority data variables to have less impact compared to other variables. As a result, the predicted outputs may lag the original profile. The third problem is the ESOINN cannot produce unique output for each input. As the original ESOINN is designed for clustering, its predicted output is based on the nearest node. Hence, two different inputs may have the same predicted output. In other words, the output is discrete and cannot produce continuous predicted values.

To solve these problems, this chapter proposed a new improved ESOINN named DB-SOINN-R. The DB-SOINN-R consists of three modifications: (i) a new density-based denoising that replaces the WT-based denoising, (ii) a new mean Euclidean distance (*mEd*) as the distance metric, and (iii) incorporates the k-nearest-neighbors inverse distance weighting (kNN-IDW) regression [69] so a unique output is obtained based on the distance. The model is first trained with the DB-SOINN. Then, the trained DB-SOINN is used alongside the kNN-IDW regression to achieve the final prediction, forming the DB-SOINN-R model.

4.2.1 Density-based Self-Organizing Incremental Neural Network (DB-SOINN)

The DB-SOINN is based on ESOINN and there are three changes. The first change is the change of merging condition. The second change is the introduction of the new density-based denoising that replaces the *WT*-based denoising of the original ESOINN. The last change is the use of a new distance metric, namely mean Euclidean distance (*mEd*) that replaces the use of Euclidean distance as the distance metric to find the distances between data. Figure 4.1 illustrates the flowchart of the proposed DB-SOINN algorithm, and Algorithm 4 in Appendix 1 shows the pseudocode for the DB-SOINN.

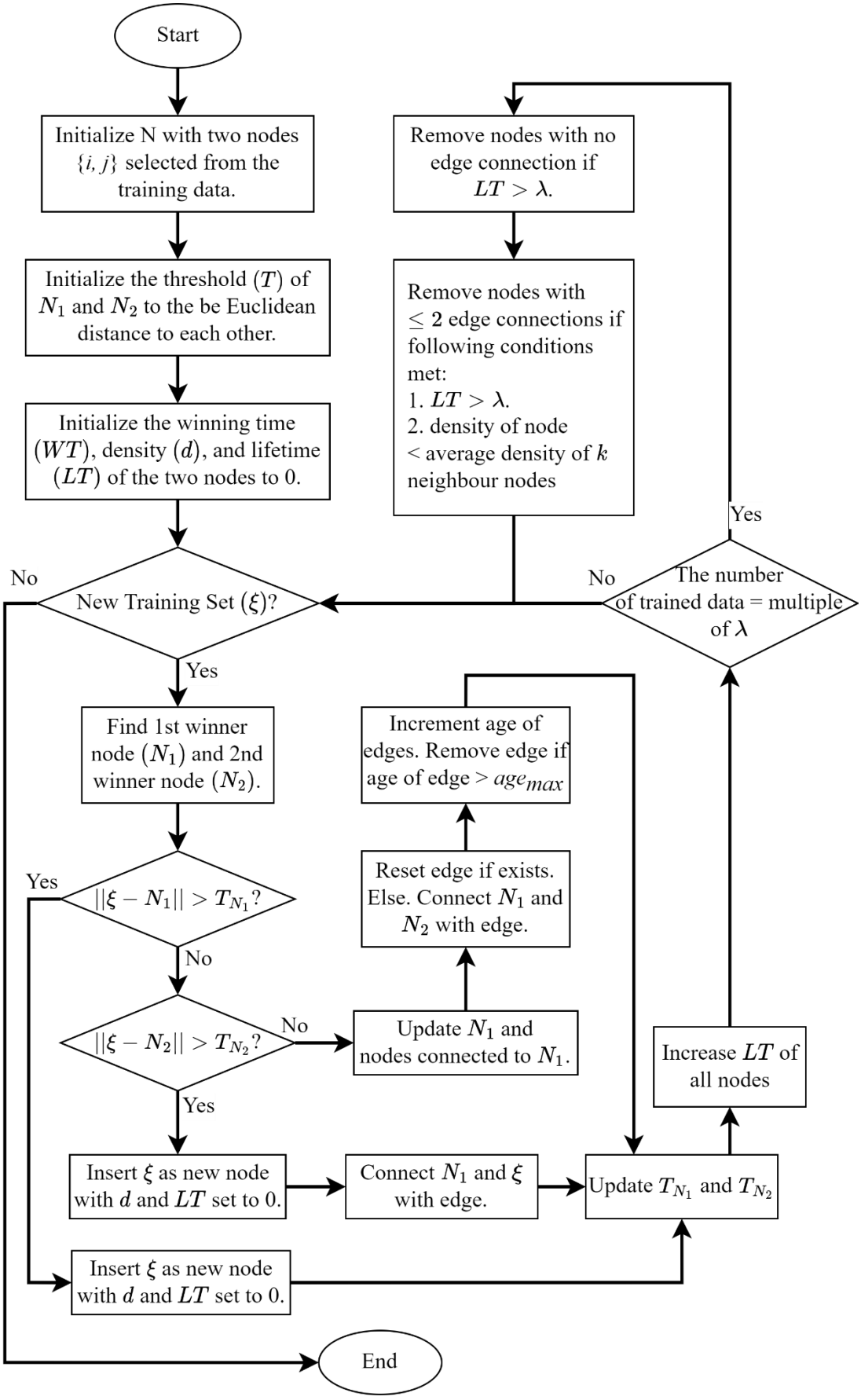


Figure 4.1: Flowchart of the DB-SOINN Algorithm

In ESOINN, the ξ is added to $\mathcal{N}$ if the distance to N_1 exceeds T_{N_1} or the distance to N_2 exceeds T_{N_2} (lines 7-8 of Algorithm 3). So, if the ξ has a distance to N_1 shorter than T_{N_1} and a distance to N_2 larger than T_{N_2}, it will still be inserted as a new node. But for such situations, the ξ should be related or similar to N_1 since it is within the T_{N_1}. Hence, a new condition for node insertion is added in DB-SOINN-R where the ξ is still appended as a new node but also connects the newly inserted node with N_1 if the distance to N_1 is lower than T_{N_1} and the distance to N_2 exceeds T_{N_2} (lines 8-12 of Algorithm 4). If the distance to N_1 exceeds T_{N_1} and the distance to N_2 exceeds T_{N_2}, the ξ is inserted as a new node without any edge connection (lines 13-16 of Algorithm 4).

The update subclass process (line 18 of Algorithm 3) is removed in DB-SOINN as separation of classes at high density overlapped regions is only useful for clustering and not concerned for prediction. Two new variables are introduced for every node and are used in new density-based denoising in the DB-SOINN. The first variable is the lifetime of the node (LT). The LT measures the longevity of the node since it is first added to the model. The node should only be eligible for removal if the node has existed in the model for iterations higher than λ. This prevents a recently added node to be removed as soon as the next training iteration. The LT of the node is initialized to 0 for new nodes (lines 11 and 16 of Algorithm 4) and is incremented by one in every training iteration (line 27 of Algorithm 4).

The second new variable is the density of node (d). A node with non-zero WT means it has merged with one or more input data, so it should be treated as a group of data instead of a single point, thus the d is used to record the density of this group of data. Instead of using the discrete value of WT, the d is introduced to record the distances to the input data that merged with N_1. A new node is initialized with $d = 0$ (lines 10 and 15 of Algorithm 4), indicating it has not been merged with any data. When the N_1 is merged with ξ, the d of N_1 (d_{N_1}) is updated (line 22 of Algorithm 4) with the distance between the ξ and N_1 as shown in equation (11). The higher the d, the lower the density of the node that represents a group of data merged into a single node. The d is used in the new density-based denoising.

$$d_{N_1} = (d_{N_1} + \|\xi - N_1\|)/2 \qquad (11)$$

As discussed earlier in the introduction of this chapter, the *WT*-based denoising of the original ESOINN has a problem which may incorrectly remove important nodes when the data has uneven density distribution. To solve this issue, the *WT*-based denoising is replaced with a new density-based denoising that compares the local density of node (*D*) of the concerning node with the average *D* of its $k_{denoise}$ neighbor nodes. The $k_{denoise}$ is a new parameter introduced by the new density-based denoising to replace the parameters C_1 and C_2 used by the original WT-based denoising of the ESOINN.

The $k_{denoise}$ determines the number of nearest neighbor nodes considered in the calculation of *D*, as shown in equation (12). The calculation of *D* differs depending on the *d* of the node. If the *d* is non-zero, it means the node has been merged with ξ before and should be treated as a group of data with a density of *d*. If the *d* is zero, it means the node is not merged with ξ and should be treated as a single node.

$$D_i = \begin{cases} \frac{1}{k_{denoise}} \sum_{1}^{k_{denoise}} \left\| N_i - N_j \right\|, (j \in \mathcal{K}_i), d_i = 0 \\ \frac{1}{k_{denoise}+1} \left(\left(\sum_{i}^{k_{denoise}} \left\| N_i - N_j \right\| \right) + d_i \right), (j \in \mathcal{K}_i), d_i > 0 \end{cases} \quad (12)$$

The calculation of $k_{denoise}$ uses the number of neighbor nodes instead of radius because there might be no nodes in the radius and thus failed to calculate the density of the node or needs to be configured as extremely big or infinite. Hence, a number of nodes are selected to calculate the neighbor nodes. The $k_{denoise}$ is recommended to be within the range of 2 – 5. A high number of $k_{denoise}$, less similar nodes may be treated as part of the group of similar nodes and reduces the accuracy of denoising, where a similar concept can be found in kNN [106].

In the density-based denoising (lines 28-36 of Algorithm 4), only nodes with $\mathcal{p} \leq 2$ are concerned. This is because nodes with two or more connections should have merged with at least two input data and are well established as part of a cluster of similar nodes, while nodes with one or fewer connections may be noise nodes that have less occurrence in the dataset. For nodes with $\mathcal{p} = 0$, they are removed if *LT* is higher than λ (line 30 of Algorithm 4). For nodes with $\mathcal{p} = 1 \wedge 2$, they need to meet two criteria to be removed. The first criterion is the node's *LT* is higher than λ. This is to ensure the node does not get removed when it is a new node that is inserted into the model recently. The second criterion is the node's *D* is higher than the average of *D* of its $k_{denoise}$ neighbor nodes (line 34 of Algorithm 4). The *D* of the concerning nodes (D_a at line 32

of Algorithm 4) and D of each $k_{denoise}$ nearest nodes of the concerning nodes (D_b at line 33 of Algorithm 4) need to be calculated. The purpose is to remove nodes that have a different density than other nodes. As previously mentioned, the density of node (d) is introduced to record the distances to the input data that merged with N_1. Hence, the higher the d, the lower the density. If the node has a higher D than the average of neighbor nodes, the node has a lower density than its neighbor nodes. Thus, it is recognized as a noise node and is removed.

As above-mentioned, the second problem of ESOINN is the inappropriate use of Euclidean distance (*Ed*) when the dimension of each input type is different. However, this is common for training data to incorporate time-series data with discrete data, such as day type, time of the day, and day of the week. These discrete input variables are usually one dimensional, while the historical time-series data has more than one dimension. For example, each training set consists of one dimension for the minute of the day and twelve dimensions for the historical time-series load demand, forming a ratio of 1:12 in terms of the number of dimensions per variable type. This imbalance in dimension may cause the predicted output to lag the actual profile. To tackle this problem, the proposed DB-SOINN uses a new distance metric named mean Euclidean distance (*mEd*) instead of *Ed* to find the distance between nodes and data. To find the *mEd*, an Ed for each variable type is first calculated, then the *Ed* of each variable type is divided by their respective dimension. Then, the *mEd* is the sum of the divided *Ed*, which is shown in equation (13). Since the *Ed* per variable type is divided by their respective dimension, each variable type is now treated as one dimension regardless of its original number of dimensions, thus all variable types will have equal impacts at the resultant distance.

$$\|\xi - N\| = mEd(\xi, N) = \sum_{i=1}^{m} \frac{1}{n_i} \sqrt{\sum_{j=1}^{n} (\xi^j - N^j)^2} \qquad (13)$$

where n and m are the data dimension per variable type and number of variable types, respectively.

4.2.2 K-nearest Neighbour Inverse Distance Weighting Regression (kNN-IDW)

Since SOINN based models, including the DB-SOINN, are unsupervised learning algorithms, they do not need a dedicated target set for model learning. For prediction, the desired output or target data is combined with the input data, forming the training data. Without kNN-IDW, the prediction output will be the target data associated with

the nearest node. This would cause two issues as illustrated in the two scenarios in Figure 4.2. Table 4.1 provides the expected predicted outputs for three inputs in Figure 4.2.

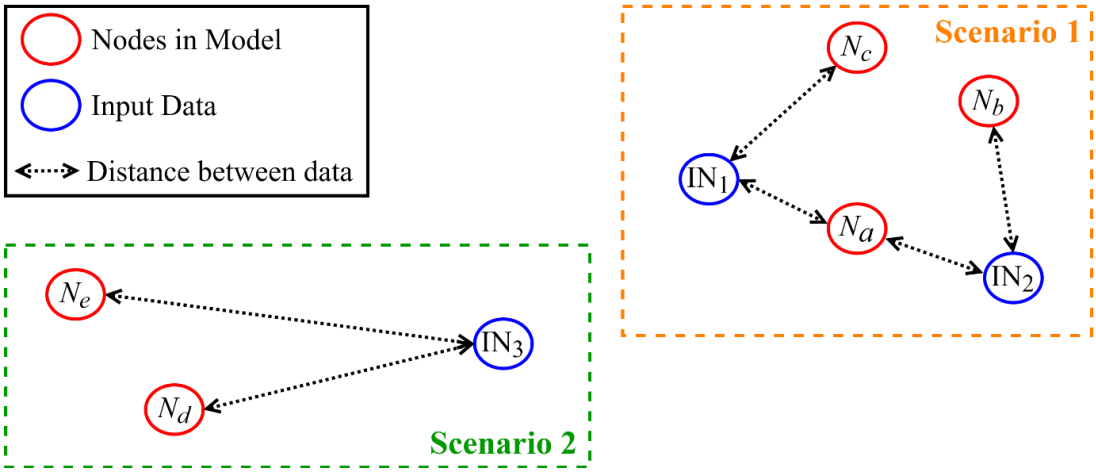


Figure 4.2: Two possible scenarios of using ESOINN for prediction

Table 4.1: Expected predicted outputs for the three inputs in Figure 4.2

Input	**Predicted Output without kNN-IDW**	**Predicted Output with kNN-IDW (k_{idw} = 2)**
IN_1	Target data associated with N_a	A unique output calculated using equation (13) with the nearest nodes: N_a and N_c
IN_2	Target data associated with N_a	A unique output calculated using equation (13) with the nearest nodes: N_a and N_b
IN_3	Target data associated with N_d	A unique output calculated using equation (13) with the nearest nodes: N_d and N_e

In scenario 1, the two different inputs (IN_1 and IN_2) have the same predicted output (target data associated with N_a) since both inputs have the N_a as the nearest node. In scenario 2, the predicted output for IN_3 is the target data associated with N_d as it is the nearest node to IN_3, despite the high distance between IN_3 and N_d (distance between IN_3 and N_d is significantly higher than the distance between IN_1 and N_a and the distance between IN_2 and N_a). The high distance means IN_3 has a low similarity with N_d. The N_d is selected because no node is closer than N_d. To tackle these problems, the kNN-IDW regression is added to the DB-SOINN, forming the DB-SOINN-R. In this chapter,

models that incorporate kNN-IDW regression will have '-R' added to the end of the models' name.

The kNN-IDW regression is not part of the DB-SOINN algorithm but is used for prediction to obtain unique outputs. The kNN-IDW adds weights that are inversely proportional to the distance. The higher the distance, the smaller the contribution to the output. The number of nodes used in the kNN-IDW regression is based on the user-defined parameter, k_{idw}. Equation (14) shows the equation to find the predicted output ($P_{predicted}$).

$$P_{predicted} = \frac{\sum_{j=1}^{k_{idw}} \frac{T_j}{dist_j}}{\sum_{j=1}^{k_{idw}} \frac{1}{dist_j}} \qquad (14)$$

where T_j is the target data associated with j-th nearest node and $dist_j$ is the distance (*Ed* for ESOINN and *mEd* for DB-SOINN) between input and the j-th nearest node.

4.3 Simulation Setup

Since each building or site has its unique characteristics, prediction models should be trained with the data collected from the site. To emulate the lack of historical data at the time of implementation, the models are first trained with a small amount of data from their respective datasets before testing on the remaining data. The first two months of data are used for the initial model training, and the remaining months of data are used for testing. There will be four months and ten months of testing data for *Dataset A* and *Dataset B*, respectively.

Since the aim of the research work is to develop an MD reduction controller that requires minimal historical data to begin operations, the load prediction model needs to be implementable with a small amount of historical data. Hence, two different scenarios are performed, with *Scenario A* to obtain the baseline results and *Scenario B* to obtain the results with incremental learning. The purposes of the two scenarios are to compare and study the effect of incremental learning on both conventional and proposed models in terms of accuracy and computational resources. Since the models are trained with only two months of data, they can also be used to study the effect of small training data on prediction accuracy.

Models in *Scenario A* performs prediction conventionally where the model is constructed using a set of train data and tested on a different set of test data, while models in *Scenario B* are updated at the end of each day using the data collected on the day throughout the testing period. For models that do not have native incremental learning, which includes the FFNN, LSTM, and SVR, their models are retrained daily with all data collected up to the day, including the data used to train the previous model. The proposed DB-SOINN-R is compared with five models: FFNN, LSTM, SVR, kNN-R, and ESOINN. For all day-ahead and one-hour-ahead models, the parameters are grid-searched and are always shown first in the result sections. To ensure the models are tested on unseen data, the first two months data of pre-training data are used for grid-search [107], [108], with the last two weeks of the two months used for validation.

Although the models are evaluated on two datasets: *Dataset A* and *Dataset B*, the detailed analysis is performed on *Dataset A* since more knowledge of the characteristics of the load profile and buildings. The proposed model is then tested on *Dataset B* to verify that it not only works on a specific dataset but also performs similarly well on a different dataset that is publicly available online.

4.3.1 Data Preparation

This chapter evaluates the prediction models on two lead times: day-ahead and one-hour-ahead. Figure 4.3 shows the block diagrams for the day-ahead and one-hour-ahead models. The day-ahead model is to predict the next day load profile, and the one-hour-ahead model is to predict the one-hour-ahead load demand. The day-ahead load prediction predicts the next day profile once per day, while the one-hour-ahead prediction predicts the one hour ahead demand every 5-min. With a data interval of 5-mins, the day-ahead model has 288 inputs and 288 outputs, whereas the one-hour-ahead model has 13 inputs and 1 output.

Both day-ahead and one-hour-ahead models use historical time-series load profiles, and only load profiles on working days are concerned, as most energy optimization tasks are necessary only with high and fluctuating demands. Most studies focus on load prediction improvement with the inclusive of complex inputs, such as weather information and building internal temperatures. But different sensors need to be installed and processed for the prediction, in which the processes are complex. The difficulty of implementing sensors for existing buildings has been studied in [74].

Moreover, the correlations between different types of variables are not necessarily the same for all buildings or sites. It is time-consuming to develop a specialized prediction model for each building. Both training and testing data are normalized using min-max normalization with the same min and max values of 0 and 60, respectively.

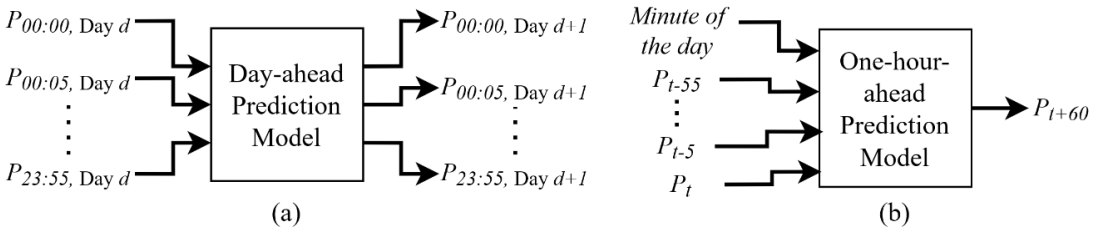


Figure 4.3: Block diagrams for the day-ahead and one-hour-ahead prediction models

4.3.2 Model Configuration

This proposed DB-SOINN-R is compared with five models: FFNN, LSTM, SVR, kNN-R, and ESOINN. The FFNN is using the scaled conjugate gradient as the training function with its number of hidden layers and its number of hidden neurons per layers grid searched. The LSTM has the LSTM layers, which are connected to a fully connected layer, and then connected to a regression layer (output layer). The LSTM is using the 'ADAM' optimizer, a learning rate of 0.005, an epoch of 250, and mean-squared-error as the loss function. The number of LSTM layers and the number of hidden units per LSTM layer are grid-searched. The SVR models use the Gaussian kernel with hyperparameters optimized using the built-in function in MATLAB [109]. For the SVR day-ahead model, one SVR model is trained for each of the 288 outputs for the day-ahead load prediction because ordinary SVR algorithms can only have one output per model. The kNN-R has its only parameter, k grid-searched. For ESOINN and DB-SOINN-R, its age_{max} is initialized to be infinite for long-term dependency, while the λ is defined based on the nature of the data. A short λ may cause frequent denoising and incorrect removal of data. For example, day-ahead models with $\lambda = 5$ mean the model is denoised every 5 days, while one-hour-ahead models with $\lambda = 1440$ mean the model is denoised every 5 days (288 data per day with a time interval of 5-min).

4.3.3 Performance Metrics

This chapter evaluates the models based on their prediction accuracy and execution time. Five commonly used metrics for prediction accuracy are selected, including mean absolute percentage error (MAPE), root mean square error (RMSE), coefficient of

variation of root mean square error (CVRMSE), mean absolute error (MAE), and coefficient of determination (R^2). RMSE and MAE measure the errors in kW, MAPE and CVRMSE are in percentage, and R^2 is unitless. The percentage unit of MAPE and CVRMSE ease cross-studies comparison. R^2 is used to evaluate the model's robustness [67].

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{|y_t|} \times 100\% \quad (15)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (16)$$

$$CVRMSE = \frac{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}}{\bar{y}} \quad (17)$$

$$MAE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|} \quad (18)$$

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{\sum_{t=1}^{n}(y_t - \bar{y})^2} \quad (19)$$

where y_t, $\hat{y}_t$, and $\bar{y}$ represents actual power, predicted power, and mean of actual power, respectively. To evaluate the execution time, the times spent at model training and prediction are measured, which include the initial training time, average prediction time, average incremental learning time, total prediction time, and total incremental learning time. Initial training time measures the time taken to pre-train the model with the first two months of data. The average prediction time is the average time spent on obtaining the output per interval. The average incremental learning time is the average time spent per day on the daily incremental learning or daily model retraining. The total prediction time and total incremental learning time are the total time spent on prediction and the total time spent on incremental learning, respectively. The data processing time is excluded since they are done before training and prediction. All executions are bounded to CPU only as not all algorithms are coded to support GPU acceleration. All models are coded in MATLAB 2019b and performed on a laptop with an Intel i7-6700HQ quad-core CPU and 16GB RAM.

4.4 Results and Discussion for *Dataset A*

This section presents the results and discussion of six models: FFNN, LSTM, SVR, kNN-R, ESOINN, and the proposed DB-SOINN-R on *Dataset A*. The grid-searched

parameters of these models are provided in Section 4.4.1. Section 4.4.2 and 4.4.3 discuss the prediction accuracies of the day-ahead and one-hour-ahead models for *Dataset A*, respectively. The execution times of the day-ahead and one-hour ahead models are shown in Section 4.4.4.

4.4.1 Grid-searched Parameters

The grid-searched parameters for the day-ahead and one-hour-ahead models in this section are given in Table 4.2 and Table 4.3, respectively.

Table 4.2: Grid-searched parameters for day-ahead models on Dataset A

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 6, Number of hidden neurons per layer = 72
LSTM	Number of LSTM layers = 1, Number of LSTM per layer = 24
SVR	*Parameters for all 288 models are optimized using built-in optimizer.*
kNN-R	k_{idw}=6
ESOINN, λ=*20*	C_1=0, C_2=0
DB-SOINN-R, λ=*20*	k_{idw}=6, $k_{denoise}$=2

Table 4.3: Grid-searched parameters for one-hour-ahead models on Dataset A

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 5, Number of neurons per layer = 6
LSTM	Number of LSTM layers = 3, Number of LSTM per layer = 9
SVR	Epsilon = 0.000744, KernelScale=0.262891, Lambda=0.0000339
kNN-R	k_{idw}=34
ESOINN, λ=*2880*	C_1=1, C_2=1
DB-SOINN-R, λ=*2880*	k_{idw}=12, $k_{denoise}$=10

4.4.2 Day-ahead Load Prediction Accuracy

Table 4.4 shows the prediction accuracy of the day-ahead models in *Scenario A* and *Scenario B*. In *Scenario A* where the models are only trained with the first two months of data and does not incremental learn daily, the best performer in MAPE, CVRMSE, R^2 are kNN-R, DB-SOINN-R, and LSTM models, respectively. With incremental learning in *Scenario B*, the proposed DB-SOINN-R model performs the best in all

metrics. All models in *Scenario B* show better accuracy than their counterparts in *Scenario A* that has no incremental learning or retraining. This shows the implementation of daily incremental learning or retraining results in better prediction accuracy.

Table 4.4: Prediction accuracy of day-ahead models in *Scenario A* and *Scenario B* on *Dataset A*

Models	***Scenario A* (without incremental learning)**					***Scenario B* (with incremental learning)**				
	MAPE (%)	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
FFNN	10.281	5.064	15.314	3.756	0.913	9.329	4.299	13.001	3.223	0.907
LSTM	9.976	4.99	15.116	3.685	0.920	9.054	4.386	13.265	3.195	0.911
SVR	9.912	4.817	14.569	3.565	0.906	7.533	3.592	10.865	2.610	0.926
kNN-R	9.188	4.489	13.575	3.278	0.915	7.203	3.171	9.500	2.378	0.928
ESOINN, $\lambda=20$	10.161	4.700	14.216	3.500	0.883	8.369	3.591	10.860	2.728	0.903
DB-SOINN-R, $\lambda=20$	9.250	4.468	13.514	3.275	0.916	6.986	2.995	9.058	2.286	0.936

Figure 4.4 presents the monthly MAPE and CVRMSE for the six day-ahead models: FFNN, LSTM, SVR, KNN-R, ESOINN (λ=20), and DB-SOINN-R (λ=20). In September, all models have similar MAPE and CVRMSE in both scenarios, with less than 0.5% and 1% differences in MAPE and CVRMSE, respectively. This is because the September data is similar to August data that is used to train the models. Hence, updating the model in *Scenario B* does not show much difference for September. The models have a similar MAPE of around 8%, except the ESOINN that has the highest MAPE of 9.2% and 8.7% in *Scenario A* and *Scenario B*, respectively. All models also have similar CVRMSE of around 10% - 11% in September, except the LSTM model that has the highest CVRMSE of 12.09% in *Scenario B*. The LSTM is the only model that has worse prediction accuracy in *Scenario B* than in *Scenario A* in September.

In October, all models except FFNN show significant differences in MAPE and CVRMSE between the two scenarios. The FFNN has 0.3% and 1.5% differences in MAPE and CVRMSE, respectively. It is insignificant compared to other models, such as LSTM has 1.8% and 3.3% differences in MAPE and CVRMSE, respectively, and DB-SOINN-R has 3% and 5.17% differences in MAPE and CVRMSE, respectively. This shows the accomplishable improvement in prediction accuracy with incremental learning.

In November, all models show significant differences in MAPE and CVRMSE between the two scenarios. However, the FFNN and LSTM have smaller differences compared to other models. In December, the differences in MAPE and CVRMSE between the two scenarios are smaller compared to October and November. The LSTM has the smallest differences in MAPE and CVRMSE, which are 0.4% and 2.2%, respectively.

Compared to other models, both FFNN and LSTM have smaller improvements in prediction accuracy with incremental learning, and they also have worse accuracy (in *Scenario B*) than other models across all months. The LSTM has worse accuracy in *Scenario B* in September, despite Table 4.4 shows LSTM has better overall accuracy in *Scenario B* than in *Scenario A*. The SVR, kNN-R, ESOINN, and DB-SOINN-R models have significantly better prediction accuracy in *Scenario B* than *Scenario A*. This indicates the incremental can improve the prediction accuracy for the appropriate models. The proposed DB-SOINN-R has the best prediction accuracy among all models

in all months. In contrast, the original ESOINN has the highest MAPE and CVRMSE in *Scenario B* among the SVR, kNN-R, ESOINN, and DB-SOINN-R for all months. This shows the effectiveness of the proposed changes in the DB-SOINN-R for day-ahead load prediction.

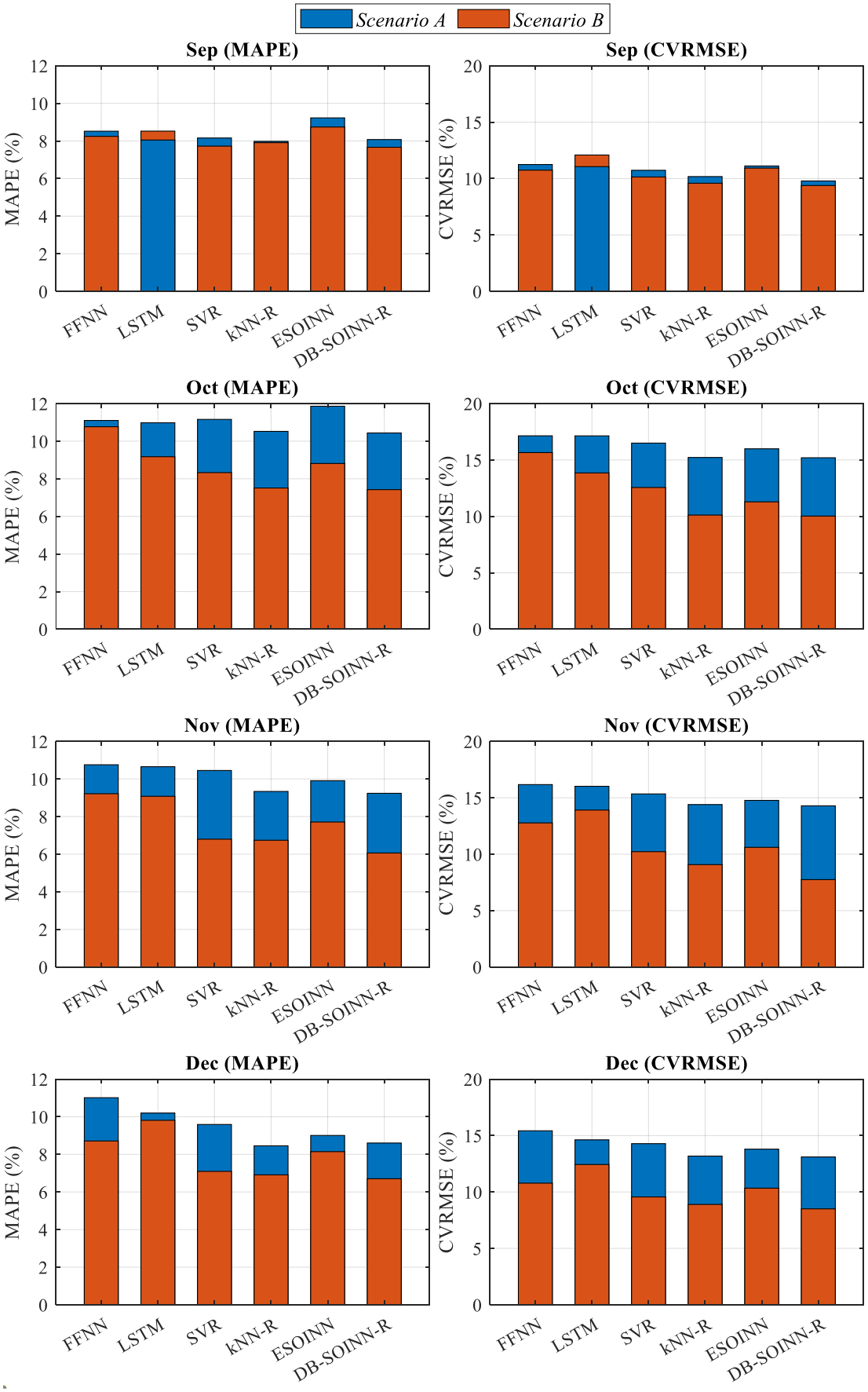


Figure 4.4: Monthly MAPE and CVRMSE for day-ahead models on Dataset A

Figure 4.5 provides the data for the daily MAPE for the six day-ahead models: FFNN, LSTM, SVR, KNN-R, ESOINN (λ=20), and DB-SOINN-R (λ=20). All models have similar MAPE between the two scenarios in September, which has been observed in Figure 4.4. From the end of September onwards, all models have their MAPE begin to increase for *Scenario A*. With incremental learning in *Scenario B*, the SVR, kNN-R, ESOINN, and DB-SOINN-R achieved a lower MAPE than in *Scenario A*. For these four models, their daily MAPE values in *Scenario B* are within the range of 5% - 10%, which is smaller than the range of 5% - 15% in *Scenario A*. In terms of 25th percentile (Q1), 75th percentile (Q3), and interquartile range (IQR) for the MAPE, the incremental learning did help to reduce these values. For example, the proposed DB-SOINN-R has Q1 of 7.586%, Q3 of 10.698%, and IQR of 3.113% in *Scenario A*. With incremental learning in *Scenario B*, the Q1, Q3, and IQR are reduced to 5.534%, 8.232%, and 2.697%, respectively. This shows a reduction of 2.054%, 2.466%, and 0.416% in Q1, Q3, and IQR, respectively. The lower Q1 and Q3 indicates higher prediction accuracy, whereas the smaller IQR indicates a more consistent prediction accuracy.

The FFNN and LSTM behave differently compared to the other four models. In September and October, the FFNN model has similar daily MAPE in both scenarios, and the daily MAPE in *Scenario B* begins to be lower than in *Scenario A* from November onwards. The LSTM model has its daily MAPE fluctuates between 5.7% and 15.8% between days 34 and 53 in *Scenario B*. This shows the LSTM model is unstable with incremental learning or daily retraining. Hence, retraining the FFNN and LSTM daily to achieve incremental learning is inappropriate.

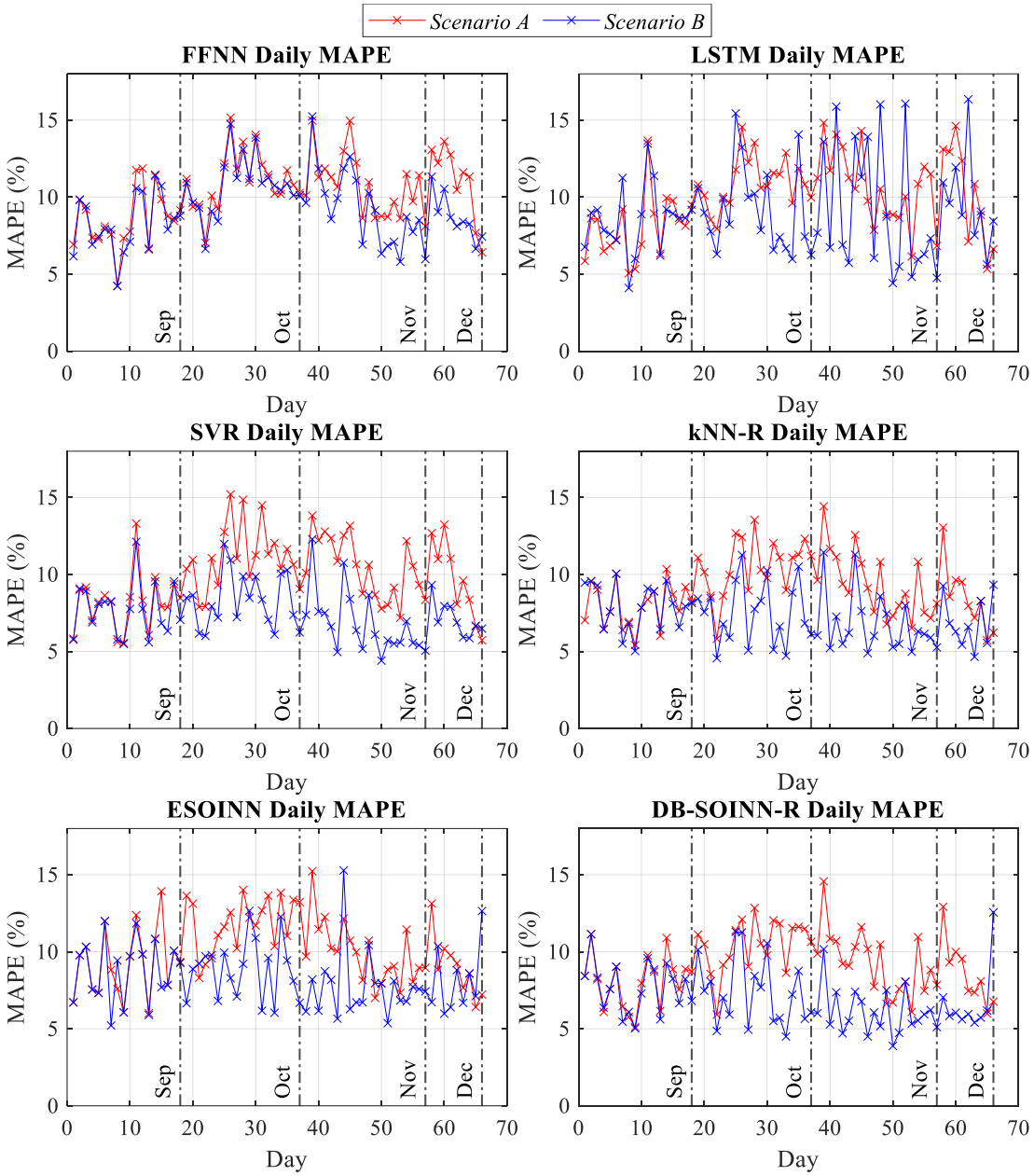


Figure 4.5: Daily MAPE for day-ahead models on Dataset A

4.4.3 One-hour-ahead Load Prediction Accuracy

Table 4.5 shows the prediction accuracy of the one-hour-ahead models in *Scenario A* and *Scenario B*. In *Scenario A*, the FFNN model performs the best among the six models, while the DB-SOINN-R model performs the best in *Scenario B*. Both FFNN and LSTM models have their accuracy metrics in *Scenario B* worse than *Scenario A*. For the SVR, kNN-R, ESOINN, and DB-SOINN-R models, they all have their prediction accuracy improved with incremental learning applied in *Scenario B*.

