Title

Autonomous and Coordinated Energy Management Systems: Implementing On-Line Prediction, Electric Vehicle Time Series Analysis to Microgrid Control

Author

Xue, Yun

Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Autonomous and Coordinated Energy Management Systems: Implementing On-Line Prediction, Electric Vehicle Time Series Analysis to Microgrid Control

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Yun Xue

December 2021

Dissertation Committee:
Dr. Alfredo A. Martinez-Morales, Co-Chairperson
Dr. Matthew J. Barth, Co-Chairperson
Dr. Hamed Mohsenian-Rad
Dr. Hamidreza Nazaripouya

The Dissertation of Yun Xue is approved:

Committee Co-Chairperson

Committee Co-Chairperson

University of California, Riverside

ACKNOWLEDGMENT

First and all, I would like to sincerely thank my Ph.D. advisor, Dr. Alfredo. A. Martinez-Morales. He has shown the greatest patience and support for my Ph.D. research and life. He has always been actively discussing researches and finding potential collaborators for me. Without his guidance, my Ph.D. journey would have had more unforeseen difficulties. He has offered me such a wonderful experience for my Ph.D. life.

I would also like to express my gratitude to Dr. Matthew J. Barth for serving as the co-chairperson of my committee, Dr. Hamed Mohsenian-Rad and Dr. Hamidreza Nazaripouya for serving as my committee members, for their valuable discussions and suggestions.

In addition, I would like to acknowledge the Department of Electrical and Computer Engineering at UCR, Center for Environmental Research and Technology (CE-CERT) at UCR, National Center for Sustainable Transportation (NCST), California Energy Commission (CEC) for fully or partially supporting my study and research in this dissertation.

Finally and foremost, I would like to express my deepest gratitude to my family: my parents, grandparents, parents-in-law, and husband. Without their support, I would not be able to come so far and end up with a Ph.D. degree. All of my success belongs to my beloved family. Special thanks to my dear husband. I could not imagine how life had become without his company during the quarantine time. He is the one who is always bringing me joy and love.

The main contents in Chapter 4 of this dissertation are a reprint or rewritten of the materials that appeared in the following publication.

[1] Y. Xue, M. Penchev, H. Xin, and A. A. Martinez-Morales, "Flow Battery Control Strategy Implementation and Cost Benefit Analysis for Microgrid under Updated Time-of-Use Periods", 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2020, pp. 1-5, doi: 10.1109/ISGT45199.2020.9087673.

[2] Y. Xue, M. Todd, S. Ula, M. J. Barth, and A. A. Martinez-Morales, "Field Implementation of a Real-time Battery Control Scheme for a Microgrid at the University of California, Riverside", 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), Chengdu, China, 2019, pp. 2859-2864. doi: 10.1109/ISGT-Asia.2019.8881076

[3] 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", 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC), Portland, OR, 2016, pp. 1875-1880. doi: 10.1109/PVSC.2016.7749947

To my father Jinhe Xue, mother Wenying Liu, husband Kaifan Xu

for all their unconditional love and support.

To my grandpa, in loving memory.

ABSTRACT OF THE DISSERTATION

Autonomous and Coordinated Energy Management System: Implementing On-Line Prediction, Electric Vehicle Time Series Analysis to Cohesive Microgrid Control

by

Yun Xue

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2021
Dr. Alfredo. A. Martinez-Morales, Chairperson

Due to the finite nature and detrimental environmental impacts of conventional fossil-fuel energy resources, renewable energy resources are promising clean and sustainable forms of energy for meeting society's future energy needs. As renewable energy continues to gain higher penetration in the electric power grid through distributed generation, microgrids represent an innovative approach towards achieving a distributed grid architecture, playing a critical role in implementing future smart grid systems.

This dissertation first introduces a novel online prediction algorithm with the forgetting factor: Affine Extreme Learning Machine with Forgetting Learning (FL-AELM). The proposed FL-AELM is applied to forecast solar generation and EV charging power. Due to

training data's non-availability and seasonality features, the FL-AELM is actively trained by the newly arrived data. Compared to other state-of-the-art prediction models, the proposed FL-AELM improves the prediction accuracy by over 8% and 13% for solar generation and EV charging power, respectively.

Additionally, this dissertation introduces the unique EV characteristic from the charging current time series: tail feature. Tail is a subsequence of the charging time series consisting of nonincreasing current in the constant voltage stage. A novel shape-wise distance – refine matrix profile similarity (RMPS) is proposed to measure the global similarity of two tails by comparing each subsequence of the two time series. By tunning the subsequence length and cluster numbers, diverse charging profiles can be classified into seven groups with unique tail templates. Combining the learned tail templates into the centralized and distributed EV smart charging, an average of 4% - 5% flexible rate or 0.7% satisfactory rate improvement can be achieved.

This dissertation further introduces two real-world microgrid applications equipped with solar generation and battery energy storage systems (BESS). Different predictive model-based energy management systems are developed and implemented to handle demand response, load shifting, and net load reduction (i.e. peak shaving). With the utilization of BESS, the microgrid systems can reduce the electricity cost and peak demand. If a system transitions to the updated time-of-use periods, the BESS can significantly contribute to more cost savings and take a more critical role in reducing the total electric bill than the legacy time-of-use periods.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Motivations

From 1882, when the first electrical power station (Pearl Street Station) was built with the capability to supply 85 customers and provide electricity to 400 lamps, to the changes resultant from Edison's direct current (DC) to Tesla's alternating current (AC) in the 19th century, the idea of a large centralized power plant sprang out of greater efficiency and cost-effectiveness [1]. Under a centralized generation paradigm, large-scale power plants are built away from end-users, whether powered by fossil fuels, nuclear or hydroelectric dams. Utilities can purchase power from centralized power plants located in other regions through connected transmission lines. Centralized electricity generation capacity has grown over time, and currently, in the United States, more than 1,100 gigawatts of electric power can be generated through large-scale centralized facilities [2].

Over the last century, centralized electricity was dispatched to most places all over the world. While in some rural areas, where transmission and distribution lines cannot reach, there is an urgent need to find onsite alternative resources, which can satisfy local electricity needs. In 1978, the world's first solar power village was operated in Schuchuli, Arizona, when no electricity could be delivered to this village [3]. According to "Renewables 2017 Global Status Report", distributed renewable energy has supplied the needs of roughly 10% of the 600 million people living off-grid on the African continent [4].

It has been documented that centralized power plants can negatively affect the environment in many ways, such as air pollutant emissions, land use, and energy loss during transmission [2]. On the other hand, onsite, distributed renewable energy generation provides a promising future. A better way to coordinate distributed energy resources (DERs) is to form a microgrid system, including the distributed generation and associated electrical loads within a localized power grid. With local energy management system (EMS) controllers for the DERs in the microgrid, the need for central dispatch is reduced or eliminated [5]. Microgrid projects are expanding rapidly worldwide. California's renewables portfolio standard provides a significant impetus for microgrid development, which has a target of obtaining 50% of the state's electricity from eligible renewable resources by 2030, 100% zero-energy buildings by 2020 for all new residential buildings, and 2030 for new commercial buildings [6].

1.2 Problem Statement and Contribution of the Dissertation Research

It is well recognized that microgrids, as the future distributed power system, have the great potential to improve grid sustainability. The real-time microgrid operation, including the online predicting of the uncertainty within the microgrids, the constraints for different battery energy storage systems (BESS), and the dynamics of the EV charging profiles, are not well discussed. This dissertation discusses the above constraints and provides solutions to those constraints, the contributions of this dissertation can be summarized as:

- An online prediction model with a forgetting learning mechanism is developed for solar generation and EV charging power prediction.
- Classification of the diverse EVs' charging profiles through the "tail" time series using the novel distance measure - Refine Matrix Profile Similarity is proposed. The learned "tail" features can be inherited into the smart charging algorithm to further increase energy scheduling efficiency.
- The distributed EV charging scheduling game is established to manage the optimal charging energy from the individual EVSE side, considering the uncertainty of EV charging power after optimal scheduling.
- Real-time battery control algorithm is proposed for real-world solar + battery microgrid arrangements to perform different microgrids' functionalities by emphasizing different systems' objectives and constraints.

1.3 Organization of the Dissertation Research

The dissertation is organized as follows: Chapter 2 describes the prediction model – Forgetting Learning – Affine Extreme Learning (FL-AELM) for the uncertainties in a microgrids system such as solar generation and electric vehicle charging loads. The FL-AELM is online trained, and any outdated data is given less importance, which shows the computational efficiency and accuracy. Chapter 3 further discusses the unique EV charging time series characteristic: tail feature. This chapter presents the novel distance measure - Refine Matrix Profile Similarity (RMPS). With the RMPS, the k-medoids can classify diverse charging profiles into only seven groups. The learned tail templates are utilized as the constraints in the EV energy scheduling to enhance the flexibility and satisfactory rates of the smart charging algorithms in both centralized and distributed control. The scheduling game for distributed EV scheduling is also introduced in this chapter. The adjusted average scheduled power by centralized control is utilized as the prediction of charging power. The unique Nash equilibrium is found when the optimal distributed schedule is achieved. Chapter 4 introduces the real-world microgrid application with the solar and BESS architecture. The battery control algorithms based on the model predictive control (MPC) schematic are proposed for microgrid energy management to handle demand response, load shifting, and peak shaving. The proposed battery control strategies are implemented for both lithium and flow BESSs, addressing the related battery constraints and system limitations.

Chapter 2 Power Prediction Model for Solar Generation and Electrical Vehicles

The intermittence of renewable energy, such as solar generation, and randomness of electric charging power pose many challenges to microgrid energy management [7]. The effectiveness of energy management, such as load dispatch, demand response, and power quality [8] in a microgrid system, is highly dependent on the accuracy of the prediction for renewable generation and electric power load.

This chapter introduces a novel online prediction model: affine extreme machine learning with a forgetting mechanism. The model is applied to solar generation and EV charging power prediction. Comprehensive prediction performance is carried out by comparison between different state-of-art prediction algorithms.

2.1 Background for Prediction Algorithms in the Microgrid System

2.1.1 Background of Solar Generation Prediction

Solar generation prediction algorithms can be classified into three categories: time series statistical, a physical-mathematical formulation including the numerical weather prediction (NWP), and the combination of both [9]. In time series statistical, machine learning algorithms, including regressions and artificial neural networks, are mainly discussed in the literature. In [10], an ensemble of M ANNs is combined to predict the 24-hour-ahead solar power and quantify the associated uncertainty at each time interval. By generating appropriate base ANN models and aggregating the results of the predictions from different ANNs trained in different weather conditions, better prediction accuracy can be achieved. However, the prediction accuracy relies on the number of base ANNs – M, and the computational burden increases while M increases. Besides, the base ANNs need comprehensive data available during the training stage to formulate different networks for weather conditions, which is not feasible for systems without well-documented data. Authors in [11] propose a swarm optimized Radial basis function (RBF) neural network to predict 15-minute ahead solar power. Considering the actual data availability for community households, only historical recursive solar power was utilized as input features. However, the solar data may become outdated, thus affecting the accuracy of the learned prediction model.

Authors in [12], [13] utilize different regression models to perform solar power prediction. In [12], the prediction algorithm, by combining the long short-term memory (LSTM) and Gaussian process regression (GPR), provides not only point prediction but also a reliable interval estimation for one-step hourly prediction. Numerical weather information and time index are chosen as the input features. Similar to [12], authors in [13] propose the grouped GPRs for the 15-minute ahead prediction. The grouped GPRs incorporate spatial dependencies between latent functions to perform prediction for multi-sites in a region. However, neither algorithms consider the solar data's timeliness. The two algorithms are computational complexity and required to be trained offline.

The physical formulation of the solar generation prediction is usually developed for solar energy resource planning purposes [14]–[16]. In [14], HELIOSAT is introduced that utilizes satellite images to calculate solar irradiance. Authors in [15] combine the NWP and satellites-based data and forecasts such as clear sky estimates to predict several hours-ahead solar generation in a region. Authors in [16] propose the physical prediction model for photovoltaic (PV) panels based on the solar irradiance on inclined surfaces and panel parameters. The physical prediction model requires comprehensive NWP information and satellite data, which are unavailable for real-time retrieval.

From the above discussion, although the solar prediction has been intensively discussed in the literature, the training solar data's timeliness, availability at training are not well addressed. The proposed prediction model in this chapter aims at addressing the two challenges.

2.1.2 Background of Electric Vehicle Charging Power Prediction

Based on different input features, the EV charging power prediction can be classified into two aspects: time series prediction and the probability distribution of charging behavior. In [17], temporal convolutional networks (TCN) are proposed for multi-steps ahead of EV load prediction. The input features are chosen as the past charging load time series. When the number of input features (past charging load time series) increases, the TCN prediction performance can be improved with more costly training computation. Authors in [18] apply the support vector machine (SVM) to prediction half-hourly and daily ahead data. By choosing the penalized kernel in SVM, the most relevant input features (past load time series) are automatically selected during the training process. Hence the minimal input features can be selected to increase the model learning speed and prediction performance. Different deep learning models for load prediction are compared in [19], with the input features selected from the past time-series data, electricity price, and day types (weekdays, weekends, and holidays).

[20]–[22] propose the EV charging load distribution statistical models for a public place. In [20], the day-ahead charging loads distribution is determined by the state of charge (SOC) distribution for a large number of EVs in a business premise. In [21], the charging load model is trained by the Gaussian mixture models (GMM) with the features of charging events, including arrival time, departure time, and charging energy. In [22], EVs' daily trip

geographical and temporal transition probabilities are considered to model charging energy consumption. The statistical model requires sufficient historical data in the modeling/ training process. Once it is established, the prediction results can be considered as deterministic values. It lacks the mobility of online adjusting with the new system data; thus, the prediction performance may degrade once the model's behavior changes, such as data's seasonality.

In this chapter, the EV charging power prediction through time series is conducted. Similar to the solar prediction research, the prediction models for the EV charging load in the literature ignore the data's timeliness and availability during the training process. The proposed prediction model in this chapter aims at addressing the two challenges.

2.2 Extreme Learning Machine

The classical Extreme Learning Machine (ELM) is illustrated in Figure 2.1. The network consists of a single hidden layer feedforward neural network (SLFN) with randomized hidden neurons. Huang el. mentioned that "ELM tends to provide the best generalization at extremely fast learning speed" [23]. Due to its computational efficiency and good generalization, it is more applicable for real-time microgrids operation.

2.2.1 Notations and Definitions

As shown in Figure 2.1, the ELM networks consist of n inputs nodes, m randomly selected hidden neurons, and n' output nodes. For any input-output set $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, the feed-forward ELM computation is illustrated in (2.1):

$$\boldsymbol{y}_i = \sum_{j=1}^{m} \beta_j \cdot g(\boldsymbol{\omega}_j \cdot \boldsymbol{x}_i + b_j) \tag{2.1}$$

where $\boldsymbol{x}_i = [x_{i1}, \dots, x_{in}] \in \mathbb{R}^n$, and $\boldsymbol{y}_i = [y_{i1}, \dots, y_{in'}]^T \in \mathbb{R}^{n'}$.

$g(\cdot)$: the activation function for the hidden layers. The typical activation functions utilized in this chapter are sine, tanh, and sigmoid functions in (2.2) – (2.4):

$$g(v) = \sin(v) \ \text{(sine function)} \tag{2.2}$$

$$g(v) = \frac{e^{2v}-1}{e^{2v}+1} \ \text{(tanh function)} \tag{2.3}$$

$$g(v) = \frac{1}{e^{-v}+1} \ \text{(sigmoid function)} \tag{2.4}$$

$\boldsymbol{\omega}_j, \boldsymbol{b_j}$: the weights and bias of the hidden neurons connecting the j^{th} hidden neuron and the input nodes, where $\boldsymbol{\omega}_j = [\boldsymbol{\omega}_{j1}, \dots, \boldsymbol{\omega}_{jn}]^T$.

m: the number of hidden neurons.

$\boldsymbol{\beta}_j$ is the output weight vector connecting the j^{th} hidden neuron and the outputs, where $\boldsymbol{\beta}_j = [\boldsymbol{\beta}_{j1}, \dots, \boldsymbol{\beta}_{jn'}]^T$.

For N arbitrary distinct inputs $\mathbf{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}^T \in \mathbb{R}^{n \times N}$ and outputs $\mathbf{Y} = \{y_1, \dots, y_N\}^T \in \mathbb{R}^{N \times n'}$, N equations can be written in (2.1) in the form of matrices in (2.5):

$$\mathbf{Y} = \mathbf{H} \cdot \boldsymbol{\beta} \tag{2.5}$$

where $\mathbf{H}$ is the hidden layer output matrix:

$$\mathbf{H} = \begin{bmatrix} g(\boldsymbol{\omega}_{11} \cdot \boldsymbol{x}_{11} + b_1) & \dots & g(\boldsymbol{\omega}_{n1} \cdot \boldsymbol{x}_{1n} + b_m) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_{1m} \cdot \boldsymbol{x}_{N1} + b_1) & \dots & g(\boldsymbol{\omega}_{nm} \cdot \boldsymbol{x}_{Nn} + b_m) \end{bmatrix}_{N \times m} = g(\mathbf{X} \cdot \mathbf{W} + \mathbf{b}),$$

$$\mathbf{W} = [\boldsymbol{\omega}_1 \dots \boldsymbol{\omega}_m]_{n \times m}, \boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\beta}_1^T \\ \vdots \\ \boldsymbol{\beta}_m^T \end{bmatrix}_{m \times n'}, \mathbf{Y} = \begin{bmatrix} \boldsymbol{y}_1^T \\ \vdots \\ \boldsymbol{y}_m^T \end{bmatrix}_{N \times n'}$$

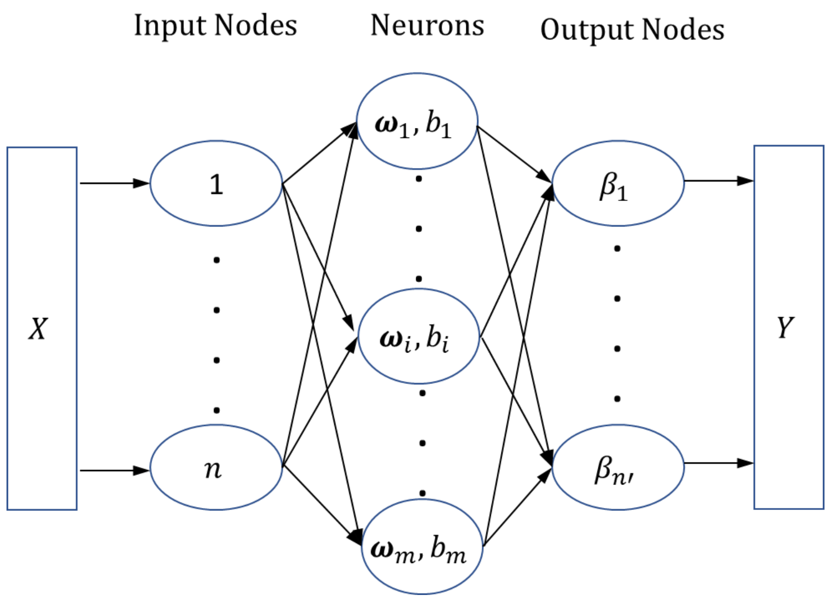


Figure 2.1: Extreme Learning Machine (ELM) networks.