Table 4.5: Prediction accuracy of one-hour-ahead models in Scenario A and Scenario B on Dataset A

Models	***Scenario A* (without incremental learning)**					***Scenario B* (with incremental learning)**				
	MAPE (%)	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
FFNN	9.355	4.573	13.829	3.316	0.878	8.281	3.972	12.014	2.777	0.889
LSTM	8.710	4.309	13.031	3.095	0.913	9.658	4.203	12.711	2.990	0.867
SVR	7.478	4.551	13.765	2.812	0.873	5.665	2.851	8.623	1.939	0.941
kNN-R	7.715	3.997	12.087	2.665	0.911	5.515	2.758	8.341	1.881	0.945
ESOINN, *λ=2880*	7.956	4.139	12.517	2.846	0.883	6.703	3.301	9.984	2.266	0.918
DB-SOINN-R, *λ=2880*	6.60	3.510	10.615	2.398	0.937	5.049	2.379	7.194	1.691	0.959

The monthly MAPE and CVRMSE for the six one-hour-ahead models: FFNN, LSTM, SVR, kNN-R, ESOINN (λ=2880), and DB-SOINN-R (λ=2880) are provided in Figure 4.6. The SVR, kNN-R, ESOINN, and DB-SOINN-R one-hour-ahead models show a similar monthly trend their day-ahead models. In September, the differences in MAPE and CVRMSE between scenarios are less than 0.2% and 0.4%, respectively. In October, November, and December, they show significant differences in MAPE and CVRMSE between the two scenarios. The DB-SOINN-R model has the best accuracy, whereas the ESOINN model has the worst accuracy among the four models. This shows the effectiveness of the proposed changes in the DB-SOINN-R for one-hour-ahead load prediction.

The FFNN and LSTM models show significantly worse accuracy than other models in all four months, even with incremental learning in *Scenario B*. Although the FFNN model has a lower MAPE and CVRMSE with incremental learning in *Scenario B* than in *Scenario A*, they are still higher compared to SVR, kNN-R, EOSINN, and DB-SOINN-R models. The LSTM model shows worse monthly MAPE in *Scenario B* than in *Scenario A* for September, October, and November. In December, LSTM models show a negligible 0.1% lower MAPE and 1.59% lower CVRMSE in *Scenario B* than in *Scenario A*. Although LSTM has higher CVRMSE in *Scenario B* than in *Scenario A* for October, November, and December, it still insufficient to justify the implementation of incremental learning for LSTM one-hour-ahead load prediction. Hence, the LSTM one-hour-ahead model is not appropriate for incremental learning or retraining.

Figure 4.7 shows the daily MAPE for the six one-hour-ahead models: FFNN, LSTM, SVR, kNN-R, ESOINN (λ=2880), and DB-SOINN-R (λ=2880). The FFNN model shows slightly lower MAPE in Scenario B than in Scenario A, but it fluctuates between days 31 and 39. The LSTM model has its daily MAPE in *Scenario B* consistently worse than in *Scenario A*, which has observed in Figure 4.6. Moreover, it has daily MAPE fluctuation between days 1 and 40. The LSTM day-ahead model has a similar issue, as shown in Figure 4.5. The LSTM has an unstable model after incremental learning is implemented in both day-ahead and one-hour-ahead load predictions.

For the SVR, kNN-R, ESOINN, and DB-SOINN-R models, their daily MAPE values in *Scenario B* are lower than in *Scenario A* from October onwards. This demonstrates a better prediction accuracy is achievable with incremental learning or retraining. These four models have most of their daily MAPE values in *Scenario A* within the range of 5% - 13%. The range is reduced to 4% - 8% in *Scenario B*. For example, the DB-SOINN-R in *Scenario A* has its daily MAPE values within the range of 4.2% - 10.1%. The range reduced to 3.3% - 7.9% in *Scenario B*, with most MAPE at around 5% consistently. In terms of Q1, Q3, and IQR for the MAPE, the proposed DB-SOINN-R has Q1, Q3, and IQR values of 5.427%, 7.767%, and 2.340%, respectively, in *Scenario A*. With incremental learning in *Scenario B*, the Q1, Q3, and IQR reduced to 4.404%, 5.373%, and 0.969%, respectively. This shows differences of 1.023%, 2.394%, and 1.371% in Q1, Q3, and IQR, respectively. The lower Q1 and Q3 means the prediction accuracy is better, whereas the smaller IQR indicates a more consistent prediction accuracy with incremental learning, which is also observed in day-ahead load prediction.

The poor performance of the FFNN and LSTM one-hour-ahead models in *Scenario B* should be due to the inappropriate configuration to handle the larger number of training data for the one-hour-ahead load prediction than the day-ahead load prediction. Two days of load profile are processed to one training set for day-ahead models, but 576 training sets for one-hour-ahead models. As the number of training data increases, the existing network configuration of the FFNN and LSTM may be inappropriate to map the relationship, since the network structure and hyperparameters are unchanged during the daily retraining. Optimizing the network structure and hyperparameters daily during the retraining process may improve the performance, but it would take a longer time.

Another possibility of the poor performance of the FFNN and LSTM models is the lack of different types of variables such as weather data and building-related data. Other studies that use FFNN and LSTM models in load prediction are often paired with different types of data. However, this research work does not consider these data due to the absence of data. Also, other studies usually use more than one year of data. Including more data may improve the accuracy of the FFNN and LSTM models, but a large amount of historical data is not always available. In conclusion, the re-training

approach for FFNN and LSTM one-hour-ahead models to achieve incremental learning is inappropriate.

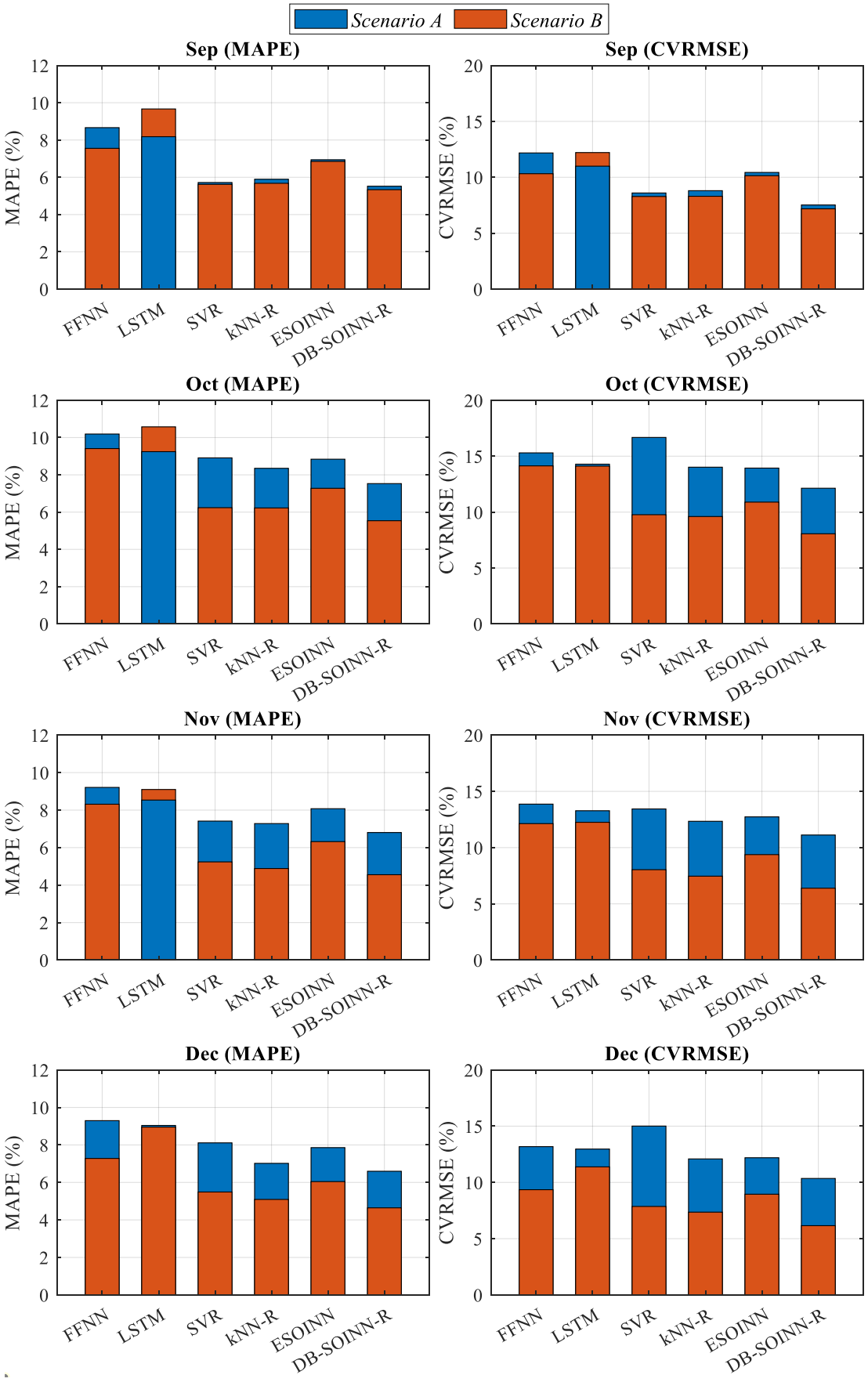


Figure 4.6: Monthly MAPE and CVRMSE for one-hour-ahead models on *Dataset A*

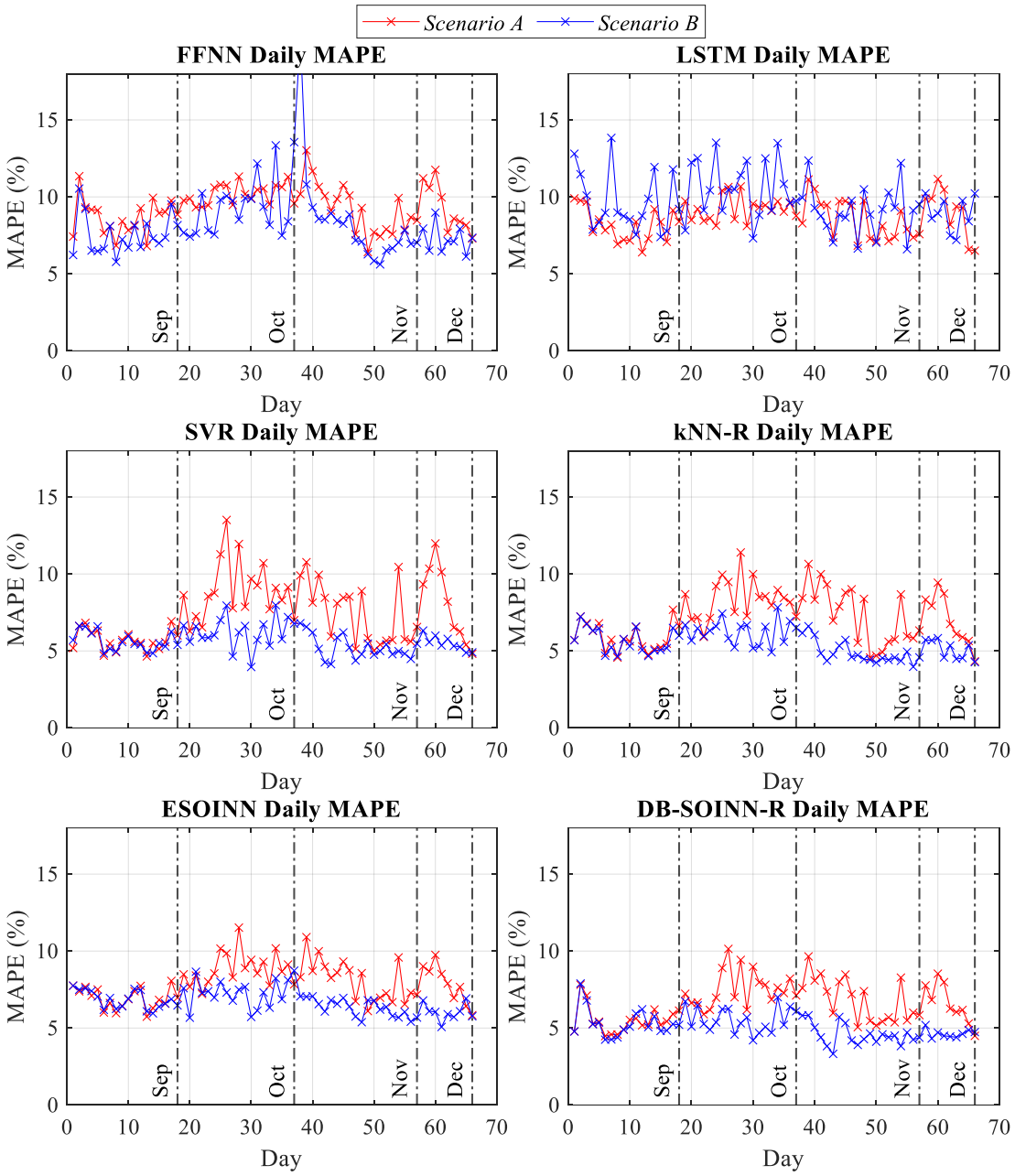


Figure 4.7: Daily MAPE for one-hour-ahead models on Dataset A

4.4.4 Execution Times

Table 4.6 and Table 4.7 show the execution times recorded in *Scenario B* for the day-ahead models and one-hour-ahead models, respectively. Among the day-ahead models, the SVR has the longest time as expected because 288 SVR models need to be trained and retrained daily throughout the testing period. For the kNN-R, it has no training time as no learning process is involved. For the ESOINN and DB-SOINN-R models, their unsupervised learning makes them notably faster than other day-ahead supervised models (FFNN, LSTM, and SVR) at both training and prediction.

Despite the fast execution times of the unsupervised day-ahead models, both FFNN and SVR models completed their initial training up to 11 times faster than the unsupervised ESOINN and DB-SOINN-R models, showing that the training times of the FFNN and SVR models are not affected by the number of training sets. However, the lack of incremental learning capability makes them slower in completing the daily re-training since the ESOINN and DB-SOINN-R models only need to update their model with the data collected on the day, while others need to retrain a new model with all collected data that includes data used to train the model in the previous day. The SVR one-hour-ahead model has a very fast total prediction time that is a mere 1.4571s slower than the original ESOINN model and 13.9047s faster than the DB-SOINN-R model. Although the FFNN one-hour model are retraining a new model daily while the DB-SOINN-R is updating its model, the retraining time of FFNN one-hour-ahead model is only 9.0403s slower than the second-fastest DB-SOINN-R model. The one-hour-ahead LSTM model has a much longer execution time than the day-ahead LSTM model. This shows that the training time is highly impacted by the number of training data. The DB-SOINN-R models do have a longer execution time than the original ESOINN model but are still faster than the three supervised models (FFNN, LSTM, and SVR).

Additionally, the three supervised models (FFNN, LSTM, and SVR) are trained using the built-in MATLAB function, which are compiled code while, the ESOINN and DB-SOINN-R are written in m-files that are interpreted. The m-files code should perform slower than the compiled built-in MATLAB function. With code optimization for speed, the execution speed of the ESOINN and DB-SOINN-R may be improved. For reference, the DB-SOINN-R has an average CPU usage of around 30%, while the SVR has an average CPU usage of around 70%, running at a quad-core CPU.

Table 4.6: Execution time for day-ahead models in Scenario B on Dataset A

Models	**Initial Train Time (s)**	**Average Daily Prediction Time (s)**	**Average Daily Training Time (s)**	**Total Daily Prediction Time (s)**	**Total Training Time (s)**	**Total Time (s)**
FFNN	4.519	0.0213	2.3789	1.407	156.997	162.923
LSTM	8.891	0.0047	7.475	0.313	493.371	502.575
SVR	108.893	0.200	148.436	13.226	9796.8	9918.919
kNN-R	-	0.000325	-	0.0215	-	0.0215
ESOINN, $\lambda=20$	0.143	0.000406	0.0018	0.0268	0.116	0.286
DB-SOINN-R, $\lambda=20$	0.207	0.000869	0.0024	0.0573	0.160	0.424

Table 4.7: Execution times for one-hour-ahead models in Scenario B on Dataset A

Models	**Initial Train Time (s)**	**Average Daily Prediction Time (s)**	**Average Daily Training Time (s)**	**Total Daily Prediction Time (s)**	**Total Training Time (s)**	**Total Time (s)**
FFNN	4.995	0.0109	2.376	206.339	156.783	368.116
LSTM	303.330	0.0035	982.976	66.806	64876	65246.14
SVR	4.484	0.000279	6.543	5.309	431.856	441.649
kNN-R	-	0.0025	-	48.026	-	48.026
ESOINN, $\lambda=2880$	19.946	0.000203	0.472	3.852	31.123	54.9201
DB-SOINN-R, $\lambda=2880$	49.714	0.0010	2.239	19.214	147.742	216.67

4.5 Detailed Analysis of the DB-SOINN-R model on *Dataset A*

As previously mentioned in Section 4.2, the proposed DB-SOINN-R model consists of three modifications to the original ESOINN: density-based denoising, mean Euclidean distance (*mEd*), and kNN-IDW regression. This section demonstrates the improvement in accuracy achieved by each proposed modification of the DB-SOINN-R model. The improvements by the changes at day-ahead load prediction are provided in Section 4.5.2. The mean Euclidean distance (*mEd*) is not tested in the day-ahead load prediction as there is only one type of data in the training set, which is the historical time-series load profile. The effect of the denoising interval λ on the prediction accuracy is also presented. For day-ahead models, their tested λ are 0, 5, and 20, which translates to no denoise, denoise every 5 days, and denoise every 20 days, respectively. Section 4.5.3 presents the improvements made by the changes at one-hour-ahead load prediction. For one-hour-ahead models, their tested λ are 0, 1440, and 2880, which translate to no denoise, denoise every 5 days, and denoise every 10 days, respectively. All models are only performed in *Scenario B*, in which the models are updated daily.

4.5.1 Grid-searched Parameters

Table 4.8 and Table 4.9 show the grid-searched parameters for the day-ahead and one-hour-ahead models in this section, respectively. There are different models that incorporate one or more of the three proposed changes to the original ESOINN, and the abbreviations for these models are shown below:

- ESOINN → Original ESOINN
- ESOINN-R → Original ESOINN with kNN-IDW regression
- EOSINN-mEd → Original ESOINN with *mEd* as the distance metric
- ESOINN-mEd-R → ESOINN-mEd with kNN-IDW regression.
- DB-SOINN-R → Proposed DB-SOINN that uses *mEd* as the distance metric, the new density-based denoising, and kNN-IDW regression.

Table 4.8: Grid-searched parameters for day-ahead models on Dataset A

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 6, Number of hidden neurons per layer = 72
LSTM	Number of LSTM layers = 1, Number of LSTM per layer = 24
SVR	*Parameters for all 288 models are optimized using built-in optimizer.*
kNN-R	k_{idw}=6
ESOINN, $\lambda=0$	-
ESOINN, $\lambda=5$	C_1=0, C_2=0
ESOINN, $\lambda=20$	C_1=0, C_2=0
ESOINN-R, $\lambda=0$	k_{idw}=6
ESOINN-R, $\lambda=5$	C_1=0, C_2=0, k_{idw}=6
ESOINN-R, $\lambda=20$	C_1=0, C_2=0, k_{idw}=6
DB-SOINN-R, $\lambda=0$	k_{idw}=6
DB-SOINN-R, $\lambda=5$	k_{idw}=6, $k_{denoise}$=2
DB-SOINN-R, $\lambda=20$	k_{idw}=6, $k_{denoise}$=2

Table 4.9: Grid-searched parameters for one-hour-ahead models on Dataset A

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 5, Number of neurons per layer = 6
LSTM	Number of LSTM layers = 3, Number of LSTM per layer = 9
SVR	Epsilon = 0.000744, KernelScale=0.262891, Lambda=0.0000339
kNN-R	k_{idw}=34
ESOINN, $\lambda=0$	-
ESOINN, $\lambda=1440$	C_1=1, C_2=0.5
ESOINN, $\lambda=2880$	C_1=1, C_2=1
ESOINN-R, $\lambda=0$	k_{idw}=28
ESOINN-R, $\lambda=1440$	C_1=0.3, C_2=0.3, k_{idw}=28
ESOINN-R, $\lambda=2880$	C_1=0.7, C_2=0.5, k_{idw}=20
ESOINN-mEd, $\lambda=0$	-
ESOINN-mEd, $\lambda=1440$	C_1=0.7, C_2=0.7
ESOINN-mEd, $\lambda=2880$	C_1=0.7, C_2=0.7
ESOINN-mEd-R, $\lambda=0$	k_{idw}=12
ESOINN-mEd-R, $\lambda=1440$	C_1=0.5, C_2=1, k_{idw}=12
ESOINN-mEd-R, $\lambda=2880$	C_1=0.5, C_2=1, k_{idw}=12
DB-SOINN-R, $\lambda=0$	k_{idw}=12
DB-SOINN-R, $\lambda=1440$	k_{idw}=12, $k_{denoise}$=10
DB-SOINN-R, $\lambda=2880$	k_{idw}=12, $k_{denoise}$=10

4.5.2 Day-ahead Load Prediction

Table 4.10 shows the prediction accuracy of different variations of ESOINN day-ahead models, including original ESOINN and the proposed DB-SOINN-R models. It is seen from Table 4.10 that the original ESOINN model has the worst accuracy among all models, despite the better accuracy with $\lambda=20$. With the kNN-IDW regression added to the original ESOINN (ESOINN-R), it shows significant improvement in all metrics. However, the ESOINN-R with $\lambda=5$ shows 0.6% and 1.5% higher MAPE and CVRMSE, respectively, than the ESOINN-R with $\lambda=0$ (no denoise). This shows using the original denoising in the ESOINN may decrease the prediction accuracy. With the new density-based denoising, the DB-SOINN-R model with $\lambda=5$ and 20, has better accuracy than the DB-SOINN-R model with $\lambda=0$. Unlike the original denoising in ESOINN that may poses worse accuracy, enabling the new density-based denoising in the DB-SOINN-R models brings improvement to the prediction accuracy, which is the expected behavior by an appropriate denoising process.

Table 4.10: Prediction accuracy of different ESOINN day-ahead models in Scenario B on Dataset A

Models	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
ESOINN, $\lambda=0$	8.519	3.639	11.007	2.776	0.900
ESOINN, $\lambda=5$	8.503	3.813	11.533	2.833	0.896
ESOINN, $\lambda=20$	8.369	3.591	10.860	2.728	0.903
ESOINN-R, $\lambda=0$	7.140	3.075	9.299	2.344	0.934
ESOINN-R, $\lambda=5$	7.739	3.574	10.808	2.633	0.923
ESOINN-R, $\lambda=20$	7.164	3.107	9.397	2.356	0.935
DB-SOINN-R, $\lambda=0$	7.120	3.077	9.305	2.347	0.934
DB-SOINN-R, $\lambda=5$	7.072	2.963	8.961	2.285	0.936
DB-SOINN-R, $\lambda=20$	6.986	2.995	9.058	2.286	0.936

4.5.3 One-hour-ahead Load Prediction

Table 4.11 provides the data for the prediction accuracy of different variations of ESOINN one-hour-ahead models, including original ESOINN and the proposed DB-SOINN-R models. As given in Table 4.11, the inclusion of kNN-IDW regression to the one-hour-ahead ESOINN models (ESOINN-R) brings significant improvement to the prediction accuracy in all metrics. Then, the use of the *mEd* (ESOINN-mEd-R) further improves the accuracy in all metrics. Figure 4.8 depicts the effect of *mEd* on the selected nodes for kNN-IDW regression.

As seen in Figure 4.8(a), the predicted profile of ESOINN-R model tends to lag the actual profile. This is because the ESOINN selected inappropriate nodes that represents data of different minute for kNN-IDW. Referring to the example given in Figure 4.8(b), the ESOINN-R uses nodes with a minute of the day from as far as 505^{th} minute when predicting the one-hour-ahead demand at 575^{th} minute. There is 70 minutes difference. With the implementation of *mEd* in DB-SOINN-R, it selected nodes with a minute of day difference of 5 or 10 minutes for kNN-IDW, eliminating the behavior of predicted profile lags actual profile, thus improving the prediction accuracy.

For the ESOINN-mEd-R with $\lambda=1440$, the prediction accuracy is less than the ESOINN-mEd-R with $\lambda=0$ in all metrics. This shows the original denoising in the ESOINN may cause a poor prediction accuracy after the implementation of kNN-IDW regression. With the new density-based denoising, the DB-SOINN-R with $\lambda=1440$ and *2880* show better prediction accuracy than the DB-SOINN-R with $\lambda=0$ (no denoise), which is the expected outcome of appropriate denoising.

Table 4.11: Prediction accuracy of different one-hour-ahead models in Scenario B on Dataset A

Models	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
ESOINN, *λ=0*	7.138	3.506	10.603	2.413	0.908
ESOINN, *λ=1440*	6.813	3.357	10.153	2.309	0.915
ESOINN, *λ=2880*	6.703	3.301	9.984	2.266	0.918
ESOINN-R, *λ=0*	5.654	2.843	8.600	1.933	0.942
ESOINN-R, *λ=1440*	5.672	2.846	8.608	1.940	0.942
ESOINN-R, *λ=2880*	5.663	2.814	8.511	1.930	0.942
ESOINN-mEd-R, *λ=0*	5.147	2.442	7.386	1.731	0.958
ESOINN-mEd-R, *λ=1440*	5.376	2.477	7.491	1.793	0.957
ESOINN-mEd-R, *λ=2880*	5.155	2.414	7.300	1.726	0.959
DB-SOINN-R, *λ=0*	5.167	2.417	7.311	1.731	0.958
DB-SOINN-R, *λ=1440*	5.041	2.381	7.200	1.693	0.959
DB-SOINN-R, *λ=2880*	5.049	2.379	7.194	1.691	0.959

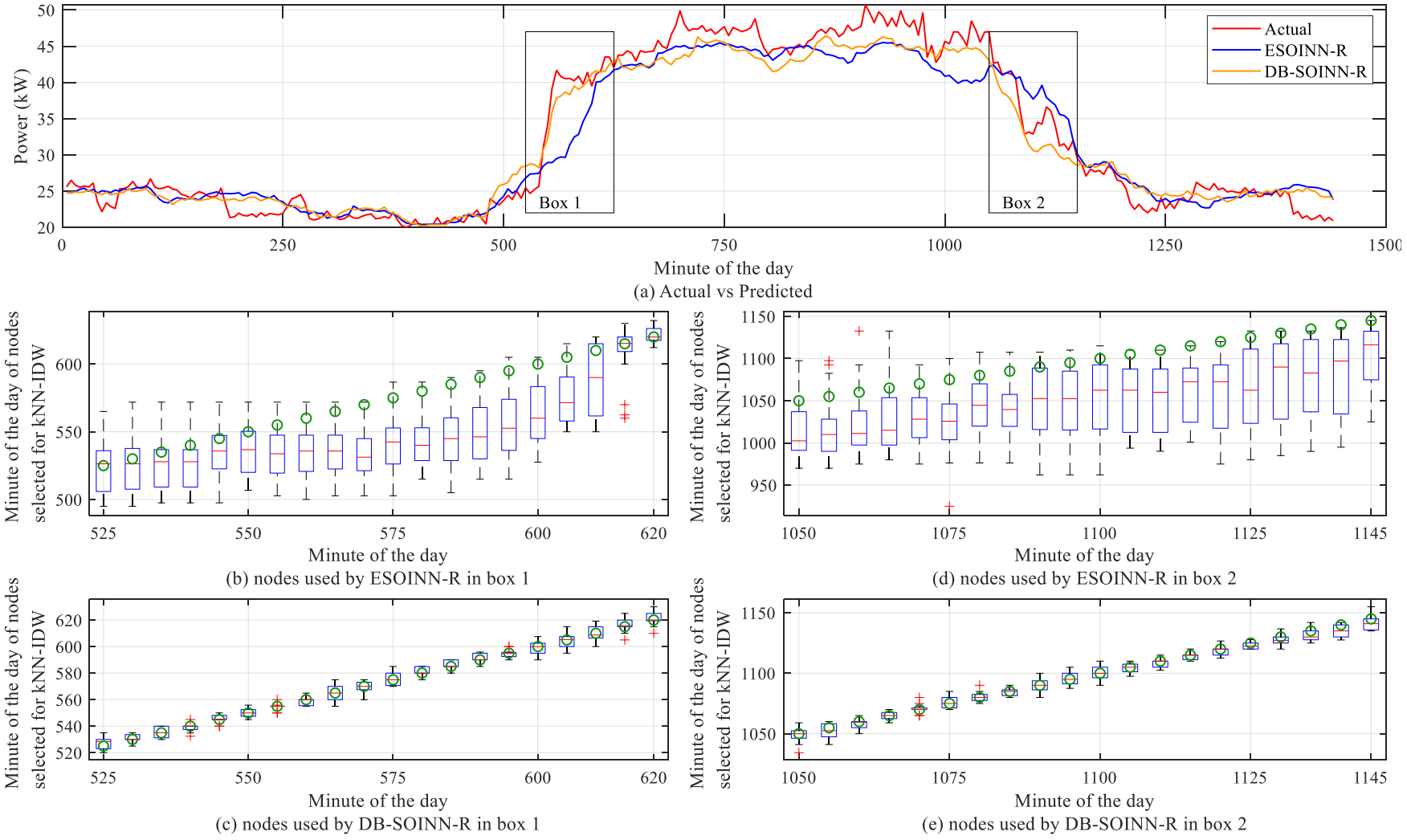


Figure 4.8: Investigation on the effect of mean Euclidean distance

4.6 Results and Discussion for *Dataset B*

This section presents the results and discussion of six models: FFNN, LSTM, SVR, kNN-R, ESOINN, and DB-SOINN-R for *Dataset B*. New parameters are grid-searched for *Dataset B* and are provided in Section 4.6.1. Section 4.6.2 and 4.6.3 discuss the prediction accuracies of the day-ahead and one-hour-ahead models for *Dataset B*, respectively.

4.6.1 Grid-searched Parameters

Table 4.12 and Table 4.13 show the grid-searched parameters for the day-ahead and one-hour-ahead models, respectively.

Table 4.12: Grid-searched parameters for day-ahead models on Dataset B

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 7, Number of hidden neurons per layer = 256
LSTM	Number of LSTM layers = 2, Number of LSTM per layer = 208
SVR	*Parameters for all 288 models are optimized using built-in optimizer.*
kNN-R	k_{idw}=4
ESOINN, λ=*20*	C_1=0.3, C_2=1
DB-SOINN-R, λ=*20*	k_{idw}=4, $k_{denoise}$=8

Table 4.13: Grid-searched parameters for one-hour-ahead models on Dataset B

Models	**Grid-searched parameters**
FFNN	Number of hidden layers = 4, Number of neurons per layer = 15
LSTM	Number of LSTM layers = 2, Number of LSTM per layer = 24
SVR	Epsilon = 0.0050513, KernelScale=2.2315, Lambda=0.0000272
kNN-R	k_{idw}=2
ESOINN, λ=*2880*	C_1=0.5, C_2=0.5
DB-SOINN-R, λ=*2880*	k_{idw}=8, $k_{denoise}$=40

4.6.2 Day-ahead Load Prediction Accuracy

Table 4.14 presents the prediction accuracy of the day-ahead models obtained using the public *Dataset B.* In *Scenario A*, both SVR and DB-SOINN-R perform similarly well and have better accuracy than other models. The SVR has 0.23% lower MAPE and 0.063kW lower MAE than the DB-SOINN-R, whereas the DB-SOINN-R has 0.012kW lower RMSE, 0.11% lower CVRMSE, and 0.002 higher R^2 than the SVR. Hence, they are considered comparable in *Scenario A.*

With incremental learning in *Scenario B*, all models have better prediction accuracy in all metrics, except LSTM. The LSTM shows 0.841% lower CVRMSE and 0.17kW lower MAE, but 0.086% higher MAPE in *Scenario B* than in *Scenario A*. Hence, no significant differences for LSTM. Other models have significant improvements, such as the DB-SOINN-R has 1.351% lower MAPE and 1.207% lower CVRMSE in *Scenario B* than in *Scenario A.* With incremental learning in *Scenario B*, the DB-SOINN-R achieved the best prediction accuracy among all models. The DB-SOINN-R also shows better accuracy than ESOINN in all metrics, especially in *Scenario B,* demonstrating the consequence of proposed changes in DB-SOINN-R.

Table 4.14: Prediction accuracy of day-ahead models in *Scenario A* and *Scenario B* on *Dataset B*

Models	***Scenario A* (without incremental learning)**					***Scenario B* (with incremental learning)**				
	MAPE (%)	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
FFNN	7.222	4.090	8.656	3.298	0.962	4.309	2.532	5.359	1.888	0.973
LSTM	5.172	3.303	6.991	2.508	0.974	5.258	2.906	6.150	2.338	0.973
SVR	3.991	2.427	5.136	1.818	0.978	2.989	1.994	4.220	1.372	0.983
kNN-R	4.333	2.388	5.054	1.930	0.980	2.954	1.818	3.849	1.340	0.986
ESOINN, $\lambda=20$	4.233	2.392	5.063	1.862	0.979	3.324	2.029	4.294	1.495	0.983
DB-SOINN-R, $\lambda=20$	4.221	2.376	5.029	1.881	0.980	2.870	1.806	3.822	1.309	0.986

4.6.3 One-hour-ahead Load Prediction Accuracy

The prediction accuracy of the one-hour-ahead models obtained using the public *Dataset B* are given in Table 4.15. Similar to the results for *Dataset A* in section 4.4.3, the FFNN and LSTM one-hour-ahead models have considerably worse accuracy compared to other models. The FFNN has the lowest accuracy in *Scenario A*, while the LSTM has the worst accuracy in *Scenario B*. The LSTM also has a lower accuracy in *Scenario B* than in *Scenario A*. This again indicates that the FFNN and LSTM are not suitable to achieve incremental learning through retraining daily.

In *Scenario A*, the DB-SOINN-R has the best prediction accuracy among all models. Although the kNN-R has a mere 0.026% lower MAPE than the DB-SOINN-R, DB-SOINN-R performs better in other metrics. All models have better prediction accuracy in *Scenario B* than in *Scenario A*, except LSTM. In *Scenario B*, the proposed DB-SOINN-R performs the best among all models in all metrics. The proposed changes in DB-SOINN-R have led to a better accuracy than the ESOINN in both scenarios.

Table 4.15: Prediction accuracy of one-hour-ahead models in *Scenario A* and *Scenario B* on *Dataset B*

Models	***Scenario A* (without incremental learning)**					***Scenario B* (with incremental learning)**				
	MAPE (%)	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**	**MAPE (%)**	**RMSE (kW)**	**CVRMSE (%)**	**MAE (kW)**	**R^2**
FFNN	4.030	2.339	4.950	1.812	0.977	2.912	1.816	3.843	1.328	0.986
LSTM	3.440	1.890	4.001	1.495	0.990	4.001	2.396	5.072	1.639	0.976
SVR	2.219	1.343	2.842	0.999	0.993	1.767	1.169	2.475	0.821	0.994
kNN-R	2.171	1.438	3.045	1.023	0.992	1.448	1.001	2.118	0.680	0.996
ESOINN, *λ=2880*	2.448	1.535	3.248	1.109	0.990	1.851	1.327	2.809	0.887	0.993
DB-SOINN-R, *λ=2880*	2.197	1.327	2.808	1.005	0.993	1.438	0.979	2.071	0.667	0.996

4.7 Summary

This chapter evaluates the effect of incremental learning on six models: FFNN, LSTM, SVR, kNN-R, ESOINN, and the proposed DB-SOINN-R, on both day-ahead and one-hour-ahead load predictions. This chapter proposes a new DB-SOINN-R model that incorporates a new density-based denoising, a new mean Euclidean distance as the distance metric, and kNN-IDW regression. To evaluate the effect of incremental learning, the models were trained with the first two months of data and tested on the remaining data, with the models updated or retrained daily throughout the testing period. Results are obtained using two different datasets, *Dataset A* and *Dataset B*. *Dataset A* is a load profile from an educational building, and *Dataset B* is a public load profile of the city of Johor in Malaysia. Results show that all models could achieve better prediction accuracy from incremental learning, except FFNN and LSTM models. The DB-SOINN-R outperforms the original ESOINN in the two datasets. Without the proposed changes in the DB-SOINN-R, the ESOINN model would not be considered as a viable model since other models such as SVR often have better accuracy at both day-ahead and one-hour-ahead load predictions. The DB-SOINN-R incremental model has the highest prediction accuracy among all models for both day-ahead and one-hour-ahead load predictions in both *Dataset A* and *Dataset B*. This shows the proposed model has good generalization. The execution times of the DB-SOINN-R day-ahead and one-hour-ahead models are also faster than the supervised FFNN, LSTM, and SVR models. Therefore, the DB-SOINN-R is considered as a viable incremental model for day-ahead and one-hour-ahead time-series load predictions.

Chapter 5 – Two-stage Maximum Demand Reduction Controller

5.1 Overview

Conventional threshold controllers that use load prediction have their peak shaving performance easily affected by the prediction accuracy of the load prediction models. Since they use predicted profiles to optimize the threshold demand (P_{Th}), an inaccurate load profile may lead to inappropriate P_{Th}, causing low or no MD reduction. To alleviate this issue, a new incremental prediction model was proposed in Chapter 4. It is interesting to note that the model specifically works with small historical data. Although the results demonstrated that the model has better prediction accuracy in comparison with other models, it does not eliminate prediction error. One day of predicted load profile with poor accuracy can cause peak demand reduction failure that may abolish the effort of peak reduction for the month. Therefore, this chapter proposes a novel dynamic two-stage MD reduction controller. Compared to other controllers, the novel controller is better at handling various prediction errors and preventing peak demand reduction failure for achieving a higher MD reduction. The proposed controller is designed to be adaptive, so it requires minimal amount of historical data to begin operation while achieving a reliable MD reduction performance.

In this chapter, the proposed two-stage controller is compared with three controllers, named fixed threshold [32], single-stage [35], and fuzzy controller [36]. All controllers, except fixed threshold controller, require a prediction model to perform the load prediction. The prediction model proposed in Chapter 4 is selected as the prediction model in this chapter. Since the evaluation of the DB-SOINN-R has been completed and presented in Chapter 4, this chapter focuses on evaluating the capability of the proposed controller at MD reduction and peak demand reduction failure prevention capability. To demonstrate its adaptiveness, the proposed two-stage controller has only 30 working days of historical data before MD reduction operations of up to ten months and are tested on the two datasets.

The structure of the proposed controller is first presented, following is the elaboration of the simulation setup used for evaluating the proposed controller, with results and discussion provided subsequently. The findings of this chapter are summarized in the last section.

5.2 Structure of the Two-stage Maximum Demand Reduction Controller

To tackle the problem of reliance on the accuracy of load prediction, some state-of-the-art controllers use different rigid parameters to reduce unnecessary BESS outputs caused by prediction error, thus prolonging the operational peak reduction period. The use of rigid parameters causes them incapable to adaptive to future variations. Additionally, a large amount of historical data to predetermine the optimal rigid parameters of the controller since they are responsible for preventing peak reduction failure. It is impractical to stagnant the implementation of the controller until the collection of sufficient historical data and validation of the parameters. This may cause a decline in users' interest and reduces the demand response participation rate. Thus, rendering these controllers infeasible for sites with no or little historical data.

To tackle the above-mentioned problem, a novel dynamic two-stage controller is proposed in this chapter. The proposed two-stage controller has three parts: prediction models, two-stage control strategy, and reserve BESS capacity. Unlike conventional controllers that use day-ahead load profiles, the two-stage controller uses two load predictions with lead times of day-ahead and one-hour-ahead. The two-stage controller adjusts the P_{Th} every 5-min. The first stage determines the threshold based on the predicted day-ahead load profile to achieve the highest possible peak reduction with the remaining BESS energy. The second stage refines the threshold based on the predicted one-hour-ahead load profile to prevent peak demand reduction failure if necessary. To handle unexpected energy usage due to unusual profiles, a small BESS is reserved.

Open literature reveals that no research is using load prediction with lead time shorter than day-ahead for peak demand reduction of buildings. Although the one-hour-ahead load prediction is applicable for other energy management tasks such as load smoothing [100], it is not used for daily peak demand reduction. With the shorter lead time of the one-hour-ahead load prediction than the day-ahead load prediction, it should be more accurate than the day-ahead load prediction. Moreover, the one-hour-ahead load prediction is constantly predicting the one hour ahead load demand every 5-min, the inclusive of latest load inputs from the predicting day allows a more consistent accuracy where there is a smaller chance of high prediction error that persists throughout the whole day.

The addition of one-hour-ahead load prediction in the two-stage controller enables it to prevent peak demand reduction failure dynamically with real-time data. It does not need to pre-determine rigid parameters that can significantly affect the probability of peak reduction failure if not selected appropriately. Thus, it needs no long-term historical data for analysis at the time of implementation. Due to its dynamic nature, the controller can adapt to load pattern variations automatically and can be easily implemented for different buildings with limited historical data at the time of implementation. This easy-to-implement controller can potentially boost consumers' DR participation.

5.2.1 Prediction Model

Figure 5.1(a) and (b) show the inputs/outputs of the day-ahead and one-hour-ahead prediction models, respectively. The inputs to the day-ahead load profile prediction model are the load profile of today ($P_{00:00}$ to $P_{23:55}$ of Day d), while the outputs are the predicted load profile of the next day ($P_{00:00}$ to $P_{23:55}$ of Day $d+1$). The inputs to the one-hour-ahead prediction model are the minute of the day and previous hour historical load demands (P_{t-55} to P_t), while the output is the one-hour-ahead load demand (P_{t+60}). The day-ahead load prediction model predicts the next day load profile once at the beginning of each day, whereas the one-hour-ahead load prediction model predicts the one-hour-ahead demand every 5-min. Both prediction models are developed using an incremental model named DB-SOINN-R that has been evaluated and presented in Chapter 4. The DB-SOINN-R has a better prediction accuracy than conventional models such as DNN and SVR, and its one-hour-ahead model has better accuracy than its day-ahead model due to its shorter lead time. Despite that, the prediction model may produce high prediction errors for certain extreme conditions that are undesirable for the controller. The incremental learning of DB-SOINN-R allows it to update its model daily with collected data. Combined with the dynamic two-stage control strategy, the proposed dynamic two-stage controller can be implemented with a small amount of historical data.

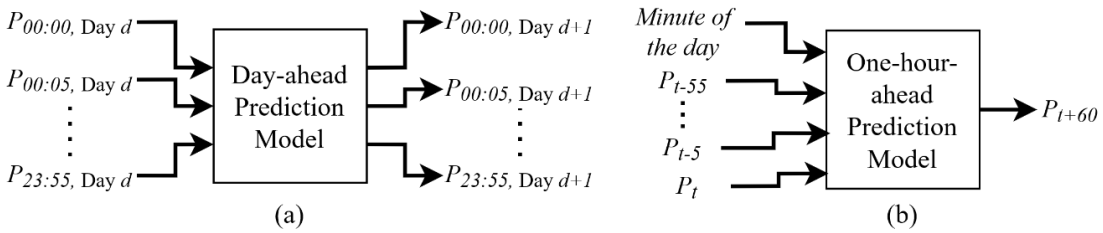


Figure 5.1: (a) Day-ahead load prediction model; (b) One-hour-ahead load prediction model

5.2.2 Interpolated One-hour-ahead Load Profile

The output of the one-hour-ahead prediction model has only one output as seen in Figure 5.1 (b), which is the predicted one-hour-ahead demand. There are two methods to get the next hour load profile. The first method (*Method A*) uses the predicted outputs from previous timesteps. For example, the current *t* is 09:00, and the current output of the one-hour-ahead model is the demand for *t+60* (10:00). The demands for 09:05 to 09:55 have already been predicted in previous timesteps when *t* is 08:05 to 08:55. The second method (*Method B*) is linear interpolating the demands between the latest load demand (P_t) and the predicted P_{t+60}. Figure 5.2(a) and Figure 5.2(b) depict the prediction error for one-hour-ahead predicted profile (*Method A*) and interpolated one-hour-ahead predicted profile (*Method B*), respectively. The amount of prediction error is represented by the shared green area. *Method B* has a significantly smaller prediction error or green shared area when compared to *Method A*. *Method B* has significantly less prediction error (green shaded area) than *Method A*. Hence, *Method B* is used in the two-stage control strategy to obtain the next hour predicted load profile.

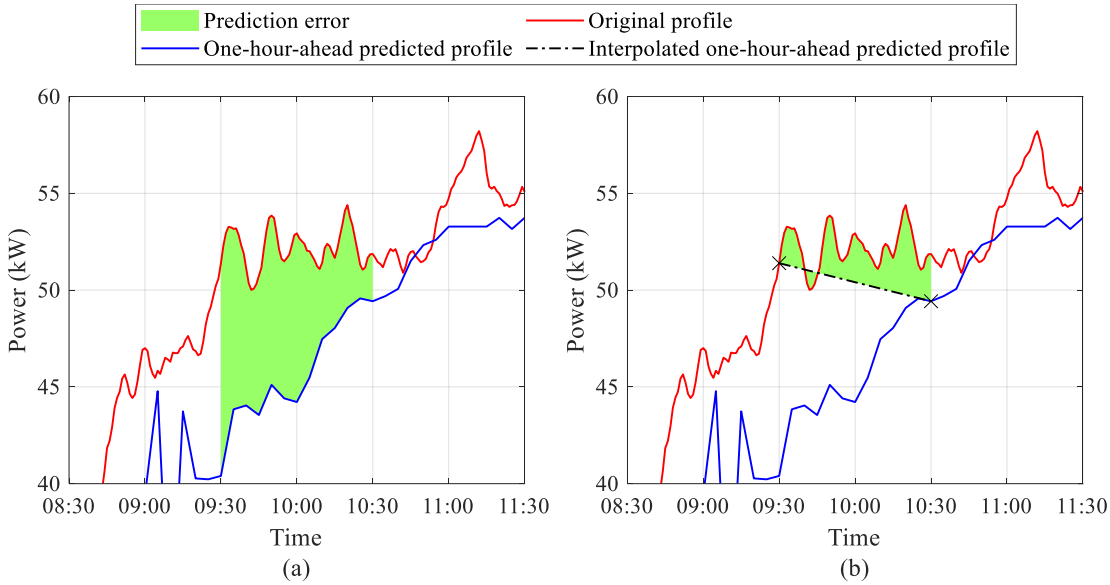


Figure 5.2: (a) Prediction errors for one-hour-ahead predicted profile (Method A) and (b) interpolated one-hour-ahead predicted profile (Method B)

5.2.3 Two-stage Control Strategy

Figure 5.3 shows the flowchart of the proposed two-stage control strategy. Table 5.1 shows the definitions of the variables used for the two-stage controller. Conventional single-stage controllers use the first stage only to find the P_{Th}. They do not have the BESS sufficiency checking and second stage threshold adjustment that use the interpolated next hour load profile. To use the short one-hour-ahead lead time for peak reduction, it utilizes the BESS discharge rate to estimate the energy usage beyond the one-hour lead time.

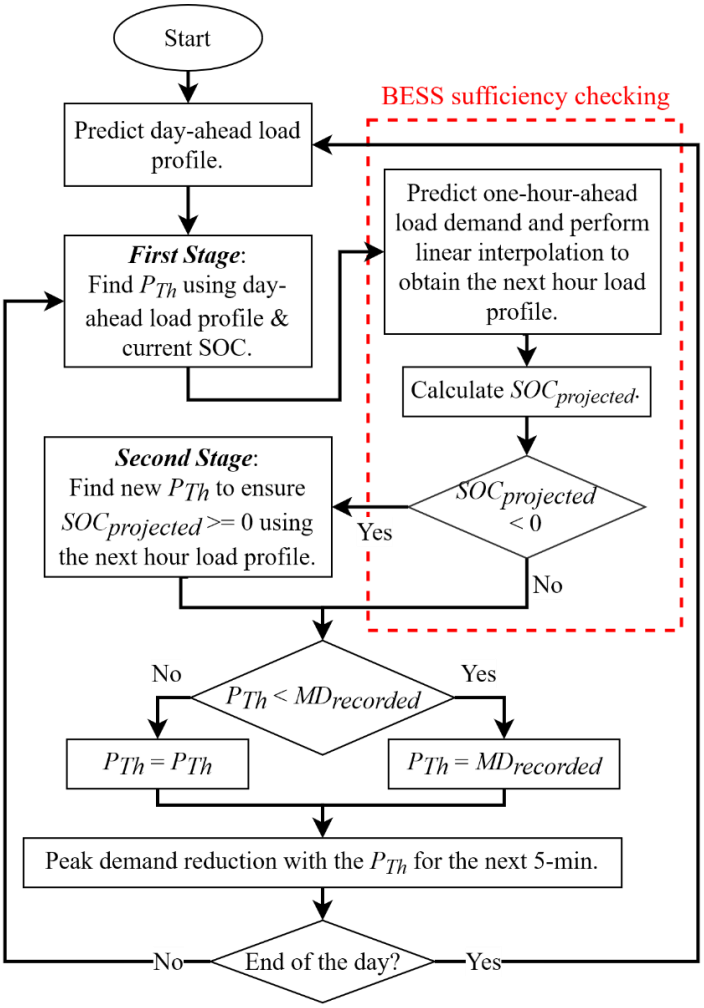


Figure 5.3: Flowchart of the two-stage control strategy

Table 5.1: Definition of variables used for the two-stage control

Variable	**Definition**
P_{Th}	Demand threshold
SOC	State-of-charge of ESS
ESS_{size}	Size of ESS
E_{used}	SOC energy used in demand reduction
$E_{projected}$	Projected energy usage for the next hour
t_{end}	Projected demand reduction end time (in minutes)
$SOC_{projected}$	Projected SOC of ESS at the t_{end}
$E_{projected}$	Projected energy usage of the next hour
$DisRate_{projected}$	Projected discharge rate of the ESS
y_p^i	The *i*-th point of the interpolated one hour-ahead predicted profile (y_p)
$MD_{recorded}$	The current MD of the day

Figure 5.4 shows an example of the operation of the two-stage controller at time *t*. The BESS sufficiency checking is done by checking the projected SOC of BESS at the projected peak demand reduction end time ($SOC_{projected}$). To get the $SOC_{projected}$, it first needs to find the BESS energy that has been used (E_{used}) and the projected energy usage of the next hour ($E_{projected}$) using equations (20) and (21), respectively.

$$E_{used} = BESS_{size} - \left(BESS_{size} \times SOC \right) \tag{20}$$

$$E_{projected} = \sum_{i=1}^{12} \left(y_{p,i} - P_{Th} \right) \times \left(\frac{5}{60} \right) \tag{21}$$

Figure 5.4 shows the E_{used} (blue area) that represents the used BESS energy up to the time *t* and the $E_{projected}$ (orange area) represents the energy between the interpolated next hour load profile and the P_{Th} from the first stage. The E_{used} and $E_{projected}$ are then used to calculate the projected discharge rate ($DisRate_{projected}$) of the BESS using equation (22).

$$DisRate_{projected} = \frac{E_{used} + E_{projected}}{\left(k + 12 \right) \times \frac{5}{60}} \tag{22}$$

where *k* is the number of intervals between the first demand reduction of the day and the current time *t*, as shown in Figure 5.4. For example, if the first demand reduction occurs at 09:00, and the current time is 11:00, the *k* is 24, for the 5-min data. Then, the $DisRate_{projected}$ is used to calculate the $SOC_{projected}$ using equation (23) with t_{end} referring to the projected demand reduction end time. The t_{end} is the time when the BESS is discharged completely based on the day-ahead predicted load profile with the initial P_{Th}, as indicated in Figure 5.4. The t_{end} is obtained using the day-ahead load profile since the one-hour-ahead lead time is too short. Since the two-stage controller only adjust the P_{Th} to remain the same or increase, the t_{end} may become earlier every 5-min if t_{end} is recalculated using the latest P_{th}. If the day-ahead prediction is inaccurate, an earlier t_{end} may cause a bigger error with the actual t_{end} that may result in peak demand reduction failure. Therefore, the maximum t_{end} is used to consider the worst-case scenario. The maximum t_{end} is calculated using the initial P_{Th} from the first stage (green line) that is found at the beginning of the day.

$$SOC_{projected} = ((BESS_{size} \times SOC) - E_{projected}) - \left(DisRate_{projected} \times \frac{(t_{end} - (k+12)) \times 5}{60} \right) \quad (23)$$

If the $SOC_{projected}$ is negative, it indicates that the current P_{Th} may not be achievable with the remaining BESS energy, thus peak demand reduction failure may occur. As shown in the example in Figure 5.4, the original projected SOC (brown line) based on the P_{Th} from the first stage shows a negative $SOC_{projected}$, which indicates more BESS energy may be required to successfully reduce the peak demand to P_{Th}. To prevent the potential peak demand reduction failure, the controller enters the second stage to find a new P_{Th} that can achieve a $SOC_{projected}$ of 0% (light blue line) with the next hour load profile. Increasing the P_{Th} lowers the $E_{projected}$ and $DisRate_{projected}$, thus closer to the desired $SOC_{projected}$, which is 0%.

The P_{Th} from the second stage is then checked to prevent non-beneficial demand reduction by restricting the P_{Th} from being lower than the $MD_{recorded}$. The $MD_{recorded}$ is the highest recorded 30-minute interval demand up to the current time of the day. For example, if the current time is 14:00, and the highest 30-minute interval demand of the day thus far is 40kW from 11:30 – 12:00, the $MD_{recorded}$ at 14:00 is 40kW. Any peak demands lower than 40kW will not contribute to lowering the MD charge.

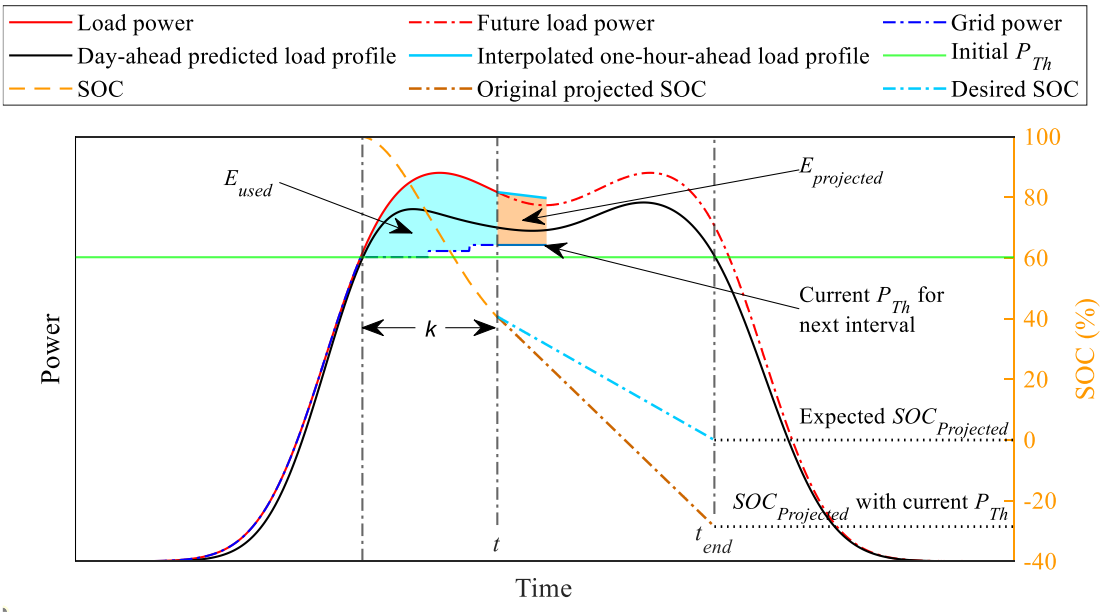


Figure 5.4: An example of the operation of the two-stage controller at time t

5.2.4 Reserved Battery Capacity

Since the t_{end} is obtained from the day-ahead predicted load profile, it can be incorrect if the day-ahead predicted profile is inaccurate. The error in t_{end} may cause BESS deficiency for demand reductions since the controller is adjusting the P_{Th} to have the BESS emptied before or at t_{end}. The example in Figure 5.4 shows the day-ahead predicted load profile (black line) leads the load power (red line). If the BESS empties at t_{end}, there will be no energy to reduce power demands that are higher than P_{Th} after t_{end}. To tackle this issue, the proposed two-stage controller reserves an extra BESS that is only used when the SOC is below 10% and remains idle if the SOC is higher than 10%. The error of t_{end} should be small unless abnormal profiles that have unusual event occurring after the typical peak period. Hence, reserving a small BESS should suffice the extra energy required by the error in t_{end}. The size of the reserved BESS is referred to as $BESS_{reserved}$ in the remaining part of this thesis.

5.3 Simulation Setup

The performance of the proposed controller is evaluated using MATLAB simulation and the results are presented in this chapter. The real-world evaluation of the developed controller will be provided in the next chapter to check if the controller is workable in real-time.

Conventionally, Simulink is used instead of MATLAB scripts. Simulink has ready-to-use models such as batteries and can construct electrical circuits to simulate the converter more realistically. However, incorporating everything into Simulink resulted in a very slow simulation time due to the lack of a high-performance computer. It takes more time to simulate than running it in real-world for the same period of data. Hence, accuracy is sacrificed for speed. Below shows the trade-offs of not using Simulink models or any realistic models:

1. It does not consider the characteristics of the system, such as the electrical noises, output fluctuation, and response time.
2. It does not consider the characteristics of the battery, such as response time and battery temperature.

There are articles that show manageable differences between simulation and real-world implementation, such as in [33]. A few assumptions are made for the simulation in this research work:

1. **The power remains constant within the data interval.**
   The data used has an interval of 1-min. Thus, the power is assumed to remain constant within the 1-min interval. For example, the power readings between 09:00:00 – 09:00:59 are the same at 40kW. In real-world, the readings may fluctuate slightly due to electrical noise and sensor resolution.
2. **The response time of the converter is instantaneous.**
   The response time of the converter is typically quick (in the range of milliseconds), so it does not necessarily need to be accounted for the daily peak reduction and monthly maximum demand reduction.
3. **The optimizations are completed instantaneously.**
   The prediction and optimization used in this research work are completed in seconds, so it is not expected to cause significant differences in the output. Since the control re-optimizes the P_{Th} every 5-mins, the peak reduction operations are not halted when it is predicting or optimizing, but it will just use the P_{Th} from the previous interval. The adjustment P_{Th} in real-world may be a few seconds slower, but not necessarily need to be concerned for the daily peak demand reduction.
4. **The battery response time is assumed to be instantaneous, and its capacity has no degradation regardless of the discharge current.**
   Batteries are usually listed based on their nominal capacity for a specific C-rate. The higher the C-rate, the smaller the actual capacity, and vice versa. However, the discharge current is not expected to be constant throughout the whole discharge cycle. Hence, it should be safe to assume the average capacity throughout the whole discharge cycle is very similar to the nominal capacity.

The simulation in this chapter considers the efficiency of converters. Figure 5.5 shows the power flow between the converter and BESS to the AC bus. With η_P be the efficiency of the converter at P power, and P_{BESS} be the available power of the BESS, the discharging power at the AC side can be obtained using equation (24), and the charging power at the DC side can be obtained using equation (25). Since the efficiency varies depending on power, it is always referring to the input power. Thus, it is referring

to the power from DC side when it is discharging and is referring to the power from AC side when it is charging. Since the controller would try to maintain the demand at P_{Th}, the BESS needs to discharge more power and less energy is charged into BESS.

$$P_{D,\,AC} = \eta_{P_{D,\,DC}} \times P_{D,\,DC} \tag{24}$$

$$P_{C,\,DC} = \eta_{P_{C,\,AC}} \times P_{C,\,AC} \tag{25}$$

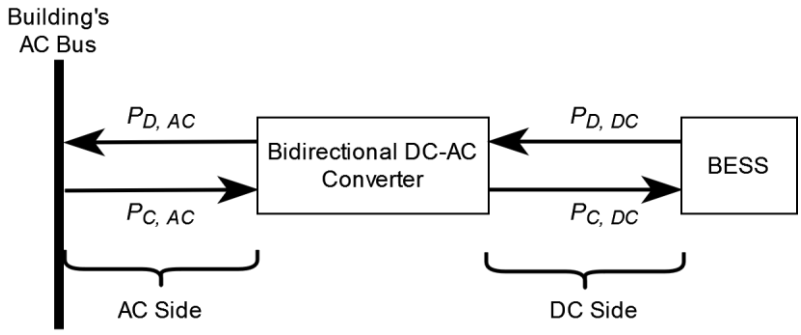


Figure 5.5: Power Flow between converter and BESS

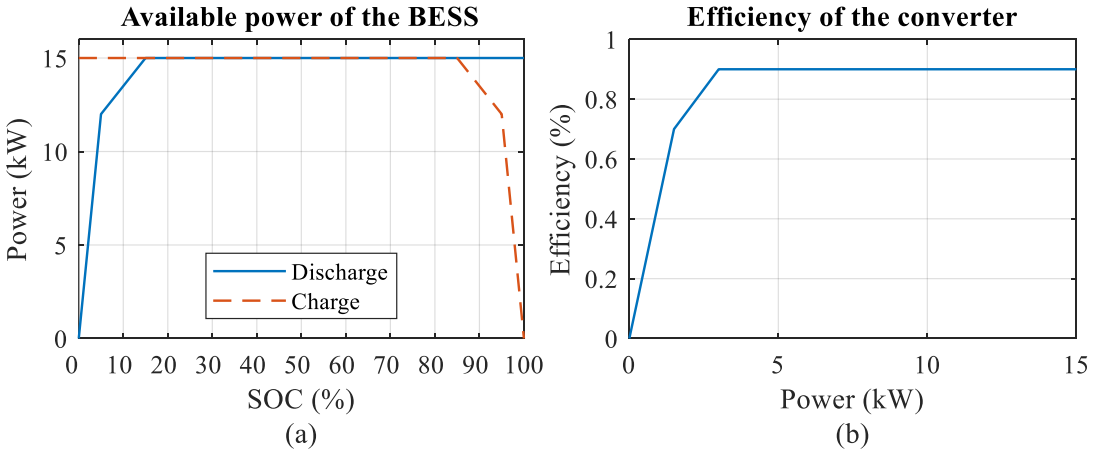


Figure 5.6: (a) Available power of the BESS, and (b) efficiency of the converter

The efficiency of converters that varies with power and maximum available power of BESS that varies with SOC, with the approach referred to [33]. It used a lookup table approach where the available output/input power of BESS and efficiency of the converter have their own curve. Figure 5.6 depicts the available power and efficiency curves used in the simulation. The available charging and discharging power of the BESS are limited by the SOC of the BESS. The peak charging and discharging powers of the BESS are assumed to be 15kW, which is 25% of the *Dataset A* and 20% of *Dataset B*. The 15kW is selected to not excessively oversizing the inverter rating and not restricting the operation of two-stage controller at the same time. If the discharge

or charge power is frequently restricted by the power rating, it may indicate the selection of inverter rating is inappropriate. Since peak demand reduction failure is avoided by P_{Th} increment, which means reduced inverter output, it is possible to avoid peak demand reduction failure by having the output power reduced according to the inverter rating. But if a peak demand reduction failure is avoided by the reduced output power by the power rating, but not by the P_{Th} adjustment of the controller, the capability of the controller may not be assessed properly. In simulations, the power rating is mostly to restrict the charging power of the battery. Otherwise, the charging power of the BESS during off-peak hours may look unusual high.

As indicated in Figure 5.6(a), the available discharge and charging powers decline when the SOC is lower than 15% and higher than 85%, respectively. This drop in BESS power restricts the discharging and charging power. However, they only occur near the end of the SOC and are not used typically. Moreover, the usable SOC can be limited so it would not limit output power when SOC is low. For example, a lead-acid battery is often restricted to no discharge below 50% since discharging below 50% can decrease the lifespan of lead-acid battery significantly [4], [35].

5.3.1 BESS sizing

Two methods are explored to find the $BESS_{size}$. The first method (*Method A*) suggests finding a BESS size that can perform peak shaving on each day of the collected historical load profiles [45]. The second method (*Method B*) suggests finding a $BESS_{size}$ with a generic load profile that is obtained by averaging the collected historical load profiles [35]. Both methods require a desired peak reduction percentage, which is selected to be 12% by referencing the average MDRP of the two months of peak reduction without PV in [4]. The calculated required $BESS_{size}$ using the two methods are provided in Table 5.2.

Table 5.2: Calculated required $BESS_{size}$ using the two methods

Dataset	**Required $BESS_{size}$ (kWh)**	
	Method A	***Method B***
Dataset A	28.262	30.466
Dataset B	49.763	46.557

As seen in Table 5.2, *Dataset A* has a higher $BESS_{size}$ with *Method B* than with *Method A*, while *Dataset B* shows a higher $BESS_{size}$ with *Method A* than with *Method*

B. Hence, instead of selecting which method to use, the higher $BESS_{size}$ is used. D*ataset A* uses 30.466kWh found using *Method B*, while *Dataset B* uses 49.763kWh found using *Method A*. Then, a margin is added to the calculated $BESS_{size}$ to cater for unforeseen additions and losses in the system [110]. A margin of 5% is selected in this research work. After the 5% margin and rounding up, the final $BESS_{size}$ is selected to be 32kWh and 53kWh for *Dataset A* and *Dataset B*, respectively.

As proposed in Section 5.2.4, the two-stage controller reserves an extra BESS to handle the underestimation of t_{end}. The $BESS_{reserved}$ is determined to be 10% of $BESS_{size}$, thus the $BESS_{reserved}$ is 3.2kWh for *Dataset A* and 5.3kWh for *Dataset B*. All controllers use the same BESS configuration for each dataset, thus the "32kWh+3.2kWh" and "53kWh+5.3kWh" configurations for *Dataset A* and *Dataset B*, respectively.

5.3.2 Preparation of controllers

The proposed two-stage controller is compared with two conventional controllers: fixed threshold controller [32] and single-stage controller [35]. To enhance the creditability of the proposed two-stage controller, it is also compared with a state-of-the-art controller, which is selected to be a fuzzy controller developed in [36]. This fuzzy controller is selected because it is developed for Malaysia load profiles and contains sufficient information to duplicate its work for comparison purposes in this research work. All controllers need to be prepared based on a certain amount of collected historical data at the time of implementation.

The fixed threshold controller needs to use the historical data to determine a desired fixed P_{Th} that is constant throughout the whole peak reduction period. The conventional single-stage, fuzzy, and proposed two-stage controllers need to train their prediction models using historical data. Like other conventional single-stage controllers, the fuzzy controller also requires day-ahead load profiles to optimize its output at a fixed interval throughout the day. To emulate limited historical data at the time of implementation, they only use the first 30 working days of historical data to find the fixed P_{Th} and train the models. This simulation considers working days only because peak reduction is not necessary for the low peak demands on holidays and weekends that do not contribute to the MD of the month.

Based on the desired peak reduction percentage of 12% and $BESS_{size}$ found in section 5.3.1, the fixed threshold controller determines a fixed P_{Th} of 43kW and 67kW for *Dataset A* and *Dataset B*, respectively. Figure 5.7 shows the flowchart of the MD reduction experiment for the proposed two-stage controller. The peak demand reduction operations begin from day 31 onwards after the prediction models are trained with the first 30 days of historical data only. At the end of each day, the prediction models are updated with the collected data of the day. The BESS always recharges to its full capacity during off-peak periods to prepare for the next day's operations. The conventional single-stage controller has a similar experiment flowchart to the two-stage controller, except that it uses the day-ahead load prediction model only.

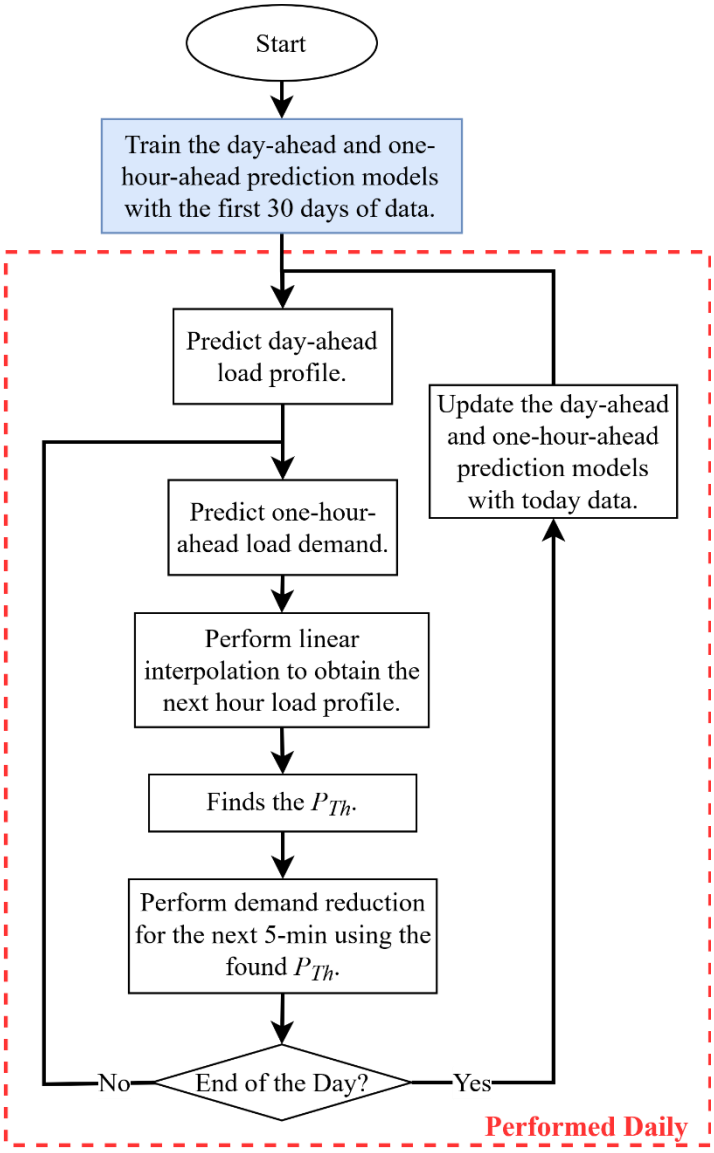


Figure 5.7: Flowchart of the MD reduction experiment for the proposed two-stage controller

As previously mentioned, the fuzzy controller developed in [36] is duplicated for comparison purposes in this research work. Figure 5.8 illustrates the flowchart of the fuzzy controller. The fuzzy controller has adopted Mamdani's fuzzy inference method as it is the most common fuzzy methodology. Unlike other controllers that optimize the demand threshold (P_{Th}) at every interval, the output of the fuzzy controller is the discharge power of BESS (P_D) for the next 5-min. The input to the fuzzy controller is the SOC of the BESS and the time of operation (t_{op}). The output of the fuzzy controller is the discharge power of the BESS. The definition of the fuzzy sets of the input and output of the fuzzy controller is provided in Table 5.3.

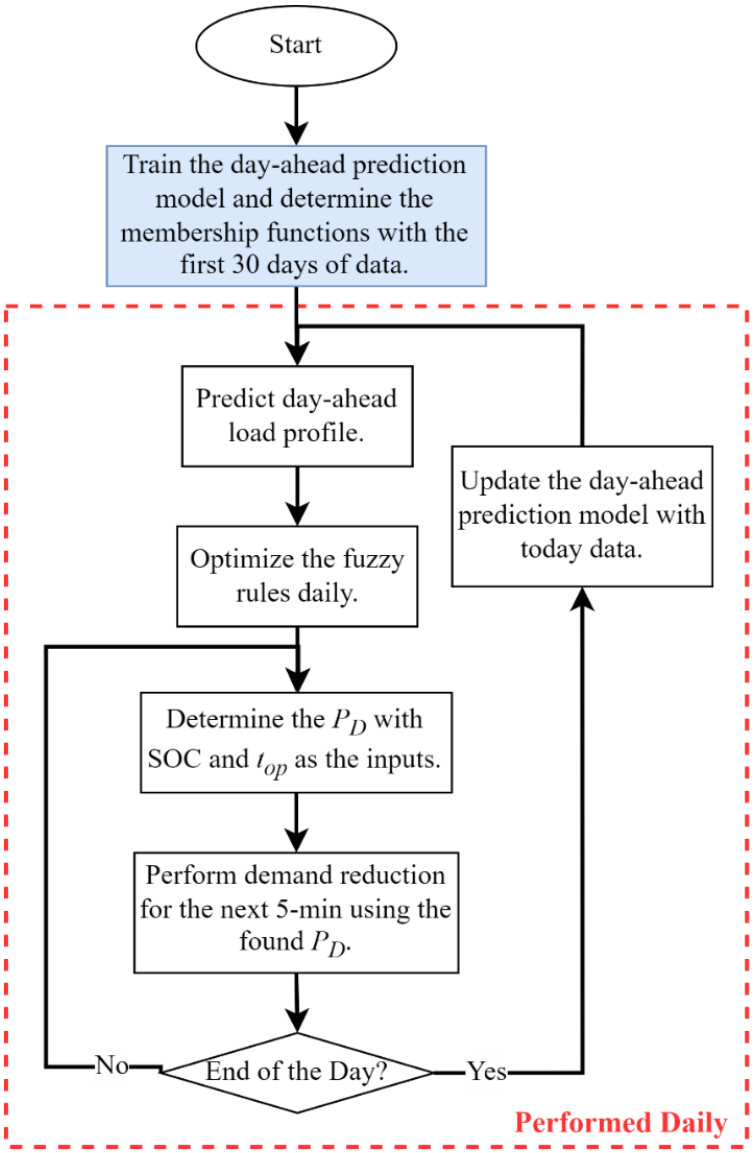


Figure 5.8: Flowchart of the Fuzzy Controller

Table 5.3: Definition of the fuzzy sets of the input and output of the fuzzy controller

SOC	t_{op}	P_D **of BESS**
EL – Extremely Low	EE – Extremely Early	VL – Very Low
VL – Very Low	VE – Very Early	L – Low
L – Low	E – Early	M – Middle
M – Middle	M – Middle	H – High
H – High	L – Late	VH – Very High
VH – Very High	VL – Very Late	
EH – Extremely High	EL – Extremely Late	

Before implementation, the fuzzy controller specifically defines the membership functions based on the characteristics of each building's load profile [36]. Figure 5.9 and Figure 5.10 illustrate the membership functions used for *Dataset A* and *Dataset B*, respectively. As depicted in Figure 5.8, the fuzzy controller uses the predicted day-ahead load profile to optimize its fuzzy rule at the beginning of each day. Table 5.4 presents an example of fuzzy rules used by *Dataset A*.

Since the fuzzy controller determines the discharge power only, it needs to predetermine a charging demand threshold to charge the BESS during off-peak hours. When the load power is below the charging demand threshold, the differences in power are recouped into the BESS. After studying the datasets, the charging demand threshold is determined to be 35kW and 40kW for *Dataset A* and *Dataset B*, respectively.

Despite the optimization of fuzzy rules, the membership functions are rigid and do not change throughout the MD reduction operations, thus cannot provide the adaptiveness required in this research work. Moreover, the determination of the membership functions and charging demand threshold need to be specifically designed for each profile, increasing the complexity of the implementation of the controller.