2.2.2 Extreme Learning Machine Approximation

The objective of the ELM is to minimize the loss between the approximation from (2.5) and observations as explicated in:

$$\ell = \min_{\boldsymbol{\omega}, \boldsymbol{b}, \boldsymbol{\beta}} \|\mathbf{Y} - \mathbf{O}\| \tag{2.6}$$

where ℓ is the loss function and $\mathbf{O} \in \mathbb{R}^{N \times n'}$ is the observation matrix. Due to the arbitrary choice of $\boldsymbol{\omega}, \boldsymbol{b}$ in the ELM architecture, to minimize the error in (2.6), only output weight $\boldsymbol{\beta}$ needs to be approximated. The equivalent of minimization (2.6) is the minimization of $\sum_{i=1}^{N}(\mathbf{Y}_i - \mathbf{O}_i)^2$, which can be written in (2.7):

$$\ell = \min_{\boldsymbol{\beta}} \|\mathbf{H}\boldsymbol{\beta} - \mathbf{O}\|_2^2 \tag{2.7}$$

can be solved by minimum norm least-squares [23] as shown in (2.8)

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{+}\mathbf{O} \tag{2.8}$$

where $\mathbf{H}^{+}$ is the Moore-Penrose generalized inverse. The approximation of $\widehat{\boldsymbol{\beta}}$ follows two principles [24], [25]:

- When the number of training samples is greater than the number of hidden nodes: $N \geq m$, (2.8) can be written as (2.9);

$$\widehat{\boldsymbol{\beta}} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}\mathbf{O} \tag{2.9}$$

- When the number of training samples is less than the number of hidden nodes: $N < m$, (2.8) can be written as (2.10).

$$\widehat{\boldsymbol{\beta}} = \mathbf{H}^{T}(\mathbf{H}\mathbf{H}^{T})^{-1}\mathbf{O} \tag{2.10}$$

2.2.3 Improvements for Extreme Learning Machine Algorithm

The approximation of $\boldsymbol{\beta}$ only considers the Empirical Risk Minimization (ERM), and it tends to overfit the data and is sensitive to the outliers in the input data samples [26].

Besides, due to the randomized choice of hidden neurons, the stability, generalization, and prediction accuracy of the ELM heavily rely on the selected weights of hidden neurons [27], the number of neurons set [28], and the input data samples scaling [29]. In this section, different ELM algorithms are presented to improve the above challenges.

(I) Regularized Factors

To solve the limitation of only considering the ERM, structural risk minimization (SRM) can be added to (2.8). Hence, ℓ can be written as (2.11):

$$\ell = \min_{\boldsymbol{\beta}}(||\mathbf{H}\boldsymbol{\beta} - \mathbf{O}||_2^2 + \gamma||\boldsymbol{\beta}||_k^2) \tag{2.11}$$

where $||\cdot||_k$ represents the k-norm.

- By applying the ℓ_1 penalty (lasso), (2.11) becomes the regularized ELM (RELM), known as ℓ_1-RELM, as shown in (2.12) with $k = 1$.

$$\ell = \min_{\boldsymbol{\beta}}(||\mathbf{H}\boldsymbol{\beta} - \mathbf{O}||_2^2 + \gamma||\boldsymbol{\beta}||_1^2) \tag{2.12}$$

  Minimizing (2.12) can reduce the computational cost by sparse weights [30]. However, (2.12) becomes non-differential at zero points. Authors in [30] combine alternative direction methods of multipliers and optimize the dual decomposition of $||\mathbf{H}\boldsymbol{\beta} - \mathbf{O}||_2^2$ and $\gamma||\boldsymbol{\beta}||_1$. $\mathbf{H}$ and $\boldsymbol{\beta}$ are recursively updated in an online fashion.

- By applying the ℓ_2 penalty (ridge regression), (2.11) becomes the ℓ_2-RELM as shown in (2.13) with $k = 2$.

$$\ell = \min_{\boldsymbol{\beta}}(\|\mathbf{H}\boldsymbol{\beta} - \mathbf{O}\|_2^2 + \gamma\|\boldsymbol{\beta}\|_2^2) \tag{2.13}$$

(2.13) becomes a convex problem. In [26], Deng et al. introduced weighting factors v to alleviate the outliers inference, where $\ell = \min_{\boldsymbol{\beta}}(\|\mathbf{D}(\mathbf{H}\boldsymbol{\beta} - \mathbf{O})\|_2^2 + \gamma\|\boldsymbol{\beta}\|_2^2)$, $\mathbf{D} = \text{dialog}(v_1^2, \dots, v_N^2)$, $\boldsymbol{\beta}$ then can be solved by quadratic optimization in (2.14). When $\mathbf{D}$ is a unit matrix $\mathbf{I}$, (2.15) represents a solution for unweighted RELM.

$$\boldsymbol{\beta} = \boldsymbol{f}_{\boldsymbol{\beta}}(\mathbf{H}, \mathbf{D}, \gamma) \tag{2.14}$$

$$\boldsymbol{\beta} = \begin{cases} (\gamma\mathbf{I} + \mathbf{H}^{\boldsymbol{T}}\mathbf{D}\mathbf{H})^{-1}\mathbf{H}^{\boldsymbol{T}}\mathbf{D}\mathbf{O}, N \geq m \\ \mathbf{H}^{\boldsymbol{T}}(\gamma\mathbf{I} + \mathbf{D}\mathbf{H}\mathbf{H}^{\boldsymbol{T}})^{-1}\mathbf{D}\mathbf{O}, N < m \end{cases} \tag{2.15}$$

- By combining ℓ_1 and ℓ_2 penalties, (2.11) becomes elastic-net RELM as shown in (2.16).

$$\ell = \min_{\boldsymbol{\beta}}(\|\mathbf{H}\boldsymbol{\beta} - \mathbf{O}\|_2^2 + \gamma_1\|\boldsymbol{\beta}\|_1^2 + \gamma_1\|\boldsymbol{\beta}\|_2^2) \tag{2.16}$$

The elastic-net RELM can overcome the stability issue caused by ℓ_1 penalty and solve (2.16) more efficiently with sparse solutions [31], [32].

Besides adding different penalties, the selection of the regularized factor is crucial for optimization accuracy. Authors in [33] map a function between the output weights and γ, where $\gamma = f(\boldsymbol{\beta})$ to adaptively adjust γ for different tasks/datasets to improve the generalization. Authors in [34], [35] apply the leave-one-out cross validation (LOO) to obtain the optimal γ.

(II) Different Random Neurons Selection

As discussed in [36], ELM tends to have more neurons to achieve accuracy. However, ELM may have redundant hidden neurons than tunning-based learning algorithms due to the random assignment of $\boldsymbol{\omega}$ and $\boldsymbol{b}$. Besides, more neurons may result in an ill-conditioned output matrix $\mathbf{H}$, which reduces the generalization of ELM. Lastly, the test prediction generality of ELM is dependent on $\boldsymbol{\omega}$ and $\boldsymbol{b}$. Different $\boldsymbol{\omega}$ and $\boldsymbol{b}$ may adapt variously for different sets of test data. To resolve the above issues, authors in [36]–[38] utilize evolutionary algorithms such as particle swarm optimization (PSO) to select the optimal neurons with more compact networks.

Furthermore, multiple ELM networks can be ensembled as a whole system to avoid individual ELM neurons' parameters deficiency, known as ensembled ELM (EELM). The output of EELM can be chosen as the mean of the outputs from individual ELMs as shown in (2.17) [39],

$$\mathbf{Y} = \frac{1}{p}\sum_{i=1}^{p} \mathbf{H}_i \boldsymbol{\beta}_i \tag{2.17}$$

where p is the number of ELMs. In addition, authors in [27] apply the cross-validation technique to the EELM in the training phase to avoid overfitting and iteratively update $\boldsymbol{\omega}$ and $\boldsymbol{b}$ at the ith ELM.

(III) Number of Neurons Selection

As discussed above, the number of neurons tends to be larger than classical tunning-based learning algorithms, while larger neurons tend to make the hidden layer output matrix ill-conditioning. How to choose the optimal number of hidden neurons is critical for the generalization of any ELM architecture. Instead of users determining the number of neurons by trial-and-error, the Incremental neurons of ELM (IELM) can be applied to find the optimal number. Authors in [28] propose the error-minimization-based method by incrementally updating the output weights when new neurons are added to the existing ELM. By adding δL_k hidden nodes to the existing ELM with $\mathbf{Y}_k = \mathbf{H}_k \boldsymbol{\beta}_k$, the number of hidden nodes become $L_{k+1} = L_k + \delta L_k$, the output weight $\boldsymbol{\beta}$ can be updated as follows when $N \geq L_{k+1}$:

$$
\begin{aligned}
\mathbf{Q}_{k+1} &= \left((\mathbf{I} - \mathbf{H}_k \mathbf{H}_k^{+})\delta \mathbf{H}_k\right)^{-1} \\
\mathbf{U}_{k+1} &= \mathbf{H}_k^{-1}(\mathbf{I} - \delta \mathbf{H}_k \mathbf{Q}_k) \\
\boldsymbol{\beta}_{k+1} &= \mathbf{H}_{k+1}^{+}\mathbf{O} = \begin{bmatrix} \mathbf{U}_{k+1} \\ \mathbf{Q}_{k+1} \end{bmatrix} \mathbf{O}
\end{aligned}
\tag{2.18}
$$

where

$$
\mathbf{H}_{k+1} = [\mathbf{H}_k, \delta \mathbf{H}_k], \delta \mathbf{H}_k = \begin{bmatrix} g\left(\boldsymbol{\omega}_{L_k+1}, b_{L_k+1}, x_1\right) & \cdots & g\left(\boldsymbol{\omega}_{L_{k+1}}, b_{L_{k+1}}, x_1\right) \\ \vdots & \ddots & \vdots \\ g\left(\boldsymbol{\omega}_{L_k+1}, b_{L_k+1}, x_N\right) & \cdots & g\left(\boldsymbol{\omega}_{L_{k+1}}, b_{L_{k+1}}, x_N\right) \end{bmatrix}_{N \times \delta L_k}.
$$

If the neurons are added one by one, $\delta\mathbf{H}_k = \left[g\left(\boldsymbol{\omega}_{L_k+1}, b_{L_k+1}, x_1\right) \quad \cdots \quad g\left(\boldsymbol{\omega}_{L_{k+1}}, b_{L_{k+1}}, x_N\right)\right]_{N\times 1}^T$, $\mathbf{Q}_k$ in (2.18) can be converted to (2.19).

$$\mathbf{Q}_{k+1} = \frac{\delta\mathbf{H}_k^T(\mathbf{I} - \mathbf{H}_k\mathbf{H}_k^+)}{\delta\mathbf{H}_k^T(\mathbf{I} - \mathbf{H}_k\mathbf{H}_k^+)\delta\mathbf{H}_k} \tag{2.19}$$

Authors in [40] combine the ℓ_2-RELM and the IELM to enhance the generalization performance of the IELM, the output weight $\boldsymbol{\beta}$ can be updated by (2.20):

$$\begin{aligned} \mathbf{Q}_{k+1} &= \frac{\delta\mathbf{H}_k^T(\mathbf{I} - \mathbf{H}_k\mathbf{P}_k)}{\delta\mathbf{H}_k^T(\mathbf{I} - \mathbf{H}_k\mathbf{P}_k)\delta\mathbf{H}_k + \gamma} \\ \mathbf{U}_{k+1} &= \mathbf{P}_k(\mathbf{I} - \delta\mathbf{H}_k\mathbf{Q}_{k+1}) \\ \boldsymbol{\beta}_{k+1} &= \mathbf{D}_k\mathbf{O} = \begin{bmatrix}\mathbf{U}_{k+1} \\ \mathbf{Q}_{k+1}\end{bmatrix}\mathbf{O} \end{aligned} \tag{2.20}$$

Instead of directly using the Moore-Penrose generalized inverse $\mathbf{H}_k^+$, $\mathbf{P}_k$ can be solved by (2.15), where $\mathbf{P}_k = (\gamma\mathbf{I} + \mathbf{H}_k^T\mathbf{H}_k)^{-1}\mathbf{H}_k^T$.

(IV) Hidden Layer Input Data Scaling

Besides selecting the number of hidden neurons, ELM is sensitive to the range of input variables. In the ELM architecture, the parameters of the hidden neurons are independent of the input data. Without tunning these parameters, the hidden layer net inputs could distribute into the activation function's saturated or linear regime [29]. Thus, ELM can be plagued by input data scaling due to the non-uniformly distributed hidden layer net inputs. Authors in [29], [41] propose to add affine transformation before the activation function to map the hidden neurons' input data to a uniformly distributed range and avoid a saturated regime. The expression of the affine transformation are explicated below:

$$\mathbf{V} = at(\mathbf{X}') = s \cdot \Gamma(\mathbf{X}') + t \tag{2.21}$$

$\mathbf{X}'$: hidden neurons' input data, $\mathbf{X}' = \mathbf{X} \times \mathbf{W} + \mathbf{b}$.

$at(\cdot)$: the function of an affine transformation.

Γ: the concatenation of all elements in $\mathbf{X}'$ in ascending order, $\Gamma(\mathbf{X}) \triangleq [x_1', \dots, x_{N\times m}']^T$ where x' is the element in $\mathbf{X}'$ and $x'_1 \leq \cdots \leq x'_{N\times m}$.

$\mathbf{V}$: a uniformly distributed vector in the range of v^l and v^u, $\mathbf{V} \triangleq [v_1, \dots, v_{N\times m}]^T$, $v_1 = v^l$ and $v_{N\times m} = v^u$.

s and t can be estimated in (2.22) by gradient descent [42].

$$\mathcal{E} = \min_{s,t} \frac{1}{2} \|g(s \cdot x' + t) - v\|_2^2 \tag{2.22}$$

For each iteration in the gradient descent with learning rate η, s and t can be updated by (2.23).

$$\begin{aligned} s(k+1) &= s(k) - \eta \frac{\partial \mathcal{E}}{\partial s(k)}, \frac{\partial \mathcal{E}}{\partial s} = \sum_{i=1}^{N\times m} [g(s \cdot x_i' + t) - v_i] \frac{\partial g}{\partial s} \\ t(k+1) &= t(k) - \eta \frac{\partial \mathcal{E}}{\partial t(k)}, \frac{\partial \mathcal{E}}{\partial t} = \sum_{i=1}^{N\times m} [g(s \cdot x_i' + t) - v_i] \frac{\partial g}{\partial t} \end{aligned} \tag{2.23}$$

where k is the iteration step. s and t stop updating when they converge, or maximal iteration step is reached. Downsampling can also be applied for $\mathbf{H}$ and $\mathbf{V}$ for efficient computing without yielding computation accuracy, where

$$\Gamma(\widetilde{\mathbf{X}}') = \left[\widetilde{x_1'}, \dots, \widetilde{x_{Nd}'}\right]^T, Nd \leq N \times m$$

$$\widetilde{\mathbf{V}} = [\tilde{v}_1, \dots, \tilde{v}_{Nd}]^T, Nd \leq N \times m$$

Learning rate η can be derived by backtracking line search (BLS) based on Armijo condition [43]:

$$\mathcal{E}(k + \eta\mathbf{p}) \leq \mathcal{E}(k) + \eta c m$$

where $\mathbf{p}$ is the search direction for local decrement, $m = \nabla\mathcal{E}^T\mathbf{p}$, c is a predefined control parameter and $c \in (0,1)$. For gradient descent, $\mathbf{p}$ can be chosen as the $-\nabla\mathcal{E}$. The computation for affine parameters is presented in Algorithm 2.1.

Algorithm 2.1: Optimized Affine Parameters

Input: $\boldsymbol{x}, f(x)$

Output: learning rate α

Choose $c \in (0,1)$, $\alpha_j = 1$, $\tau \in (0,1)$, $j = 0$

Calculate search direction $\mathbf{p} = -\nabla f(\boldsymbol{x})$, $m = -\nabla f(\boldsymbol{x})^T \times \nabla f(\boldsymbol{x})$

while $f(\boldsymbol{x} + \alpha\mathbf{p}) > f(\boldsymbol{x}) + \alpha \cdot c \cdot m$ and $j \leq Iter_{max}$

    $j = j + 1, \alpha_j = \tau \cdot \alpha_{j-1}$

end while

return α

2.2.4 Learning Scheme

In many real-world applications, the computation of the learning process is slow due to an extensive training dataset. Sometimes, a complete set of data is usually not available at the training stage. Besides, when the training datasets increase over time, the datasets may include outdated data that affect the accuracy of the trained networks. In this section, various learning processes are presented to tackle the issues described above.