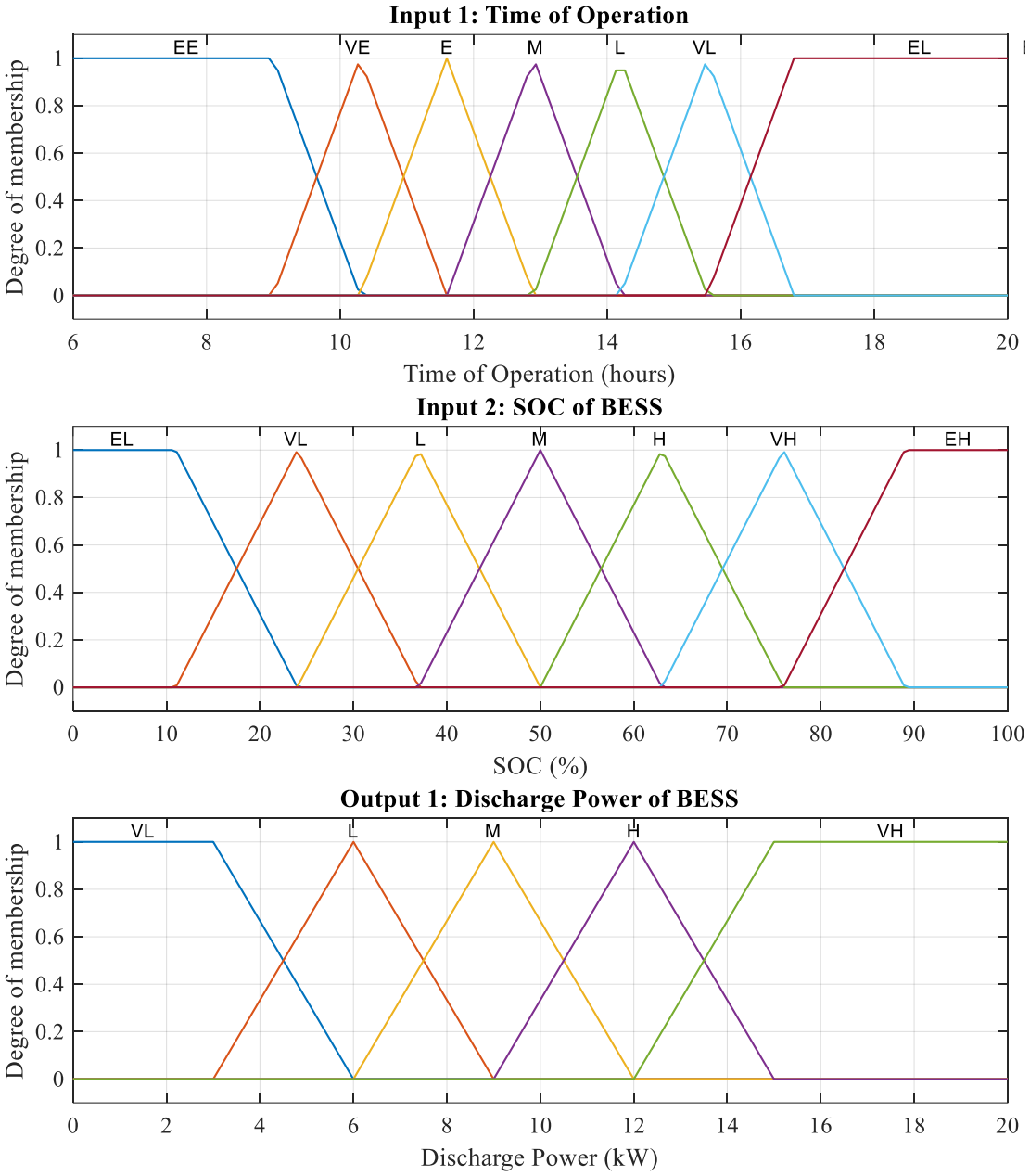


Figure 5.9: Membership functions for Dataset A

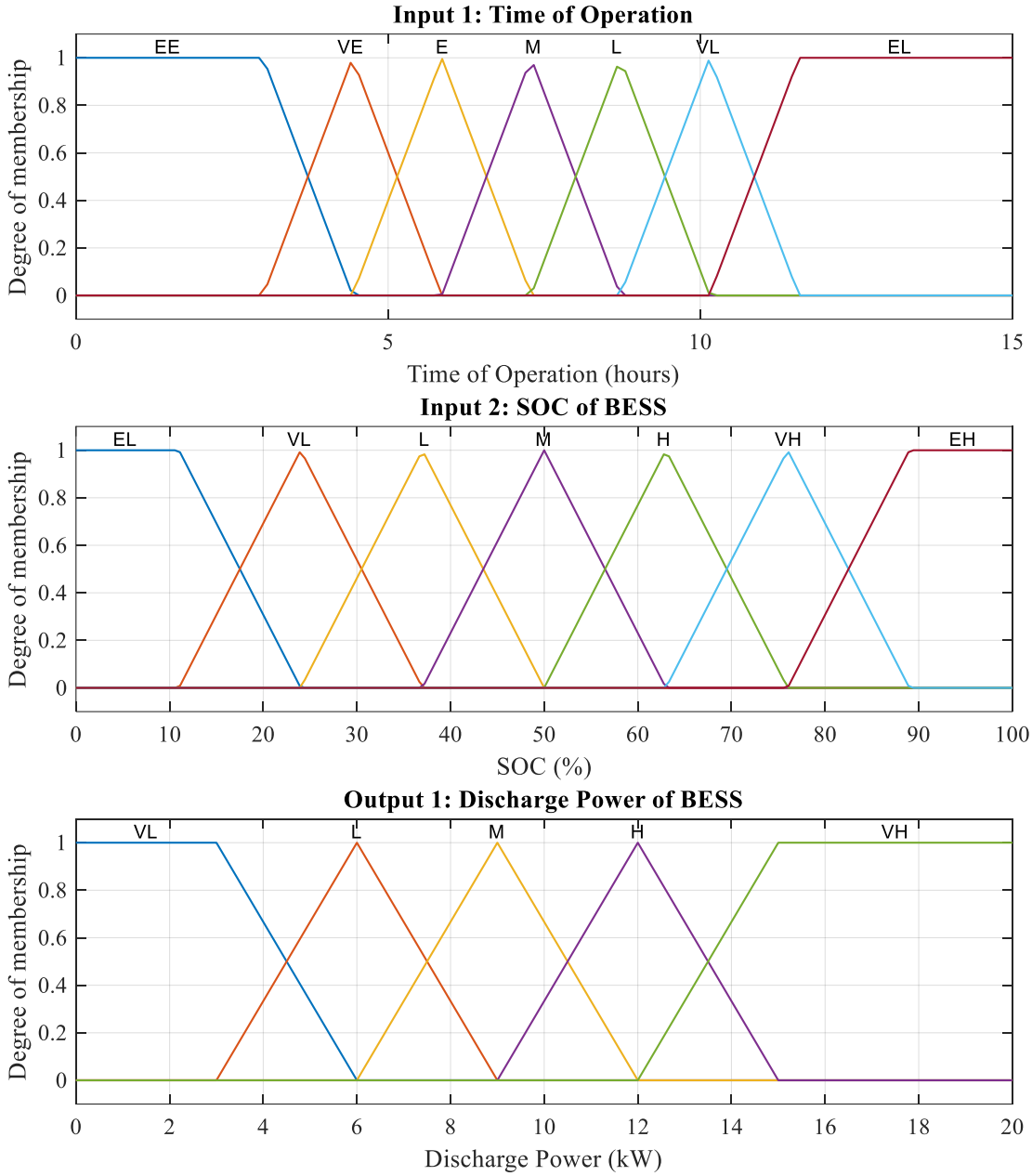


Figure 5.10: Membership function for Dataset B

Table 5.4: An example of fuzzy rules adopted by Dataset A

t_{op}	**SOC**						
	EL	VL	L	M	H	VH	EH
EE	VL	VL	L	L	L	M	M
VE	VL	VL	L	L	M	M	M
E	VL	L	L	L	M	M	H
M	VL	L	L	M	M	H	H
L	VL	L	M	M	H	H	VH
VL	VL	L	M	H	VH	VH	VH
EL	L	H	VH	VH	VH	VH	VH

5.3.3 Performance Metrics

This chapter evaluates the performance of the controllers with three metrics: maximum demand reduction percentage (MDRP), daily peak demand reduction percentage (PDRP), and peak demand reduction failure rate (PDR_{FR}). The accuracy of the day-ahead and one-hour-ahead predicted profiles are evaluated using mean absolute percentage error (MAPE). The MDRP evaluates monthly MD reduction performance, where a higher MDRP indicates higher MD and MD charge reductions. The MDRP is calculated using equation (26):

$$MDRP = \left(\frac{MD_{original} - MD_{reduced}}{MD_{original}} \right) \times 100\% \quad (26)$$

where $MD_{original}$ denotes the monthly MD before reduction and $MD_{reduced}$ denotes the monthly MD after reduction. Differs from the monthly MDRP, the PDRP calculates the daily peak demand reduction performance by using the highest energy consumption of any consecutive 30-minute interval of the day, instead of the month. To measure the occurrence of peak demand reduction failure, peak reduction failure rate (PDR_{FR}) is used. It is the ratio of the number of days with peak demand reduction failure to the total number of working days of the month. If the day has its BESS emptied before successfully reducing all anticipated peak demands, and has a PDRP of lower than 5%, the day is considered as peak demand reduction failure. If the day has a negligible PDRP of lower than 1%, it is also considered as peak demand reduction failure. This mostly occurs when the fixed threshold controller has its fixed P_{Th} higher or close to the peak demand of the day. Equation (27) is used to calculate the MD reduction saving cost.

$$MD_{cost} = MD_{rate} \times MD \quad (27)$$

where MD_{cost} is the MD charge of the month in RM, MD_{rate} is the MD charge rate in RM/kW, and MD is the maximum demand of the month in kW. In this chapter, the MD charge rate is assumed to be RM45.10/kW, which is rates for Tariff C2 set by TNB Malaysia. To evaluate the prediction accuracy of day-ahead and one hour-ahead load predictions, MAPE shown in equation (28) is used.

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{|y_t|} \times 100\% \quad (28)$$

where y_t, $\hat{y}_t$, $\bar{y}$, and n represent actual demand, predicted demand, mean of actual demand, and the number of data points in y_t, respectively.

5.4 Results and Discussion for *Dataset A*

Table 5.5 presents the monthly day-ahead MAPE, one-hour-ahead MAPE, MDRP, and PDR_{FR} for the three controllers on *Dataset A*. The one-hour-ahead load prediction has its MAPE lower than the day-ahead load prediction in all five months. This shows that the one-hour-ahead load prediction is more accurate than the day-ahead load prediction. Despite the relatively good average MAPE of 6.922% for day-ahead load prediction, Figure 5.14(c) shows there are seven days of under-forecasted load profiles with MAPE higher than 10%. Regardless of the seven days of under-forecasted load profile, the proposed two-stage controller has the highest average MDRP among the four controllers and 0% PDR_{FR}, showing its effectiveness at preventing peak demand reduction failure caused by BESS energy deficiency when the day-ahead load profile is under-forecasted.

Table 5.5: Day-ahead MAPE, one-hour-ahead MAPE, MDRP and PDR_{FR} per month on Dataset A

Month	Day-ahead MAPE (%)	One-hour-ahead MAPE (%)	Max MAPE (%)	MDRP (%)				PDR_{FR} (%)			
				Fixed threshold	Single-stage	Fuzzy	Two-stage	Fixed threshold	Single-stage	Fuzzy	Two-stage
Aug	6.757	5.293	10.726	13.752	1.666	6.904	9.413	7.143	35.714	0	0
Sep	7.666	5.348	11.140	7.323	2.385	8.255	9.242	0	44.444	0	0
Oct	7.422	5.540	11.292	1.773	1.822	6.707	8.428	68.421	47.368	0	0
Nov	6.062	4.556	10.161	3.586	3.995	5.315	7.601	40	10	0	0
Dec	6.706	4.643	7.042	1.148	1.150	7.309	10.433	50	50	0	0
Average	6.922	5.076	10.072	5.516	2.204	6.898	9.024	32.911	35.443	0	0

The fixed threshold controller has the highest MDRP in August compared to the four controllers because the original MD of August is similar to July data that is used to find the fixed P_{Th} of 43kW. In September, the fixed threshold controller has 1.919% lower MDRP than the proposed two-stage controller but 4.938% higher than the single-stage controller. For the remaining months, the fixed threshold controller has the lowest MDRP and the highest PDR_{FR} among the four controllers. As indicated in Table 3.2, *Dataset A* has its MD in October increased 9.56% compared to the MD in September, and the high MD maintained for the remaining two months. The fixed P_{Th} is too low for the high MD in October, resulting in frequent peak demand reduction failure caused by insufficient BESS energy. The single-stage controller has the lowest average MDRP and the highest PDR_{FR} among the four controllers. The underperforming of the single-stage controller is due to its inappropriate P_{Th} adjustment for compensating the under-forecasted day-ahead load profiles. The fuzzy controller has 0% PDR_{FR}, but its MDRP is lower than the proposed two-stage controller in all months. The proposed two-stage controller has the highest average MDRP among the three controllers and 0% PDR_{FR}, showing its effectiveness at preventing peak demand reduction failure caused by BESS energy deficiency when the day-ahead load profile is under-forecasted.

Figure 5.11 and Figure 5.12 show the performance of the four controllers on days 39 and 40, respectively. Both fixed threshold and single-stage controllers have peak demand reduction failure on these two days. Although the single-stage controller consistently increases the P_{Th}, the increment is insufficient to compensate for the high under-forecast day-ahead error. The day-ahead MAPE on days 39 and 40 is 11.29% and 11.26%, respectively. The proposed two-stage controller has a PDRP of 11.79% on day 39 and a PDRP of 9.069% on day 40, which are significantly higher than the low PDRP of between 0.501% and 4.351% for fixed threshold and single-stage controllers. Same as the proposed two-stage controller, the fuzzy controller has no peak reduction failure on days 39 and 40 with a PDRP of 7.4% and 5.764%, respectively. The PDRP of the fuzzy controller on these two days are higher than the conventional fixed threshold and single-stage controllers but are lower than the proposed two-stage controller.

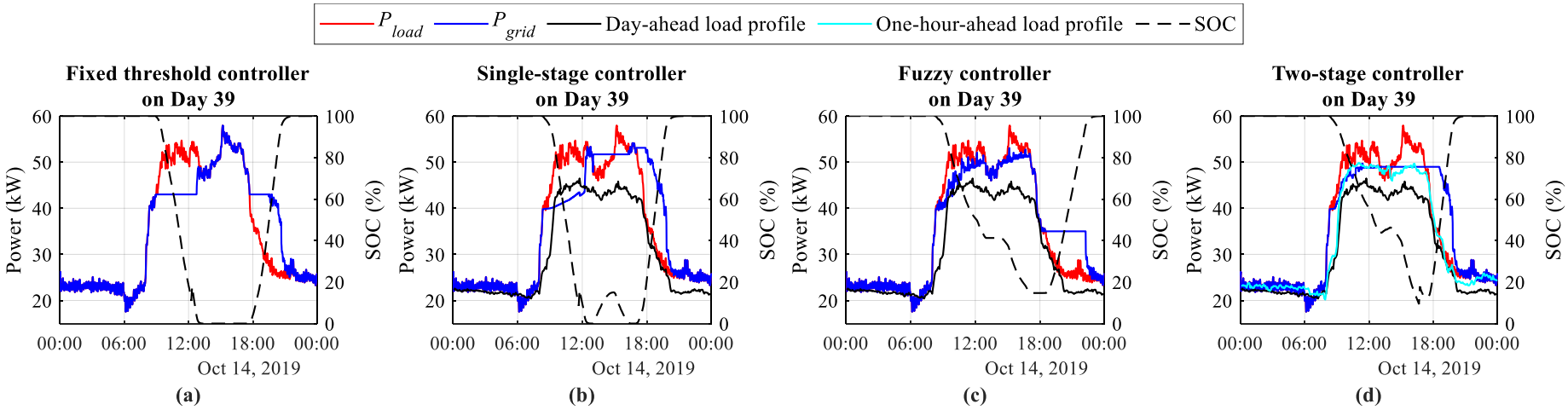


Figure 5.11: Performance of (a) fixed threshold, (b) single-stage, (c) fuzzy, and (d) two-stage controllers on day 39 for Dataset A

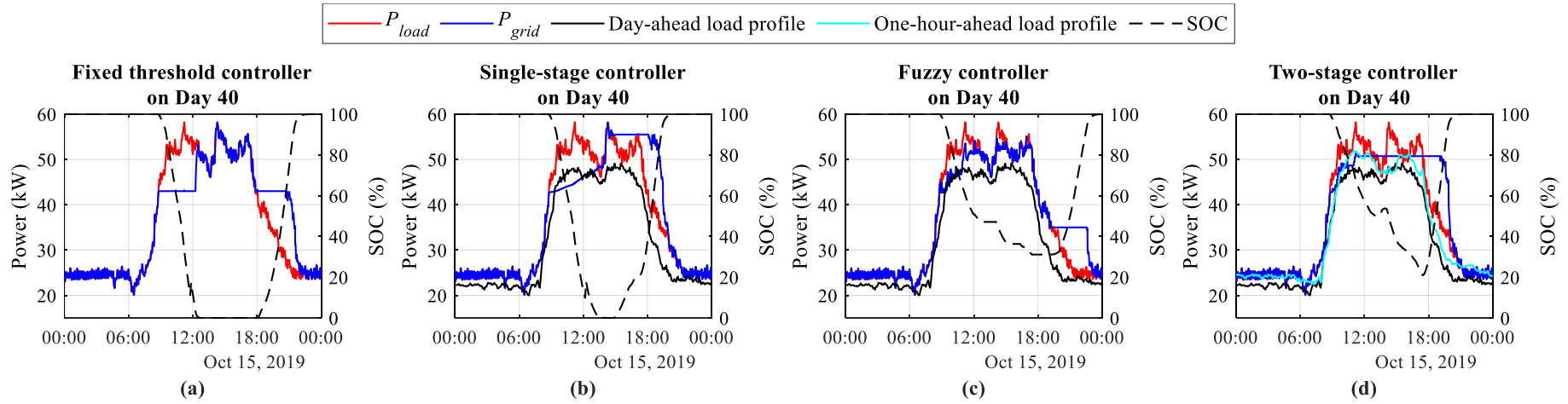


Figure 5.12: Performance of (a) fixed threshold, (b) single-stage, (c) fuzzy, and (d) two-stage controllers on day 40 of Dataset A

The conventional single-stage controller can have a higher PDRP than the proposed two-stage controller, such as on day 12 in Figure 5.13. Day 12 has a relatively accurate day-ahead load profile with a MAPE of 6.8%. Despite that, it fails to predict the high demands between 11:00 to 12:30. The conventional single-stage controller only has a small P_{Th} increment to compensate for the unpredicted peak demand and has no peak demand reduction failure. This is because the day-ahead load profile is over-forecasted at 12:30 – 14:20, which effectively counterbalance the excessive energy usage by the under-forecast.

The proposed two-stage controller overreacted to the unpredicted high demand by having a high P_{Th} adjustment. As a result, the two-stage controller has a PDRP of 10.55% that is lower than the PDRP of 18.33% for the conventional single-stage controller. However, the overreaction of the two-stage controller is expected. Since the controller would never know the exact period of the unpredicted high demand at that time, the overreaction can help prevent peak demand reduction failure when the worst-case scenario occurs, such as if the unpredicted high demand has a longer period.

The overreaction behavior of the proposed two-stage controller can help to achieve a higher PDRP, as shown on day 13. The single-stage controller has its BESS emptied and fails to reduce the peak demands at 17:00 – 18:00, resulting in a PDRP of 9.308% that is lower than the PDRP of 16.96% for the proposed two-stage controller. The proposed two-stage controller has adjusted its P_{Th} at 12:00 and needs no adjustment for the later unpredicted peak demands at 14:20 – 15:30, thus the higher PDRP. Despite the higher PDRP of the single-stage controller than the proposed two-stage controller on day 12, its MDRP for that month (August) is very low at 1.9% due to its high PDR_{FR} of 35.714%. In comparison, the two-stage controller has an MDRP of 10.135% and PDR_{FR} of 0% in August. This shows the importance of peak demand reduction prevention for the monthly MD reduction.

Figure 5.14 shows that the two-stage controller has its lowest PDRP of 3.919% on day 63, which is not considered as peak demand reduction failure because its BESS is not exhausted. Due to the over-forecasted day-ahead load profile, the controller found a P_{Th} that is higher than the actual peak demand of the day. The actual peak demand of the day is usually low at 46.6kW. This is acceptable as significantly over-forecasted

day-ahead load profiles usually have low peak demands that do not contribute to the MD of the month.

Although the proposed two-stage controller has a lower MDRP than the fixed threshold controller in August, it has significantly higher MDRP for the remaining months. The proposed two-stage controller outperforms other controllers in total MD charge saving. Assuming the MD charges have a fixed rate of RM45.10/kW, the proposed two-stage controller shows a total MD charge saving of RM1091.62, which is 40% higher than the total saving of RM833.31 for the fuzzy controller, 69.61% higher than the total saving of RM643.59 for the fixed threshold controller, and 306.23% higher than the total saving of RM268.72 for the single-stage controller.

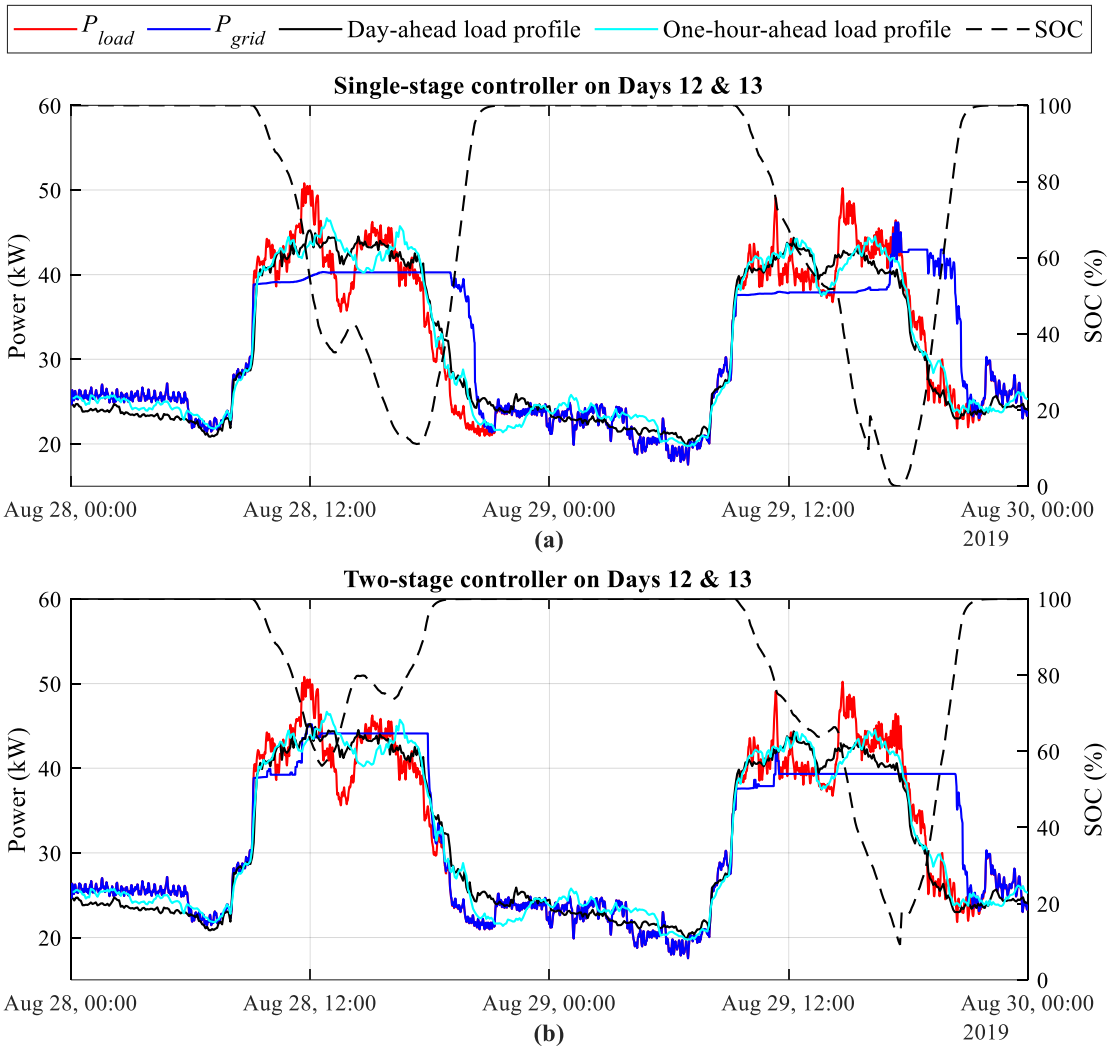


Figure 5.13: Performance of (a) single-stage and (b) two-stage controllers on days 12 and 13 of Dataset A

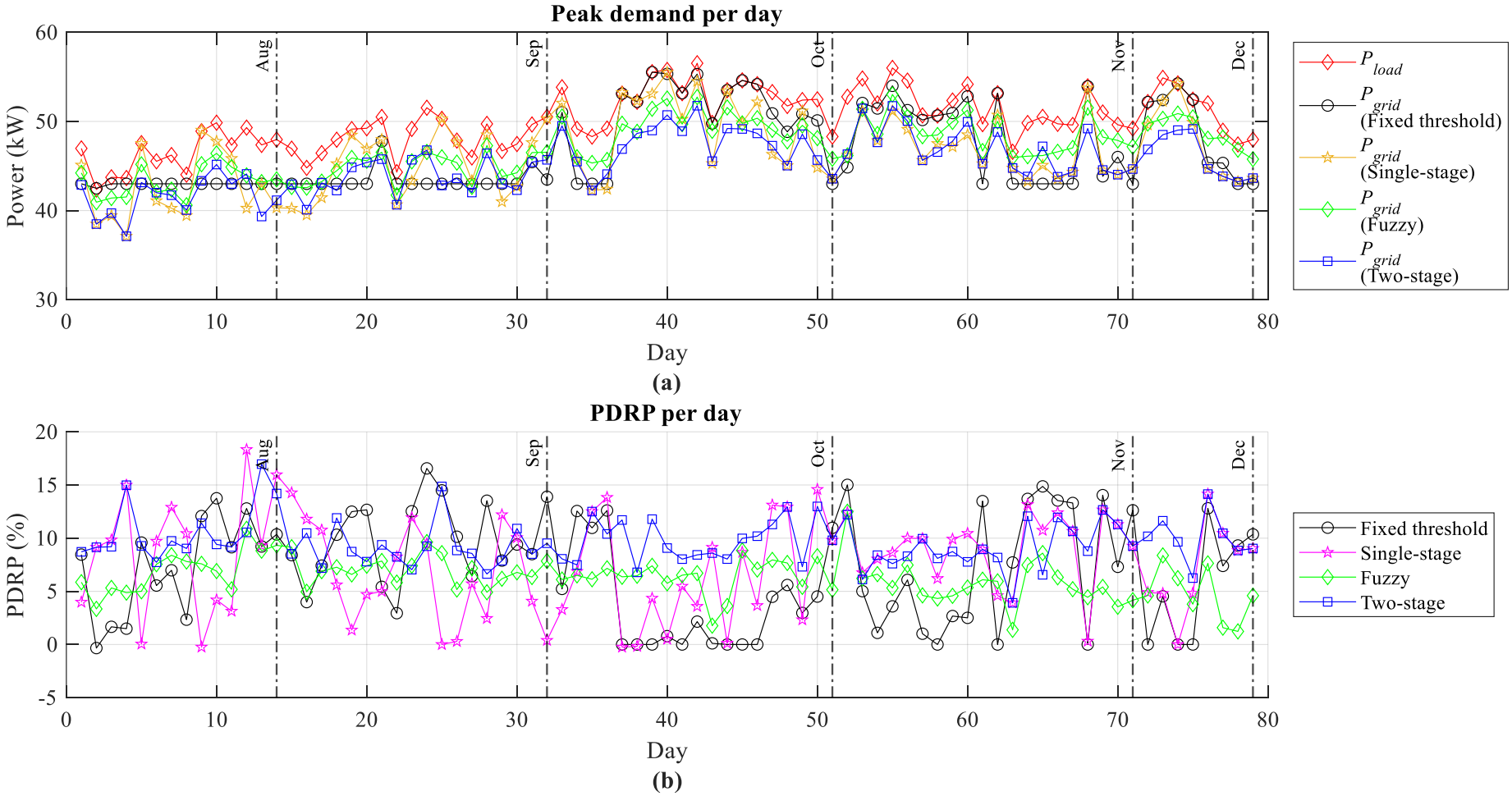


Figure 5.14: Daily performance for the controllers on Dataset A

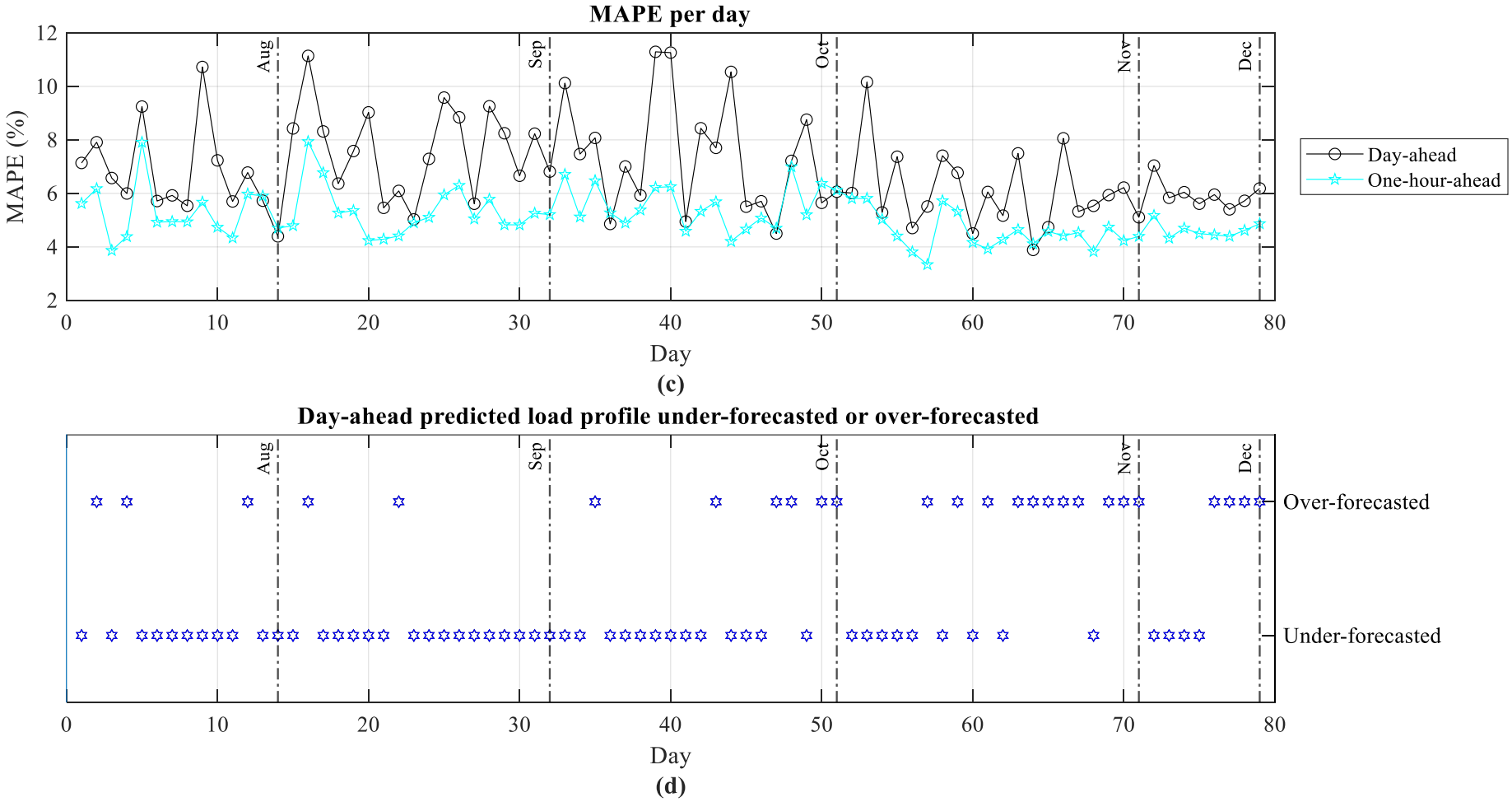


Figure 5.14(Continued): Daily performance for the controllers on Dataset A

Figure 5.15 depicts the boxplot of daily PDRP for the four controllers on *Dataset A*. The purpose of the boxplot is to evaluate the performance of the controller in terms of daily PDRP and to study its relationship with monthly MD reduction. In terms of median, the proposed two-stage controller has the highest median at 9.18%. The conventional single-stage controller has a slightly lower median than the proposed two-stage controller at 8.84%. The fixed threshold controller has a median of 7.30% and the fuzzy controller has the lowest median at 6.345%.

Both the conventional single-stage and fixed threshold controllers have their lowest PDRP at 0%, which indicates there are days with peak demand reduction failure. On the other hand, the fuzzy and proposed two-stage controllers have 0% PDR_{FR}, which means no days with peak reduction failure. They always trying to achieve a certain amount of peak reduction, instead of outright failure that results in no peak reduction. The proposed two-stage controller has the highest minimum PDRP of 3.92%, while the fuzzy controller has a minimum PDRP of 1.232%.

As seen in the boxplots in Figure 5.15, the proposed two-stage controller has a narrower IQR of 2.64 (8.19% - 10.84%), compared to 6.76 (4% - 10.76%) for single-stage controller and 10.6 (1.79% - 12.39%) for fixed threshold controller. Other than the IQR, the proposed two-stage controller also has higher Q1 and Q3 values than the other controllers. This indicates the proposed two-stage controller can achieve a more consistent peak demand reduction and higher PDRP when compared to the other two conventional controllers with the same predicted day-ahead load profiles. Although the fuzzy controller has 0% PDR_{FR} and a slightly narrower IQR of 2.38 (5.16% - 7.53%) than the proposed two-stage controller, its Q1 and Q3 are significantly lower. The Q3 of the fuzzy controller is lower than the Q1 of the proposed two-stage controller. The fuzzy controller is not a suitable candidate for daily peak reduction for *Dataset A*.

It is understood from the above analysis that a good overall daily PDRP performance does not mean good MDRP and vice versa. The fuzzy controller has the worse median in daily PDRP despite its average MDRP being higher than the two conventional fixed threshold and single-stage controllers. The single-stage controller might have the lowest average MDRP, its median in daily PDRP is very close to the best performing two-stage controller. This is because MDRP requires consistent instead of average PDRP performance. The proposed two-stage controller has the highest

minimum PDPR, the highest Q1 and Q3, and the highest median among the four controllers. As a result, it has the highest MDRP.

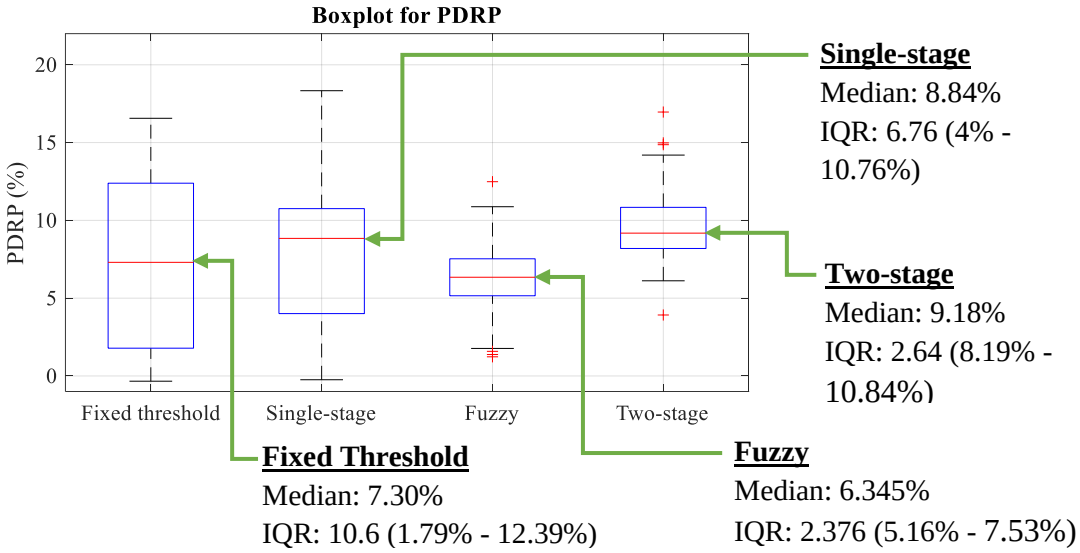


Figure 5.15: Boxplot of daily PDRP for the four controllers on Dataset A

5.4.1 Example of the activation of the additional reserved BESS

There is an additional reserved BESS implemented for all controllers. The additional reserved BESS is to tackle the potential scenario of inaccurate t_{end} for the proposed two-stage controller. Figure 5.16 presents an example of the activation of the additional reserved BESS for the two-stage controller on day 25. When the SOC is below 10%, the reserved BESS activates, which is indicated by the small increment in SOC as shown in the green dotted box. It helps to prevent the potential peak demand reduction failure caused by the underestimation of t_{end}. The day-ahead load profile (black line) is under-forecasted, and it is leading the load power (red line) at around 18:00. With the additional BESS energy, the controller reduces all demands successfully with a remaining SOC of 0.59% and a PDRP of 14.86%. Without the additional reserved BESS, peak demand reduction failure may have occurred.

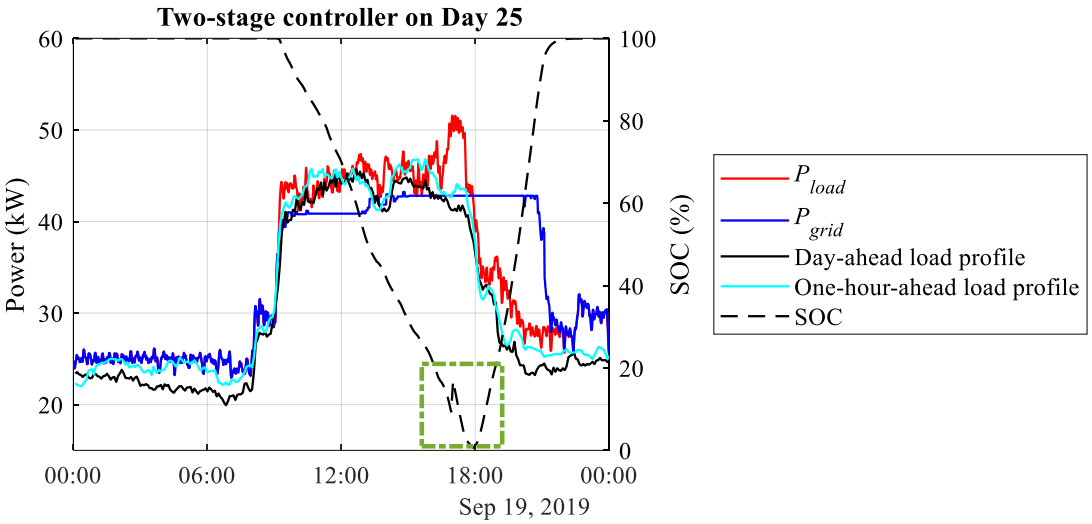


Figure 5.16: An example of the activation of the additional reserved ESS

5.4.2 MD reduction with doubled the $BESS_{size}$

To demonstrate the adaptability of the proposed two-stage controller, this section presents the results for *Dataset A* with the $BESS_{size}$ doubled from 32kWh to 64kWh, and the 10% reserved capacity doubled from 3.2kWh to 6.4kWh. The result is provided in Table 5.6. With the $BESS_{size}$ doubled, the single-stage controller has a lower MDRP and lower PDR_{FR} than with the smaller 32kWh+3.2kWh BESS configuration. This may be due to the larger BESS has a higher tolerance to prediction errors. Both the fuzzy and proposed two-stage controllers do have higher MDRP with the larger $BESS_{size}$. The proposed two-stage controller still has the highest MDRP compared to other controllers on *Dataset A*. Assuming an MD charge of RM45.10/kW, the proposed two-stage controller shows a total MD charge saving of RM1791.05 on *Dataset A*, which is 35.17% higher than the total saving of RM1325 for the fuzzy controller and 222.6% higher than the total saving of RM555.12 of the single-stage controller.

Table 5.6: MDRP and PDR_{FR} per month on Dataset A with 64kWh + 6.4kWh BESS configuration

Month	MDRP (%)			PDR_{FR} (%)		
	Single-stage	**Fuzzy**	**Two-stage**	**Single-stage**	**Fuzzy**	**Two-stage**
Aug	3.565	11.328	15.674	7.143	0	0
Sep	3.575	12.384	15.238	11.111	0	0
Oct	2.050	10.212	13.915	21.053	0	0
Nov	7.214	10.439	13.207	5	0	0
Dec	6.357	10.440	16.000	0	0	0
Average	4.552	10.960	14.807	8.861	0	0

5.5 Results and Discussion for *Dataset B*

Table 5.7 presents the monthly day-ahead MAPE, one-hour-ahead MAPE, Max MAPE, MDRP, and PDR_{FR} for the four controllers on *Dataset B*. Compared to *Dataset A*, the day-ahead MAPE is significantly lower in *Dataset B*. With better accuracy, the single-stage controller has a 0.849% higher average MDRP than the fixed threshold controller but still has low MDRP for some months due to the non-zero PDR_{FR}. The MDRP of the fuzzy controller is comparable to the proposed two-stage controller and has also achieved 0% PDR_{FR}. This shows that the single-stage and fuzzy controllers need accurate day-ahead load profiles for better performance. Despite the better performing single-stage and fuzzy controllers on *Dataset B* than on *Dataset A*, the proposed two-stage controller still has the highest MDRP among all controllers on *Dataset B*.

Table 5.7: Day-ahead MAPE, one-hour-ahead MAPE, MDRP and PDR_{FR} per month on Dataset B

Month	Day-ahead MAPE (%)	One-hour-ahead MAPE (%)	Max MAPE (%)	MDRP (%)				PDR_{FR} (%)			
				Fixed threshold	Single-stage	Fuzzy	Two-stage	Fixed threshold	Single-stage	Fuzzy	Two-stage
Mar	4.122	1.680	7.723	10.307	1.187	8.441	10.528	19.048	38.095	0	0
Apr	3.271	1.580	7.204	9.532	0.459	7.256	7.911	0	19.048	0	0
May	2.574	1.228	5.458	4.956	10.016	9.843	9.259	20	0	0	0
Jun	2.563	1.344	5.587	9.269	11.349	10.976	11.484	33.333	0	0	0
Jul	3.033	1.448	9.643	4.270	5.594	10.296	10.538	82.609	0	0	0
Aug	2.873	1.449	6.822	2.942	3.022	8.818	9.790	60	15	0	0
Sep	3.577	1.957	7.872	3.139	8.860	8.455	9.381	65	0	0	0
Oct	2.269	1.341	5.599	3.682	3.676	8.357	9.651	50	4.546	0	0
Nov	2.463	1.345	5.137	1.174	9.831	8.627	9.437	90	0	0	0
Dec	1.968	1.026	3.950	0.825	4.599	8.620	9.931	100	4.762	0	0
Average	2.871	1.440	6.500	5.010	5.859	8.969	9.791	52.153	8.134	0	0

The fixed threshold controller records very high PDR_{FR} and very low MDRP from July onwards, because the fixed P_{Th} of 67kW that determined from the first 30 days of data, is higher than the daily peak demands of less than 70kW from July onwards as shown in Figure 5.20(a). The fixed threshold controller has 0% PDRP on day 99 as indicated in Figure 5.17(a), which is caused by day 99 having a peak demand 68kW that is lower than the fixed P_{Th} of 67kW. In contrast, both the single-stage and two-stage controllers have the same PDPR of 8.738% on day 99. Day 99 has an over-forecasted day-ahead load profile with the highest MAPE of 8.716%. With the over-forecasted day-ahead profile, the two-stage controller does not need to trigger the second stage for further P_{Th} adjustment and thus has the same PDRP as the single-stage controller. The fuzzy controller proposed in [36] requires an upper limit that is determined using the day-ahead load profile to restrict excessive discharge. Since the day-ahead profile is over-forecasted, the upper limit is high and thus the low PDRP.

As presented in Figure 5.18(a), the fixed threshold controller is having a PDRP of 9.269% on day 65, while the single-stage controller and two-stage controller have PDRP of 16.21% and 13.89%, respectively. Similar to day 12 on *Dataset A* in Figure 5.13. the proposed two-stage controller overreacted to the unpredicted peak demand, resulting in a lower PDRP than the conventional single-stage controller.

However, the overreaction of the proposed two-stage controller does help to prevent peak demand reduction failure on day 164 as shown in Figure 5.19. On day 164, both the fuzzy and the proposed two-stage controllers adjusted their P_{Th} in response to the unpredicted high demand at 10:14. Although the time is close to the end of peak reduction and there should be sufficient remaining energy in BESS , the fuzzy controller has a much higher P_{Th} adjustment compared to the proposed two-stage controller. This resulted in a PDRP of 8.34% for the fuzzy controller on day 164. The proposed two-stage controller estimates the peak reduction end time (t_{end}) in its P_{Th} optimization. Since the unpredicted peak demand at 10:14 is near the expected t_{end}, the proposed two-stage controller made a very small P_{Th} adjustment only. As a result, the proposed two-stage controller has the highest PDRP of 10.11% compared to other controllers.

On day 164, the single-stage controller has its BESS emptied at 11:45, resulting in a low PDRP of 3.676%. It is also the only day with peak demand reduction failure that the single-stage controller has in October. Thus, the single-stage controller has the

lowest MDRP in October among all controllers as seen in Table 5.7. There are similar observations in August and December, where the MDRP of the single-stage controller can be improved significantly if the PDRP of one or two days is improved. This shows the importance of preventing peak demand reduction failure for the monthly MD reduction.

Figure 5.20 provides the data for daily performance for the controllers on *Dataset B* that includes peak demand per day, PDRP per day , MAPE per day and indication of under-forecast or over-forecast for the day-ahead predicted load profile. It is clearly seen from Figure 5.20(b) that the proposed two-stage controller has a more consistent PDRP with no PDPR lower than 5%. Although the fuzzy controller shows comparable performance to the proposed two-stage controller in this *Dataset B* due to the better day-ahead accuracies, the proposed two-stage controller has a higher average MDRP and total MD charge saving. Assuming the MD charges have a fixed rate of RM45.10/kW, the total MD charge saving for the two-stage controller is RM3119.78, which is 9.2% higher than the total saving of RM2857.1 of the fuzzy controller, 68.2% higher than the total saving of RM1854.78 for the single-stage controller and 91.08% higher than the total saving of RM1632.70 for the fixed threshold controller.

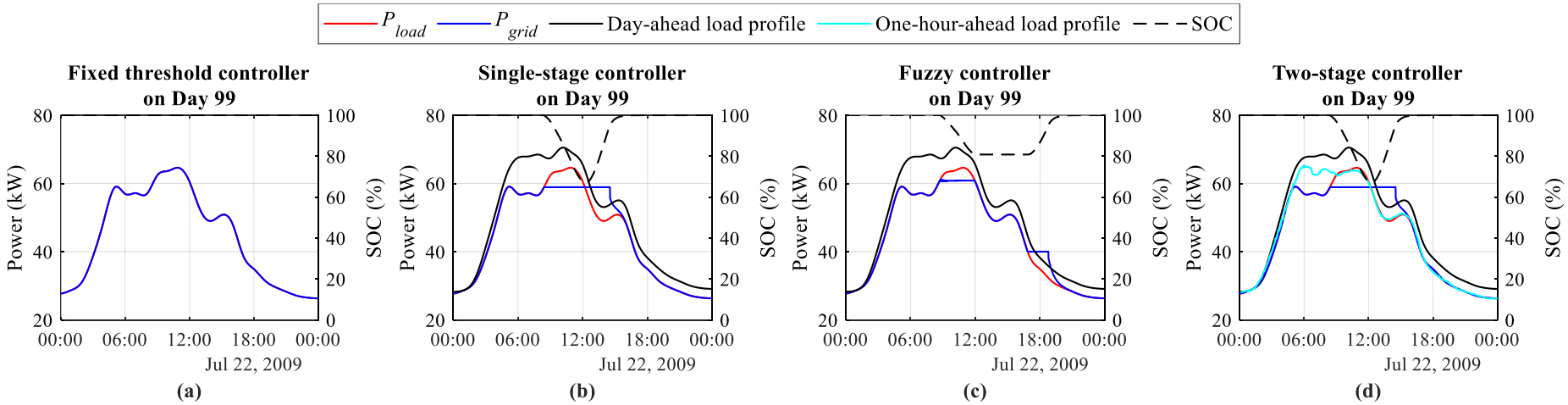


Figure 5.17: Performance of (a) fixed threshold, (b) single-stage, (c) fuzzy, and (d) two-stage controllers on day 99 of Dataset B

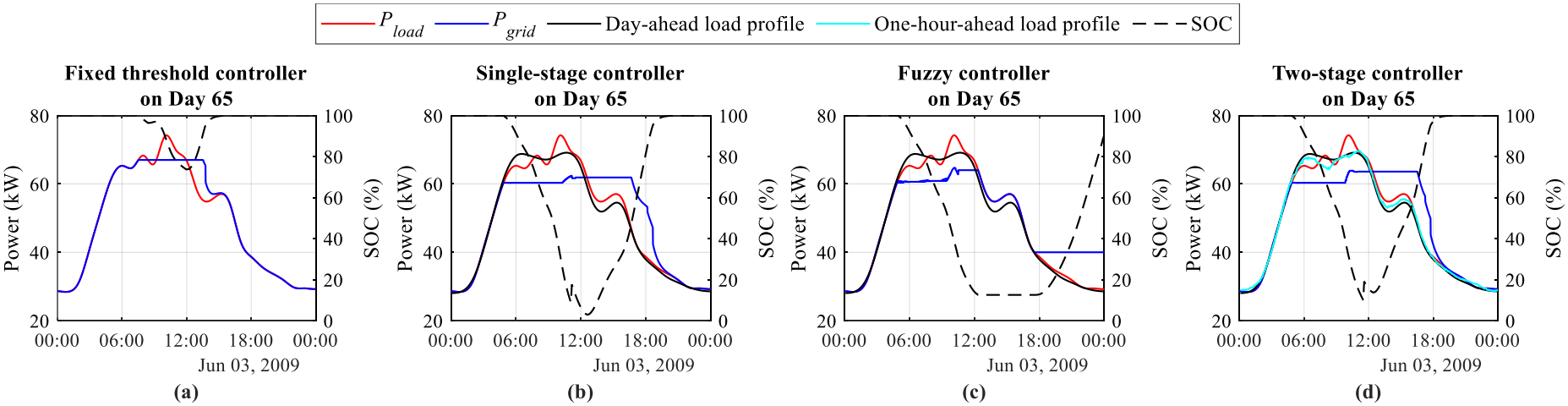


Figure 5.18: Performance of (a) fixed threshold, (b) single-stage, (c) fuzzy, and (d) two-stage controllers on day 65 of Dataset B

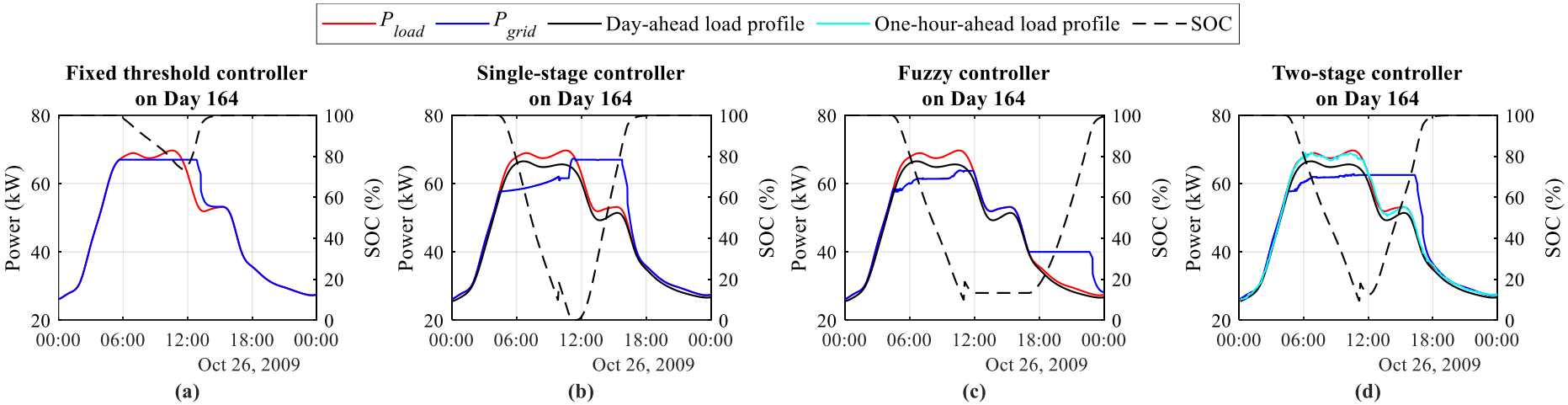


Figure 5.19: Performance of (a) fixed threshold, (b) single-stage, (c) fuzzy, and (d) two-stage controllers on day 164 of Dataset B

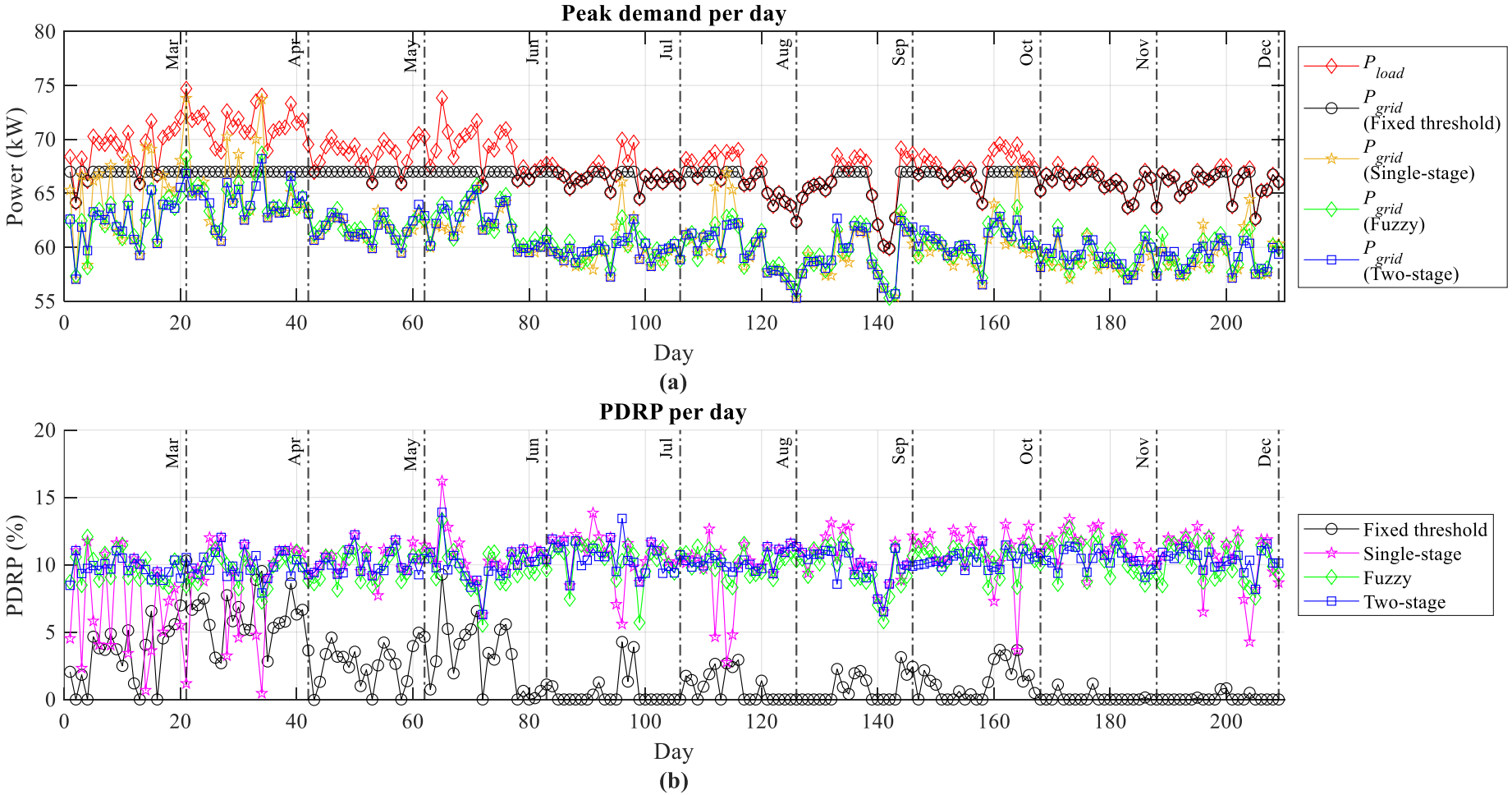


Figure 5.20: Daily performance for the controllers on Dataset B

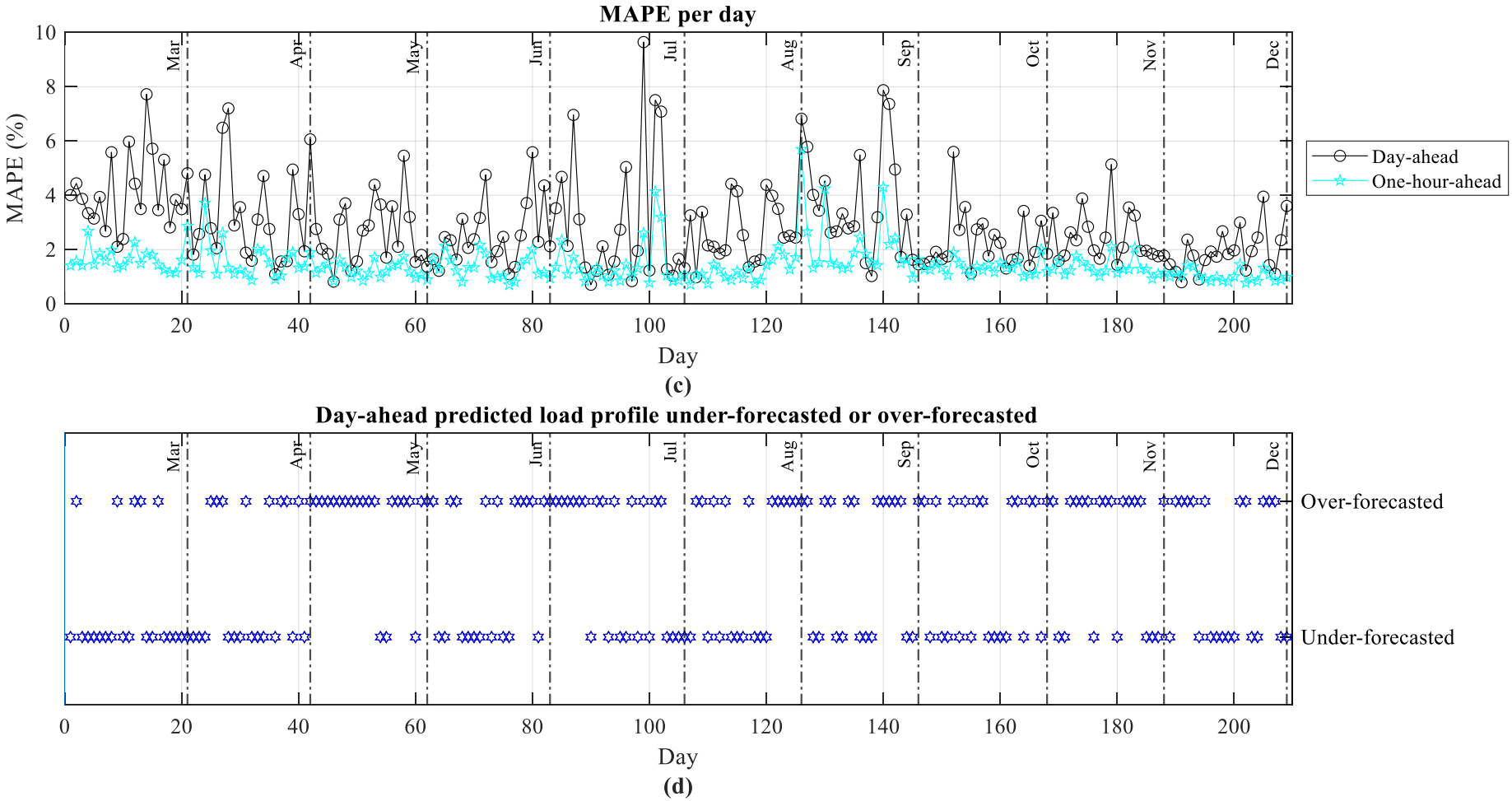


Figure 5.20 (Continued): Daily performance for the controllers on Dataset B

Figure 5.21 shows the boxplot of daily PDRP for the four controllers to evaluate their daily PDRP performance and to study the relationship between daily PDRP and monthly MD reduction. As seen from Figure 5.21, the performance of the fixed threshold controller is significantly weaker than others. Its highest PDRP is 10.3% only, which is only around the median of other controllers. The median of daily PDRP for the fixed-threshold, single-stage, fuzzy, and the proposed two-stage controllers are 0.82%, 10.68%, 10.3%, and 10.18%, respectively. Although the proposed two-stage has the lowest median among the three controllers, the difference is small.

As indicated in Figure 5.21, the conventional single-stage controller has an IQR of 2.08 (9.52% - 11.60%), which is narrower, higher Q1, and higher Q3 in *Dataset B* than in *Dataset A*. This is contributed by the lower overall MAPE of the day-ahead load prediction in *Dataset B* than in *Dataset A*. This shows the conventional single-stage controller can have good daily peak demand reduction performance with accurate day-ahead load profiles. The conventional single-stage controller does have a higher Q1 and Q3 than the proposed two-stage controller, but it has a high PDR_{FR} and a significantly higher number of days with PDRP lower than 6%, as indicated by the high number of outliers (red +) in Figure 5.21. The PDRP for the single-stage controller can be as low as 0.46%, while the lowest PDRP for the proposed two-stage controller is 6.31%. This is important for monthly MD reduction because the MDRP can be easily impacted by just one or two days of peak demand reduction failure. Therefore, the conventional single-stage controller has lower overall MDRP and total MD charge saving than the proposed two-stage controller.

The proposed two-stage controller has an IQR range of 1.28 (9.58% - 10.86%), while the fuzzy controller has an IQR of 2.00 (9.02% - 11.02%). The fuzzy controller and the proposed two-stage controller do have a similar median with only a 0.12% difference, but the proposed two-stage controller has a narrower IQR and higher Q1. Moreover, the minimum daily PDRP for the fuzzy controller is 5.54%, which is 0.77% lower than the 6.31% for the proposed two-stage controller. The higher minimum and more consistent daily PDRP allows the proposed two-stage controller to have a slightly higher overall MDRP and total MD charge saving than the fuzzy controller.

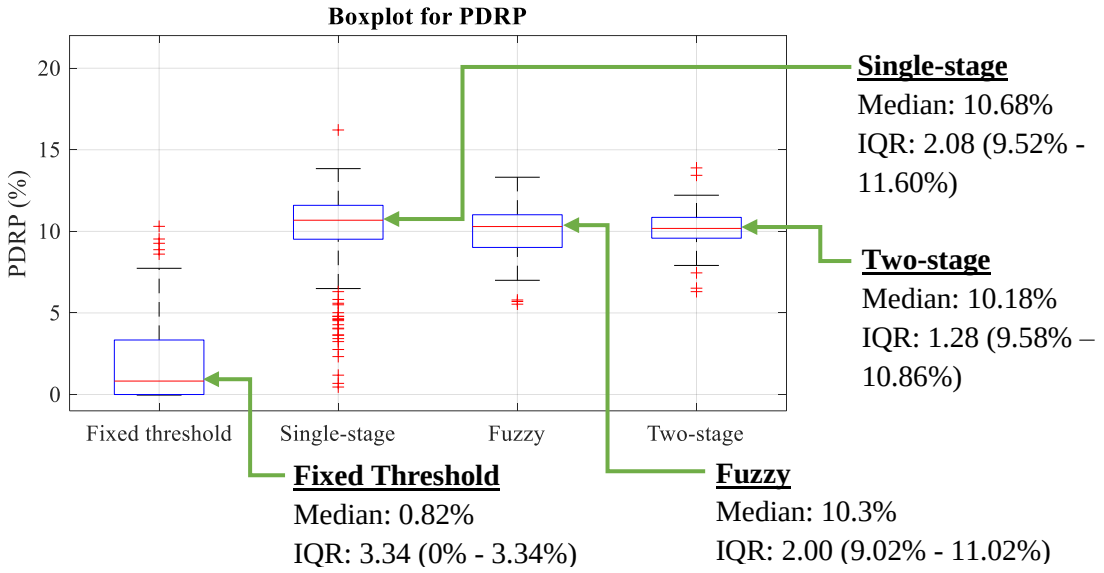


Figure 5.21: Boxplot of daily PDRP for the controllers on Dataset B

5.6 Summary

This chapter covered the monthly MD reduction performance of four controllers: conventional fixed threshold, conventional single-stage, fuzzy, and the proposed two-stage controllers. The conventional fixed threshold controller has a good MD reduction performance when the load profile is very similar to the historical profiles that are used to find the fixed threshold. Although the proposed two-stage controller has the highest MDRP in *Dataset B*, the results show that the conventional single-stage and fuzzy controllers can have better peak demand reduction performances with more accurate day-ahead load profiles. The single-stage controller has a higher average MDRP and a lower PDR_{FR} on *Dataset B* than on *Dataset A*. The fuzzy controller does have comparable performance to the proposed two-stage controller on *Dataset B* due to the more accurate day-ahead load prediction, but the proposed two-stage controller has 2.13% higher average MDRP and 40% higher total MD charge savings on *Dataset A*.

It is noteworthy that the proposed two-stage controller performs equally well on both datasets with no peak demand reduction failure (0% PDR_{FR}) and is not severely affected by the load prediction accuracy. The proposed two-stage controller has the highest MDRP and the highest total MD charge saving among the four tested controllers on both datasets, even with the doubled $BESS_{size}$ on *Dataset A*. The consistent performance of the proposed two-stage controller has demonstrated the peak demand reduction prevention capability of the proposed controller works on different datasets, even with different BESS configurations.

The advantages of the dynamic two-stage controller are its capability to provide consistent MD reduction for different profiles, ease of implementation, and requires only a small amount of historical data for preparation before implementation. The proposed controller does not use rigid parameters that need to be pre-determined from long-term historical data analysis at the time of implementation. It automatically finds and adjusts the P_{Th} in real-time based on available BESS energy. The proposed controller uses only 30 days of historical data for preparation. Then, the controller can be implemented and continuously improve itself to adapt to future load profiles automatically.

The implementation of the proposed two-stage controller is comparably straightforward. The same configuration is used for both *Dataset A* and *Dataset B*, except where the prediction models are trained using their respective data. Despite that, the proposed DB-SOINN-R is also implementable with 30 days of historical data. Since the proposed two-stage controller uses only simple data, it should be implementable for other industries or sectors.

The downside of the proposed controller is that it may have a slightly lower MDRP than other controllers in some months due to an excessive increase of the demand threshold to prevent peak demand reduction failure. However, this behaviour is expected by the proposed algorithm because a lower MDRP is better than no MD reduction due to BESS exhaustion. As a result, the proposed two-stage controller achieved a higher average MDRP and higher total MD charge savings than other controllers in both datasets.

Chapter 6 – Experimental Testing and Evaluation

6.1 Overview

The load predictions and MD reduction controller were presented and analysed in Chapter 4 and Chapter 5 focusing on simulation. This chapter is dedicated to check the viability of the proposed controller in an actual environment. A scaled down experiment setup is used. The setup is scaled down to a single-phase 240VAC setup with peak power of 1.8kW.

6.2 Experiment Setup

Figure 6.1 shows the experiment setup that resembles the block diagram shown in Figure 3.1. There are three sides, which are the load, grid, and BESS side. The grid side is a 240VAC network that supplies power to the loads. The load side is the Chroma 63800 programmable AC load that is used to simulate the load profile of Block D at the University of Nottingham Malaysia (*Dataset A*). The BESS side is the batteries that store and discharge energy to achieve peak shaving. The experiment in this chapter uses two different two converters with one responsible for charging and another one for discharging. A programmable AC-DC power supply Chroma 62024P-80-60 is used to charge the batteries from the AC grid, while a DC-AC grid-tied inverter SUN-2000GTIL2 is used to discharge the battery power to the AC grid.

Different sensing instruments are used to measure the different readings at different parts of the circuit connections. All the sensing devices are controlled by a node.js local server run in a low-power processing unit, which is a raspberry pi 4 4GB. The raspberry pi 4 is the central of all data that handles all the sensors readings such as process, logging and, transmitting to different scripts for other purposes, such as load predictions and the two-stage control strategy. More details on the data interfacing and controlling are discussed in sections 6.2.3 and 6.2.7, respectively.

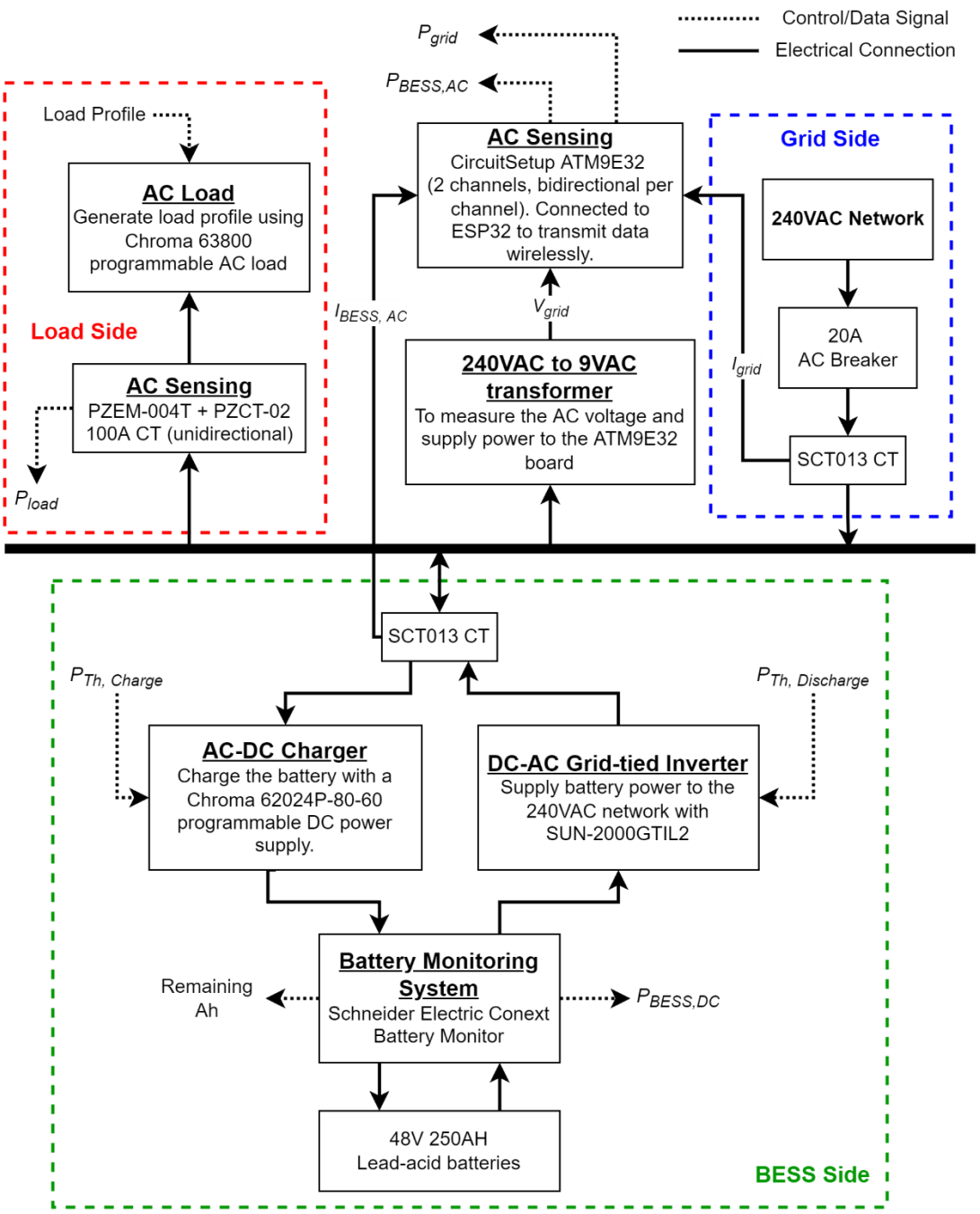


Figure 6.1: Experiment setup

6.2.1 Equipment

The experiment in this chapter uses four different equipment. Table 6.1 summarizes the purpose, control interface, and rating of the four different equipment. The datasheets of these equipment are provided in Appendix 1.

Table 6.1: Purpose, interface, and rating of the equipment

Equipment	**Purpose**	**Interface (protocol)**	**Rating (Range)**
Chroma 63800 Programmable AC Load	To simulate the load profile of a building.	RS232 port (Serial command)	Voltage (50 ~ 350Vrms) Current (0 ~ 18Arms) Power (0 ~ 1.8kW) Frequency (45 ~ 440Hz)
Chroma 62024P-80-60 Programmable DC Power Supply	To charge the batteries from the grid.	RS232 port (Serial command)	Voltage (0 ~ 80VDC) Current (0 ~ 60ADC) Power (0 ~ 2.4kW) Efficiency (85% at 2.4kW)
DC-AC Grid-tied inverter SUN-2000GTIL2	To discharge the power from batteries to the grid for peak reduction.	Digital signal	Output Voltage (185~265Vrms) Input Voltage (45 ~ 90VDC) Power (0 ~ 2kW) Power (Peak 2kW, Continuous 1.8kW) Efficiency (up to 92%)
Valve regulated maintenance-free lead-acid batteries	The BESS for peak reduction. Four 12V blocks in series forming 48V battery bank.	N/A	For each block: Voltage 12V Nominal temperature 25°C Max discharge current 1800A (5sec) Max charging current 75A 250Ah at 10C-rate Internal R= 2.6mΩ 100% capacity at 25°C Operating temperature range: -15°C ~ 40°C

The output power of DC-AC Grid-tied inverter SUN-2000GTIL2 is adjustable via its LCD panel or using an external limiter device that connects to the external limiter port of the inverter. The output power cannot be adjusted easily through programming. The original purpose of the adjustable output power of the inverter is to prevent exports to grid. This is because some old power meters do not consider power flow direction and count power exports as power imports. Thus, the user is paying for generated powers instead of getting paid for the exported power. For the experiment in this chapter, this function is modified to achieve real-time output power control for the inverter.

The output signal of a limiter has been interpreted in [111]. The external limiter port of the inverter consists of two wires only, with one wire carrying digital HIGH or LOW signals, and another wire is the ground wire. The inverter will ramp up or down when the signal is high (3.3V/5V) or low (0V), respectively. Hence, an ESP32 is used to produce the signal to the inverter. More details are discussed in section 6.2.7.

6.2.2 Sensing Instruments

Instead of using sensors that output analog signals that require further processing, such as analog-to-digital converter (ADC), this experiment uses different sensing instruments that measure the readings in real-time and, the readings can be fetched via different communication protocols. This simplifies the sensing processes and provides more accurate readings. Table 6.2 shows the purpose, interface, and measurement range of the four sensing instruments used in the experiment.

Table 6.2: Purpose, interface, and measurement range of the sensing instruments

Instruments	**Purpose**	**Interface (protocol)**	**Measurement Range**
PZEM-004T - Using PZCT-02 100A current transformer for stepping down and indirect AC current sensing.	AC Sensing	TTL (Modbus)	AC RMS Voltage (80V ~ 260V) AC RMS Current (0A ~ 100A) Active power (0kW ~ 23kW) Power Factor (0.00 ~ 1.00) Frequency (45Hz ~ 65Hz) Active Energy (0kWh ~ 9999.99kWh)
CircuitSetup ATM9E32 - Has two channels and both channels can measure the direction of AC current/power flow - Using SCT013 100A CT per channel.	AC Sensing	TTL (SPI)	AC RMS Voltage (0V ~ 655.35V) AC Current (0A ~ 100A) Power Factor (-1.000 ~ 1.000) Frequency (45Hz ~ 65Hz)
Schneider Conext battery monitor - Capable of measuring bi-directional current flow.	Battery monitoring	RS485 (Modbus)	DC Voltage (0V ~ 70V) DC Current (0 ~ 200A) Remaining Ah (0Ah – 9990Ah) SOC of battery (0% ~ 100%) Temperature (-20°C – 50°C)

Table 6.3 summarizes the parameters measured per sensing instrument or sensor. These sensing instruments can also measure frequency and power factors, which can be used to verify the effect on the grid stability and check for any unexpected impact on the grid. Both the PZEM-004T and ATM9E32 are for AC sensing. The PZEM-004T is capable of measuring unidirectional current flow only, while the ATM9E32 has two channels with each capable of measuring bidirectional current flow. Therefore, the PZEM-004T is used for measuring the AC parameters of the load side, as the load draws power only and never supplies power. The ATM9E32 is used to measure the grid and BESS sides that has bidirectional power flows. As shown in Figure 6.1, only one CT is used to measure the current ($I_{BESS,AC}$) from the AC-DC charger and DC-AC grid-tied inverter. This is because both devices should not switch on at the same time. Hence, the CT is measuring the current flowing through the inverter when the BESS is discharging and is measuring the current through the charger when charging the BESS. If both converters are switched ON at the same time, both converters will work normally, but the CT will be capturing the resultant power.

The Conext battery monitor can measure bidirectional DC current flow, so it is used to measure the discharge and charging DC powers of the battery. The meter does not have power readings but can be obtained easily by multiplying the DC voltage and DC current. The Conext battery monitor is also used to retrieve battery readings such as remaining Ah and SOC. The remaining energy in the battery is necessary for the two-stage controller for P_{Th} adjustment. Typically, SOC is used to calculate the remaining energy using the nominal capacity. The Conext battery monitoring system measures both the SOC and remaining Ah with a resolution of 1% and 1Ah, respectively. However, 1% of 250Ah means 2.5Ah, which means it has a lower resolution than the remaining Ah reading. Hence, this experiment uses the remaining Ah reading to calculate the SOC and remaining energy instead of using the SOC reading directly.