(I) Batch Learning (BL)

During the training process, a Batch Learning (BL) algorithm uses the entire training set to generate an output hypothesis, which is a function that maps instances in $\mathbf{X}$ to labels in $\mathbf{Y}$ [44]: $\mathbb{Y} = f_{\boldsymbol{BL}}(\mathbb{X})$, where $f_{\boldsymbol{BL}}(\cdot)$ is the trained BL model, $\mathbf{X} \in \mathbb{X}, \mathbf{Y} \in \mathbb{Y}$. BL is also called offline training. $f_{\boldsymbol{BL}}(\cdot)$ is expected to predict any y accurately with unseen x, where $x \in \mathbb{X}, x \notin \mathbf{X}$, and $y \in \mathbb{Y}, y \notin \mathbf{Y}$. If new data samples come to the trained model, this model must be retrained from scratch using both previous and new data samples [45].

(II) Online Learning (OL)

An Online Learning (OL) algorithm receives a sequence of data (either individually or in mini-batches) and updates the model one by one: $\mathbb{Y} = f_{\boldsymbol{OL}}(\mathbb{X})$, where $\mathbb{X}$ and $\mathbb{Y}$ are incremental data samples and $f_{\boldsymbol{OL}}(\cdot)$ is the trained OL model. $f_{\boldsymbol{OL}}(\cdot)$ is an internal hypothesis [44] that is continuously updated with new sequences during the OL training process. OL is ideal when data is generated continuously in real-time, and the entire data is unavailable at the training stage.

(III) Forgetting Learning (FL)

As discussed above, during the training process, the entire data sometimes is not available. In some cases, the trained model $f_{\boldsymbol{tr}}(\cdot)$ could be less relevant to the new data samples, which is due to the weights of $f_{tr}(\cdot)$ are changed to meet the objectives of new data samples [46]. Those outdated data may make $f_{tr}(\cdot)$ less accurate [47]. A Forgetting Learning (FL) in this section receives data sequentially and updates the model with adaptively adjust the outdated data weights: $\mathbb{Y} = f_{FL}(\mathbb{X})$, where $\mathbb{X}$ and $\mathbb{Y}$ are relevent data samples and $f_{FL}(\cdot)$ is the trained FL model. FL is ideal when data is time sensitive and only valid within a certain time period [47].

2.3 Solar Generation Prediction

Short-term solar generation prediction is critical for efficient real-time microgrid operation. Solar generation data is time-sensitive, as illustrated in Figure 2.2 – Figure 2.3. Data is collected from the 100 kW solar PV system located at the College of Engineering Center for Environmental Research and Technology (CE−CERT) facility of the University of California – Riverside (UCR). Figure 2.2 depicts the monthly solar generation, and Figure 2.3 depicts the average hourly solar generation in different months of 2019. It can be found that the generated energy during cold months (such as November – February) is less than the hot months (such as May – August). Overall, the highest solar generation happens in the months when the days are long and sunny with the cooler ambient temperature (i.e. June). Due to data loss from September 14^{th} – September 18^{th}, the sum of solar generation in September is less than its standard generation. Due to the daytime savings that started in March and ended in November, the hourly solar generation curves are shifted by one hour. Overall, the solar generation shows distinct seasonal patterns. Hence, short-term solar generation prediction based on the ELM algorithm with the forgetting learning is applied and presented in this section.

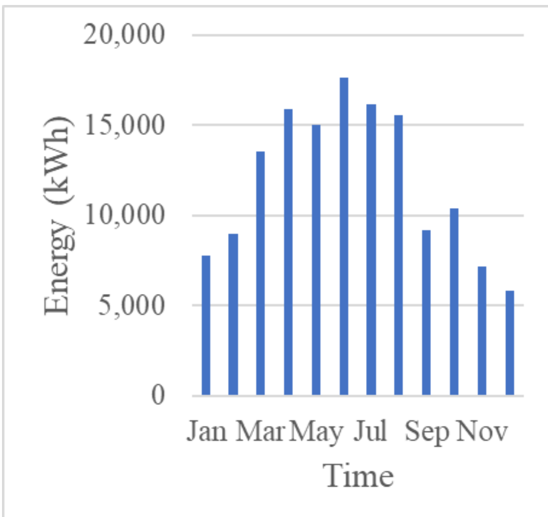


Figure 2.2: Sum of solar generation in different months of 2019.

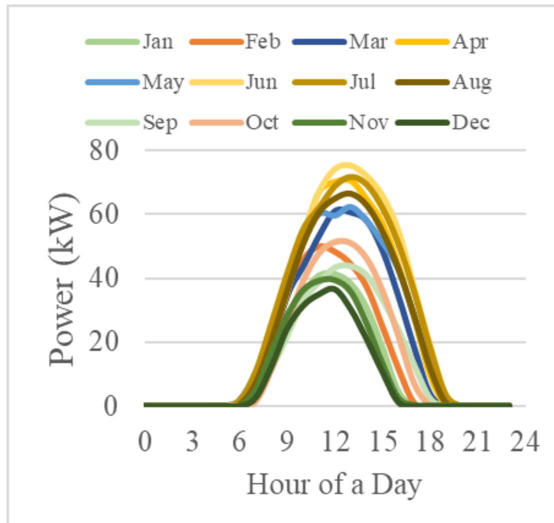


Figure 2.3: Average of hourly solar generation in different months of 2019.

2.3.1 Algorithm Description

(I) Background

Authors in [47] utilize the classific ELM with the forgetting mechanism to predict the short-term solar generation. However, as discussed in Section 2.2 and Figure 2.4, different settings of the neurons of ELM may behave differently. In Figure 2.4, 15-minute solar generation data (details of the training data sets are discussed in Section 2.3.2) of the previous seven days is utilized for training the ELM model, and solar generation of the next day is used for testing. Two separate ELM trials with different randomized neurons are tested. In Trial 1, the ELM model fits the data very well with the testing error of Mean Square Error (MSE) = 0.47. However, prediction performance in Trial 2 is worse with MSE = 6.08. In this example, the drawback of ELM is discussed. The following section illustrates the details of the proposed ELM with forgetting learning and affine transformation.

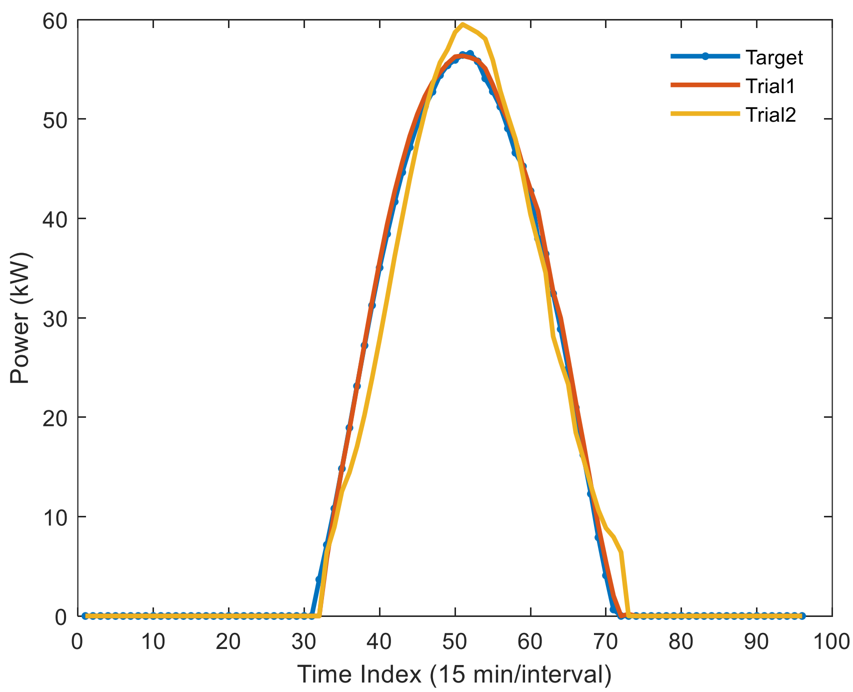


Figure 2.4: ELM prediction performance with two random neurons trials.

(II) Forgetting Learning Formulation

For the online training process with the growing input data size, only $N \geq m$ is considered where $rank(\mathbf{H}) = m$. During the online training process, at step k, $\boldsymbol{\beta}_k$ can be estimated by (2.15) where $\boldsymbol{\beta}_k = \mathbf{P}_k \mathbf{H}_k^T \mathbf{O}_k$, and $\mathbf{P}_k$ can be written in (2.24):

$$\begin{gathered}\mathbf{P}_k = (\gamma \mathbf{I} + \mathbf{H}_k^T \mathbf{H}_k)^{-1} \\ \mathbf{K}_k = \mathbf{P}_k^{-1}\end{gathered} \tag{2.24}$$

At step $k+1$, incremental data $(\delta \mathbf{X}_k, \delta \mathbf{O}_k) = \left\{(x_i, o_i), i = (\sum_{j=0}^{k} N_j) + 1, \dots, (\sum_{j=0}^{k+1} N_j)\right\}$ is added to (2.5), the hidden layer output matrix $\mathbf{H}_{k+1} = \begin{bmatrix} \mathbf{H}_k \\ \delta \mathbf{H}_k \end{bmatrix}$, where the update process can be illustrated below [48]:

$$\begin{gathered}\delta \mathbf{H}_k = g(\delta \mathbf{X}_k, \boldsymbol{\omega}, \boldsymbol{b}) \\ \mathbf{K}_{k+1} = \gamma \mathbf{I} + \mathbf{H}_{k+1}^T \mathbf{H}_{k+1} = \gamma \mathbf{I} + [\mathbf{H}_k^T \quad \delta \mathbf{H}_k^T] \begin{bmatrix} \mathbf{H}_k \\ \delta \mathbf{H}_k \end{bmatrix} \\ \mathbf{K}_{k+1} = \mathbf{K}_k + \delta \mathbf{H}_k^T \delta \mathbf{H}_k\end{gathered} \tag{2.25}$$

$$\begin{aligned}\boldsymbol{\beta}_{k+1} &= \mathbf{K}_{k+1}^{-1} [\mathbf{H}_k^T \quad \delta \mathbf{H}_k^T] \begin{bmatrix} \mathbf{O}_k \\ \delta \mathbf{O}_k \end{bmatrix} \\ &= \mathbf{K}_{k+1}^{-1} (\mathbf{H}_k^T \mathbf{O}_k + \delta \mathbf{H}_k^T \delta \mathbf{O}_k) \\ &= \mathbf{K}_{k+1}^{-1} (\mathbf{K}_k \mathbf{K}_k^{-1} \mathbf{H}_k^T \mathbf{O}_k + \delta \mathbf{H}_k^T \delta \mathbf{O}_k) \\ &= \mathbf{K}_{k+1}^{-1} (\mathbf{K}_k \boldsymbol{\beta}_k + \delta \mathbf{H}_k^T \delta \mathbf{O}_k)\end{aligned}$$

Apply (2.25) to the above equation:

$$\begin{aligned}\boldsymbol{\beta}_{k+1} &= \mathbf{K}_{k+1}^{-1} \left((\mathbf{K}_{k+1} - \delta \mathbf{H}_k^T \delta \mathbf{H}_k) \boldsymbol{\beta}_k + \delta \mathbf{H}_k^T \delta \mathbf{O}_k \right) \\ &= \boldsymbol{\beta}_k + \mathbf{K}_{k+1}^{-1} \delta \mathbf{H}_k^T (\delta \mathbf{O}_k - \delta \mathbf{H}_k \boldsymbol{\beta}_k)\end{aligned}$$

$\mathbf{K}_{k+1}^{-1}$ can be derived by Woodbury matrix identity [49] and $\boldsymbol{\beta}_{k+1}$ is updated in (2.26):

$$\begin{aligned}
\mathbf{K}_{k+1}^{-1} &= \mathbf{K}_k^{-1} - \mathbf{K}_k^{-1}\delta\mathbf{H}_k^T(\mathbf{I} + \delta\mathbf{H}_k^T\mathbf{K}_k^{-1}\delta\mathbf{H}_k^T)^{-1}\delta\mathbf{H}_k\mathbf{K}_k^{-1} \\
\mathbf{P}_{k+1} &= \mathbf{P}_k - \mathbf{P}_k\delta\mathbf{H}_k^T(\mathbf{I} + \delta\mathbf{H}_k\mathbf{P}_k\delta\mathbf{H}_k^T)^{-1}\delta\mathbf{H}_k\mathbf{P}_k \\
\boldsymbol{\beta}_{k+1} &= \boldsymbol{\beta}_k + \mathbf{P}_{k+1}\delta\mathbf{H}_k^T(\delta\mathbf{O}_k - \delta\mathbf{H}_k\boldsymbol{\beta}_k)
\end{aligned} \tag{2.26}$$

Since the online sequential trained model is a combination of old data and new data, instead of direct disregarding the old data illustrated in [48], an adaptive weight can be added to the old data to emphasize the timeliness of the data samples, wherein (2.25) $\mathbf{K}_{k+1}$ can be updated by:

$$\mathbf{K}_{k+1} = \lambda\mathbf{K}_k + \delta\mathbf{H}_k^T\delta\mathbf{H}_k$$

(2.26) then can be updated in (3.30).

$$\begin{aligned}
\mathbf{P}_{k+1} &= \frac{1}{\lambda}\mathbf{P}_k - \frac{1}{\lambda^2}\mathbf{P}_k\delta\mathbf{H}_k^T\left(\mathbf{I} + \frac{1}{\lambda}\delta\mathbf{H}_k\mathbf{P}_k\delta\mathbf{H}_k^T\right)^{-1}\delta\mathbf{H}_k\mathbf{P}_k \\
\boldsymbol{\beta}_{k+1} &= \boldsymbol{\beta}_k + \mathbf{P}_{k+1}\delta\mathbf{H}_k^T(\delta\mathbf{O}_k - \delta\mathbf{H}_k\boldsymbol{\beta}_k)
\end{aligned} \tag{2.27}$$

λ can be adaptively adjusted by a bounded monotonic function such as in [50], $\lambda = \min(1, \lambda - \tan^{-1}(E))$, where E is the difference between the forecast error with $\boldsymbol{\beta}_{k+1}$ and $\boldsymbol{\beta}_k$. λ can also be retrieved by solving optimization: $\min\limits_{0<\lambda<1}\frac{1}{N_{k+1}}(\delta\mathbf{O}_k - \delta\mathbf{H}_k\boldsymbol{\beta}_{k+1})^T(\delta\mathbf{O}_k - \delta\mathbf{H}_k\boldsymbol{\beta}_{k+1})$ in [51].

(III) Affine ELM with Forgetting Learning (FL-AELM)

In real-time operation, the entire solar generation data or weather information is not always available at the beginning of training. Besides, due to the seasonality of the solar generation, training data may include outdated data that affect the prediction performance. Hence, the online forgetting training process is utilized. The training model f_{AFELM} is updated periodically whenever new data is available. New data can come chunk-by-chunk, such as hourly data points, or one-by-one, such as a single 15-minute data point. The principles of the FL-AELM algorithm can be built as follows:

(i) Initialization: at $k = 0$, get f_{AFELM}^{o} with the historical training solar data sets $(\mathbf{X}_0, \mathbf{O}_0) = \{(x_i, o_i), i = 1, \dots, N_0\}$, and retrieve the valid data parameters $\lambda_0 = 1$.

a. Randomly generalize $\boldsymbol{\omega}_j$ and b_j, $j = 1, \dots, m$, m is the number of hidden neurons, and $m < N_0$.

b. Calculate the hidden layer output matrix $\mathbf{H}_o$:

$$\mathbf{H}_o = \begin{bmatrix} g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_1 + b_1) & \dots & g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_1 + b_m) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_1 \cdot \boldsymbol{x}_{N_0} + b_1) & \dots & g(\boldsymbol{\omega}_m \cdot \boldsymbol{x}_{N_0} + b_m) \end{bmatrix}_{N_o \times m}$$

c. Calculate the affine transformation parameters s and t based on (2.21) – (2.23).

d. Apply affine transformation to $\mathbf{H}_o$ to retrieve the uniformly distributed hidden layer output matrix $\mathbf{V}_o = g(at(\mathbf{H}_0, s_0, t_0))$.

e. Estimate the output weight vector $\boldsymbol{\beta}_0$ based on ℓ_2-RELM optimization and (2.15), where

$$\boldsymbol{\beta}_0 = (\gamma \mathbf{I} + \mathbf{V}_0^T \mathbf{V}_0)^{-1} \mathbf{V}_o^T \mathbf{O}$$

$$\mathbf{P}_0 = (\gamma \mathbf{I} + \mathbf{V}_0^T \mathbf{V}_0)^{-1}$$

γ can be optimized by LOO error-based algorithm similar to [34].

f. Get the $f_{\boldsymbol{AELM}}^{o}(\mathbf{H}_o, \boldsymbol{\beta}_o, s_0, t_0, \gamma)$

(ii) FL: update $f_{\boldsymbol{FL-AELM}}^{k}(\mathbf{H}_k, \boldsymbol{\beta}_k, s_k, t_k, \gamma)$ whenever new data is available.

a. Retrieve output weight matrix $\boldsymbol{\beta}_k$ and $\mathbf{P}_k$ at k.

b. Wait until the data chunk from the $k+1$ time interval is available, the new data chunk $(\delta\mathbf{X}_k, \delta\mathbf{O}_k) = \left\{(x_i, o_i), i = (\sum_{j=0}^{k} N_j) + 1, \dots, (\sum_{j=0}^{k+1} N_j)\right\}$.

c. Calculate the hidden layer output matrix for the outdated and new data chunk: $\tilde{\delta}\mathbf{H}_k, \delta\mathbf{H}_k$, and the affine transformation $\delta\boldsymbol{\mathcal{V}}_k = g(at(\delta\mathbf{X}_k, s_k, t))$.

d. If $\max(\delta\boldsymbol{v}_k) > v^u$ or $\min(\delta\boldsymbol{v}_k) < v^l$, update s_{k+1}, t_{k+1} based on (2.21) – (2.23); else, $s_{k+1} = s_k, t_{k+1} = t_k$.

e. Update $\boldsymbol{\beta}_{k+1}$ and $\mathbf{P}_{k+1}$ based on (3.30) where:

$$\mathbf{P}_{k+1} = \frac{1}{\lambda_k}\mathbf{P}_k - \frac{1}{\lambda_k^2}\mathbf{P}_k\delta\boldsymbol{\mathcal{V}}_k^T\left(\mathbf{I} + \frac{1}{\lambda}\delta\boldsymbol{\mathcal{V}}_k\mathbf{P}_k\delta\boldsymbol{\mathcal{V}}_k^T\right)^{-1}\delta\boldsymbol{\mathcal{V}}_k\mathbf{P}_k$$

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \mathbf{P}_{k+1}\mathbf{P}_{k+1}\delta\boldsymbol{\mathcal{V}}_k^T(\delta\mathbf{O}_k - \delta\boldsymbol{\mathcal{V}}_k\boldsymbol{\beta}_k)$$

f. Update $\lambda_{k+1} = \lambda_k - \left(\frac{1}{5\pi}\right)\tan^{-1}\left(l_{\boldsymbol{FL-AELM}}^{k+1} - l_{\boldsymbol{FL-AELM}}^{k}\right), 0 \le \lambda_{k+1} \le 1$[50].

g. Set $k = k+1$ and go back to (ii).