Table 6.3: Instruments or sensors per parameter

Sensing Instruments	Parameter	Abbreviation
PZEM-004T + PZCT-02 100A CT	Load Voltage	V_{load}
	Load Current	I_{load}
	Load Power	$\underline{P}_{load}$
	Load Frequency	F_{load}
	Load Power Factor	PF_{load}
CircuitSetup ATM9E32 (common)	Grid/BESS Voltage	V_{grid}
	Grid/BESS Frequency	F_{grid}
CircuitSetup ATM9E32 (Channel 1 with SCT013 100A CT)	Grid Current	I_{grid}
	Grid Power	$\underline{P}_{grid}$
	Grid Power Factor	PF_{grid}
CircuitSetup ATM9E32 (Channel 2 with SCT013 100A CT)	BESS AC Current	$I_{BESS,AC}$
	BESS AC Power	$P_{BESS,AC}$
	BESS AC Power Factor	$PF_{BESS,AC}$
Schneider Conext battery monitor	BESS DC Voltage	$V_{BESS,DC}$
	BESS DC Current	$I_{BESS,DC}$
	Remaining Ah	Ah

6.2.3 Data interfacing

Figure 6.2 shows the data flow between different programs. This experiment uses a raspberry pi 4 4GB model and an ESP32 microcontroller. The raspberry pi 4 is also known as a single-board computer, so it is more capable than a microcontroller and can handle multiple tasks at once with its four CPU cores. The raspberry pi 4 has a Node.js local server, an MQTT broker, and four python scripts running at the same time. The Node.js local server is hosted as the central of all data. It receives data from the different sensors every 5s. Then, it processes the data to 1-min data by considering data loss. All data are logged into file for future reference and analysis. It also transmits necessary readings to other blocks, such as sending the load data and battery readings to the Python (A) block. An MQTT broker is installed in the raspberry pi to handle MQTT transmissions.

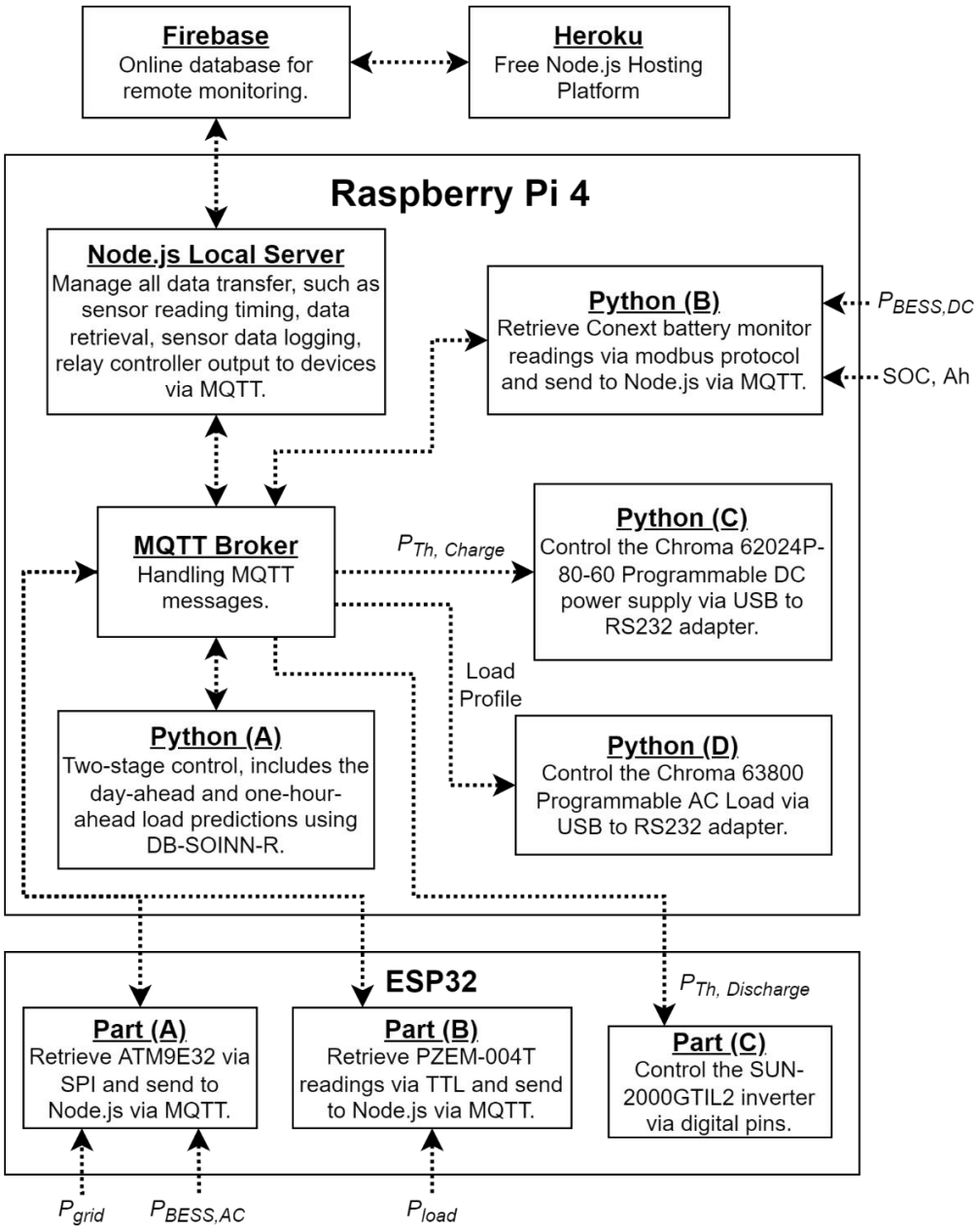


Figure 6.2: Data exchange block diagram

There are four python scripts running simultaneously in the raspberry pi 4. Python (A) runs the two-stage control that includes the day-ahead and one-hour-ahead DB-SOINN-R load predictions. Python (B) retrieves the Conext battery monitor readings via USB-RS485 adapter. Python (C) interfaces with the Chroma 62024P-80-60 programmable DC power supply for charging the 48V battery bank. Python (D) controls the Chroma 63800 programmable AC load to simulate the load profile via a USB-RS232 adaptor. Table 6.4 summarizes the interface, protocol, and coding language used per sensing instrument and equipment.

Table 6.4: Interface, protocol, and coding language used per sensing instrument and equipment

Instrument/Equipment	**Interface**	**Protocol**	**Coding Language**	**Block**
Conext battery monitor	USB-RS485	Modbus	Python	Python (B)
Chroma 62024P-80-60	USB-RS232	Serial	Python	Python (C)
Chroma 63800	USB-RS232	Serial	Python	Python (D)
ATM9E32	TTL	SPI	Arduino IDE	Part (A)
PZEM-004T	TTL	Modbus	Arduino IDE	Part (B)
SUN-2000GTIL2	TTL	-	Arduino IDE	Part (C)

MQTT is selected as the communication protocol between scripts and between raspberry pi 4 and ESP32. MQTT is a lightweight protocol that is designed for low overhead and small data transmission. Although HTTP is simpler for requests that require responses, preliminary testing shows that HTTP tends to have more response timeout compared to MQTT. Hence, MQTT is used for most of the communication, while HTTP is used for intermediate requests or requests that can tolerate request failure.

Python (A) is the python code for the two-stage control and DB-SOINN-R load predictions, which are translated from the MATLAB code for the DB-SOINN-R and two-stage control used in Chapter 4 and Chapter 5. Python is more RAM-efficient, which is good for implementation on a low-power device such as the raspberry pi 4. The outputs of the code have been verified and performed identically to the MATLAB code.

Python (B) retrieves data from the Conext battery monitor when receiving a request message from the local server. Python (C) performs charging based on received $P_{Th,Charge}$ message from the local server. Python (D) sets the AC load to change its power load every 1-min based on *Dataset A*, with more detailed discussion is covered in section 6.2.5.

Unlike raspberry pi 4 that can run multiple programs simultaneously, the ESP32 can only run one program. The ESP32 program has three parts: 1) interface with the CircuitSetup ATM9E32 power meter board via SPI, 2) interface with the PZEM-004T

via Modbus TTL, and 3) control the SUN2000GTIL2 inverter via digital pin. The library codes for ATM9E32 and PZEM-004T are referenced from [112] and [113], respectively. Since the PZEM-004T is designed to work with 5V, it is modified to be 3.3V compatible by adding a resistor [114]. One digital pin of the ESP32 is used to control the inverter. The communication between raspberry pi 4 and ESP32 is done wirelessly through a local network, thus low latency and allowing more flexible device placement.

To enable remote monitoring, the data needs to be accessible over the internet. Hence, the local server updates the latest data to Firebase, which is an online database by Google. Then, a separate Node.js server is hosted in a free hosting platform named Heroku to serve different web pages for displaying data and errors. The website is available at https://peak-shaving.herokuapp.com. The website is protected by login credentials to prevent unauthorized access.

6.2.4 Data processing

The local server polls the readings at a rate of 5-second. It is then processed into data of two different intervals: 1) 1 minute, and 2) 5 minutes. Figure 6.3 shows the flowchart of the three main loops for processing the sensor data. Due to the asynchronous nature of Node.js, these three loops are executed simultaneously, but they are triggered at different timing due to the different intervals. All the 5s data, 1-min data, and 5-min data are logged into csv files and saved in the SD card of the raspberry pi 4.

Loop A requests readings from all sensing instruments every 5 seconds. The 5s data are stored temporarily for processing at Loop B. Conventionally, the sensing instrument may send data to the local server automatically at a predefined sampling interval. However, this may result in different timestamps for the readings when there are multiple sensing instruments. Therefore, the scripts for sensing instruments are coded to wait for request messages from the local server before retrieving readings from sensors and send back to the local server. This ensures minimal time difference between readings from the different sensing instruments. It also allows easy detection of sensor failure, which is when the sensor does not reply to the request message.

Loop B triggers every 1-min to average the collected 5s data. Assuming there is no data loss, there will be 12 sets of 5s readings every time Loop B triggers. Since Python (D) sets the AC load to change its power load every 1-min, the measured 5s

readings within the minute should be very similar. Averaging the readings suppresses weird outliers and helps to handle response timeout failures from the sensing instruments. With 5 seconds sampling interval, it allows up to 11 response timeout failures with a minimum of one success read. Loop C triggers every 5-min to average the 1-min data into 5-min data and is then sent to the Python (A) for load predictions and two-stage control.

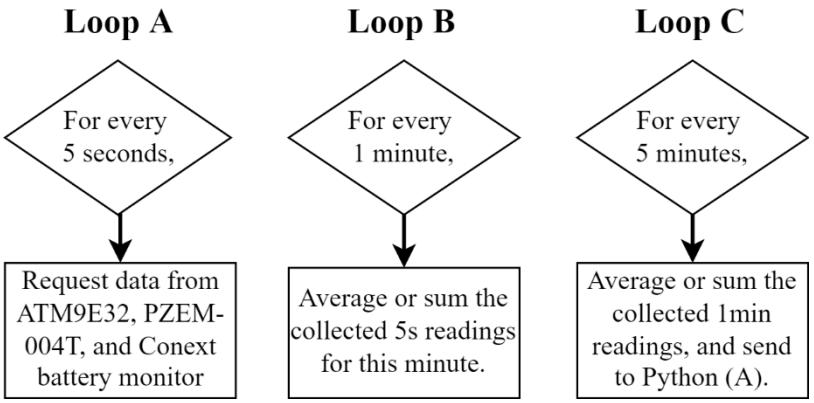


Figure 6.3: Loops for processing sensor data

6.2.5 Load profile

Unlike the simulations done in Chapter 3 and Chapter 4 that used up to 10 months of data, the purpose of this experiment is to validate the real-world performance of the proposed two-stage controller and to observe the differences with simulation results. This experiment only performs peak reductions for four days of the load profile of Block D at the University of Nottingham Malaysia (*Dataset A*), which are from 13th August 2019 (Tuesday) – 16th August 2019 (Friday). If there are negligible differences in these four days, the difference is assumed to be negligible for other days.

As shown in Table 6.1, the programmable AC load has a peak power of 1.8kW only. To maximize the usable range, an offset is deducted from the original *Dataset A* profile to remove the constant power drawing throughout the four days. For the first four days of data, the minimum and maximum powers are 20.1333kW and 50kW, respectively. Hence, the power in kW is converted to the range of 0-1.8kW using equation (29). For example, with the original power of 50kW, the scaled power for the AC load is 1800W.

$$scaled\ power\ in\ W = \big((power\ in\ kW) - 20\big) \times 60 \qquad (29)$$

The prediction models are pre-trained with the same 30 days of working day data that are used in the MATLAB simulation as discussed in section 5.3. Then, it starts the peak shaving operations for the next four days. To reduce unnecessary waiting time for day 1, the data between 00:00 and 08:30 of day 1 are preloaded in Python (A) and the AC load starts to output the power at 08:30 of day 1 instead of the power at 00:00.

6.2.6 Battery sizing

As discussed in section 5.3.1, *Dataset A* uses a BESS configuration of "32kWh + 3.2kWh". Since the load profile is scaled down to 1.8kW, the required BESS size is also scaled down to 40Ah for the 48V battery. As shown in Figure 6.1, the battery bank has a capacity of 250Ah. Hence, the usable Ah range is limited to 181Ah – 225Ah to form a "40Ah + 4Ah" configuration.

6.2.7 Inverter and charger control

Although only one threshold (P_{Th}) is mentioned in Section 3.2 and Chapter 5, Figure 6.1 shows two different thresholds ($P_{Th,Charge}$ and $P_{Th,Discharge}$). The $P_{Th,Charge}$ and $P_{Th,Discharge}$ have the same value in Chapter 5. They are separated in this chapter to differentiate the threshold values send to the Python (D) and ESP32 Part (C) for controlling the AC-DC charger ($P_{Th,\ Charge}$) and DC-AC Grid-tied inverter ($P_{Th,\ Charge}$ and $P_{Th,\ Discharge}$), respectively.

Every 5-min, Python (A) is triggered and responds with the P_{Th} for the next 5-min. In the simulation, the P_{Th} is compared directly with P_{load} and the converter will discharge or change accordingly. In actual implementation, the P_{load} measurement is not as stable as in simulation where the measurement fluctuates with differences of not more than 5W. The difference is small but may still cause constant switching between charging and discharging if P_{load} is very close to P_{Th}, causing both converters to turn on at the same time. Hence, a hysteresis area of 5W is added when performing the comparison as shown in lines 1 and 4 of Algorithm 1.

A negative value is sent to disable either the charger or inverter as stated in lines 3 and 5 of Algorithm 1. As shown in lines 2 and 5 of Algorithm 1, 10W is deducted from the P_{Th} before sending to the charger or inverter. This is because the power control of the equipment is not ideal and may cause the P_{grid} to exceed the desired P_{Th}, which is particularly apparent for the inverter. Since the inverter is not designed for accurate

output power control, its output tends to have a power difference of less than 10W around the P_{Th}.

Algorithm 1: Inverter and charger control		
1:	**if** $P_{load} > (P_{Th} + 5)$ **then**	
2:	$P_{Th,Discharge} = P_{Th} - 10W$	
3:	$P_{Th,Charge} = -10W$	// negative value will disable the charger.
4	**elseif** $P_{load} < (P_{Th} - 5)$ **then**	
5:	$P_{Th,Discharge} = -10W$	// negative value will disable the inverter.
6:	$P_{Th,Charge} = P_{Th} - 10W$	

For power controls of the charger and inverter, they need P_{grid} with an interval shorter than the 5s. Hence, the ESP32 reads the P_{grid} from the ATM9E32 constantly at every ~1ms and is then published via MQTT so the Python (C) can use this real-time P_{grid} for charger control. This real-time P_{grid} reading is not accessed or logged by the local server but is only used by the charger and inverter for power control. The power control of the charger is done using K-controller to maintain the P_{grid} at the desired $P_{Th,Charge}$. The inverter achieves power control by comparing the real-time P_{grid} with the $P_{Th,Charge}$. If P_{grid} is higher than $P_{Th,Charge}$, it outputs 3.3V or HIGH. If P_{grid} is lower than $P_{Th,Charge}$, it outputs 0V or LOW.

A hysteresis setting is added to prevent triggering battery charging in the early morning before any peak reduction operations for the day. The battery is often fully charged as soon as possible without exceeding the P_{Th}. However, the inverter draws a small 0.1A or 5W of power from the battery during idle. This caused a slow discharge overnight and may trigger battery charging again. To prevent this, Algorithm 2 is used to check its fully charged status. The battery will always charge up to 226Ah and raise a *fully_charged_flag*. Due to the small 5W discharge, it will quickly drop to 225Ah, and the remaining capacity needs to drop to 224Ah to trigger charging again. The maximum DC charging power is limited to 500W in this experiment to prevent overheating of the batteries. The ambient temperature is relatively high at around 30°C due to the enormous amount of heat generated by the programmable AC load.

Algorithm 2: Inverter and charger control	
1:	**if** *fully_charged_flag* == 0 **and** Ah > 225 **then**
2:	*fully_charged_flag* = 1
3:	Stop charging.
4	**if** *fully_charged_flag* == 1 **and** Ah < 225 **then**
5:	*fully_charged_flag* = 0
6:	Start charging with P_{grid} <= $P_{Th,Charge}$

6.3 Results and Discussion

Figure 6.4 shows the experiment and simulation results on four days of *Dataset A*. It is to check if there is any abnormal behaviour in the peak demand reduction operations and the relationship between different parameters in both experiment and simulation. The differences in each parameter between the experiment and simulation results will be discussed later in Section 6.3.1. Both have different scales as the experiment scaled down to a peak power of 1.8kW. Overall, the P_{load}, P_{grid}, predicted day-ahead, and one-hour-ahead load profiles are very similar in both experiment and simulation results. Despite the high similarity, there are three significant differences between them.

First, the charging power after all peak reduction operations is reduced in the experiment. As mentioned in section 6.2.7, the maximum DC charging power is limited to 500W in this experiment. Since it only affects the charging after all peak reduction operations, it does not affect the PDRP.

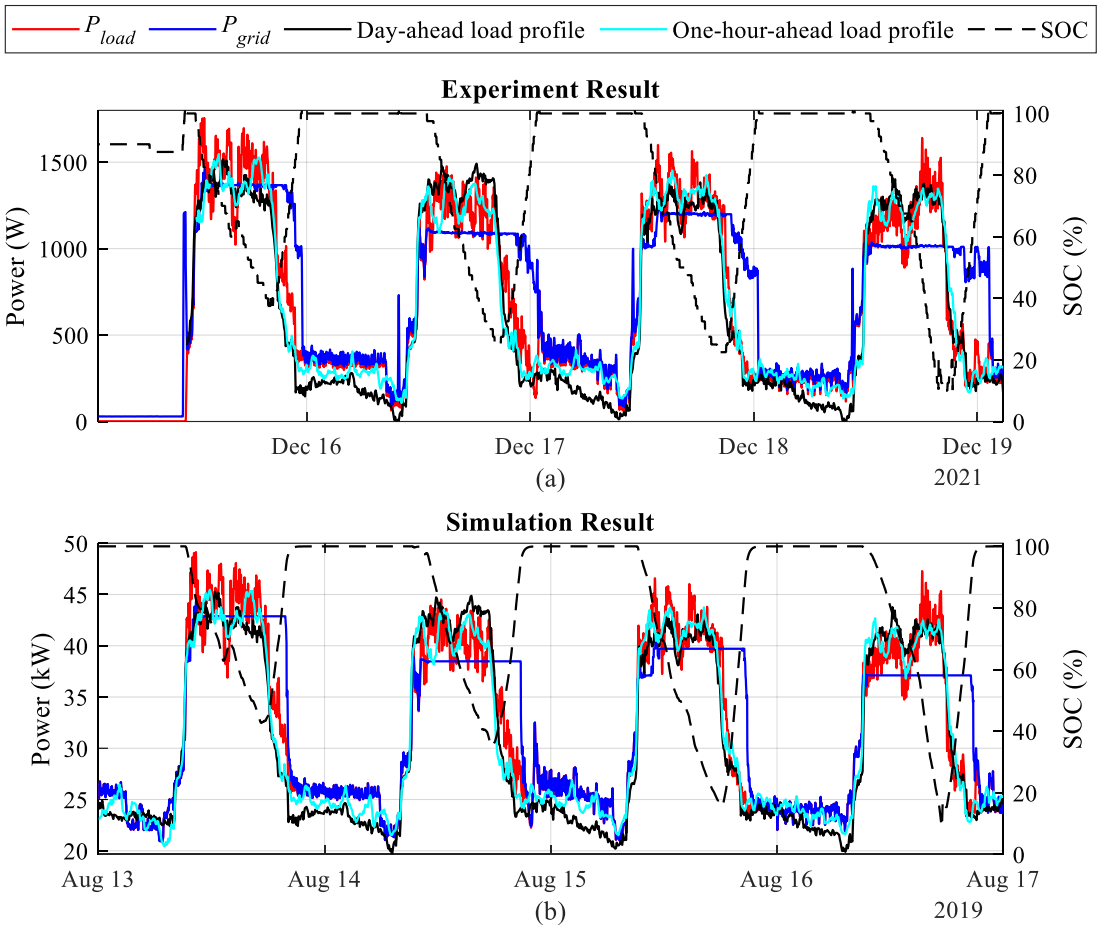


Figure 6.4: (a) Experiment and (b) Simulation results on four days of Dataset A

Second, the experiment results in Figure 6.4(a) has no power until 11:00 on 15th Dec 2021 because the experiment starts at 08:30 of *Dataset A* and has pre-load the 13th Aug 2019 00:00 – 08:30 load profile of *Dataset A* into Python (A). As shown in Figure 6.5(a), the battery is not fully charged at the beginning of the day. It needs to be fully charged before starting to ensure the battery is always fully charged before the next day in simulation. Hence, the battery is charged at 20ADC from 220Ah to 225Ah between 10:49 – 11:00, as shown in Figure 6.5(b). During this charging period, the $P_{BESS,AC}$ and $P_{BESS,DC}$ measures at 1217W and 1060.2W, respectively. This results in an efficiency of 87% at around 1kW output for the DC charger. As mentioned in Table 6.1, the AC-DC charger has an efficiency of 85% at 2.4kW. Since a typical efficiency curve of a converter is not flat but concave with peak efficiency at around 50% load, this slightly higher efficiency is expected. After the battery is fully charged, all systems are shut down at 11:02 and start at 11:10, which is when the 08:30 of day 1 of *Dataset A* begins.

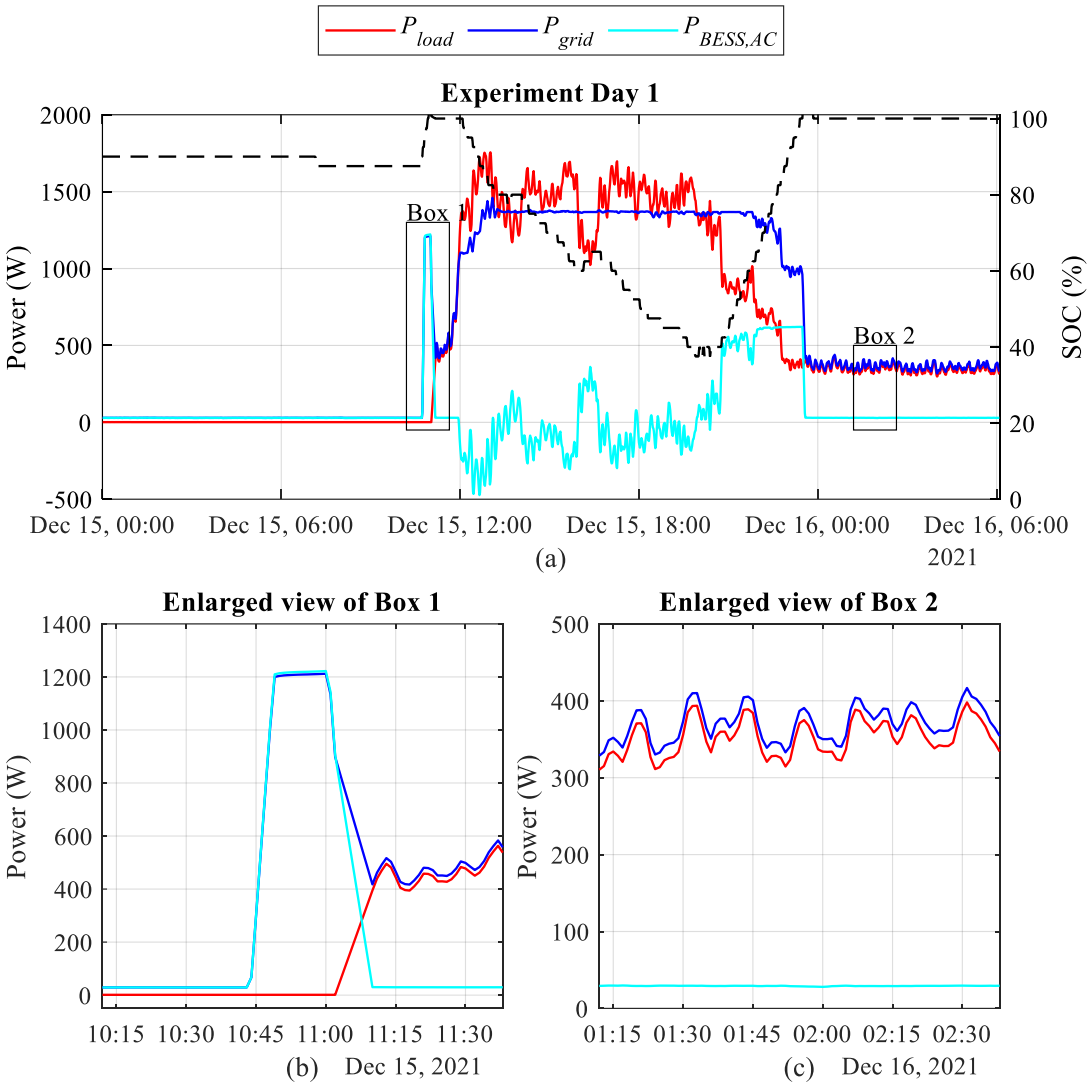


Figure 6.5: (a) Day 1 of Experiment result; (b) Enlarged view of Box 1 in (a); (c) Enlarged view of Box 2 in (a)

Third, the SOC curves are slightly different, especially the minimum SOC. The following factors might have played a role on the SOC curve: 1) the efficiency of the converters is lower than expected, 2) the 10W reduction in P_{Th} and, 3) the high 30W idle power of the AC-DC charger. As shown in Figure 5.6(b), the peak efficiency of the converter in simulation is assumed to be 90% and it stays constant until 30% of the converter's rated output power. As shown in Figure 6.4(a), the typical discharge and charge powers are within the range of 0-500W. However, both the SUN-2000GTIL2 inverter and Chroma 62024P-80-60 charger used in this experiment have relatively low efficiency at this range. Table 6.5 shows the SUN-2000GTLI2 has an efficiency range of 80% - 88% for an output power range of 100 – 500W. Table 6.6 shows the Chroma 62024P-80-60 has an efficiency range of 45% - 82% for an output power range of 50 –

500W. For a modern commercial hybrid inverter/charger that can perform discharge and charge, its efficiency is typically higher at higher than 90%. For example, MultiPlus-II by Victron Energy stated up to 96% efficiency in its datasheet [115].

Table 6.5: SUN-2000GTIL2 Inverter Efficiency Evaluation (without charger turned OFF)

Output Power (W)	**Input Power (W)**	**Efficiency (%)**
99.016	123.367	80.262
198.817	233.000	85.329
300.199	345.560	86.873
499.227	566.650	88.101

Table 6.6: Chroma 62024P-80-60 Programmable DC Power Supply Efficiency Evaluation

Output Power (W)	**Input Power (W)**	**Efficiency (%)**
50	109.786	45.543
100	162.522	61.530
200	265.204	75.414
300	384.380	78.048
400	495.381	80.746
500	604.319	82.738

Another thing to concern is the idle power of the equipment. As shown in Figure 6.5(c), the P_{grid} is slightly higher than P_{load} when both the inverter and charger are idling. Figure 6.6(a) shows the captured idling power readings at 5s intervals. The P_{grid} measures a 0.14W positive offset at no load. This may be due to the poor calibration for the CT measuring the P_{grid}. Since the offset is negligibly small, it is ignored.

As illustrated in Figure 6.6(b), the programmable AC load turns on at 09:33:00, and both the P_{grid} and P_{load} measures a draw of 1.2A. At 09:33:15, the inverter turns on and the $P_{BESS,AC}$ measures a small power of ~0.84A. However, the total idle power draw measured at P_{grid} is around 1.46A only, where it should be around 2A. This may be due to the CT being unable to capture small currents accurately. However, the idle powers of the programmable AC load and inverter are low, thus not concerned.

At 09:34:20, the AC-DC charger turned on, and both the P_{grid} and $P_{BESS,AC}$ measure a power draw of around 30W. This 30W idle power of the charger is

significantly higher than other equipment. This causes the inverter needs to output an extra 30W to compensate for this idle power during peak reductions, thus more BESS energy is required than in simulation.

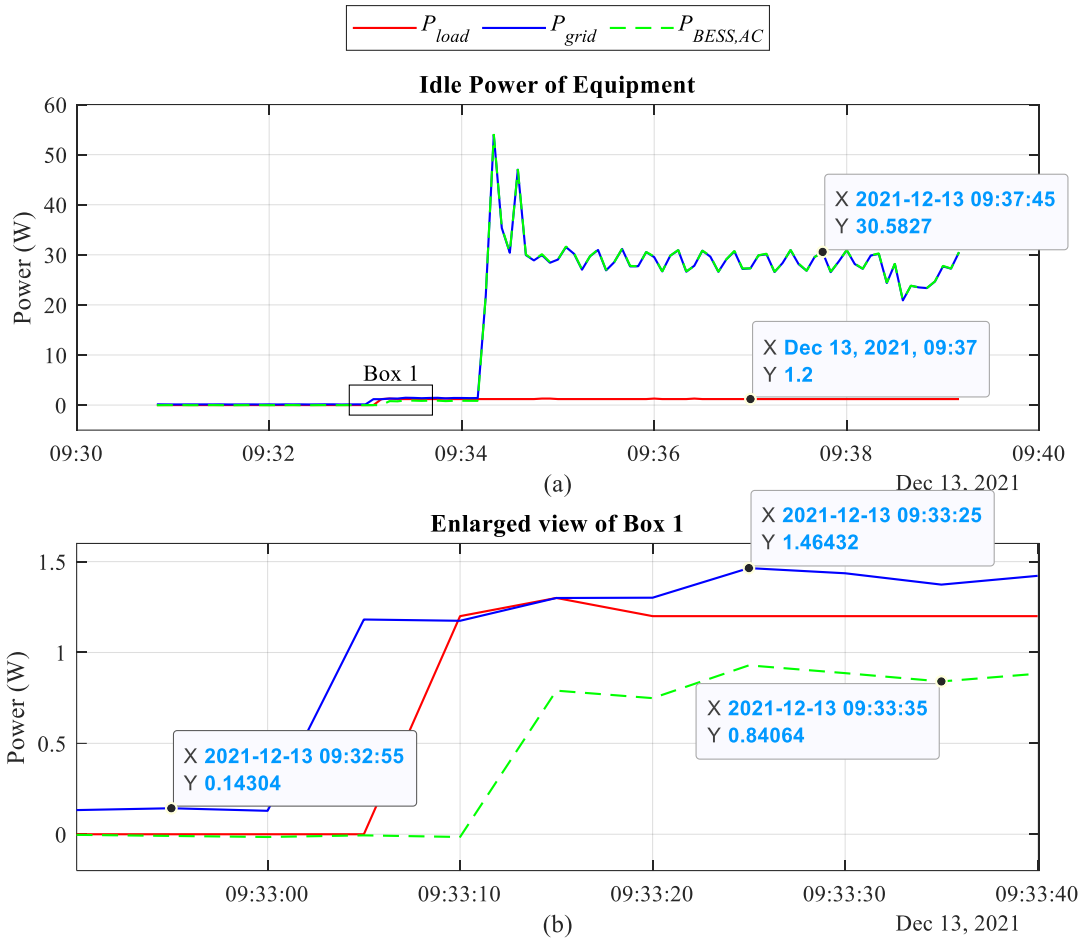


Figure 6.6: (a) Idle power of equipment; (b) Enlarged view of Box 1 in (a)

6.3.1 Validation of experiment and simulation results

The experiment and simulation results are validated through comparison. Both the experiment and simulation results should have negligible difference if correct. For direct comparison between experiment and simulation that has different power scales, the experiment readings are scaled back to kW using equation (29). Figure 6.7 shows the power curve between P_{load}, predicted dayahead load profile, predicted one-hour-ahead load profile, and P_{grid}. R^2 is used to quantify the similarity between experiment and simulation, with results shown in Table 6.7.

Table 6.7 shows the P_{load}, predicted dayahead load profile, and predicted one-hour-ahead load profile have almost negligible differences with very high R^2 values of

greater than 0.99. The P_{grid} has an R^2 value at 0.91, which is relatively low when compared to the other three measurements. The lower R^2 might be due to the difference in battery charging after peak reduction operations since this experiment has implemented a 500W DC charging power limit.

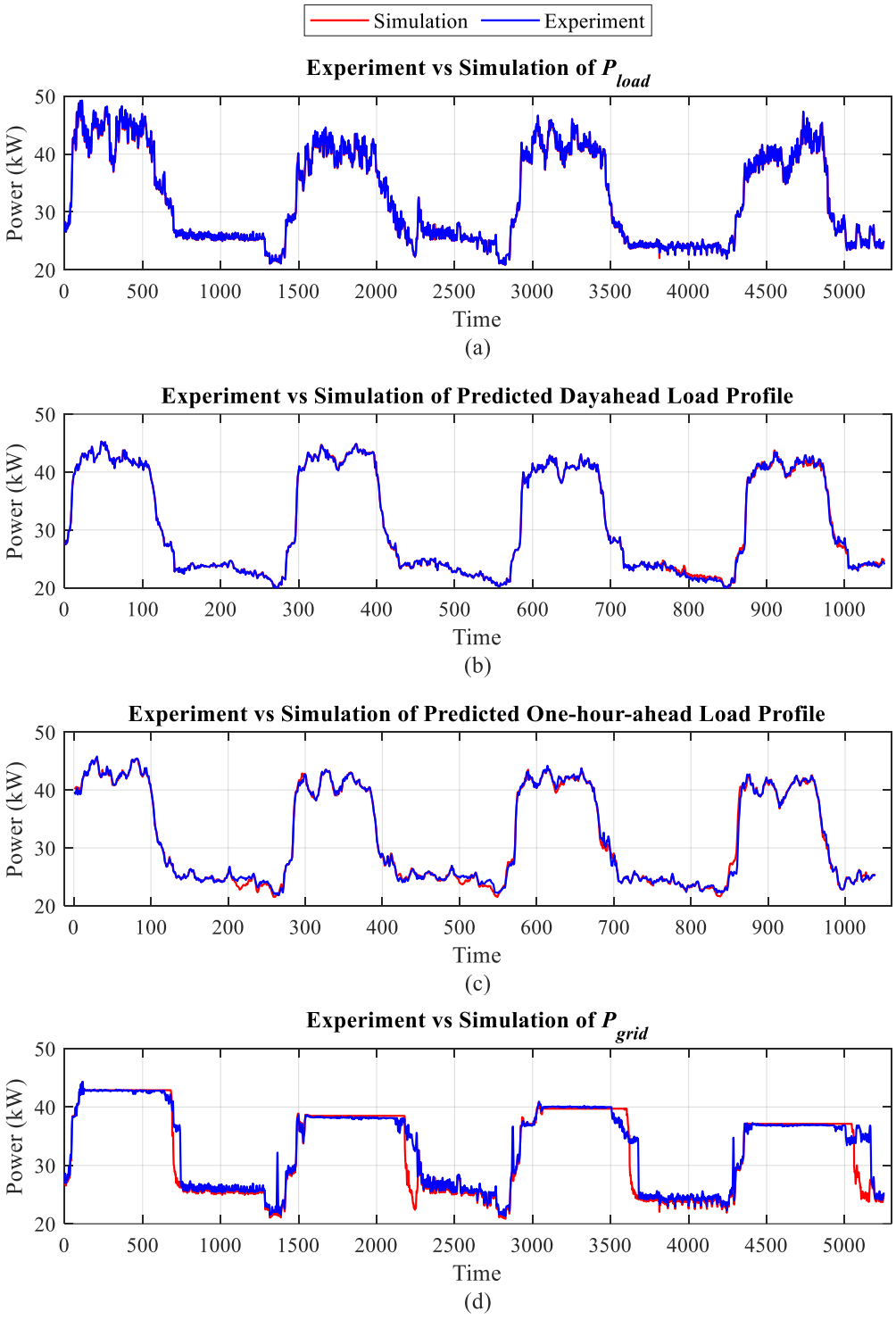


Figure 6.7: Experiment vs Simulation for (a) P_{load}, (b) predicted dayahead load profile, (c) predicted one-hour-ahead load profile, and (d) P_{grid}

Table 6.7: R^2 for measuring similarity between Experiment and Simulation

Measurements	**R^2**
Load Power	0.9999
Predicted Dayahead Load Profile	0.9993
Predicted One-hour-ahead Load Profile	0.9956
Grid Power	0.9082

Table 6.8 presents the differences in peak P_{grid} and peak P_{load} between experiment and simulation. For direct comparison between experiment and simulation that has different power scales, the experiment readings are scaled back to kW using equation (29). The differences are very small at less than 0.5kW. The highest difference is on day 3 where both the P_{grid} and P_{load} have differences of greater than 0.3kW.

The differences in PDRP and minimum SOC are given in Table 6.9. Although the experiment results show a higher PDRP than in simulation in 3 days, the differences in PDRP for all four days are negligible small at less than 0.5%. The minimum SOC has more apparent differences, especially for day 2 that has a difference of 10%. This may be due to the three factors discussed in Section 6.3. Nonetheless, there is no peak reduction failure and is acceptable.

Table 6.10 provides the differences in day-ahead MAPE and one-hour-ahead MAPE between experiment and simulation results. Since the experiment has no measured P_{load} or P_{grid} for load profile before 08:30 of day 1 of Dataset A. Both the MAPE of the experiment and simulation on day 1 are calculated using the profile from 08:30 – 11:59 only. The differences are negligible small with no differences higher than 0.5% MAPE.

As indicated in Table 6.8, Table 6.9, and Table 6.10, the differences between experiment and simulation are small and negligible. The only notable difference is in the minimum SOC. However, it is due to the worse than expected efficiency and unexpectedly high idle power of the converters. Using a better converter may be able to reduce the minimum SOC. Nonetheless, it has demonstrated the developed two-stage controllers not only work in simulation but also work similarly well in real-world applications.