2.3.2 Parameters Selection

(I) Input Parameters

As discussed in [47], the solar generation of PV arrays is affected by the size and efficiency of the PV array, solar radiation, cell temperature, and ambient temperature. Since the size and efficiency of any given PV system are physically fixed and cannot change the output anymore, variations of cell and ambient temperature and solar radiation can be utilized as the input parameters. Solar radiation is usually forecast by numerical weather prediction (NWP), including atmospheric conditions such as cloud motion [52]. Such historical information is not available for public free access. Instead of solar radiation, solar elevation angle and weather information are combined to represent the weather information. The solar elevation angle is calculated by PV Performance Modeling Toolbox developed by Sandia National Laboratories [53], and the weather information represents simple weather conditions. The weather information can be obtained freely from any public database, in which the weather type is simply coded by indicating different weather conditions by a number value as follows: clear – 10, scattered clouds – 9, partly cloudy – 8, mostly cloudy – 7, overcast – 6, haze – 5, fog – 4, light rain – 3, rain – 2, thunderstorm – 1, no sun (after sunset and before sunrise) – 0. The sunset and sunrise time is computed using JPL ephemerides [54]. As displayed in Figure 2.5, the difference between actual and calculated sunrise and sunset time is negligible. Root Mean Square Error (RMSE) is 0.015

and 0.009, respectively. The actual sunrise and sunset data is retrieved from the Sunrise-Sunset website [55]. In addition to the weather information, historical data such as power values in the previous one hour and day-ahead are also included as the input parameters.

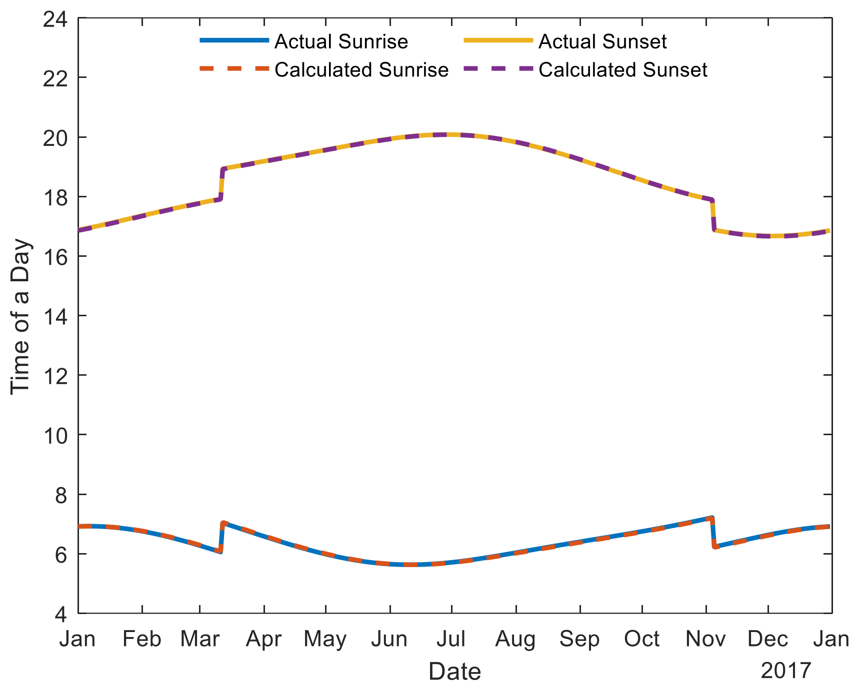


Figure 2.5: Comparison between the actual and calculated sunrise and sunset time.

The solar data is collected from the Sustainable Integrated Grid Initiative (SIGI) database and is resampled to a 15-minute interval with linear interpolation. Input parameters are listed in Table 2.1, where the historical information is utilized, j is the 15-minute duration of historical data. For example, $j = k$ represents $k \times$15-minute ahead solar power. The sampling rate for the cloud coverage data that can be retrieved from the public site is 1 hour. It is resampled to 15 minutes/data by linear interpolation.

Fields	Input Parameters
t	Current time index: $0 < t \leq 96$
p_{t-j}	Historical solar power
wt	Weather type: $0 \leq wt \leq 10$
T	Temperature
cc	Cloud coverage (%)
se	Solar elevation

Table 2.1: Input parameters for solar prediction.

(II) Data Preparation

As shown in Table 2.1, the input parameters vector is $\mathbf{X} = \{\boldsymbol{x}_i | \boldsymbol{x}_i \in (t, p_{t-j}, wt, T, se)\}^T$. The typical data normalization/scaling is depicted in (3.30). The first and second equations normalize the input data to $[-1,1]$ and $[0,1]$, and the last equation standardizes the input data to zero mean and unit deviation (also called z-score normalization). In this section, the input data $\mathbf{X}$ is normalized to $[-1,1]$. During the testing process, data values that are less than -1 or in the time range $\{t | t < t_{rise}\ or\ t > t_{set}\}$ are tuned to -1, and data values that are larger than 1 are tuned to 1.

$$\hat{\boldsymbol{x}}_i = 2\frac{\boldsymbol{x}_i - \min \boldsymbol{x}_i}{\max \boldsymbol{x}_i - \min \boldsymbol{x}_i} - 1, \hat{\boldsymbol{x}}_i \in [-1,1]$$

$$\hat{\boldsymbol{x}}_i = \frac{\boldsymbol{x}_i - \min \boldsymbol{x}_i}{\max \boldsymbol{x}_i - \min \boldsymbol{x}_i}, \hat{\boldsymbol{x}}_i \in [0,1] \qquad (2.28)$$

$\hat{\boldsymbol{x}}_i = \frac{x_i - \bar{x}_i}{x_i - \sigma_i}, \bar{x}_i, \sigma_i$ is the mean and standard deviation of $\boldsymbol{x}_i$

(III) Neurons Number

As discussed in Section 2.2.3, improper selection of the number of hidden neurons can make the output matrix ill-conditioning. In this selection, IELM is applied for choosing the neuron number. To be more specific, neurons are added one by one during the training process, where the output weight matrix $\boldsymbol{\beta}$ can be updated by (2.20). In this section, the training data is selected as the 1-month ahead data, and the test data is the following 5-day data. The activation function is chosen as sigmoid in (2.4). The training and testing data is selected between April – May, as shown in Figure 2.6.

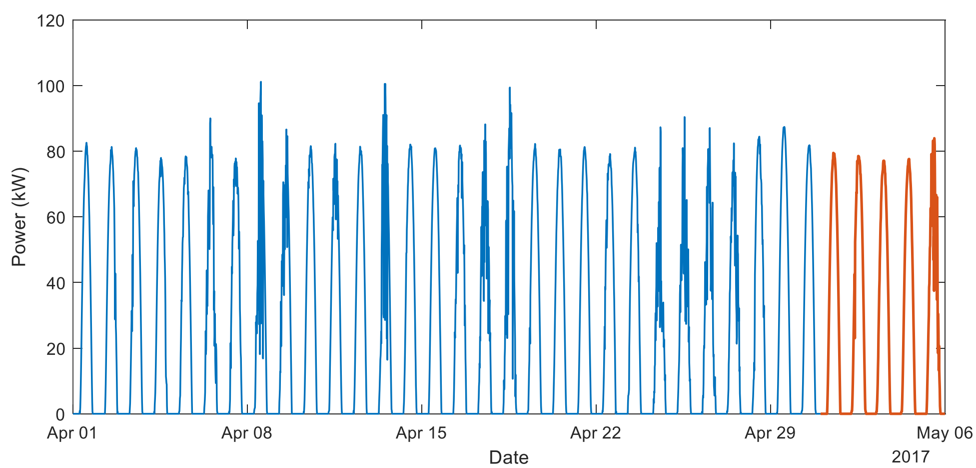


Figure 2.6: April – May training and testing data.

Figure 2.7 compares neurons selection for different algorithms. The neurons are incremented from 10 to 200, and the activation function is chosen as the sigmoid in (2.4). Each algorithm runs for 50 trails, and the final RMSE curves in Figure 2.7 are the average of the 50 trails. As shown in Figure 2.7 (a) – (c), RMSE converges at around 50 neurons, and minor improvement is recognized by adding more neurons for the three algorithms. Among the three algorithms, the ELM algorithm becomes unstable when the neurons are increased more than 160. In contrast, RELM and AELM algorithms maintain more stability with new neurons added.

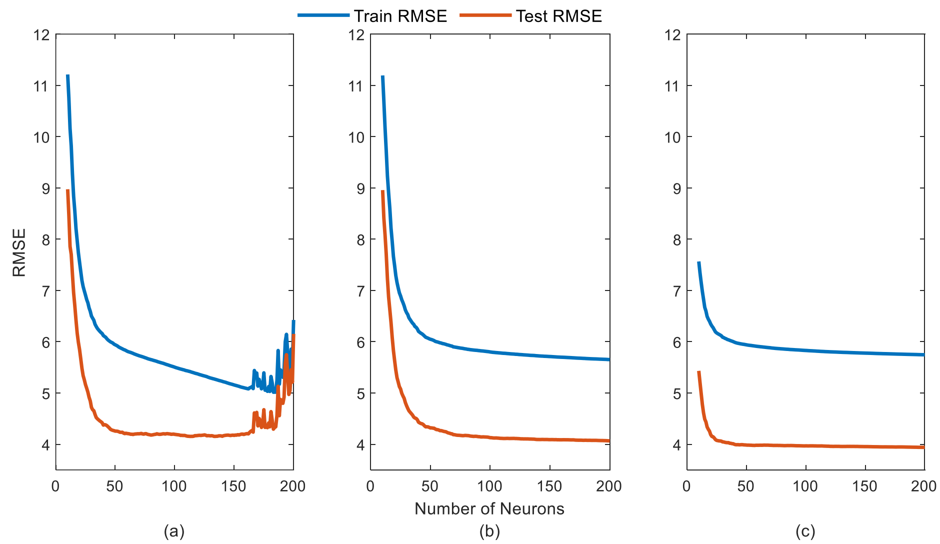


Figure 2.7: RMSE for the different number of neurons by IELM: (a) ELM algorithm, (b) RELM algorithm, and (c) AELM algorithm.

For the sigmoid function, any value out the range of $[-4, +4]$ is saturated and out of the linear regime. As illustrated in Figure 2.8 (a), when more neurons are added to the network, the possibility of elements in the hidden layer input matrix falling out of $[-4, +4]$ increases. Without affine transformation, these elements are directly transferred to $\mathbf{H}$ and saturated. Thus during the training process, instability occurs in Figure 2.7 (a) for ELM. However, Figure 2.8 (b) shows that the hidden layer input matrix is mapped to a more linear regime after the affine transformation.

With proper neurons selected, the ELM and RELM algorithms perform almost equally, and AELM performs better when the number of neurons is small. Besides, due to the randomized nature of the ELM algorithm, stability can be increased by the affine transformation to the input matrix. Overall, AELM outperforms the other two with lower RMSE and a smaller number of neurons. In the following sections, otherwise stated, the number of neurons is selected as 50 for solar prediction simulation and evaluation.

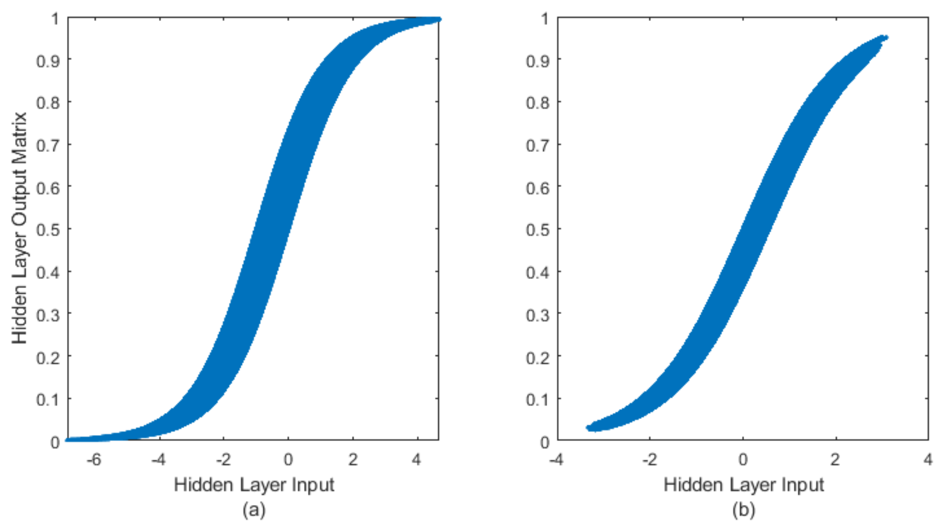


Figure 2.8: Hidden layer output matrix illustration (neurons = 200): (a) without affine transformation, and (b) with affine transformation.

2.3.3 Performance Evaluation

In this section, different scenarios of solar prediction are discussed.

(I) Prediction on Different Weather Conditions

Figure 2.9 illustrates a one-month solar power prediction using the prior month as the training sets via the FL-ELM algorithm. The 'No Update' curves represent the training process without online updating. The model is trained every month. The 'OL' curves represent the online updating based on (2.26), and the 'FL' curves represent the online updating with forgetting weights based on (3.30) and the FL-AELM algorithm. The training data covers most sunny conditions, which results in the prediction results with different learning schemes for the sunny day in Figure 2.9 (b) are similar to each: $RMSE_{NU} = 1.26, RMSE_{OL} = 1.25, RMSE_{FL} = 1.25$. In Figure 2.9 (c), the weather is more cloudy. As the cloud coverage data sample rate is 1 hour, the prediction can not accurately find the 15-minute variation. However, the hourly power change can be predicted. The prediction errors for different learning schemes are: $RMSE_{NU} = 7.38, RMSE_{OL} = 7.00, RMSE_{FL} = 6.30$. The improvement ratio of online updating is 5% and 15% for OL and FL, respectively. The month-long prediction is shown in Figure 2.9 (a) with $RMSE_{NU} = 6.25, RMSE_{OL} = 6.21, RMSE_{FL} = 6.19$. When comparing to ELM and RELM in Table 2.2, better prediction performance always lies in AELM with online updating scheme.

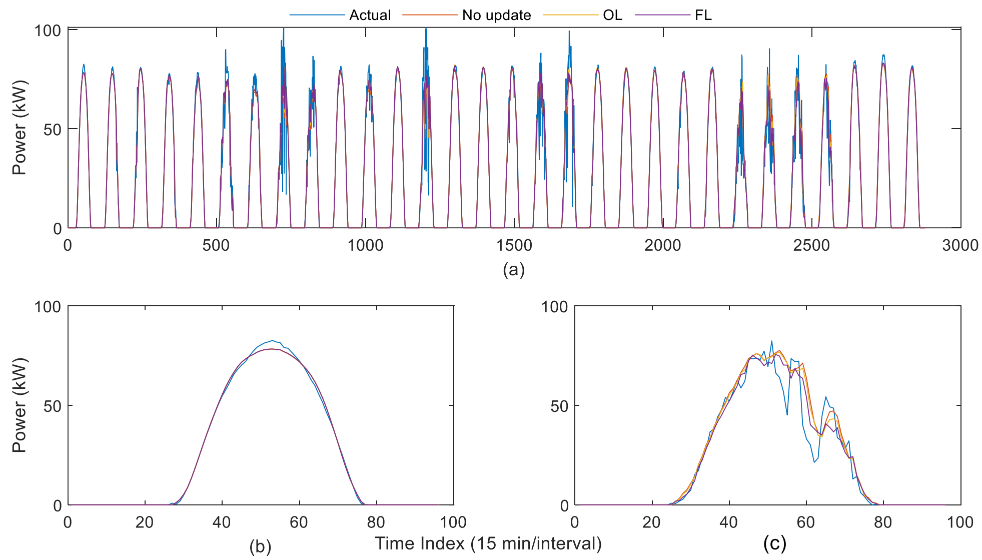


Figure 2.9: Prediction performance for April 2017: (a) month-long prediction; (b) sunny day prediction on April 10th; and (c) cloudy day prediction on April 18th.

Algorithms	Training	No Update	OL	FL
ELM	5.53	6.84	6.62	6.62
RELM	5.59	6.73	6.58	6.53
AELM	**5.34**	**6.25**	**6.21**	**6.19**

Table 2.2: RMSE for solar prediction performance on April 2017 via different algorithms.

(II) Different Settings of Training Data Lengths

In this section, different lengths of the training samples are selected to evaluate the model's accuracy and robustness to different training lengths. The prediction is performed one step ahead, which is 15-minute ahead.

Figure 2.10 displays the prediction comparison between different algorithms. The training data is selected as the one month prior to the testing data. For testing data in January, the training data is selected as the December data. The new data chunk is selected as 1 hour.

The RMSE error is the difference between non-normalized data. From Figure 2.10 (a), the FL-AELM can achieve the lowest $RMSE_{AELM} = 5.07$ for most of the months. RELM and ELM share similar RMSE with $RMSE_{ELM} = 5.54, RMSE_{RELM} = 5.49$. In Figure 2.10 (b) – (d), individual algorithms with different learning schemes are compared. The lowest RMSE can always be achieved by online updating. For one-month training data, the difference between OL and FL is negligible. During the summer months (June – August) and winter months (November – December), the prediction performance achieves the best among the other months due to consistent weather conditions for most of the days in the month.