Table 6.8: Differences in peak P_{grid} and peak P_{load} between experiment and simulation

Day	Peak P_{grid} (kW)			Peak P_{load} (kW)		
	Experiment	**Simulation**	**Difference**	**Experiment**	**Simulation**	**Difference**
1	42.957	42.876	0.081	47.227	46.970	0.257
2	38.290	38.482	-0.192	42.339	42.355	-0.016
3	40.152	39.707	0.445	44.059	43.727	0.332
4	37.018	37.111	-0.093	43.612	43.651	-0.039

Table 6.9: Differences in PDRP and minimum SOC between experiment and simulation

Day	PDRP (%)			Minimum SOC (%)		
	Experiment	**Simulation**	**Difference**	**Experiment**	**Simulation**	**Difference**
1	9.041	8.716	0.325	37.5	42.665	-5.165
2	9.564	9.144	0.419	25	35.084	-10.084
3	8.868	9.193	-0.326	22.5	16.154	6.346
4	15.120	14.982	0.137	10	8.949	1.051

Table 6.10: Differences in day-ahead MAPE and one-hour-ahead MAPE between experiment and simulation

Day	Day-ahead MAPE (%)			One-hour-ahead MAPE (%)		
	Experiment	**Simulation**	**Difference**	**Experiment**	**Simulation**	**Difference**
1	7.959	7.804	0.155	5.967	5.754	0.213
2	7.827	7.913	-0.086	5.873	6.178	-0.305
3	6.643	6.574	0.069	3.694	3.867	-0.173
4	6.467	5.996	0.471	4.767	4.385	0.382

6.3.2 Impact on the grid

Figure 6.8 depicts the PF_{grid}, PF_{load}, and V_{grid} recorded in this experiment. The purpose of this section is to demonstrate the impact on the grid after the implementation of the peak reduction system with the proposed two-stage controller . Figure 6.8(b) shows the PF_{load} has a constant PF of 1, which is in line with the configuration of the AC load. The PF_{grid} is always higher than 0.94 during peak reduction operations. After the BESS is fully charged, the PF_{grid} drops slightly when the inverter and charger are idling, such as between 15th Dec 2021 23:30 and 16th Dec 2021 08:31. However, the PF_{grid} is always higher than 0.8 throughout the low demand periods. The PF_{grid} drops to the range of 0.3 - 0.8 when the P_{grid} is between 100W - 200W between 16th Dec 08:31 and 16th Dec 10:30. This significant decrease in PF is normal with lower demand. As seen in Figure 6.8(c), the V_{grid} is between 230V - 240V throughout the experiment, which is also within the reasonable range for a 240VAC network. Hence, the peak reduction using BESS has not caused any abnormal behaviour.

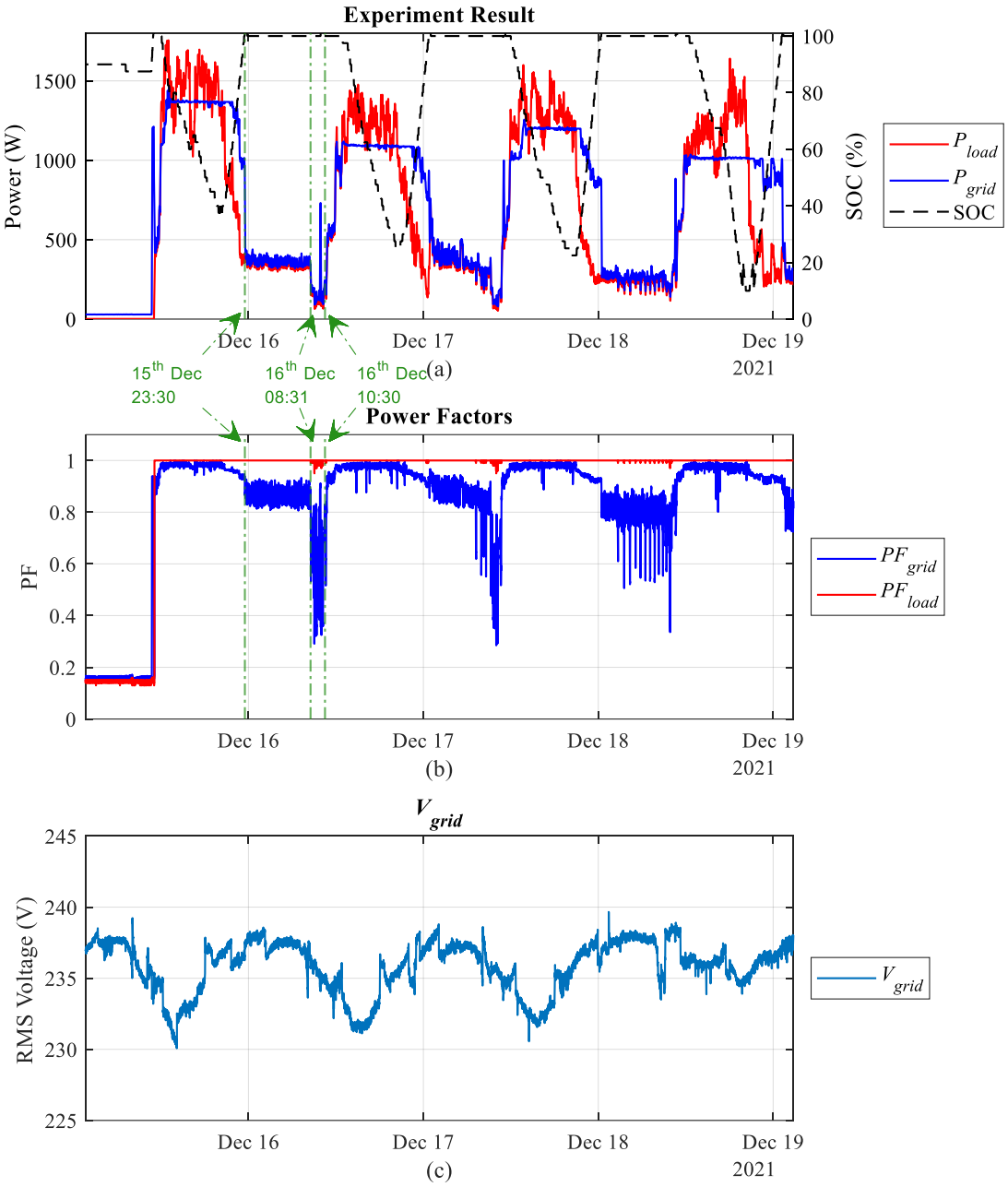


Figure 6.8: V_{grid} and Power Factors recorded in the experiment

6.4 Summary

The results obtained from the experiment in this chapter demonstrated maximum differences of 0.419%, 0.471%, and 10.084% in PDRP, MAPE, and minimum SOC, respectively. Therefore, it is concluded that there are negligible differences in the PDRP and MAPE between simulation and experiment results, except for the minimum SOC. The high difference in the minimum SOC may be due to the poor efficiency of the charger and inverter used in the experiment in this chapter. Nonetheless, the difference could be reduced further by using a more efficient converter. Apart from this, there are negligible differences in the predicted day-ahead and one-hour-ahead load profiles, with a very high R2 of higher than 0.99. These results show that there is a high similarity between the experiment and simulation results. The PDRP and MAPE for the four days show a difference of less than 0.5% between experiment and simulation results for the load profile of Block D at the University of Nottingham Malaysia (*Dataset A*). Therefore, this shows the simulation results are good enough to resemblance real-world condition and has shown the developed two-stage controllers not only works in simulation but also works similarly well in real-world applications.

Chapter 7 – Conclusion

7.1 Overview

To accomplish the research aims, this research work proposed one novel incremental load prediction model named DB-SOINN-R and one novel two-stage MD reduction controller. Both are required to achieve an adaptive MD reduction controller that is easy to implement and requires a small amount of historical data at the time of implementation. The proposed DB-SOINN-R prediction model and two-stage controller are first tested in simulation. They are tested on two different datasets to show their performance on various profiles. Then, their real-world performances were validated in a scaled-down experiment setup.

7.2 Conclusion

The simulation results demonstrate the proposed DB-SOINN-R outperforms FFNN, LSTM, SVR, kNN-R, and ESOINN on both day-ahead and one-hour-ahead load predictions on two datasets. The execution times of the day-ahead and one-hour-ahead DB-SOINN-R models are also faster than supervised FFNN, LSTM, and SVR models. Hence, the DB-SOINN-R is an efficient incremental model for day-ahead and one-hour-ahead time-series load predictions. With its unsupervised learning and incremental learning capability, it uses less computing resource and can adapt to new profile automatically as time progress.

The proposed DB-SOINN-R is then used as the prediction model for evaluating the performance of four controllers: fixed threshold, conventional single-stage, fuzzy, and the proposed two-stage controllers. Same as the evaluation of prediction models, the controllers are evaluated on two datasets. To demonstrate the adaptiveness of the controller, results that uses a larger BESS size on *Dataset B* are collected. The simulation results show the proposed-two stage controller has the highest MDRP and the highest total MD charge saving among the four tested controllers on both datasets, even with the doubled $BESS_{size}$ on *Dataset A*. The proposed two-stage controller has significantly better performance in all scenarios with up to 306.23% on *Dataset A* and 68.2% higher total MD charge saving on *Dataset B*. The proposed two-stage controller has a higher and more consistent MD reduction performances in all scenarios.

Finally, the completed proposed two-stage controller with the DB-SOINN-R as the prediction model is implemented in a scaled-down experiment setup to validate its real-world performance. Four days of experiment results are collected and compared with simulation results. The experiment results show negligible small differences of not more than 0.5% in daily PDRP and daily MAPE. The only significant difference observed is the experiment shows 10.084% lower minimum SOC on Day 2. The difference should be due to the poor efficiency of the converters used in the experiment, and the negative offset that is applied to the P_{Th} found by the two-stage controller to overcome the inaccurate power adjustment of the inverter. Overall, the experiment and simulation results are very similar. Thus, the developed two-stage controller should work similarly well in real-world applications. The day-ahead DB-SOINN-R prediction model, the one-hour-ahead DB-SOINN-R prediction model, and the two-stage control strategy are successfully implemented on a low-power processing unit, which is a Raspberry Pi 4.

7.3 Future Works

The research work demonstrated the capability of the proposed DB-SOINN-R incremental prediction model and the proposed dynamic two-stage controller in terms of ease-of-implementation and adaptability at different scenarios. However, further enhancements can be done to improve the performance of the proposed control strategy and minimize the complexity of the overall system in the future.

To validate its long-term real-world performance, the proposed controller can be implemented in an actual building using state-of-the-art converters that has a better efficiency with accuracy output power control. Although the proposed DB-SOINN-R shows a shorter training and prediction time than other supervised models, it can be further improved by adding parallel computing to utilize the multi-core CPU architecture of modern computing devices. The Raspberry pi 4 has a quad-core CPU, but the DB-SOINN-R only uses one core. Thus, adding parallel computing should be able to further reduce the learning time.

Since the applications of PV panels are gaining popularity, the application of the proposed two-stage controller can also be tested on a micro-grid system with PV to check its compatibility and perform modifications to the algorithm if needed. As PV panels have intermittent output power, it introduces more variations to the final load profile pattern seen by the grid operator. The proposed incremental DB-SOINN-R and dynamic two-stage controller should be useful at adapting to the latest load patterns automatically.

The drawback of the proposed MD reduction implementation is the initial cost of the BESS. Users may be reluctant to install BESS-based demand reduction systems due to the high initial cost and uncertainty in actual savings. Therefore, a business model can be applied where the demand reduction solution provider will install the complete systems, including inverters and BESS, for free. Then, the solution provider will get part of the monthly electricity bill savings as returns. This method allows users to enjoy savings without high upfront costs and risk but also expresses the confidence of solution providers in their system as it determines their profit margin and payback period.

Reference

[1] Suruhanjaya Tenaga, "Electricity - Final Electricity Consumption," *Malaysia Energy Information Hub*. https://meih.st.gov.my/statistics?p_auth=U6m5QNeu&p_p_id=Eng_Statistic_WAR_STOASPublicPortlet&p_p_lifecycle=1&p_p_state=maximized&p_p_mode=view&p_p_col_id=column-1&p_p_col_pos=1&p_p_col_count=2&_Eng_Statistic_WAR_STOASPublicPortlet_execution=e1s1&_Eng_Statistic_WAR_STOASPublicPortlet__eventId=ViewStatistic3&categoryId=4&flowId=7 (accessed Jun. 30, 2021).

[2] K. H. Chua, Y. S. Lim, and S. Morris, "Cost-benefit assessment of energy storage for utility and customers: A case study in Malaysia," *Energy Convers. Manag.*, vol. 106, pp. 1071–1081, Dec. 2015, doi: 10.1016/j.enconman.2015.10.041.

[3] M. Uddin, M. F. Romlie, M. F. Abdullah, S. Abd Halim, A. H. Abu Bakar, and T. Chia Kwang, "A review on peak load shaving strategies," *Renew. Sustain. Energy Rev.*, vol. 82, no. March 2018, pp. 3323–3332, 2018, doi: 10.1016/j.rser.2017.10.056.

[4] L. C. Hau, Y. S. Lim, and S. M. S. Liew, "A novel spontaneous self-adjusting controller of energy storage system for maximum demand reductions under penetration of photovoltaic system," *Appl. Energy*, vol. 260, p. 114294, Feb. 2020, doi: 10.1016/j.apenergy.2019.114294.

[5] J. Thakur and B. Chakraborty, "Demand side management in developing nations: A mitigating tool for energy imbalance and peak load management," *Energy*, vol. 114, pp. 895–912, Nov. 2016, doi: 10.1016/J.ENERGY.2016.08.030.

[6] N. G. Paterakis, O. Erdinç, and J. P. S. Catalão, "An overview of Demand Response: Key-elements and international experience," *Renewable and Sustainable Energy Reviews*, vol. 69. Elsevier Ltd, pp. 871–891, Mar. 01, 2017, doi: 10.1016/j.rser.2016.11.167.

[7] A. Mirakhorli and B. Dong, "Model predictive control for building loads connected with a residential distribution grid," *Appl. Energy*, vol. 230, pp. 627–642, Nov. 2018, doi: 10.1016/j.apenergy.2018.08.051.

[8] "Maximum Demand - Tenaga Nasional Berhad."

https://www.tnb.com.my/commercial-industrial/maximum-demand (accessed Jan. 06, 2020).

[9] "TNB Tariff Booklet." 2006.

[10] "Pricing & Tariffs - Tenaga Nasional Berhad." https://www.tnb.com.my/commercial-industrial/pricing-tariffs1 (accessed Jun. 25, 2021).

[11] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using Battery Storage for Peak Shaving and Frequency Regulation: Joint Optimization for Superlinear Gains," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 2882–2894, May 2018, doi: 10.1109/TPWRS.2017.2749512.

[12] A. S. Melinamane, S. Gadad, and S. Melinamani, "Smart Maximum Demand Controller for Consumer Loads," *Proc. B-HTC 2020 - 1st IEEE Bangalore Humanit. Technol. Conf.*, Oct. 2020, doi: 10.1109/B-HTC50970.2020.9297920.

[13] O. Ayan and B. Turkay, "Energy management algorithm for peak demand reduction," Aug. 2018, doi: 10.1109/SIELA.2018.8447088.

[14] E. Shirazi and S. Jadid, "Cost reduction and peak shaving through domestic load shifting and DERs," *Energy*, vol. 124, pp. 146–159, Apr. 2017, doi: 10.1016/J.ENERGY.2017.01.148.

[15] R. Faia, P. Faria, Z. Vale, and J. Spinola, "Demand response optimization using particle swarm algorithm considering optimum battery energy storage schedule in a residential house," *Energies*, vol. 12, no. 9, p. 1645, Apr. 2019, doi: 10.3390/en12091645.

[16] A. Roy, F. Auger, F. Dupriez-Robin, S. Bourguet, and Q. T. Tran, "A multi-level Demand-Side Management algorithm for offgrid multi-source systems," *Energy*, vol. 191, p. 116536, Jan. 2020, doi: 10.1016/J.ENERGY.2019.116536.

[17] P. Sheikhahmadi, R. Mafakheri, S. Bahramara, M. Y. Damavandi, and J. P. S. Catalaõ, "Risk-Based Two-Stage Stochastic Optimization Problem of Micro-Grid Operation with Renewables and Incentive-Based Demand Response Programs," *Energies 2018, Vol. 11, Page 610*, vol. 11, no. 3, p. 610, Mar. 2018, doi: 10.3390/EN11030610.

[18] H. Qi, H. Yue, J. Zhang, and K. L. Lo, “Optimisation of a smart energy hub with integration of combined heat and power, demand side response and energy storage,” *Energy*, vol. 234, p. 121268, Nov. 2021, doi: 10.1016/J.ENERGY.2021.121268.

[19] T. P. Imthias Ahamed and V. S. Borkar, “An efficient scheduling algorithm for solving load commitment problem under Time of Use Pricing with bound on Maximum Demand,” Feb. 2014, doi: 10.1109/PEDES.2014.7041986.

[20] A. J. Van Staden, J. Zhang, and X. Xia, “A model predictive control strategy for load shifting in a water pumping scheme with maximum demand charges,” 2009, doi: 10.1109/PTC.2009.5282271.

[21] S. Q. Ali, S. D. Maqbool, T. P. I. Ahamed, and N. H. Malik, “Pursuit algorithm for optimized load scheduling,” in *2012 IEEE International Power Engineering and Optimization Conference, PEOCO 2012 - Conference Proceedings*, 2012, pp. 193–198, doi: 10.1109/PEOCO.2012.6230859.

[22] S. Ahmad, A. Ahmad, M. Naeem, W. Ejaz, and H. S. Kim, “A Compendium of Performance Metrics, Pricing Schemes, Optimization Objectives, and Solution Methodologies of Demand Side Management for the Smart Grid,” *Energies 2018, Vol. 11, Page 2801*, vol. 11, no. 10, p. 2801, Oct. 2018, doi: 10.3390/EN11102801.

[23] M. Jacob, C. Neves, and D. Vukadinović Greetham, “Short Term Load Forecasting,” in *Forecasting and Assessing Risk of Individual Electricity Peaks*, Cham: Springer International Publishing, 2020, pp. 15–37.

[24] A. Arshad, H. M. Ali, A. Habib, M. A. Bashir, M. Jabbal, and Y. Yan, “Energy and exergy analysis of fuel cells: A review,” *Therm. Sci. Eng. Prog.*, vol. 9, pp. 308–321, Mar. 2019, doi: 10.1016/J.TSEP.2018.12.008.

[25] S. C. Kwan *et al.*, “Health impacts from TRAPs and carbon emissions in the projected electric vehicle growth and energy generation mix scenarios in Malaysia,” *Environ. Res.*, vol. 216, p. 114524, Jan. 2023, doi: 10.1016/J.ENVRES.2022.114524.

[26] Laws of Malaysia, “Environmental Quality Act 1974,” 1974.

[27] O. F. Ayodele, B. V. Ayodele, S. I. Mustapa, and Y. Fernando, “Effect of activation function in modeling the nexus between carbon tax, CO2 emissions, and gas-fired power plant parameters,” *Energy Convers. Manag. X*, vol. 12, Dec. 2021, doi: 10.1016/J.ECMX.2021.100111.

[28] G. G. Dranka, P. Ferreira, and A. I. F. Vaz, “Integrating supply and demand-side management in renewable-based energy systems,” *Energy*, vol. 232, p. 120978, Oct. 2021, doi: 10.1016/J.ENERGY.2021.120978.

[29] A. Nawaz, M. Zhou, J. Wu, and C. Long, “A comprehensive review on energy management, demand response, and coordination schemes utilization in multi-microgrids network,” *Appl. Energy*, vol. 323, p. 119596, Oct. 2022, doi: 10.1016/J.APENERGY.2022.119596.

[30] C. S. Ioakimidis, D. Thomas, P. Rycerski, and K. N. Genikomsakis, “Peak shaving and valley filling of power consumption profile in non-residential buildings using an electric vehicle parking lot,” *Energy*, vol. 148, pp. 148–158, Apr. 2018, doi: 10.1016/J.ENERGY.2018.01.128.

[31] A. Behzadi, S. M. Alirahmi, H. Yu, and S. Sadrizadeh, “An efficient renewable hybridization based on hydrogen storage for peak demand reduction: A rule-based energy control and optimization using machine learning techniques,” *J. Energy Storage*, vol. 57, p. 106168, Jan. 2023, doi: 10.1016/J.EST.2022.106168.

[32] L. C. Hau and Y. S. Lim, “A real-time active peak demand reduction for battery energy storage with limited capacity,” *J. Commun.*, vol. 11, no. 9, pp. 841–847, Sep. 2016, doi: 10.12720/jcm.11.9.841-847.

[33] C. Lange, A. Rueß, A. Nuß, R. Öchsner, and M. März, “Dimensioning battery energy storage systems for peak shaving based on a real-time control algorithm,” *Appl. Energy*, vol. 280, p. 115993, Dec. 2020, doi: 10.1016/j.apenergy.2020.115993.

[34] M. Shin, S. So, and K. Kim, “A Control Approach of Battery Energy Storage Systems to Reduce kW Demand,” *MATEC Web Conf.*, vol. 68, p. 10001, Aug. 2016, doi: 10.1051/matecconf/20166810001.

[35] K. H. Chua, Y. S. Lim, and S. Morris, “Energy storage system for peak shaving,”

Int. J. Energy Sect. Manag., vol. 10, no. 1, pp. 3–18, 2016, doi: 10.1108/IJESM-01-2015-0003.

[36] K. H. Chua, Y. S. Lim, and S. Morris, "A novel fuzzy control algorithm for reducing the peak demands using energy storage system," *Energy*, vol. 122, pp. 265–273, 2017, doi: 10.1016/j.energy.2017.01.063.

[37] R. Hanna, J. Kleissl, A. Nottrott, and M. Ferry, "Energy dispatch schedule optimization for demand charge reduction using a photovoltaic-battery storage system with solar forecasting," *Sol. Energy*, vol. 103, pp. 269–287, May 2014, doi: 10.1016/j.solener.2014.02.020.

[38] Y. Xue, M. Todd, S. Ula, M. J. Barth, and A. A. Martinez-Morales, "A comparison between two MPC algorithms for demand charge reduction in a real-world microgrid system," in *Conference Record of the IEEE Photovoltaic Specialists Conference*, Nov. 2016, vol. 2016-November, pp. 1875–1880, doi: 10.1109/PVSC.2016.7749947.

[39] D. C. Wu, A. Amini, A. Razban, and J. Chen, "ARC algorithm: A novel approach to forecast and manage daily electrical maximum demand," *Energy*, vol. 154, pp. 383–389, Jul. 2018, doi: 10.1016/J.ENERGY.2018.04.117.

[40] B. Yildiz, J. I. Bilbao, and A. B. Sproul, "A review and analysis of regression and machine learning models on commercial building electricity load forecasting," *Renew. Sustain. Energy Rev.*, vol. 73, no. February, pp. 1104–1122, 2017, doi: 10.1016/j.rser.2017.02.023.

[41] S.-K. Kim, J.-Y. Kim, K.-H. Cho, and G. Byeon, "Optimal Operation Control for Multiple BESSs of a Large-Scale Customer Under Time-Based Pricing," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 803–816, May 2017, doi: 10.1109/tpwrs.2017.2696571.

[42] R. T. De Salis, A. Clarke, Z. Wang, J. Moyne, and D. M. Tilbury, "Energy storage control for peak shaving in a single building," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2014-Octob, no. October, pp. 1–5, 2014, doi: 10.1109/PESGM.2014.6938948.

[43] E. Reihani, M. Motalleb, R. Ghorbani, and L. Saad Saoud, "Load peak shaving

and power smoothing of a distribution grid with high renewable energy penetration," *Renew. Energy*, vol. 86, pp. 1372–1379, Feb. 2016, doi: 10.1016/j.renene.2015.09.050.

[44] M. Rowe, T. Yunusov, S. Haben, W. Holderbaum, and B. Potter, "The real-time optimisation of DNO owned storage devices on the LV network for peak reduction," *Energies*, vol. 7, no. 6, pp. 3537–3560, 2014, doi: 10.3390/en7063537.

[45] C. Lu, H. Xu, X. Pan, and J. Song, "Optimal sizing and control of battery energy storage system for peak load shaving," *Energies*, vol. 7, no. 12, pp. 8396–8410, 2014, doi: 10.3390/en7128396.

[46] G. Bao, C. Lu, Z. Yuan, and Z. Lu, "Battery energy storage system load shifting control based on real time load forecast and dynamic programming," *IEEE Int. Conf. Autom. Sci. Eng.*, pp. 815–820, 2012, doi: 10.1109/CoASE.2012.6386377.

[47] D. Dongol, T. Feldmann, and E. Bollin, "A model predictive control based peak shaving application for a grid connected household with photovoltaic and battery storage," in *SMARTGREENS 2018 - Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems*, 2018, vol. 2018-March, pp. 54–63, doi: 10.5220/0006685300540063.

[48] Y. Hida, R. Yokoyama, J. Shimizukawa, K. Iba, K. Tanaka, and T. Seki, "Load following operation of NAS battery by setting statistic margins to avoid risks," *IEEE PES Gen. Meet. PES 2010*, pp. 1–5, 2010, doi: 10.1109/PES.2010.5588170.

[49] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, "Short term load forecasting and the effect of temperature at the low voltage level," *Int. J. Forecast.*, vol. 35, no. 4, pp. 1469–1484, Oct. 2019, doi: 10.1016/J.IJFORECAST.2018.10.007.

[50] M. Uddin, M. F. Romlie, M. F. Abdullah, C. K. Tan, G. M. Shafiullah, and A. H. A. Bakar, "A novel peak shaving algorithm for islanded microgrid using battery energy storage system," *Energy*, vol. 196, p. 117084, Apr. 2020, doi: 10.1016/j.energy.2020.117084.

[51] S. Yilmaz, J. Chambers, and M. K. Patel, "Comparison of clustering approaches

for domestic electricity load profile characterisation - Implications for demand side management," *Energy*, vol. 180, pp. 665–677, Aug. 2019, doi: 10.1016/J.ENERGY.2019.05.124.

[52] Z. Wang, T. Hong, and M. A. Piette, "Building thermal load prediction through shallow machine learning and deep learning," *Appl. Energy*, vol. 263, p. 114683, Apr. 2020, doi: 10.1016/j.apenergy.2020.114683.

[53] A. Zeng, S. Liu, and Y. Yu, "Comparative study of data driven methods in building electricity use prediction," *Energy Build.*, vol. 194, pp. 289–300, Jul. 2019, doi: 10.1016/j.enbuild.2019.04.029.

[54] T. Liu, Z. Tan, C. Xu, H. Chen, and Z. Li, "Study on deep reinforcement learning techniques for building energy consumption forecasting," *Energy Build.*, vol. 208, p. 109675, Feb. 2020, doi: 10.1016/j.enbuild.2019.109675.

[55] R. D. Rathor and A. Bharagava, "Short Term Load Forecasting of a Region of India using Generalized Regression Neural Network," *Glob. J. Res. Eng.*, vol. 17, no. 7, pp. 15–22, 2017, Accessed: Oct. 03, 2019. [Online]. Available: https://engineeringresearch.org/index.php/GJRE/article/view/1714/1645.

[56] H. Chitsaz, H. Shaker, H. Zareipour, D. Wood, and N. Amjady, "Short-term electricity load forecasting of buildings in microgrids," *Energy Build.*, vol. 99, pp. 50–60, 2015, doi: 10.1016/j.enbuild.2015.04.011.

[57] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, vol. 11, no. 7. 2018, doi: 10.3390/en11071636.

[58] D. Koschwitz, J. Frisch, and C. van Treeck, "Data-driven heating and cooling load predictions for non-residential buildings based on support vector machine regression and NARX Recurrent Neural Network: A comparative study on district scale," *Energy*, vol. 165, pp. 134–142, 2018, doi: 10.1016/j.energy.2018.09.068.

[59] C. Fan, J. Wang, W. Gang, and S. Li, "Assessment of deep recurrent neural network-based strategies for short-term building energy predictions," *Appl.*

Energy, vol. 236, pp. 700–710, Feb. 2019, doi: 10.1016/j.apenergy.2018.12.004.

[60] M. Cai, M. Pipattanasomporn, and S. Rahman, "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques," *Appl. Energy*, vol. 236, no. December 2018, pp. 1078–1088, 2019, doi: 10.1016/j.apenergy.2018.12.042.

[61] J. Bedi and D. Toshniwal, "Deep learning framework to forecast electricity demand," *Appl. Energy*, vol. 238, pp. 1312–1326, Mar. 2019, doi: 10.1016/j.apenergy.2019.01.113.

[62] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019, doi: 10.1109/TSG.2017.2753802.

[63] Y. Fu, Z. Li, H. Zhang, and P. Xu, "Using Support Vector Machine to Predict Next Day Electricity Load of Public Buildings with Sub-metering Devices," *Procedia Eng.*, vol. 121, pp. 1016–1022, 2015, doi: 10.1016/j.proeng.2015.09.097.

[64] L. W. Chong, D. Rengasamy, Y. W. Wong, and R. K. Rajkumar, "Load prediction using support vector regression," in *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, Dec. 2017, vol. 2017-December, pp. 1069–1074, doi: 10.1109/TENCON.2017.8228016.

[65] Y. Chen, P. Xu, Y. Chu, W. Li, Y. Wu, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings," *Appl. Energy*, vol. 195, pp. 659–670, Jun. 2017, doi: 10.1016/j.apenergy.2017.03.034.

[66] J. Moon, J. Park, E. Hwang, and S. Jun, "Forecasting power consumption for higher educational institutions based on machine learning," *J. Supercomput.*, vol. 74, no. 8, pp. 3778–3800, Aug. 2018, doi: 10.1007/s11227-017-2022-x.

[67] R. Wang, S. Lu, and W. Feng, "A novel improved model for building energy consumption prediction based on model integration," *Appl. Energy*, vol. 262, p. 114561, Mar. 2020, doi: 10.1016/j.apenergy.2020.114561.

[68] N. J. Johannesen, M. L. Kolhe, and M. Goodwin, “Smart load prediction analysis for distributed power network of Holiday Cabins in Norwegian rural area,” *J. Clean. Prod.*, vol. 266, p. 121423, Sep. 2020, doi: 10.1016/j.jclepro.2020.121423.

[69] G.-F. Fan, Y.-H. Guo, J.-M. Zheng, and W.-C. Hong, “Application of the Weighted K-Nearest Neighbor Algorithm for Short-Term Load Forecasting,” *Energies*, vol. 12, no. 5, p. 916, Mar. 2019, doi: 10.3390/en12050916.

[70] A. D. Pham, N. T. Ngo, T. T. Ha Truong, N. T. Huynh, and N. S. Truong, “Predicting energy consumption in multiple buildings using machine learning for improving energy efficiency and sustainability,” *J. Clean. Prod.*, vol. 260, p. 121082, Jul. 2020, doi: 10.1016/j.jclepro.2020.121082.

[71] X. J. Luo, L. O. Oyedele, A. O. Ajayi, C. G. Monyei, O. O. Akinade, and L. A. Akanbi, “Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands,” *Adv. Eng. Informatics*, vol. 41, no. March, p. 100926, 2019, doi: 10.1016/j.aei.2019.100926.

[72] J. Verhelst, G. Van Ham, D. Saelens, and L. Helsen, “Model selection for continuous commissioning of HVAC-systems in office buildings: A review,” *Renewable and Sustainable Energy Reviews*, vol. 76. Elsevier Ltd, pp. 673–686, Sep. 01, 2017, doi: 10.1016/j.rser.2017.01.119.

[73] X. J. Luo, L. O. Oyedele, A. O. Ajayi, C. G. Monyei, O. O. Akinade, and L. A. Akanbi, “Development of an IoT-based big data platform for day-ahead prediction of building heating and cooling demands,” *Adv. Eng. Informatics*, vol. 41, Aug. 2019, doi: 10.1016/j.aei.2019.100926.

[74] N. Aste, M. Manfren, and G. Marenzi, “Building Automation and Control Systems and performance optimization: A framework for analysis,” *Renewable and Sustainable Energy Reviews*, vol. 75. Elsevier Ltd, pp. 313–330, Aug. 01, 2017, doi: 10.1016/j.rser.2016.10.072.

[75] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, and Y. C. Tseng, “Statistical investigations of transfer learning-based methodology for short-term building energy predictions,” *Appl. Energy*, vol. 262, p. 114499, Mar. 2020, doi: 10.1016/j.apenergy.2020.114499.

[76] J. Moon, J. Kim, P. Kang, and E. Hwang, “Solving the Cold-Start Problem in Short-Term Load Forecasting Using Tree-Based Methods,” *Energies*, vol. 13, no. 4, p. 886, Feb. 2020, doi: 10.3390/en13040886.

[77] X. Li, Y. Grandvalet, and F. Davoine, “A baseline regularization scheme for transfer learning with convolutional neural networks,” *Pattern Recognit.*, vol. 98, p. 107049, Feb. 2020, doi: 10.1016/j.patcog.2019.107049.

[78] S. Furao and O. Hasegawa, “An incremental network for on-line unsupervised classification and topology learning,” *Neural Networks*, vol. 19, no. 1, pp. 90–106, Jan. 2006, doi: 10.1016/j.neunet.2005.04.006.

[79] S. Furao, T. Ogura, and O. Hasegawa, “An enhanced self-organizing incremental neural network for online unsupervised learning,” *Neural Networks*, vol. 20, no. 8, pp. 893–903, Oct. 2007, doi: 10.1016/j.neunet.2007.07.008.

[80] F. Shen and O. Hasegawa, “A fast nearest neighbor classifier based on self-organizing incremental neural network,” *Neural Networks*, vol. 21, no. 10, pp. 1537–1547, Dec. 2008, doi: 10.1016/j.neunet.2008.07.001.

[81] F. Shen, H. Yu, K. Sakurai, and O. Hasegawa, “An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network,” *Neural Comput. Appl.*, vol. 20, no. 7, pp. 1061–1074, Oct. 2011, doi: 10.1007/s00521-010-0428-y.

[82] H. Zhang, X. Xiao, and O. Hasegawa, “A load-balancing self-organizing incremental neural network,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 6, pp. 1096–1105, 2014, doi: 10.1109/TNNLS.2013.2287884.

[83] Y. Nakamura and O. Hasegawa, “Nonparametric density estimation based on self-organizing incremental neural network for large noisy data,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 28, no. 1, pp. 8–17, Jan. 2017, doi: 10.1109/TNNLS.2015.2489225.

[84] C. Wiwatcharakoses and D. Berrar, “SOINN+, a Self-Organizing Incremental Neural Network for Unsupervised Learning from Noisy Data Streams,” *Expert Syst. Appl.*, vol. 143, p. 113069, Apr. 2020, doi: 10.1016/j.eswa.2019.113069.

[85] T. Avdeenko and O. Khateev, “Taxi service pricing based on online machine

learning," in *Communications in Computer and Information Science*, Jul. 2019, vol. 1071, pp. 289–299, doi: 10.1007/978-981-32-9563-6_30.

[86] K. W. Kow, Y. W. Wong, R. Rajkumar, and D. Isa, "An intelligent real-time power management system with active learning prediction engine for PV grid-tied systems," *J. Clean. Prod.*, vol. 205, pp. 252–265, Dec. 2018, doi: 10.1016/j.jclepro.2018.09.084.

[87] B. K. Puah, L. W. Chong, Y. W. Wong, K. M. Begam, N. Khan, M. A. Juman, and R. K. Rajkumar, "A regression unsupervised incremental learning algorithm for solar irradiance prediction," *Renew. Energy*, vol. 164, pp. 908–925, Feb. 2021, doi: 10.1016/j.renene.2020.09.080.

[88] M. H. Pham, T. A. T. Vu, T. H. Dang, D. Q. Nguyen, V. H. Dang, N. T. Nguyen, T. V. Nguyen, and V. D. Pham, "An Effective Approach to ANN-Based Short-Term Load Forecasting Model Using Hybrid Algorithm GA-PSO," *Proc. - 2018 IEEE Int. Conf. Environ. Electr. Eng. 2018 IEEE Ind. Commer. Power Syst. Eur. EEEIC/I CPS Eur. 2018*, no. April, 2018, doi: 10.1109/EEEIC.2018.8493908.

[89] B. Ekici, C. Cubukcuoglu, M. Turrin, and I. S. Sariyildiz, "Performative computational architecture using swarm and evolutionary optimisation: A review," *Build. Environ.*, vol. 147, pp. 356–371, Jan. 2019, doi: 10.1016/J.BUILDENV.2018.10.023.

[90] K. A. Barber and M. Krarti, "A review of optimization based tools for design and control of building energy systems," *Renew. Sustain. Energy Rev.*, vol. 160, p. 112359, May 2022, doi: 10.1016/J.RSER.2022.112359.

[91] F. D. Wihartiko, H. Wijayanti, and F. Virgantari, "Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem," *{IOP} Conf. Ser. Mater. Sci. Eng.*, vol. 332, p. 12020, Mar. 2018, doi: 10.1088/1757-899x/332/1/012020.

[92] Y. Ding, W. Zhang, L. Yu, and K. Lu, "The accuracy and efficiency of GA and PSO optimization schemes on estimating reaction kinetic parameters of biomass pyrolysis," *Energy*, vol. 176, pp. 582–588, Jun. 2019, doi: 10.1016/J.ENERGY.2019.04.030.

[93] S. M. H. Mousakazemi, “Computational effort comparison of genetic algorithm and particle swarm optimization algorithms for the proportional–integral–derivative controller tuning of a pressurized water nuclear reactor,” *Ann. Nucl. Energy*, vol. 136, p. 107019, Feb. 2020, doi: 10.1016/J.ANUCENE.2019.107019.

[94] T. Y. Lee, “Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: A multipass iteration particle swarm optimization approach,” *IEEE Trans. Energy Convers.*, vol. 22, no. 3, pp. 774–782, Sep. 2007, doi: 10.1109/TEC.2006.878239.

[95] L. W. Chong, Y. W. Wong, R. K. Rajkumar, and D. Isa, “An adaptive learning control strategy for standalone PV system with battery-supercapacitor hybrid energy storage system,” *J. Power Sources*, vol. 394, pp. 35–49, Aug. 2018, doi: 10.1016/j.jpowsour.2018.05.041.

[96] M. R. Bonyadi and Z. Michalewicz, “Particle Swarm Optimization for Single Objective Continuous Space Problems: A Review,” *Evol. Comput.*, vol. 25, no. 1, pp. 1–54, Mar. 2017, doi: 10.1162/EVCO_R_00180.