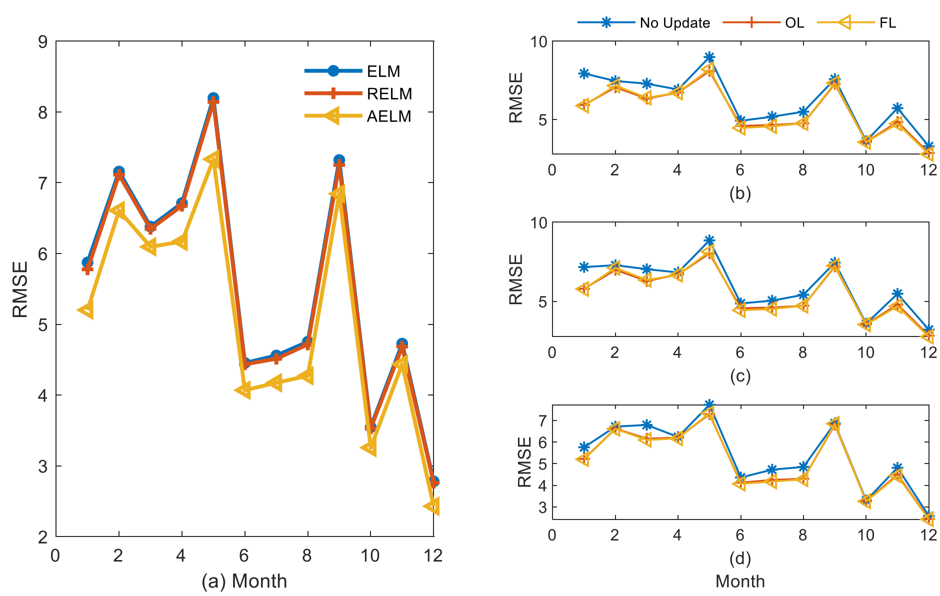


Figure 2.10: Monthly solar prediction with 1-month training data: (a) comparison between different algorithms with FL; (b) – (d) comparisons between different learning schemes of ELM, RELM, and AELM, respectively.

Figure 2.11 shows the prediction results with the training data length of three months. The testing data is post-1-month data. The AELM algorithm can achieve the lowest RMSE for most of the months: $RMSE_{AELM} = 5.12, RMSE_{ELM} = 5.50, RMSE_{RELM} = 5.49$.

Figure 2.12 shows the prediction results with the training data length of six months. The testing data is post-1-month data. With more training data added, the difference between different algorithms for most months becomes smaller. The monthly average $RMSE_{AELM} = 5.11, RMSE_{ELM} = 5.50, RMSE_{RELM} = 5.49$. FL slight improves during June – August with less RMSE.

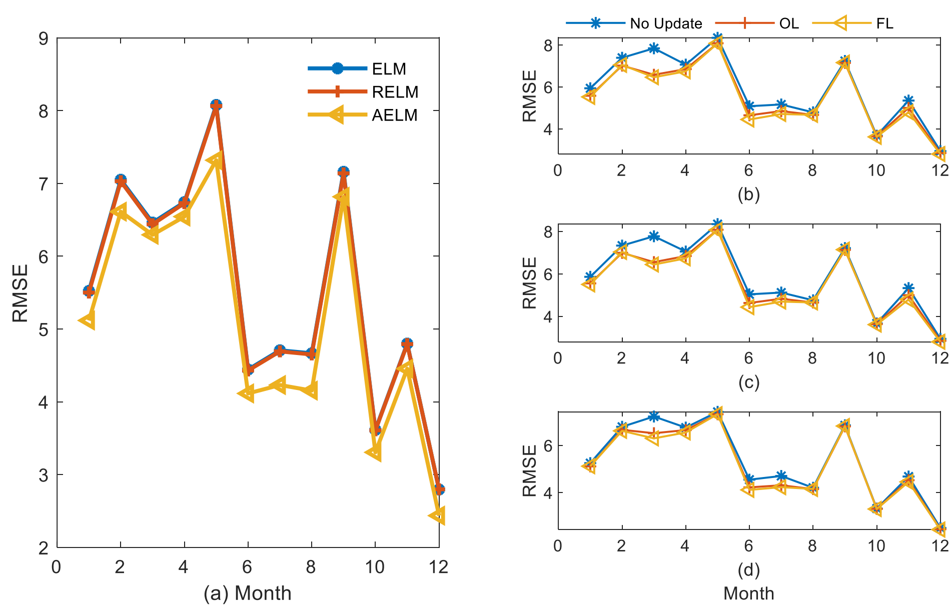


Figure 2.11: Monthly solar prediction with 3-month training data: (a) comparison between different algorithms with FL; (b) – (d) comparisons between different learning schemes of ELM, RELM, and AELM, respectively.

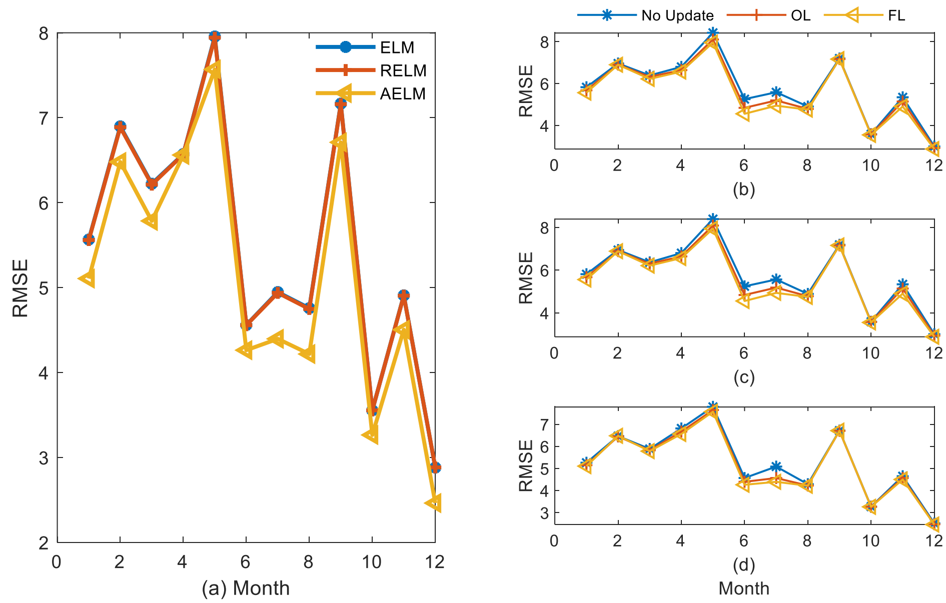


Figure 2.12: Monthly solar prediction with 6-month training data: (a) comparison between different algorithms with FL; (b) – (d) comparisons between different learning schemes of ELM, RELM, and AELM, respectively.

For short-term prediction (15-minute), the length of training data cannot improve the model accuracy when the data length is more than one month, and the prediction model can be updated every month. Besides, the FL slightly improves the prediction. Figure 2.13 displays the RMSE boxplot for the 20 testing trails for each month, where the training and testing data features are the same as Figure 2.10. The AELM always performs the best and most stable compared to ELM and RELM, and online/forgetting learning schemes are always more robust than the 'No Update' scheme.

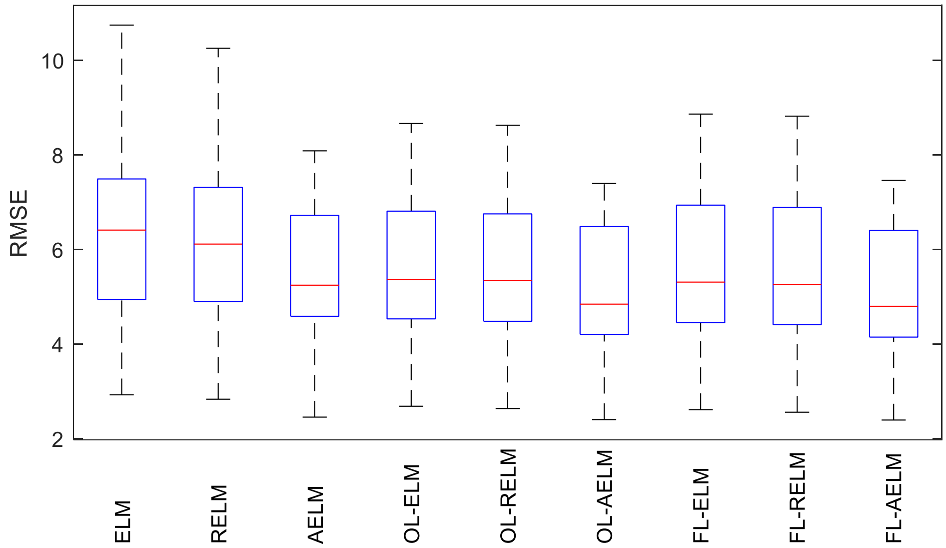


Figure 2.13: Boxplot for year-round testing RMSE with one-month prior training data.

2.3.4 Comparison with State-of-Art Machine Learning Algorithms

In this section, different state-of-art machine learning models (ML) are presented.

- Linear Regression (LR)

A general linear regression model can be modeled as $y_i = \beta_0 + \sum_{j=1}^{m} \beta_j \cdot X_{ij} + \epsilon_i, i = 1, \ldots, N$, where $f(\cdot)$ is a scalar-valued function of the independent variables X_{ij}, and ϵ_i is the noise [56]. An interaction (IA) variation of LR is when one variable has a different effect on the outcome depending on another variable, $y_i = \beta_0 + \sum_{j=1}^{m} \beta_j \cdot X_{i.j} + \sum_{k=1,k\neq k'}^{m} \beta_k \cdot X_{i,k} \cdot X_{i,k'} + \epsilon_i$. A robust variation of LR is to iteratively tune $\omega \cdot \sum_{y=1}^{N} \|y_i - \hat{y}_i\|^2$ to compute $\boldsymbol{\beta}$. A general weight function $\omega(\cdot)$ is 'biweight' [57]. A stepwise (SW) variation of LR is to find the most significant variables by adding and removing terms from the general LR model [58].

- Support Vector Machine (SVM)

A support vector machine algorithm is to find a function $f(x)$ that closes to y_i, where $|f(x) - y_i| < \varepsilon$ and the deviation for each $x \in X$ is as flat as possible [59]. The SVM utilizes different kernel functions to project $X_{i.j}$ into a high-dimensional space. In this section, three different kernels are utilized: 1) linear: $K(x_i, x_j) = x_i^T x_j$; 2) Gaussian: $K(x_i, x_j) = e^{(-\|x_i - x_j\|^2)}$; 3) polynomial (poly): $K(x_i, x_j) = (1 + x_i^T x_j)^q, q = 2,3, \ldots$.

- Gaussian Progress Regression (GPR)

GPR extends the general LR to a probabilistic model by introducing latent variables $f(x)$, and basis function h, where $f(x) \sim GP(0, K(x, x^T))$. $GP(\cdot)$ is a Gaussian process (GP) that random variables have joint Gaussian distribution. The general LR then transforms to $h(x)^T \beta + f(x)$, The probability of response $P(y_i | f(x_i), x_i) \sim N(y_i | h(x_i)^T \beta + f(x_i), K(x, x^T))$ [60]. The covariance (kernel) functions $K(x, x^T)$ in this section are selected as: 1) exponential (Exp) 2) Matern 5/2; 3) rational quadratic (RA); 4) squared exponential (SE) [61].

- Ensembled Tree (ET)

The ensembled tree algorithm extends the decision tree regression by ensembling several decision trees to produce the final prediction. Two ensemble algorithms are utilized in this section: Least-squares boosting (LS) and bootstrap aggregation (Bag) [62].

Three error matrices are compared: RMSE, Mean Absolute Error (MAE), and R-squared score R^2. RMSE and MSE measure the average prediction errors between the actual and predicted values. R^2 measures how well the predictions approximate the actual data points.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |o_i - \hat{o}_i|$$

$$R^2 = 1 - \frac{\sum_i (o_i - \hat{o}_i)^2}{\sum_i (o_i - \bar{o}_i)^2}$$

Table 2.3 displays the prediction accuracy for individual months with the training data selected as the one-month prior data. The highest prediction accuracy for each month is marked bold. The ‘Type’ is selected as the highest prediction accuracy among different variations or kernels for each model. Prediction accuracy in December is the highest while in May is the lowest among all models. FL-AELM has the highest prediction accuracy in 6 months with an average of 9.87%, 8.28%, 13.31%, and 10.65% RMSE reduction compared to LR, SVM, GPR, and ET, respectively.

Testing Month	LR				SVM				GPR				Ensembled Tree				FL-AELM		
	RMSE	MAE	$R2$	Type	RMSE	MAE	$R2$	Type	RMSE	MAE	$R2$	Type	RMSE	MAE	R^2	Type	RMSE	MAE	$R2$
Jan	5.89	2.28	0.89	Linear	5.78	2.05	0.90	Linear	6.04	2.54	0.89	Exp	5.97	2.63	0.89	Bag	**5.18**	**1.97**	**0.92**
Feb	**6.61**	**2.53**	**0.90**	SW	6.67	2.42	0.90	Poly	6.74	2.78	0.90	Exp	6.56	2.72	0.90	Bag	6.62	2.80	0.90
Mar	5.87	2.37	0.96	SW	6.13	2.29	0.95	Poly	5.81	2.27	0.96	Exp	**5.85**	**2.23**	**0.96**	Bag	6.03	2.53	0.95
Apr	6.58	2.66	0.95	Linear	6.64	2.34	0.95	Poly	6.75	2.51	0.95	Exp	6.25	2.30	0.96	Bag	**6.18**	**2.36**	**0.96**
May	8.15	3.40	0.93	SW	7.89	3.21	0.93	Poly	8.14	3.29	0.93	Exp	7.97	3.30	0.93	Bag	**7.30**	**2.86**	**0.94**
Jun	4.39	1.75	0.98	Linear	4.48	1.72	0.98	Poly	4.52	1.87	0.98	Exp	**3.97**	**1.25**	**0.98**	Bag	4.06	1.37	0.98
Jul	4.65	1.73	0.97	Linear	4.43	1.75	0.97	Linear	5.16	1.74	0.96	Exp	4.86	1.84	0.97	Bag	**4.18**	**1.46**	**0.98**
Aug	4.58	1.81	0.98	IA	4.57	1.98	0.98	Linear	5.24	2.40	0.97	Exp	5.11	2.60	0.97	Bag	**4.27**	**1.54**	**0.98**
Sep	7.01	2.62	0.93	SW	7.53	2.74	0.92	Linear	7.09	2.70	0.93	Exp	**6.74**	**2.47**	**0.93**	Bag	6.83	2.56	0.93
Oct	3.25	0.96	0.98	IA	3.40	0.91	0.98	Poly	**3.21**	**1.00**	**0.98**	Exp	3.40	1.15	0.98	Bag	3.25	1.00	0.98
Nov	4.97	2.25	0.90	Linear	4.99	1.96	0.90	Linear	5.19	2.49	0.89	Exp	5.30	2.65	0.89	Bag	**4.45**	**1.81**	**0.92**
Dec	2.72	1.15	0.97	SW	2.51	0.92	0.98	Poly	2.80	1.08	0.97	Exp	**2.42**	**0.89**	**0.98**	Bag	2.43	0.91	0.98

Table 2.3: Monthly solar generation prediction comparisons via different algorithms.

2.4 Electric Vehicle Charging Load Prediction

Over the last three decades, the greenhouse gas (GHG) emissions from the transportation sector increased by over 22% in the United States. In 2019, the GHG emission from the transportation section produced the most significant GHG emission, accounting for 29% of the total yearly generated GHG emission [63]. The electrification of transportation, including adopting a larger low/zero-carbon electric vehicles (EVs) population on the road, provides a significant key to GHG reduction. During the last decade, the population of EVs has been increasing rapidly worldwide, registered as the year-on-year growth of 14% in the United States and 40% globally, respectively [64]. As a result, the EVs' charging load inevitably increases dramatically. The yearly UCR's EV load from 2015 – 2019 is shown in Figure 2.14. The charging load incremental rate is around 31% – 51% and reached an average of 33,254 kWh increased energy per year.

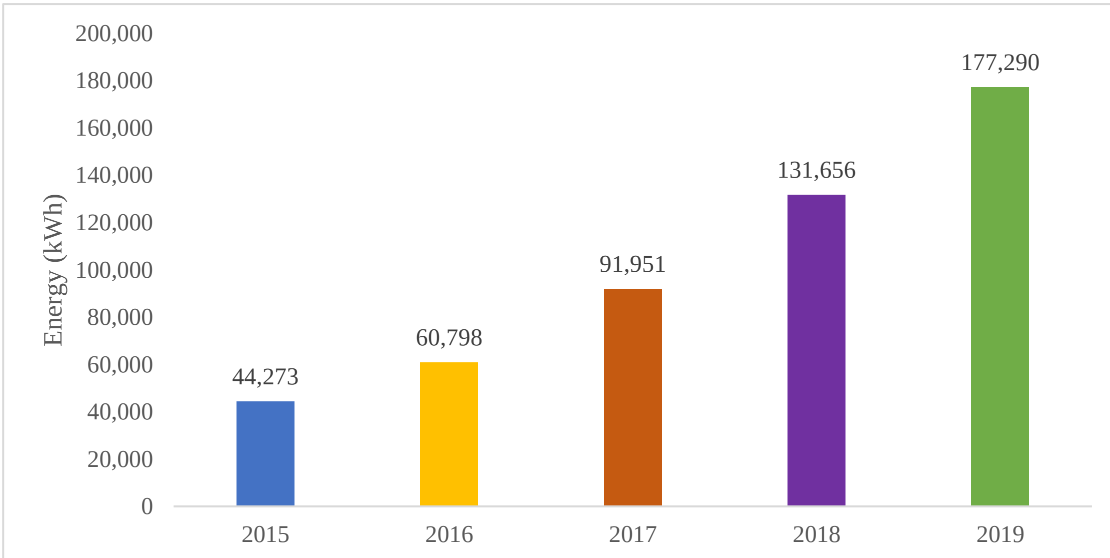


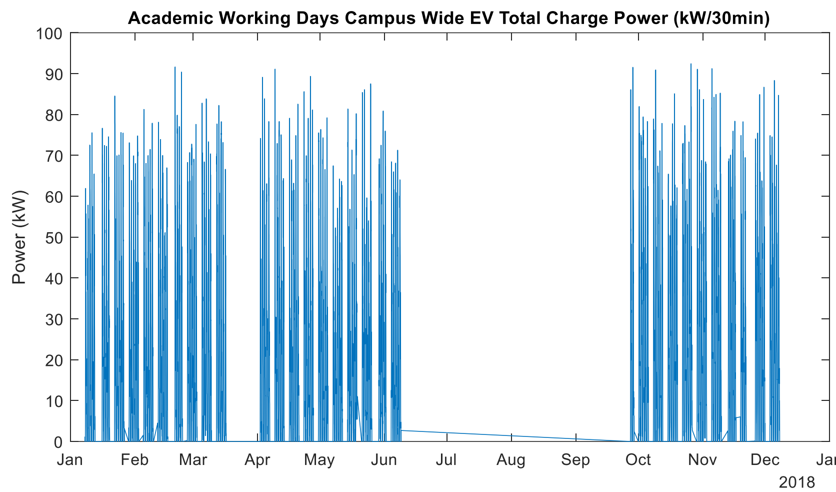
Figure 2.14: Yearly EV charging energy from 2015 – 2019.