[97] M. S. Wasim, M. Amjad, S. Habib, M. A. Abbasi, A. R. Bhatti, and S. M. Muyeen, “A critical review and performance comparisons of swarm-based optimization algorithms in maximum power point tracking of photovoltaic systems under partial shading conditions,” *Energy Reports*, vol. 8, pp. 4871–4898, Nov. 2022, doi: 10.1016/J.EGYR.2022.03.175.

[98] S. N. Ghoreishi, A. Clausen, and B. N. Joergensen, “Termination criteria in evolutionary algorithms: A survey,” *IJCCI 2017 - Proc. 9th Int. Jt. Conf. Comput. Intell.*, pp. 373–384, 2017, doi: 10.5220/0006577903730384.

[99] X. Kan, L. Reichenberg, and F. Hedenus, “The impacts of the electricity demand pattern on electricity system cost and the electricity supply mix: A comprehensive modeling analysis for Europe,” *Energy*, vol. 235, p. 121329, Nov. 2021, doi: 10.1016/J.ENERGY.2021.121329.

[100] E. Reihani, M. Motalleb, R. Ghorbani, and L. Saad Saoud, “Load peak shaving and power smoothing of a distribution grid with high renewable energy penetration,” *Renew. Energy*, vol. 86, pp. 1372–1379, 2016, doi: 10.1016/j.renene.2015.09.050.

[101] R. W. Ng, K. M. Begam, R. K. Rajkumar, Y. W. Wong, and L. W. Chong, “An improved self-organizing incremental neural network model for short-term time-series load prediction,” *Appl. Energy*, vol. 292, p. 116912, Jun. 2021, doi: 10.1016/j.apenergy.2021.116912.

[102] L. C. Hau and Y. S. Lim, “Proposed method for evaluating controllers of battery-based storage system in maximum demand reductions,” *J. Energy Storage*, vol. 46, p. 103850, Feb. 2022, doi: 10.1016/J.EST.2021.103850.

[103] F. Guimaraes and H. Javedani Sadaei, “Data for: Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series,” vol. 1, 2019, doi: 10.17632/F4FCRH4TN9.1.

[104] Y. Ding, Q. Zhang, T. Yuan, and K. Yang, “Model input selection for building heating load prediction: A case study for an office building in Tianjin,” *Energy Build.*, vol. 159, pp. 254–270, Jan. 2018, doi: 10.1016/j.enbuild.2017.11.002.

[105] J. Zhao and X. Liu, “A hybrid method of dynamic cooling and heating load forecasting for office buildings based on artificial intelligence and regression analysis,” *Energy Build.*, vol. 174, pp. 293–308, 2018, doi: 10.1016/j.enbuild.2018.06.050.

[106] M. Fopa, M. Gueye, S. Ndiaye, and H. Naacke, “A parameter-free KNN for rating prediction,” *Data Knowl. Eng.*, vol. 142, Nov. 2022, doi: 10.1016/J.DATAK.2022.102095.

[107] S. Farah, W. David A, N. Humaira, Z. Aneela, and E. Steffen, “Short-term multi-hour ahead country-wide wind power prediction for Germany using gated recurrent unit deep learning,” *Renew. Sustain. Energy Rev.*, vol. 167, p. 112700, Oct. 2022, doi: 10.1016/J.RSER.2022.112700.

[108] J. Lin, J. A. Fernández, R. Rayhana, A. Zaji, R. Zhang, O. E. Herrera, Z. Liu, and W. Mérida, “Predictive analytics for building power demand: Day-ahead forecasting and anomaly prediction,” *Energy Build.*, vol. 255, p. 111670, Jan. 2022, doi: 10.1016/J.ENBUILD.2021.111670.

[109] “Fit a support vector machine regression model - MATLAB fitrsvm.” https://www.mathworks.com/help/stats/fitrsvm.html#d122e366104 (accessed

Sep. 19, 2020).

[110] H. J. Khasawneh, A. Mondal, M. S. Illindala, B. L. Schenkman, and D. R. Borneo, “Evaluation and sizing of energy storage systems for microgrids,” *2015 IEEE/IAS 51st Ind. Commer. Power Syst. Tech. Conf. I CPS 2015*, pp. 1–8, 2015, doi: 10.1109/ICPS.2015.7266408.

[111] “Watts-on’s DIY GTIL Controller | Second Life Storage & Solar.” https://secondlifestorage.com/index.php?threads/watts-ons-diy-gtil-controller.9265/#pid63179 (accessed Dec. 30, 2021).

[112] “Split-Single-Phase-Energy-Meter/Home_Assistant_MQTT.ino at master · CircuitSetup/Split-Single-Phase-Energy-Meter · GitHub.” https://github.com/CircuitSetup/Split-Single-Phase-Energy-Meter/blob/master/Software/examples/Home_Assistant_MQTT/Home_Assistant_MQTT/Home_Assistant_MQTT.ino (accessed Sep. 10, 2021).

[113] “JCSCatena/PZEM004Tv30_MODBUS: Enables communication to Peacefair PZEM-004T v3.0 Power and Energy monitor with the Modbus protocol.” https://github.com/JCSCatena/PZEM004Tv30_MODBUS (accessed Jan. 03, 2022).

[114] “PZEM-0xx power monitor - Tasmota.” https://tasmota.github.io/docs/PZEM-0XX/#hardware-connections (accessed Jan. 03, 2022).

[115] “\ MultiPlus-II Inverter/Charger.” https://www.victronenergy.com/upload/documents/Datasheet-MultiPlus-II-inverter-charger-EN.pdf (accessed Jan. 08, 2022).

Appendix 1 – Algorithm Pseudocode

Algorithm 3: ESOINN

Initialize $\mathcal{N}$ with two nodes $\{i, j\}$ selected from the training data.
Initialize $T_a = \|i - j\|, (\forall a \in \mathcal{N})$
Initialize $WT_a = 0, (\forall a \in \mathcal{N})$
for each new training set (ξ) **do**
    $N_1 = \min_{a \in \mathcal{N}} \|\xi - N_a\|$ // find 1st winner
    $N_2 = \min_{a \in \mathcal{N} \setminus \{N_1\}} \|\xi - N_a\|$ // find 2nd winner
    if $\|\xi - N_1\| > T_{N_1}$ **or** $\|\xi - N_2\| > T_{N_2}$ **then**
        $\mathcal{N} = \mathcal{N} \cup \xi$ // ξ inserted as new node
    else
        $N_1 = N_1 + \frac{1}{WT_{N_1}}(\xi - N_1)$
        $N_i = N_i + \frac{1}{(100)WT_{N_1}}(\xi - N_i), (\forall N_i \in \mathcal{P}_{N_1})$
        $WT_{N_1} = WT_{N_1} + 1$
        Connect N_1 and N_2 with edge. If edge exists, reset age of the edge.
        Increase age of all edges associated to N_1.
        Remove edge if age of edge > age_{max}
    end if
    Update T_{N_1} and T_{N_2} using equations (6) and (7), respectively.
    if the number of trained data is a multiple of λ **then**
        Update subclass.
        Remove node a if $\mathcal{p}_a = 0, (\forall a \in \mathcal{N})$
        $WT_{mean} = \frac{1}{|\mathcal{N}|}\sum_i^{|\mathcal{N}|} WT_i$
        Remove node b if $\mathcal{p}_b = 1$ and $WT_b < C_1 \times WT_{mean}, (\forall b \in \mathcal{N})$
        Remove node c if $\mathcal{p}_c = 2$ and $WT_c < C_2 \times WT_{mean}, (\forall c \in \mathcal{N})$
    end if
end for

Algorithm 4: Proposed DB-SOINN

Initialize $\mathcal{N}$ with two training sets $\{i, j\}$ selected from the training data sets.
Initialize $T_a = \|i - j\|, (\forall a \in \mathcal{N})$
Initialize $WT_a = 0, (\forall a \in \mathcal{N})$
Initialize $LT_a = 1, (\forall a \in \mathcal{N})$
for each new training set (ξ) **do**
    $N_1 = \min_{a \in \mathcal{N}} \|\xi - N_a\|$ // find 1st winner
    $N_2 = \min_{a \in \mathcal{N} \setminus \{N_1\}} \|\xi - N_a\|$ // find 2nd winner
    if $\|\xi - N_1\| \leq T_{N_1}$ **and** $\|\xi - N_2\| > T_{N_2}$ **then**
        $\mathcal{N} = \mathcal{N} \cup \xi$ // ξ inserted as new node
        $d_\xi = 0$ // initialize *d* to 0 for new node
        $LT_\xi = 0$ // initialize LT to 0 for new node
        Connect N_1 and ξ with edge.
    elseif $\|\xi - N_1\| > T_{N_1}$ **or** $\|\xi - N_2\| > T_{N_2}$ **then**
        $\mathcal{N} = \mathcal{N} \cup \xi$ // ξ inserted as new node
        $d_\xi = 0$ // initialize *d* to 0 for new node
        $LT_\xi = 0$ // initialize LT to 0 for new node
    elseif $\|\xi - N_1\| < T_{N_1}$ **and** $\|\xi - N_2\| < T_{N_2}$ **then**
        $N_1 = N_1 + \frac{1}{WT_{N_1}}(\xi - N_1)$
        $N_i = N_i + \frac{1}{(100)WT_{N_1}}(\xi - N_i), (\forall N_i \in \mathcal{P}_{N_1})$
        $WT_{N_1} = WT_{N_1} + 1$
        Connect N_1 and N_2 with edge. If edge exists, reset age of the edge.
        Update d_{N_1} with equation (11) // update node density of N_1
        Increase age of all edges associated to N_1.
        Removes edge if age of edge > age_{max}
    end if
    Update T_{N_1} and T_{N_2} using equations (6) and (7), respectively.
    $LT_a = LT_a + 1, (\forall a \in \mathcal{N})$ // increase the lifetime of all nodes.
    if the number of trained data is the multiple of λ **then**
        for $\forall a \in \mathcal{N}$ **then**
            Removes node *a* **if** $\wp_a$ = 0 **and** $LT_a > \lambda$
            for node *a* with $\wp_a \leq 2$ **and** $LT_a > \lambda$ **then**
                Calculate D_a using equation (12).
                Calculate $D_b, (\forall b \in \mathcal{K}_a)$ using equation (12).
                Remove node *a* **if** D_a > average(D_b)
            end for
        end for
    end if
end for

Appendix 2 – Datasheets

SUN-2000GTIL2

G2 Sun Series Grid Tie Inverter Models

Sun series grid tie inverters include several models, refer to table 1.

Table 1, G2 Sun Series Grid Tie Inverter Models

Model	Rated Power (Max./Continuous)	DC Input Voltage	AC Output Voltage	AC Output Frequency	Max. Efficiency	Night Power Consumption
SUN-1000G2-M	1000W / 900W	22V~65V	95V~265V	45Hz~64Hz	90%	1.0W
SUN-1000G2-H	1000W / 900W	45V~90V	95V~265V	45Hz~64Hz	92%	1.5W
SUN-2000G2-H	2000W / 1800W	45V~90V	185V~265V	45Hz~64Hz	92%	1.5W

Model Name description:

SUN-XXX G2-X

- DC Input Voltage Range, "M" is 22V~65V, "H" is 45V~90V
- "G2" is the Abbreviation of Generation Two
- The Rated Output Power
- The Title Name of Sun Series Products

3

Technical Data of SUN Series Grid Tie Power Inverter

Table 3. Common Specifications for SUN Series Grid Tie Power Inverter

INPUT DATA (DC)	SUN-XXXG2-M	SUN-XXXG2-H
Maximum Input DC Voltage	65 V	90 V
MPPT Voltage Range	25V - 60 V	50 V - 90 V
Operating DC Voltage Range	22 V - 65 V	45 V - 90 V
Startup Voltage	25V	49V
OUTPUT DATA (AC)	**SUN-1000G2-X**	**SUN-2000G2-X**
Nominal Voltage (Range)	110V(95-140V)/230V(185-265V)	230V(185-265V)
Frequency (Range)	50Hz(45-55 Hz)/60Hz(56-64Hz)	50Hz(45-55 Hz)/60Hz(56-64Hz)
Power Factor	>0.95	>0.95
Output Waveform	Pure Sine Wave	Pure Sine Wave
CHARACTERISTIC DATA	**SUN-XXX-X-X**	
MPPT Efficiency	99%	
Over Current Protection Over	Yes	
Temperature Protection	Yes	
Reverse Polarity Protection	No	
Anit-Island Protection	Yes	
Stackable	Just for AC Output	
Operating Temperature Range	-20 C – 45 C	
Storage Temperature Range	-40 C – 65 C	

There are some common specifications of SUN Series Grid Tie Power Inverter shown in Table 3.
Other electrical specifications of every model are listed in Table 1.

Weight and Dimension of SUN series Grid Tie Power Inverter

Model	SUN-1000G-X-X	SUN-2000G-X-X
Net Weight	4.0Kg	5.8Kg
Gross Weight	5.0Kg	7.0Kg
Dimension(Package)	430mm×310mm×155mm	540mm×310mm×155mm

Outline Drawing of G2 SUN Series Grid Tie Power Inverter

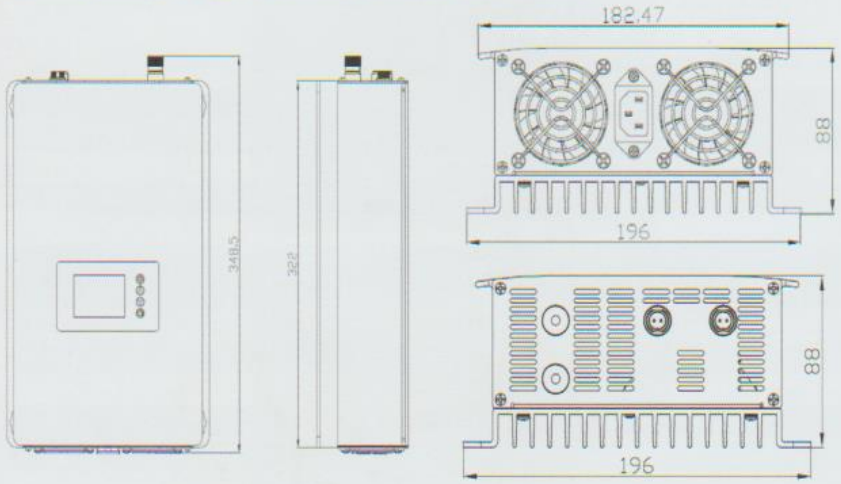


Fig. 11 SUN-1000G2-X Outline Drawing

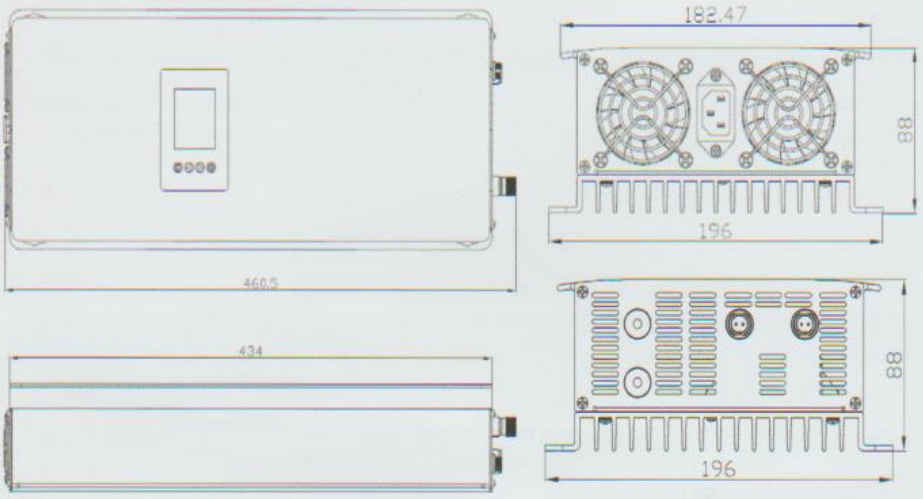


Fig. 12 SUN-2000G2-X Outline Drawing

POWERBATT 12V 250Ah Rechargeable VRLA Sealed Lead Acid AGM Battery

Battery Profiles	
Rated capacity	: 250Ah
Nominal voltage	: 12V
Cells per unit	: 6
Design life span	: 10 Years
Warranty	: 1 Year
Manufactured	: China
Brand	: POWERBATT
Series	: PK Series

Battery Outer Dimensions (mm) & Weight (kg)

Total height (with terminal) : 208 mm (approximately)
Height : 203 mm
Length : 520 mm
Width : 269 mm
Weight : 72.60 kg (approximately)

Battery Discharge Constant Current Characteristics Amperes at 25°C

F.V/TIME	5min	10min	15min	30min	60min	3h	5h	10h	20h
1.60V	775	525	415	265	163	63.9	45.7	25.4	13.4
1.70V	736	499	398	254	156	63.0	45.0	25.2	13.3
1.75V	692	469	378	242	149	61.7	44.1	25.0	13.2

Battery Discharge Constant Power Characteristics Watts at 25°C

F.V/TIME	5min	10min	15min	30min	60min	3h	5h	10h	20h
1.60V	8138	5670	4557	2973	1858	752	540	303	161
1.70V	7731	5387	4374	2854	1784	741	532	300	158
1.75V	7267	5063	4156	2712	1695	726	521	300	158

Battery Capacity at 25°C

20h rate : 264 Ah
10h rate : 250 Ah
5h rate : 220.5 Ah
1h rate : 149 Ah
15min rate : 94.5 Ah

Battery Internal Resistance Fully Charged at 25°C

Approximately 2.6mΩ

Battery Capacity Affected by Temperature 20h rate

40 °C : 102 %
25 °C : 100 %
0 °C : 85 %
-15 °C : 65 %

Battery Self-Discharge at 25°C

3 month : Remaining Capacity 91.5%
6 month : Remaining Capacity 82.5%
12 month : Remaining Capacity 65.4%
Environmental temperature affects the life span of battery. POWERBATT battery can be stored for more 6 months at 25°C.

Battery Nominal Operating Temperature

25°C ± 3°C

Battery Operating Temperature Range

Discharge : -15°C ~ 50°C
Charge : -15°C ~ 40°C
Storage : -15°C ~ 40°C

Battery Maximum Discharge Current at 25°C

1800A (5 Sec.)

Battery Maximum Charging Current at 25°C

75A

Battery Float Charging Voltage at 25°C

14.4 ~ 14.7 VDC/unit Average

Battery Cyclic Charging Voltage at 25°C

13.6 ~ 13.8 VDC/unit Average

Battery Technical Characteristics

CHART 1

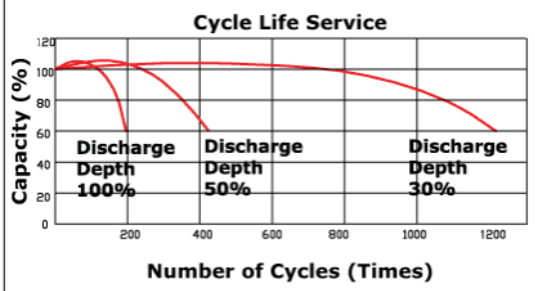


CHART 2

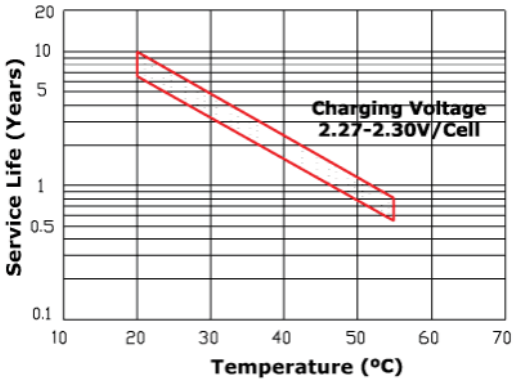


CHART 3

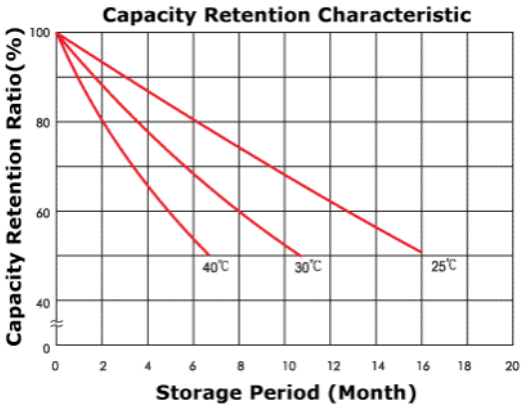


CHART 4

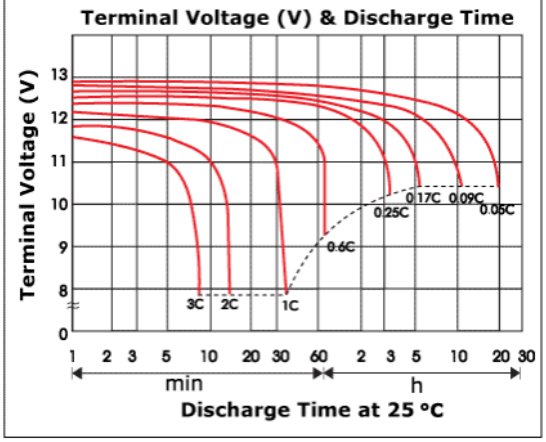


CHART 5

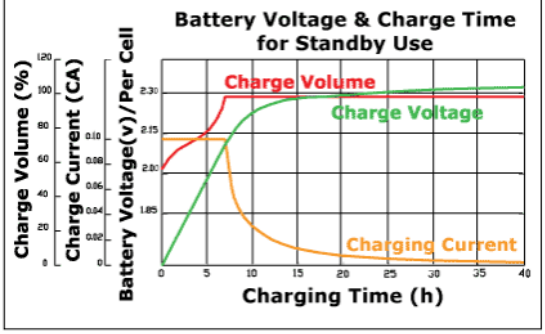
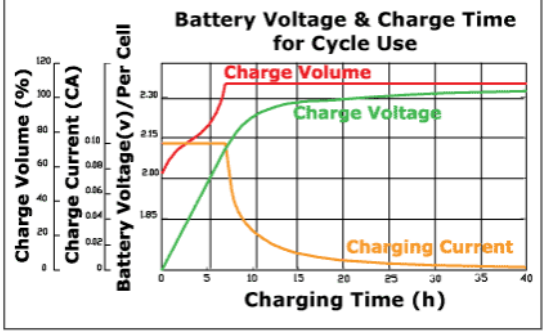


CHART 6



Battery Construction

Component	Raw Material
Cover	ABS
Container	ABS
Terminal	Copper
Safety Valve	Neoprene
Positive Plate	Lead Dioxide
Negative Plate	Lead
Separator	Fiberglass
Electrolyte	Sulphuric Acid

Chroma 63800 Programmable AC Load

SPECIFICATION

Model	63802	63803	63804
Power	1,800W	3,600W	4,500W
Current	0 ~ 18A_{rms} (54 A_{peak})	0 ~ 36A_{rms} (108 A_{peak})	0 ~ 45A_{rms} (135 A_{peak})
Voltage	50 ~ 350V_{rms} (500 V_{peak})	50 ~ 350V_{rms} (500 V_{peak})	50 ~ 350V_{rms} (500 V_{peak})
Frequency	45 ~ 440Hz, DC	45 ~ 440Hz, DC	45 ~ 440Hz, DC
AC Section			
Constant Current Mode			
Range	0 ~ 18A_{rms}, Programmable	0 ~ 36A_{rms}, Programmable	0 ~ 45A_{rms}, Programmable
Accuracy*1	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.
Resolution	2mA	5mA	5mA
Constant Resistance Mode			
Range	2.77Ω ~ 2.5kΩ, Programmable	1.39Ω ~ 2.5kΩ, Programmable	1.11Ω ~ 2.5kΩ, Programmable
Accuracy	0.5% + 0.5%F.S.	0.5% + 0.5%F.S.	0.5% + 0.5%F.S.
Resolution*4	20μS	50μS	50μS
Constant Power Mode			
Range	1,800W, Programmable	3,600W, Programmable	4,500W, Programmable
Accuracy	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.
Resolution	0.375W	1.125W	1.125W
Crest Factor (under CC, CP modes)			
Range*2	1.414 ~ 5.0, Programmable	1.414 ~ 5.0, Programmable	1.414 ~ 5.0, Programmable
Accuracy	(0.5% / I_{rms}) + 1% F.S.	(0.5% / I_{rms}) + 1% F.S.	(0.5% / I_{rms}) + 1% F.S.
Resolution	0.005	0.005	0.005
Power Factor			
Range	0 ~ 1 lead or lag, Programmable	0 ~ 1 lead or lag, Programmable	0 ~ 1 lead or lag, Programmable
Accuracy	1%F.S.	1%F.S.	1%F.S.
Resolution	0.001	0.001	0.001
Rectified Load Mode			
Operating Frequency	45Hz ~ 70Hz		
RLC mode	Parameter: I_p(max), R_s, L_s, C, R_L		
Constant Power Mode	Parameter: I_p(max), Power setting = 200W ~ 1,800W, PF = 0.4 ~ 0.75	Parameter: I_p(max), Power setting = 200W ~ 3,600W, PF = 0.4 ~ 0.75	Parameter: I_p(max), Power setting = 200W ~ 4, 500W, PF = 0.4 ~ 0.75
Inrush Current Mode	Parameter: I_p(max), R_s, L_s, C, R_L, Phase		
	80A (peak current)	160A (peak current)	200A (peak current)
R_s Range	0 ~ 9.999Ω	0 ~ 9.999Ω	0 ~ 9.999Ω
L_s Range	0 ~ 9999μH	0 ~ 9999μH	0 ~ 9999μH
C Range	100 ~ 9999μF	100 ~ 9999μF	100 ~ 9999μF
R_L Range	2.77 ~ 9999.99Ω	1.39 ~ 9999.99Ω	1.11 ~ 9999.99Ω
DC Section			
Voltage Range	7.5V ~ 500V	7.5V ~ 500V	7.5V ~ 500V
Current Range	0A ~ 18A	0A ~ 36A	0A ~ 45A

Slew Rate	4 ~ 240A/ms	4 ~ 600A/ms	4 ~ 600A/ms
Rise Time	75µs	75µs	75µs
Min. Operating Voltage	7.5V	7.5V	7.5V
Operating Mode	CC, CV, CR, CP, DC Rectified		
Short Circuit Simulation	Use the CR mode loading under max. power rating.		
Constant Current Mode			
Range	0 ~ 18A, Programmable	0 ~ 36A, Programmable	0 ~ 45A, Programmable
Accuracy*1	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.
Resolution	2mA	5mA	5mA
Constant Resistance Mode			
Range	2.22Ω ~ 1kΩ, Programmable	1.25Ω ~ 1kΩ, Programmable	1.00Ω ~ 1kΩ, Programmable
Accuracy	0.5% set + 0.5%F.S.	0.5% set + 0.5%F.S.	0.5% set + 0.5%F.S.
Resolution*4	20µS	50µS	50µS
Constant Power Mode			
Range	1,800W, Programmable	3,600W, Programmable	4,500W, Programmable
Accuracy	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.
Resolution	0.375W	1.125W	1.125W
Constant Voltage Mode			
Range	7.5V ~ 500V, Programmable	7.5V ~ 500V, Programmable	7.5V ~ 500V, Programmable
Accuracy	0.3% + 0.3%F.S.	0.3% + 0.3%F.S.	0.3% + 0.3%F.S.
Resolution	0.01V	0.01V	0.01V
Measurement Section			
DVM Range	$350V_{rms}$ ($500\ V_{peak}$)	$350V_{rms}$ ($500\ V_{peak}$)	$350V_{rms}$ ($500\ V_{peak}$)
DVM Accuracy	0.1% + 0.1%F.S.	0.1% + 0.1%F.S.	0.1% + 0.1%F.S.
DVM Resolution	10 mV	10 mV	10 mV
DAM Range	$18A_{rms}$ ($80\ A_{peak}$)	$36A_{rms}$ ($160.00\ A_{peak}$)	$36A_{rms}$ ($200.00\ A_{peak}$)
DAM Accuracy (<70Hz)	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.	0.1% + 0.2%F.S.
DAM Accuracy (>70Hz) *1	$0.1\%(1+CF^2\times kHz)$ + 0.2%F.S.	$0.1\%(1+CF^2\times kHz)$ + 0.2%F.S.	$0.1\%(1+CF^2\times kHz)$ + 0.2%F.S.
DAM Resolution	1.0 mA	2.5 mA	2.5 mA
True Power			
Range	0 ~ 1,800W	0 ~ 3,600W	0 ~ 4,500W
Accuracy	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.
Apparent Power			
Range	0 ~ 1,800VA	0 ~ 3,600VA	0 ~ 4,500VA
Accuracy	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.	0.2% + 0.3%F.S.
Others parameters*3	Q(VAR), CF, PF, Freq, R, I_p-, I_p+, THDv		
Others			
Vmonitor	±500V/±10V (Isolated)	±500V/±10V (Isolated)	±500V/±10V (Isolated)
Imonitor	±80A/±10V (Isolated)	±200A/±10V (Isolated)	±200A/±10V (Isolated)
Protection	OCP: $19.2A_{rms}$; OVP: $360V_{rms}$ (DC: $510V_{DC}$);	OCP: $38.4A_{rms}$; OVP: $360V_{rms}$ (DC: $510V_{DC}$);	OCP: $48A_{rms}$; OVP: $360V_{rms}$ (DC: $510V_{DC}$);

1-4

	OPP: 1,920W; OTP	OPP: 3,840W; OTP	OPP: 4,800W; OTP
Parallel Ability	Yes	Yes	Yes
Remote Interface	GPIB, RS-232C		
AC Input			
Operable Voltage	90-132/180-264V_{rms}	90-132/180-264V_{rms}	90-132/180-264V_{rms}
Fuse	2A, 250V	3A, 250V	3A, 250V
Operable Frequency	47 ~ 63Hz	47 ~ 63Hz	47 ~ 63Hz
Maximum VA	150VA	250VA	250VA
General			
Dimension (H×W×D)	177×440×595 mm 7.0×17.3×23.4 inch	310×440×595 mm 12.2×17.3×23.4 inch	310×440×595 mm 12.2×17.3×23.4 inch
Weight	37kg / 81.57lbs	66kg / 145.5lbs	66kg / 145.5lbs

All specifications are subject to change without notice.

Note:

*1: Additional Spec: 0.1%×CF^2×kHz

*2: The load value of CF is related to characteristic, frequency and output impedance of UUT. When frequency is 440Hz, the load value of CF and the accuracy of the measurement value above 4.3 will over the specification.

*3: CF = 0.5% / I_{rms} + 1%F.S.

*4: S(Siemens) indicates conductance, that is the inverse of resistance.

Chroma 62024P-80-60 Programmable DC Power Supply

Model	62006P-100-25	62006P-300-8	62012P-80-60	62012P-100-50	62012P-600-8	62024P-80-60	62024P-100-50	62050P-100-100
Output Ratings								
Output Voltage	0~100V	0~300V	0~80V	0~100V	0~600V	0~80V	0~100V	0~100V
Output Current	0~25A	0~8A	0~60A	0~50A	0~8A	0~60A	0~50A	0~100A
Output Power	600W	600W	1200W	1200W	1200W	2400W	2400W	5000W
Line Regulation								
Voltage	0.01%+6mV	0.01%+18mV	0.01%+8mV	0.01%+10mV	0.01%+18mV	0.01%+8mV	0.01%+10mV	0.01%+10mV
Current	0.01%+5mA	0.03%+20mA	0.01%+10mA	0.01%+12mA	0.03%+20mA	0.01%+10mA	0.01%+12mA	0.01%+12mA
Load Regulation								
Voltage	0.01%+10mV	0.01%+50mV	0.01%+12mV	0.01%+18mV	0.01%+50mV	0.01%+12mV	0.01%+18mV	0.01%+18mV
Current	0.01%+5mA	0.03%+40mA	0.01%+20mA	0.01%+28mA	0.03%+40mA	0.01%+20mA	0.01%+28mA	0.01%+28mA
Voltage Measurement								
Range	20V/100V	60V/300V	16V/80V	20V/100V	120V/600V	16V/80V	20V/100V	20V/100V
Accuracy	0.05% + 0.05%F.S.							
Current Measurement								
Range	5A/25A	1.6A/8A	12A/60A	10A/50A	1.6A/8A	12A/60A	10A/50A	20A/100A
Accuracy	0.1% + 0.2%F.S.	0.1% + 0.1%F.S.						
Output Noise (0 ~ 20MHz)								
Voltage Ripple (P-P)	85 mV	180 mV	100 mV	100 mV	180 mV	100 mV	100 mV	125 mV
Voltage Ripple (rms)	10 mV	90 mV	10 mV	15 mV	90 mV	10 mV	15 mV	20 mV
Current Ripple (rms)	10 mA	60 mA	30 mA	20 mA	60 mA	30 mA	20 mA	30 mA
OVP Adjustment Range	110% of Vset to 110% of Vmax							
Efficiency	0.75	0.75	0.8	0.8	0.8	0.85	0.85	0.85
Drift (8 hours)								
Voltage	0.02% of Vmax							
Current	0.04% of Imax							
Temperature Coefficient								
Voltage	0.02% of Vmax/ °C							
Current	0.04% of Imax/ °C							
Transient Response Time	3 mS	3mS	3 mS	3 mS	3mS	3mS	3mS	3mS
10 % step change	180 mV	600 mV	250 mV	250 mV	600 mV	250 mV	250 mV	250 mV
AC Input Voltage	95 to 250Vac					190 to 250Vac (Single phase)	190 to 250Vac (Single phase)	190 to 250Vac (3phase 4 wire, Delta connection) or 342 to 440Vac (3phase 5 wire, Y connection)
Weight	13kg	13kg	13kg	13kg	13kg	13kg	13kg	25kg
Operating Temperature	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C	0~40°C
Dimensions (HxWxD) mm	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	88 x 428 x 425	177 x 428 x 425

All specifications are subject to change without notice.

ORDERING INFORMATION

62006P-100-25 : Programmable DC Power Supply, 100V / 25A / 600W
62006P-300-8 : Programmable DC Power Supply, 300V / 8A / 600W
62012P-80-60 : Programmable DC Power Supply, 80V / 60A / 1200W
62012P-100-50 : Programmable DC Power Supply, 100V / 50A / 1200W
62012P-600-8 : Programmable DC Power Supply, 600V / 8A / 1200W
62024P-80-60 : Programmable DC Power Supply, 80V / 60A / 2400W
62024P-100-50 : Programmable DC Power Supply, 100V / 50A / 2400W
62050P-100-100 : Programmable DC Power Supply, 100V / 100A / 5000W
A620004 : GPIB Interface for Model 62000P Series
A620006 : Rack Mounting Kit for Model 62000P Series
A620009 : Softpanel for 62000P Series

OTHER SPECIFICATIONS

Programming &Measurement Resolution	
Voltage (Front Panel)	10 mV
Current (Front Panel)	10 mA
Voltage (Remote Interface)	0.003% of Vmax
Current (Remote Interface)	0.002% of Imax
Voltage (Analog Programming Interface)	0.04% of Vmax
Current (Analog Programming Interface)	0.04% of Imax
Programming Accuracy	
Voltage Programming (Front Panel and Remote Interface)	0.1% of Vmax
Voltage Programming (Analog Programming Interface)	0.2% of Vmax
Current Programming (Front Panel and Remote Interface)	0.3% of Imax
Current Programming (Analog Programming Interface)	0.3% of Imax
Programming Response Time	
Rise Time: For a programmed 5% to 95% step of rated voltage. (Full Load)	10 ms
Rise Time: For a programmed 5% to 95% step of rated voltage. (No Load)	10 ms
Fall Time: For a programmed 95% to 5% step of rated voltage. (Full Load)	60 ms
Fall Time: For a programmed 95% to 5% step of rated voltage. (No Load)	840 ms (max.) / 4S for 600V models
Vout setting (GPIB send command to DC Power Supply receiver)	20 ms
?Volt, ? Current (under GPIB command using Fetch)	25 ms
?Volt, ? Current (under GPIB command using Measure)	70 ms
Analog Programming Interface	
Voltage and Current Programming inputs	0~10Vdc or 0~5Vdc of F.S.
Voltage and Current monitor	0~10Vdc or 0~5Vdc of F.S.
Isolation: Maximum working voltage of any analog programming signal with respect to chassis potential	70 Vdc
Auxiliary Power Supply	
Output Voltage	12 Vdc
Maximum current source capability	10 mA
Remote Inhibit Function	
Use to disable the output of DC Power Supply; Active Low	TTL
DC-ON Output Signal	
Indicate the output status, Active High	TTL
Fault Output Signal	
Indicate if there is a fault/protection occurred, Active Low	TTL
Series & Parallel operation function with Master / Slave control	
Voltage limit @ Series Mode. (Model 62012P-600-8)	800 Volt
Voltage limit @ Series Mode (Refer to Ground)	240 Volt
Number of DC Power Supplies allowed @ master / slave control mode	5
Auto Sequencing Programmable Function	
Number of program	10
Number of sequence	100
Time Range	5 ms ~ 15000 S
TTL signal out	8 bits
TTL source capability	7 mA
Slew Rate Control Function	
Voltage slew rate range (The fall rate will be affected by the discharge rate of the output capacitors especially under no load condition.)	0.01V ~ 10V/ms
Current slew rate range of current	0.01A ~ 1A/ms
Minimum transition time	0.5 ms
Remote Sense	
Line loss compensation	5V

All specifications are subject to change without notice.

Developed and Manufactured by :
CHROMA ATE INC.
致茂電子股份有限公司

HEADQUARTERS
66, Hwa-Ya 1st Rd., Hwa-Ya Technology Park, Kuei-Shan Hsiang, Taoyuan Hsien 33383, Taiwan
Tel: +886-3-327-9999
Fax: +886-3-327-8898
http://www.chromaate.com
E-mail: chroma@chroma.com.tw

U.S.A.
CHROMA ATE INC. (U.S.A.)
7 Chrysler Irvine, CA 92618
Tel: +1-949-421-0355
Fax: +1-949-421-0353
Toll Free: +1-800-478-2026

EUROPE
CHROMA ATE EUROPE B.V.
Morseestraat 32, 6716 AH Ede, The Netherlands
Tel: +31-318-648282
Fax: +31-318-648288

CHINA
CHROMA ELECTRONICS (SHENZHEN) CO., LTD.
8F, No.4, Nanyou Tian An Industrial Estate, Shenzhen, China PC: 518054
Tel: +86-755-2664-4598
Fax: +86-755-2641-9620

Distributed by:

Worldwide Distribution and Service Network

62000P-200711-PDF