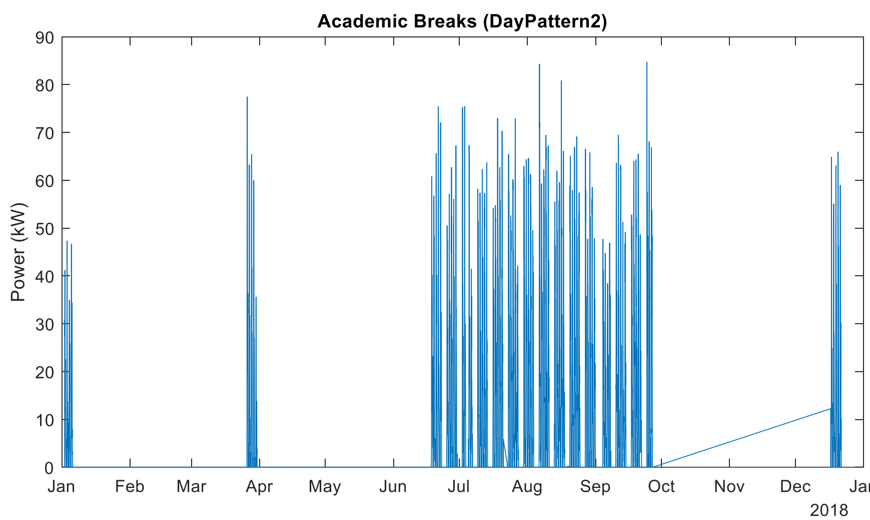
Similar to Section 2.3, this section applies the FL-AELM to predict the charging loads for UCR. Numeric analyses are performed and discussed to show the effectiveness of the proposed prediction algorithm.

2.4.1 Parameters Selection

(I) Input Data

The charging loads are heavily impacted by different day patterns of users' charging behavior on campus, as shown in Figure 2.15. During typical academic working days (DayPattern 1), charging events are the highest, while charging events on non-working days are the lowest (DayPatten 3 – 5). During academic breaks, such as summer and winter breaks, the charging loads share a similar pattern with DayPattern 1 but are more sparse. Charging loads for DayPattern 1 and 2 are more variable and share more uncertainty. Hence, in this section, charging data used for training and testing are retrieved in DayPattern 1 and 2. When retrieving the historical training data in Table 2.4, data is within the same DayPattern groups.





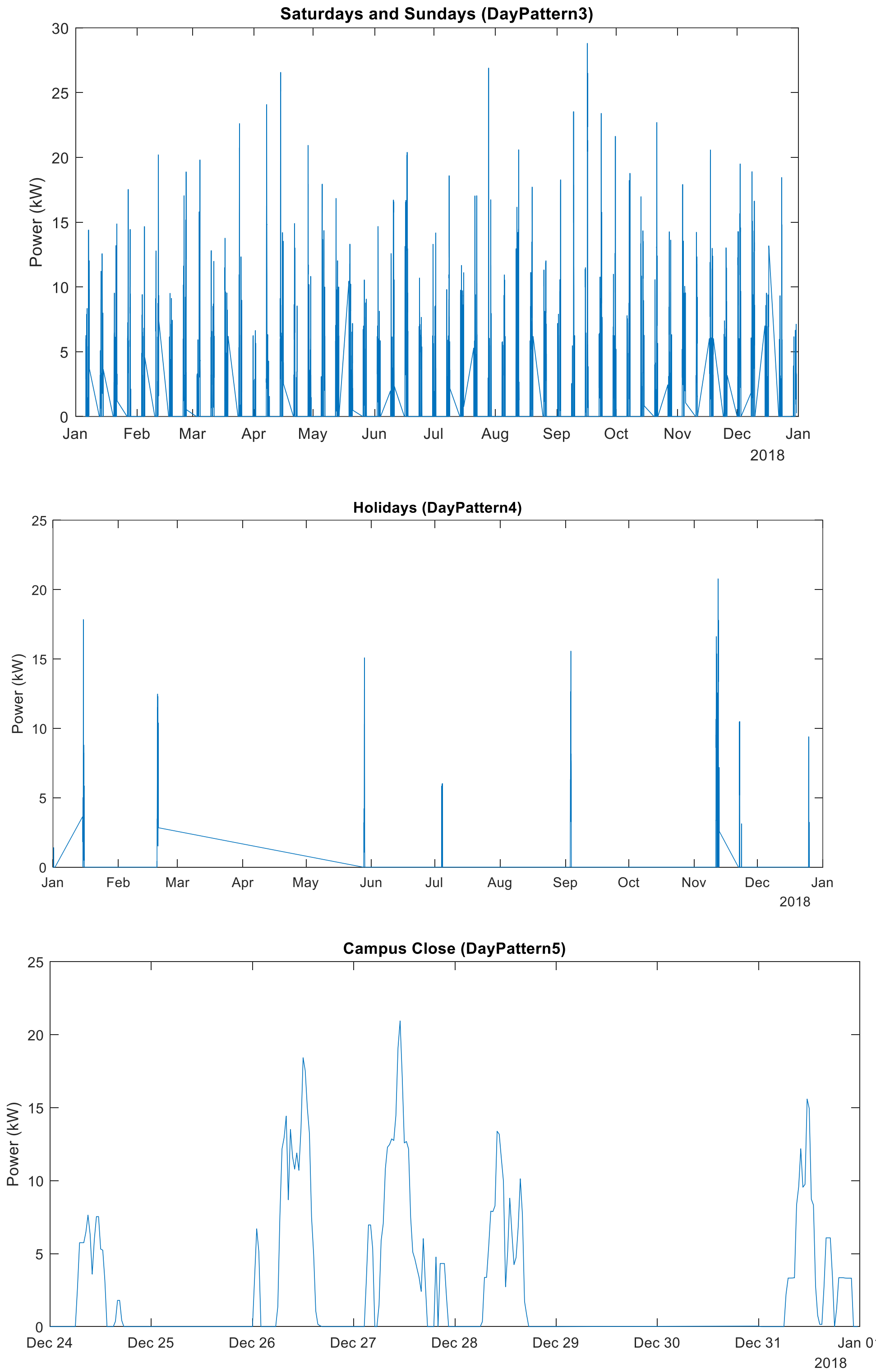


Figure 2.15: Different day patterns of EV charging loads.

Fields	Input Parameters
t	Current time index: $0 < t \leq 48$
p_{t-j}	Historical charging power
dp	Day patterns
nc	Number of connection

Table 2.4: Input parameters for EV charging loads prediction.

(II) Neurons Number

Similar to solar prediction, the number of hidden neurons is selected by IELM. RMSE for different algorithms is shown in Figure 2.16, where the training data is from the academic working days of April 2018, and testing data is from the following week. The number of the optimal hidden neurons for AELM is the smallest when comparing the other two algorithms. All three algorithms can achieve better Test RMSE when the number of neurons reaches 100. Thus, neurons number is set to 100 in this section for different prediction scenarios.

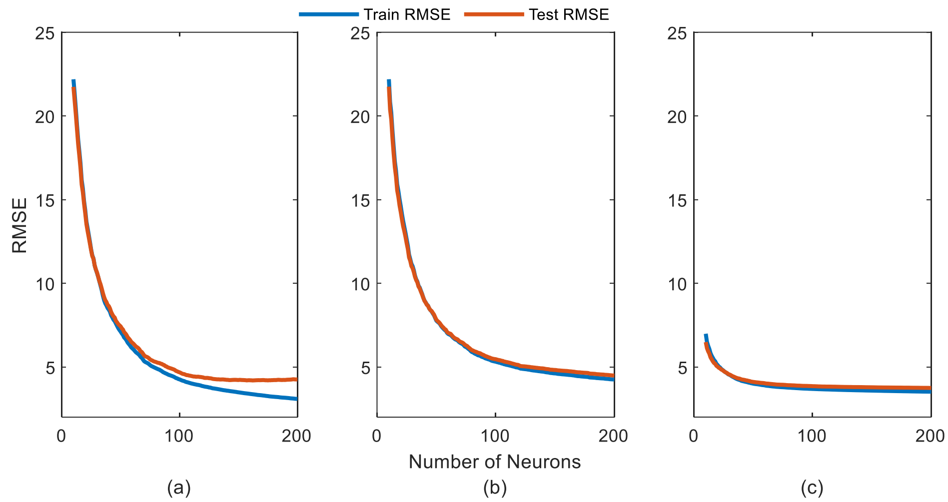


Figure 2.16: RMSE for the different number of neurons by IELM for EV loads prediction: (a) ELM algorithm, (b) RELM algorithm, and (c) AELM algorithm.

2.4.2 Simulation Results and Performance Evaluation

Different algorithms are assessed and compared in this section, as shown in Table 2.5 – Table 2.6. The training parameters are selected as: p_{t-j} and the number of connected chargers in $t-1$. The training data is selected as the entire year of 2018, and the testing data is selected from January – April 2019. In the training process, the RMSE is calculated by cross-validation of 10-folds. The EGPR is the best regression model learned by the MATLAB Regression Learner [65]. Since most of the charging events ended in 2 – 3 hours, the training parameters are selected as (I): $p_{t-7\cdot 48}$, $p_{t-14\cdot 48}$, $p_{t-21\cdot 48}$, $p_{t-28\cdot 48}$, p_{t-1}, p_{t-2}, p_{t-3}, p_{t-4}, p_{t-5}, p_{t-6} and (II) $p_{t-7\cdot 48}$, $p_{t-14\cdot 48}$, $p_{t-21\cdot 48}$, $p_{t-28\cdot 48}$, p_{t-1}, p_{t-2}, p_{t-3}, p_{t-4}. During the training process, different algorithms share similar results. The EGPR performs the worst. The online learning scheme always performs better during the testing process than other algorithms, and the FL-ELM algorithm performs the best under (I) and (II). The training time for ELM algorithms is the total time of the 20 trials, and for the EGPR algorithm is the total time for 5-folds training validation. The best testing results always fall into the FL-AELM algorithm, which is improved by 53.89% compared to the classic ELM, 18.80% compared to the AELM without forgetting learning, and 13.98% compared to EGPR, the best state-of-art regression model learned by the software. An example of the six-week prediction performance from 4/22/2019 – 4/29/2019 of different algorithms is shown in Figure 2.17.

Training	RMSE I	MAE I	R^2 I	RMSE II	MAE II	R^2 II	Training Time
ELM	7.4	5.46	0.91	5.38	3.73	0.95	0.42 s
RELM	7.44	5.52	0.91	5.41	3.77	0.95	5.36 s
AELM	4.98	3.28	0.96	4.97	**3.26**	0.96	10.49 s
EGPR	4.89	**3.21**	0.96	**4.95**	3.27	0.96	45.59 s

Table 2.5: EV charging loads training performance comparisons.

Testing	RMSE I	MAE I	R^2 I	RMSE II	MAE II	R^2 II
ELM	13.21	8.13	0.82	7.62	4.83	0.94
RELM	12.9	8.04	0.83	7.49	4.81	0.94
AELM	7.5	4.55	0.94	7.24	4.40	0.95
OL-ELM	9.55	6.88	0.91	6.60	4.40	0.96
OL-RELM	9.48	6.68	0.91	6.58	4.41	0.96
OL-AELM	6.4	4.02	0.96	6.29	3.95	0.96
FL-ELM	9.0	6.47	0.92	6.46	4.36	0.96
FL-RELM	8.96	6.47	0.92	6.44	4.37	0.96
FL-AELM	**6.09**	**3.89**	**0.96**	**5.99**	**3.83**	**0.96**
EGPR	7.08	4.34	0.95	7.01	4.33	0.95

Table 2.6: EV charging loads testing performance comparisons.

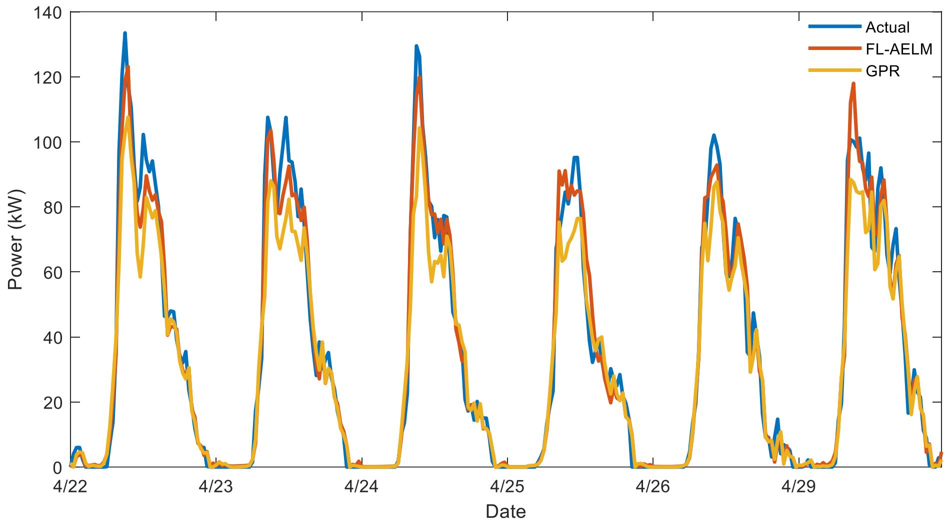


Figure 2.17: Prediction performance comparison for different algorithms.

2.5 Conclusion

In this chapter, the novel FL-AELM is proposed and applied to predict solar and EV power accurately. In Section 2.2, different ELM variations and limitations are discussed. Due to the randomized nature of the ELM algorithm and the non-uniformly distributed hidden layer net inputs, the hidden layer output could drop into the activation function's saturated or linear regime. By applying the affine transformation, the hidden layer net inputs can be projected into a non-saturated zone uniformly, thus improving the model stability.

In real-time operation, the entire solar/EV information is not always available at the beginning of training, and training data may become outdated due to seasonality. To further improve the prediction accuracy, the forgetting learning scheme is combined with the

AELM. In Sections 2.3 and 2.4, the accurate FL-AELM prediction model is achieved by first finding the optimal number of neurons via IELM, then applying the FL-AELM to update the prediction model online. By comparing to the other state-of-the-art ML algorithms, the improved accuracy is further validated.

Chapter 3 Electric Vehicles Charging Characteristics

The beneficial environmental impacts of EVs with less GHG production draw the world's attention, and nationwide, the population of EVs has increased rapidly in recent years. As a result, the number of electric vehicle supply equipment (EVSE) has also proliferated, especially for public EVSEs, nearly doubled between 2015 and 2019 [66]. This trend has led to significant attention in different aspects of EV charging management, including drivers charging behavior modeling and prediction [67], the model-free EV charging coordination by artificial intelligence [68], the recommendation of charging schedule for the sequential arrived EVs in a network of charging station [69], and Vehicle-to-Grid (V2G) considering battery-aging [70]. The critical challenge for EV charging management in public places lies in the limitation of non-exchangeable battery information between EV and EVSEs, hindering the formulation of the EV battery charging behavior dynamics from the EVSE side.

In this chapter, the unique EV charging characteristics: tail feature is discussed. Diverse charging current profiles can be classified into seven groups through the novel distance measure: refine matrix profile similarity. This chapter also discusses EV charging scheduling through centralized and distributed algorithms. With the consideration of charging tail characteristics, the EV smart charging algorithm shows more efficient energy scheduling.

3.1 Background for Electric Vehicles Charging Characteristics

3.1.1 Electric Vehicle Charging Dynamics

The charging process is generally divided into two stages for a general EV charging load profile: the constant charging current (CC) and the constant charging voltage (CV) stage. During the CC stage, the charging current maintains constant, while in the CV stage, the charging current gradually decreases to 0 till the full SOC of the battery. Some works consider parameters such as the SOC, and the battery capacity is perceived to the EVSE operator [70] – [71] without establishing any battery charging model. Other works simplify the battery charging model by assuming a constant charging power pattern during the entire charging process [68] – [69] or considering a predetermined charging power threshold to identify the CV stage [67]. However, the simplification of the battery model inhibits the accuracy of real-time EV management and scheduling problem.

There are two common approaches to formulate the EV charging loads dynamics: model-based and data-driven approaches. Authors in [72] utilize the ZIP model (impedance Z, current I, and power P) to formulate the charging load equivalent circuit systematically. The ZIP parameters are estimated by curve fitting to the actual experimental data. However, the available battery capacity is a prerequisite in the ZIP model. Therefore, the ZIP parameters for different batteries may vary significantly. In [73], the authors formulate the charging current change in the CV stage as a simple linear decreasing function. The decrement/slope of the current is a fixed value based on the EV battery characteristics. Similar to [73], authors in [74] model the charging power in the CV stage as a staircase curve with multiple discrete average power values. The mono-modeling of the charging current/power variation results in the model's non-scalability to a charging infrastructure with diverse EVs.

On the other hand, authors in [75] develop the non-linear EV charging model by machine learning with the charging time series data. The neural network is utilized for training the regression model of the relationship between the charging power and the EV charging time series with the inputs of different EV models, SOC and the arrival time. However, the EV models and SOC are not commonly available for the public EVSEs, the modeling of the SOC and charging power is not feasible. Authors in [76] apply a more generalized model to learn the battery behavior by classifying the large-scale charging current time series in the CV stage into small groups. The cluster framework is utilized by iteratively extracting and clustering the current time series in the CV stage. However, the iterative framework ignores the inaccurate extraction of the charging time series when the external EVSEs control the EV battery.

Different battery charging characteristics in the CV stage may behave differently. However, the diverse battery types may have small groups of charging behavior dynamics. The charging current time series, especially the subsequence in the CV stage, can represent a unique feature for identifying the battery charging model. Similar to [76], this chapter introduces the data-driven battery charging model by employing the extraction and classification for the charging time series in the CV stage. The extraction of the charging time series is directly through the time series with the consideration of operational noise, including the external charging schedule.

3.1.2 Electric Vehicle Charging Energy Management

An increasing number of the campus EV charging ports and uncoordinated charging events can negatively impact the campus demand load and potentially create an enormous burden for the substation. EV charging energy scheduling, also known as smart charging, is an effective way to coordinate charging events and maintain stable charging profiles. It has been conducted in many published works [71], [77]–[81]. The grid to vehicle (G2V) and vehicle to grid (V2G) technologies are utilized to smooth out the load variance and minimize the peak-to-average ratio [77], [80], minimize the total daily cost [71], [77]–[79], mitigate the impacts on the utility grid [78], balance the fluctuation of RESs [79] and reduce the emissions.

EV charging management can be classified into two categories: centralized and distributed control approaches. [82]–[84] propose centralized EV smart charging algorithms in order to find the global system objectives. In [82], the smart charging objective aims at maximizing the charging station's capacity and minimizing the delay of EVs by selecting the charging mechanisms between CCCV and multi-stage constant current (MSCC). Authors in [83] develop a satisfaction fairness parameter and embed this parameter into the objective function to reduce the system's overall charging electricity while maintaining high fairness among users. A bi-level smart charging schematic is [84], where the upper-level controller (aggregator) can coordinate the temporal charging schedules for each EV to shift load along the control horizon, and the lower-level controller (each EVSEs) can coordinate the spatial schedules to minimize the peak at the current time.

The centralized smart charging requires the entire system and the individual EVSE's dynamics to be known at each control step. It requires sophisticated communication and may jeopardize the privacy issue. Besides, the computation burden for optimization may increase dramatically with the number of EVSEs; thus, the scalability of centralized control cannot be maintained. The distributed control algorithms are proposed to handle the above limitations. In [85], a general framework of distributed control optimization is proposed with the objectives of load shifting and minimizing the charging electricity cost. Authors in [86] propose a multiagent framework to minimize energy costs, avoid transformer overloads, and allow EV recharging. Each agent (EVSE) learns the cooperative, selfish and final cooperation criteria policies via reinforcement learning.

However, none of the works of literature considers the actual dynamic EV charging characteristics and inherits them to the charging constraints. Besides, many works ignore the uncertainty of future charging events in the distributed control. The proposed centralized and distributed control algorithms in this section aim at tackling the above two challenges.

3.2 Electric Vehicle Charging Time Series Classification via Refine Matrix Profile Similarity

Electric vehicles charging time series can be divided into constant current and constant voltage (tail) stages. During the latter stage, different types of the battery may behave differently in their charging time series. The tail is a unique feature to identify the battery information and the real-time charging stage, contributing to more effective charging scheduling. This chapter introduces a novel tail extraction algorithm and distance measure – Refine Matrix Profile Similarity ($RMPS$). This section first classifies the variable-length, noisy time series in the extracted tail stage by k-medoids. The tails are clustered into seven groups with distinct representatives (medoids). Then the comprehensive classification evaluation by different criteria is conducted. The classification performance of the proposed $RMPS$ surpasses other distance measures, including Ecludience distance and dynamic time warping. Lastly, the potential applications of the learned clusters and representatives are illustrated.

3.2.1 Charging Time Series Introduction

In this section, EV charging profiles are obtained from the Adaptive Charging Networks (ACN)-Database. ACN-Data is a public and dynamic EV charging dataset collected from the real-world workplace EV charging data [21]. There are 15 fields in the ACN-data to record the details of each charging session [87]. The focus is on the charging time series of each charging session. Therefore, three fields in the ACN-Data are targeted: userID (the unique identifier of the user), chargingCurrent (time series of the measured current received from the EVSE), and the pilotSignal (time series of the pilot signals sent to the EVSE). We collect all data from the same site (i.e., Caltech). The ACN-data includes high-resolution real-time current measurements. Therefore, all data samples are preprocessed by resampling them to 5s/data with linear interpolation, aligning charging and pilot time series with the same timestamp, and adding a moving average window length of 20 to filter out any high-resolution noise.

Over 80% of the public and workplace EVSE in the United States are level 2, which utilize the J1772 protocol to communicate with the plugged EVs [88]. The J1772 defines the pilot control signal as the communication between EVSE and EV. The pilot signal (current) sets the boundary of the maximum charging power delivered to the plugged EVs [89]. Charging current during the CC stage remains constant, while different EV batteries may perform significantly different during the CV stage. Typically, the time series data in the CV stage is more valuable to analyze. Hence, $\boldsymbol{tail}$ is defined as a subsequence of the charging time series consisting of a set of consistently nonincreasing current in the CV stage in (3.1):

$$\boldsymbol{tail} = \{c_i | c_{i-1} \geq c_i, \forall i\} \tag{3.1}$$

where c_i is the charging current at time index i. In practice, due to the different charging behavior of the battery and the scheduled control in the ACN-data by pilot signal, the $\boldsymbol{tail}$ is not always a decreasing sequence. To mitigate the inference from the pilot signal, we adopt the piloted points definition from [76]. Any piloted point at t is defined as the charging current at t which is controlled by the pilot signal by scheduling in (3.2). In addition to the piloted point, (3.3) locates the charging current points continuously controlled by the pilot signal during the CV stage.

$$\boldsymbol{pts}(t) - \boldsymbol{cts}(t) \leq \boldsymbol{pts}(t-1) - \boldsymbol{cts}(t-1) \tag{3.2}$$

$$\boldsymbol{pts}(t) - \boldsymbol{cts}(t) < 0 \tag{3.3}$$

where $\boldsymbol{pts}$ represents for the pilot current time series, $\boldsymbol{cts}$ represents for the current time series.

Figure 3.1 exhibits two charging events with the charging current characteristic influenced by the pilot current from the same userID. The top figure is a typical charging current profile with the CC-CV stages. During the CV stage, the $\boldsymbol{tail}$ current gradually decreases from the constant charging current and eventually reaches 0. The bottom figure displays an example of the charging current controlled by the pilot current. The charging current is always less than the pilot current. During the end stage of charging sessions, the $\boldsymbol{tail}$ current also gradually decreases but no longer follows the typical tail patterns of the battery. It is essential to extract the tails directly with the given $\boldsymbol{cts}$ and $\boldsymbol{pts}$, while without the knowledge of the battery information or historical charging data of the battery.

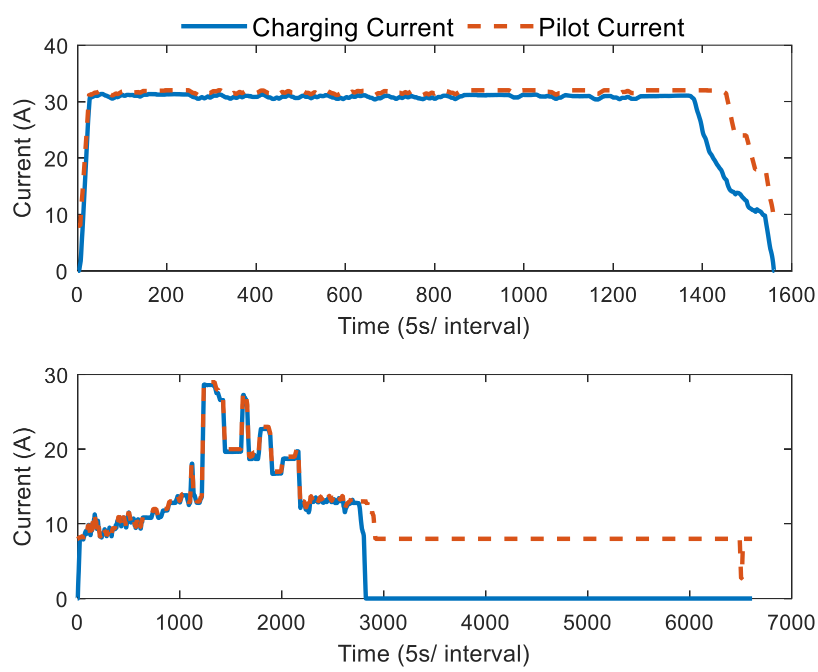


Figure 3.1: Charging current and pilot current for two charging events: top). charging current not controlled by the pilot current; bottom) charging current controlled by the pilot current.

3.2.2 Tail Extraction Algorithm

Tail is a non-increasing portion in cts regardless of the pilot current. Therefore, the extracted tail time series ($\boldsymbol{tts}$) consistently follows a non-increasing pattern without the piloted points (external control) and gradually decreases to zero. Algorithm 3.1 finds the non-increasing portion of $\boldsymbol{cts}$ by locating the starting and ending points of $\boldsymbol{tts}$. It first finds ending point t_e of the tail by iteratively finding the last non-increasing points of the $\boldsymbol{cts}$. It then locates the the tail starting point t_s at local maximum of the $\boldsymbol{tts}$ starting from t_e. In line 1, $\boldsymbol{cts}$ is reversed to $\boldsymbol{cts}'$ and the first-order derivate $\boldsymbol{c\dot{t}s}'$ is differentiated. Lines 2 – 6 use while loop to continuously finding $\boldsymbol{cts}'$ increasing indices, where ϵ_1 is the threshold for finding the maximum value in $\boldsymbol{cts}$ and $\epsilon_1 \in (0,1]$. In lines 3 and 4, the first increasing index and its corresponding local maximum of $\boldsymbol{cts}'$ are recorded in t_{e_o} and t_{s_o}, respectively. Line 5 records all t_{s_o} in the vector LOC. In line 7, the tail ending point t_e is the last index of $\boldsymbol{cts}'$subject to $\boldsymbol{cts}'(t_e) \leq \boldsymbol{cts}'(LOC) + \epsilon_2$. A small value ϵ_2 is added to the minimal value of $\boldsymbol{cts}'(LOC)$ to indicate the completion of the charging session, and $t_e \geq 1$. Lines 8 – 10 find the tail starting point t_s not created by of the pilot signal, where ϵ_3 in line 8 is the tolerance range for continuously piloted points.

Algorithm 3.1: Direct tail extraction

Input: any charging and pilot time series: $\boldsymbol{cts}$ and $\boldsymbol{pts}$ with length N
Output: tail time series: $\boldsymbol{tts}$

$\boldsymbol{cts}' \leftarrow \boldsymbol{cts}(N, N-1, \dots, 1), \dot{\boldsymbol{cts}}' \leftarrow \frac{d\boldsymbol{cts}'}{dt}, t_{s_o}, t_{e_o} \leftarrow 1$

while $\boldsymbol{cts}'(t_{s_o}) < \epsilon_1 \cdot \max(\boldsymbol{cts})$

$\quad t_{e_o} \leftarrow t_{s_o} + t - 1,\ t \leftarrow \min_{t \geq t_{s_o}} \{t | \dot{\boldsymbol{cts}}'(t) > 0\}$

$\quad t_{s_o} \leftarrow t_{e_o} + t - 1,\ t \leftarrow \min_{t \geq t_{e_o}} \{t | \dot{\boldsymbol{cts}}'(t) = 0\}$

$\quad LOC \leftarrow [LOC, t_{s_o}]$

end while

$t_e = \max\{1, LOC(t)\}$,
$t \leftarrow \max\{t | \boldsymbol{cts}'(LOC) \leq \min \boldsymbol{cts}'(LOC) + \epsilon_2\}$

$t_{s_1} \leftarrow t_{s_o} + k + \epsilon_3 - 1,\ k$ is the continuously k points satisfied for (3.3).

$t_{s_2} = \begin{cases} t_{s_1},\ (3.2)\ is\ satisfied\ for\ t \in [t_{s_o}, t_{s_1}] \\ t_{s_o},\ else \end{cases}$

$t_s = t_{s_2} + \operatorname{argmax}_{t \in [t_{s_2}, t_e]} \boldsymbol{cts}(t) - 1$

return $\boldsymbol{tts} \leftarrow \boldsymbol{cts}(t_s : t_e)$

Figure 3.2 demonstrates tail extraction results from two distinctive users based on Algorithm 1, where the parameters ϵ_1, ϵ_2, and ϵ_3 are chosen as $\epsilon_1 = 0.9, \epsilon_2 = 0.5, \epsilon_3 = 5$. It is observed that different vehicles equipped with different batteries may behave distinct charging tails with different lengths from comparing two different users. In contrast, charging tails remain the same, with various required charging energy for the same type of vehicle. Although the duration for the tails of the same *userID* is different, the tail structure (shape) remains similar. Hence, tails can be chosen as the critical EV features to classify different vehicles on the market.

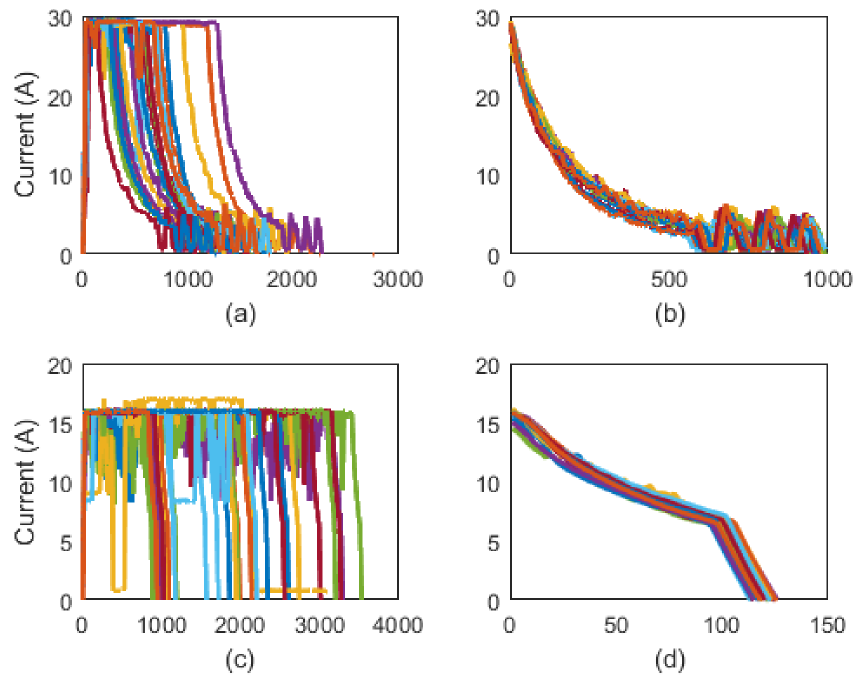


Figure 3.2: Tail extraction for two distinct users. (a). partial charging curves of userID 115; (b) tails extraction from (a); (c). partial charging curves of userID 135; (d). tails extraction from (c).

Data samples are selected from September 2018 – July 2020 from the ACN-Data. There are charging events that do not include any of the tails, resulting in the charging current dropping directly from the constant current to zero in a short period, commonly less than two minutes. The situations are caused by two reasons: 1) the EV battery directly disconnected from the EVSEs before it was charged; 2) the nonexistence of CV stage for such battery. Another common situation is that only a partial tail exists due to the charging sessions controlled by the pilot signal, such as the one shown in Figure 3.1 (bottom). To acquire suitable tails that show actual tails' characteristics without external inference, the chosen tail samples are selected by first excluding samples whose lengths median absolute deviations are outside the percentile threshold of [10,90]. The chosen tails samples are called partial samples in this paper. The partial samples are then partitioned into nine bins with uniform bin widths based on their lengths. The partial tail samples are further narrowed down to the set which has the largest number of samples. The value of nine bins is selected to primarily represent the tail duration distribution and guarantee adequate tail samples from various users. There are a total of 28 frequent users, and 903 partial tails are selected. The selected tails are shown in Figure 3.3 with data scaled to [0,1]. The following section presents the formulation of the classification algorithm by considering variate lengths and noise of tail time series.

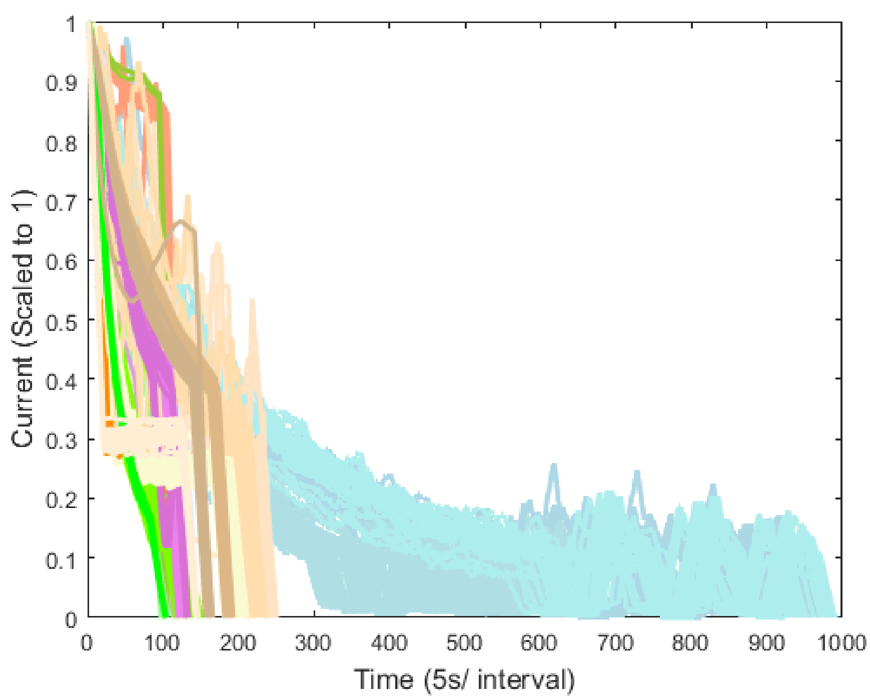


Figure 3.3: Selected partial tails extracted by Algorithm 3.1.

3.2.3 Tail Classification Algorithm

For the time series classification, the key is to find the similarity/distance among different time series and iteratively locate each time series into its nearest neighbor until it converges. The purpose is to cluster tails into small groups to find possible representatives of battery features regardless of the location, lengths of different tails. This section first introduces the background of the Matrix Profile (MP) and then elaborates the proposed distance measure: $RMPS$ for the tail classification formulation.

(I) Background

The Matrix Profile (MP) of a time series $\boldsymbol{T}$ is defined as a vector of the Euclidean distances between the z-score all-subsequences of $\boldsymbol{T}$ and its nearest non-trivial neighbor of z-scored all-subsequences with length L. To make it more straightforward, this section begins by giving related definitions of MP, and more details can be found in [90] – [92].

Definition 1. An all-subsequence $\mathcal{S}$ of time series $\boldsymbol{T}$ with length $|\boldsymbol{T}|$ is defined as:

$$\mathcal{S} = \{\boldsymbol{T}_{1,L}, \boldsymbol{T}_{2,L}, \dots, \boldsymbol{T}_{N-L+1}\},$$

$$\boldsymbol{T}_{\boldsymbol{i},\boldsymbol{L}} = \boldsymbol{T}(i, i+1, \dots, i+L-1)$$

where i is the index of T. The length of subsequence $\boldsymbol{T}_{\boldsymbol{i},\boldsymbol{L}}$ is L.

Definition 2. Distance $d_i$of $\mathcal{S}_i$ is defined as the nearest non-trivial Euclidean distance within each pair of z-score normalized all-subsequences.

$$d_i = \min_{\forall j \leq SL, j \neq i} ED(s_i, s_j) \tag{3.4}$$

where s_i, s_j is the z-score normalized subsequence of $\mathcal{S}_i$ and $\mathcal{S}_j$, calculated in (3.30). SL is the size of $\mathcal{S}$ equal to $|\boldsymbol{T}| - L + 1$. $ED(\cdot)$ calculates the Euclidean distance between s_i and s_j. Any $j \neq i$ is the non-trivial neighbor of s_i.

Definition 3. *Matrix Profile* ($\mathcal{mp}$) is a vector that stores all $d_i, i = 1, \ldots, SL$. Thus, a $\mathcal{mp}$ of two time series $\boldsymbol{A}$ and $\boldsymbol{B}$ can be defined as

$$\mathcal{mp}_{\boldsymbol{AB}} = [d_{\boldsymbol{A}_1}, \ldots, d_{\boldsymbol{A}_{|\boldsymbol{A}|-L+1}}],$$

$$d_{\boldsymbol{A}_i} = \min_{\forall j \leq |\boldsymbol{B}|-L+1} ED(\boldsymbol{A}_i, \boldsymbol{B}_j)$$

Please note that in general $\mathcal{mp}_{\boldsymbol{AB}} \neq \mathcal{mp}_{\boldsymbol{BA}}$.

Definition 4. *Join Matrix Profile* of $\boldsymbol{A}$ and $\boldsymbol{B}$ ($\mathcal{mp}_{\boldsymbol{ABBA}}$) is a vector that stores all $d_{\boldsymbol{A}_i}\ and\ d_{\boldsymbol{B}_j}, \forall\, i \leq |\boldsymbol{A}| - L + 1, \forall j \leq |\boldsymbol{B}| - L + 1$.The size of $\mathcal{mp}_{\boldsymbol{ABBA}}$ is $|\boldsymbol{A}| + |\boldsymbol{B}| - 1$.

Figure 3.4 is the demonstration of $\mathcal{mp}_{\boldsymbol{ABBA}}$ calculation between two different tails with equal length $N = 124$ with the subsequence length $L = 20$. The size of all-subsequences of $\boldsymbol{tail}_1$ and $\boldsymbol{tail}_2$ is $SL = 105$. There is a gap within the range of $[105, 124]$ and $[229, 248]$, marked in the green line when the d_i calculation reaches beyond the last subsequence of each tail.

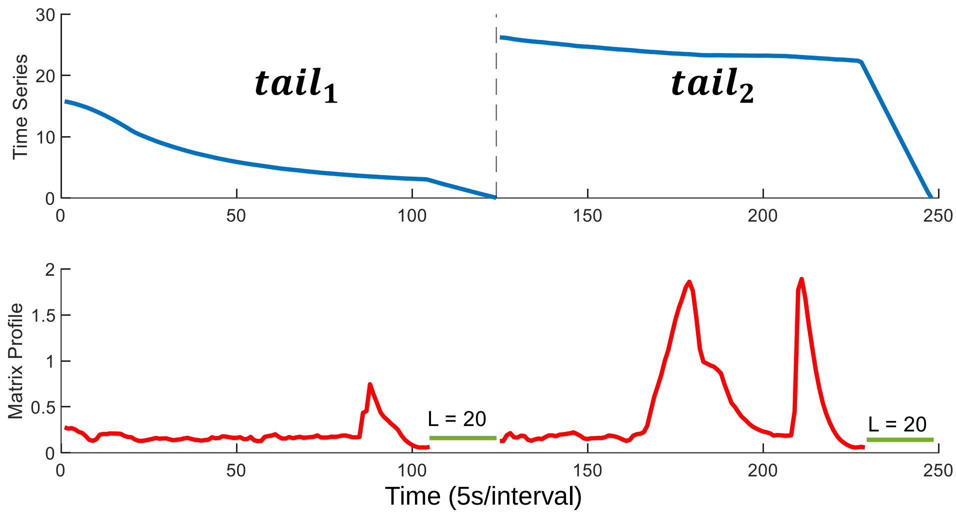


Figure 3.4: top) The concatenation of two tails $\boldsymbol{tail}_1, \boldsymbol{tail}_2$. bottom) $\mathcal{mp}_{\boldsymbol{ABBA}}$ of $\boldsymbol{tail}_1$ *and* $\boldsymbol{tail}_2$ with $L = 20$.

(II) Refine Matrix Profile Similarity

By putting all distance values to $\mathcal{mp}_{\boldsymbol{ABBA}}$, each subsequence distance with the nearest neighbor for two time series is calculated. Intuitively, if two time series are similar, they share multiple similar subsequences, where elements of $\mathcal{mp}_{\boldsymbol{ABBA}}$ are small. Hence, $\mathcal{mp}_{\boldsymbol{ABBA}}$ can be utilized to measure the similarity of two time series. Authors in [91] proposed to use the matrix profile distance, which is defined as the kth smallest values value of the sorted $\mathcal{mp}_{\boldsymbol{ABBA}}$ to measure the distance of two time series. However, the classification accuracy of choosing the appropriate k is non-quantified, especially when the lengths of the two time series are different. Besides, the kth value tends to find the local similarity, and there is no way to quantify any dissimilarity of two time series with a small k. To overcome the above issues, the Refine Matrix Profile Similarity ($RMPS$) is formulated in (3.6). The selection of k is avoided by choosing the larger value between the median and mean of $\mathcal{D}$ in (3.5) as the global distance of any two time series. Besides, the global $RMPS$ omits the temporal locality since more considerable dissimilarity can always be found in (3.6) than the matrix profile distance when any short and long time series sharing a similar shape. The proposed $RMPS$ is computationally efficient, robust, and capable of comparing time series with various lengths.

$$\mathcal{D} = sort(\mathcal{mp}_{\boldsymbol{ABBA}} | \mathcal{mp}_{\boldsymbol{ABBA}} > 0) \tag{3.5}$$

$$RMPS = \max\left(median(\mathcal{D}), mean(\mathcal{D})\right) \tag{3.6}$$

(III) k-Medoids Classification with Refine Matrix Profile Similarity

The k-medoids algorithm is used to cluster samples into the nearest medoids in the data samples. k-Medoids classification is formulated in (3.30), where $\boldsymbol{m}_k$ is the kth medoid of the tail time series $\boldsymbol{tts}$, $dist(\cdot)$ is the distance between any $\boldsymbol{tts_i}$ and $\boldsymbol{m}_k$ calculated by (3.7). Partition around medoids (PAM) algorithm [93] and k-means++ [94] initialization are utilized for the classification and Scalable Time series Anytime Matrix Profile (STAMP) algorithm [90] are utilized to calculate $\mathcal{mp}_{\boldsymbol{ABBA}}$ efficiently.

$$\min \sum_{k=1}^{K} \sum_{i=1}^{|\boldsymbol{tts}|} dist(\boldsymbol{tts_i}, \boldsymbol{m}_k) \tag{3.7}$$

3.2.4 Classification Results

Figure 3.5 displays the classification result with $RMPS$ similarity with subsequence length $L = 80$, and $K = 7$.

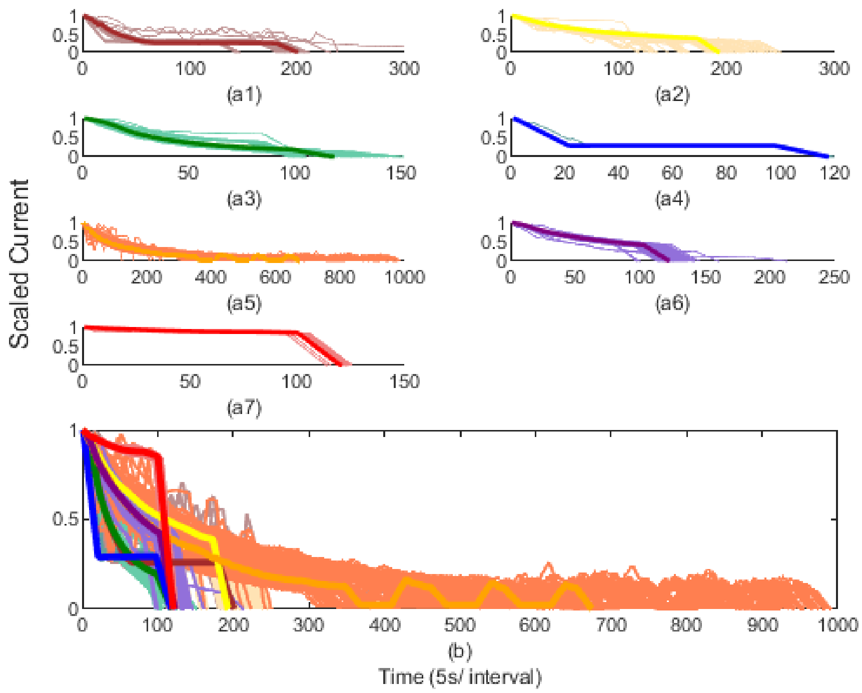


Figure 3.5: Classification via $RMPS$. (a1) – (a7) are 7 clusters with the medoids are highlighted; (b) is the overview of the entire tails.

3.2.5 Classification Evaluation

(I) Internal Indices

Without any information on the battery types in the ACN-Data, it is impossible to use the ground truth of the external index to evaluate the accuracy of the classification. Hence, this section utilizes three internal indices: Sum of the Squared Error (SSE), Silhouette Coefficient (SC), and Davies Bouldin (DB) [95] to validate the proposed $RMPS$ and comparing it with other state-of-art distance measure. The data samples utilized for analysis are shown in Figure 3.3.

This section adopts the shape-based distance (SBD) [96] in (3.8) as the error measure. SBD can differentiate two similar time series with distortions in amplitude and phase, and is in the range of $[0,2]$. WCD in (3.9) and BCD in (3.10) represent within and between cluster distances, respectively. SSE index evaluates the total dispersion within the same clusters, which is the squared distances between $\boldsymbol{tts}_i$ and its medoids $\boldsymbol{m}_k$. SC index in (3.12) evaluates the mean similarity of each point to points in its own cluster versus points in other clusters. a_i and b_i are the similarity and dissimilarity measure for any $\boldsymbol{tts}_i$ within and between the clusters, where a_i is the mean distance between $\boldsymbol{tts}_i$ and $\forall \boldsymbol{tts}_j$ within its cluster $\mathcal{M}_k$, and b_i is the smallest mean distance between $\boldsymbol{tts}_i$ and $\forall \boldsymbol{tts}_j$ in cluster $\mathcal{M}_{k'}$. DB in (3.13) evaluates the mean ratio of the similarity of each cluster with its most similar cluster. Generally, a lower SSE, DB and a higher SC represent a good classification of $\boldsymbol{tts}$.

$$SBD(\vec{x},\vec{y}) = 1 - \max_{w}\left(\frac{CC_w(\vec{x},\vec{y})}{\sqrt{R_0(\vec{x},\vec{x}) \cdot R_0(\vec{y},\vec{y})}}\right) \tag{3.8}$$

$$WCD_k^{SBD}(i,j) = dist_{SBD}\left(\boldsymbol{tts}_{k,i}, \boldsymbol{tts}_{k,j}\right), \forall i,j \in \mathcal{M}_k, i \neq j \tag{3.9}$$

$$BCD_k^{SBD}(i,k') = dist_{SBD}\left(\boldsymbol{tts}_{k,i}, \boldsymbol{tts}_{k',j}\right), \forall i \in \mathcal{M}_k, j \in \mathcal{M}_{k'}, k \neq k' \tag{3.10}$$

$$SSE = \sum_{k=1}^{K} \sum_{i=1}^{|\mathcal{M}_k|} WCD_k(i, m_k)^2 \tag{3.11}$$

$$SC = \frac{1}{K} \sum_{i=1}^{K} \bar{s}_i \, , s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{3.12}$$

$$a_i = \overline{WCD}_k(i, j), \;\; b_i = \min_{k\prime \neq k} \overline{BCD}_k(i, k')$$

$$DB = \frac{1}{K} \sum_{i=1}^{K} \delta_k \, , \delta_k = \max_{k' \neq k} \frac{\overline{WCD}_k(i, m_k) + \overline{WCD}_{k\prime}(i, m_{k\prime})}{BCD_k(m_k, m_{k'})} \tag{3.13}$$

For the $RMPS$ distance measure, two parameters needed to be tuned: the subsequence length L and the number of clusters K. Figure 3.6 demonstrates the selection of a different number of clusters and lengths of subsequence. By fixing the subsequence length $\mathrm{L} = 80$, the SSE decreases when K increases, but the SC reaches to the maximum value at $K = 7$ then decreases. A larger number of clusters makes b_i of each cluster very small, resulting to a small SC. The optimum $K = 7$ was chosen for different lengths of subsequence evaluation, as shown on the left side of Figure 3.6. A short subsequence tends to make each time series very similar; thus, the classification performance is poor. The adequate L should include the unique shape features of each tail. In our evaluation, L ranging in [70, 85] can achieve comparable classification. From the above discussion, $K = 7$ and $L = 80$ are chosen for the rest of the chapter for classification evaluation.

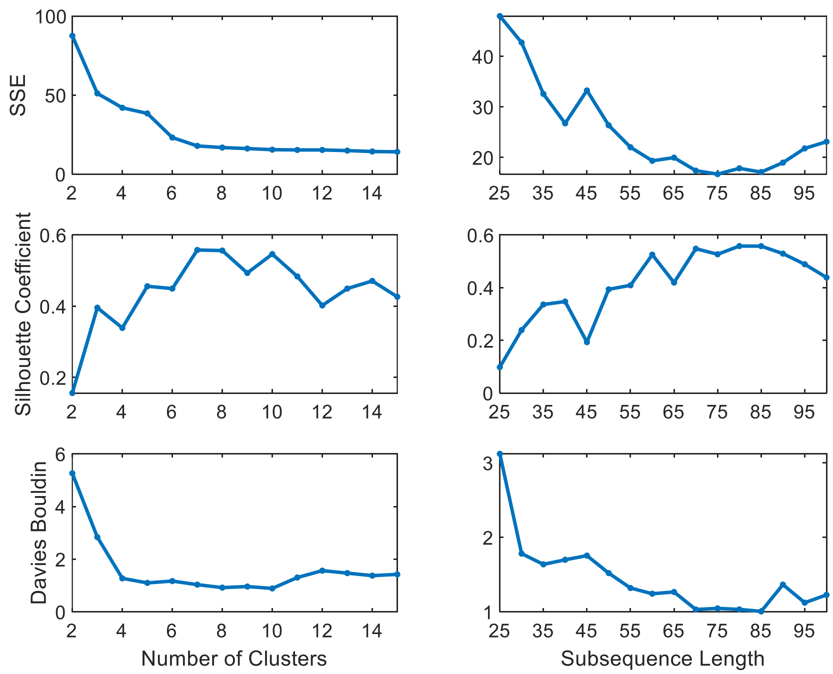


Figure 3.6: Evaluation the classification of $RMPS$ with different choices of left) different cluster number K (fix $L = 80$), and right) different lengths of subsequence L (fix $K = 7$).

(II) Comparison between Different Distance Measure

This section compares the classification performance of $RMPS$ with other distance measure methods: Euclidean distance (padding shorter tails with zeros at the end), Dynamic Time Warping (DTW), Modified Euclidean Distance (MED) defined in [76]. The evaluation of different distance measures with variable cluster numbers is shown in Figure 3.7. For $RMPS$, the choice of K is critical where a larger K makes SC/DB decreases/ increases. The three indices converge to their local optimal when $K = 7$. For ED, with the increment of the number of clusters, three indices converge at a higher $K = 10$. For MED, the SSE and SC indices are the lowest/highest with a smaller K when compared to other distance measures. When increasing the number of K, the three indices performance decrease. The optimal K for MED can be chosen at $K = 4$. Similar to MED, the three indices of DTW can achieve the optimal at small $K = 3$. The three indices values for the corresponding optimal K_{opt} are shown in Table 3.1.

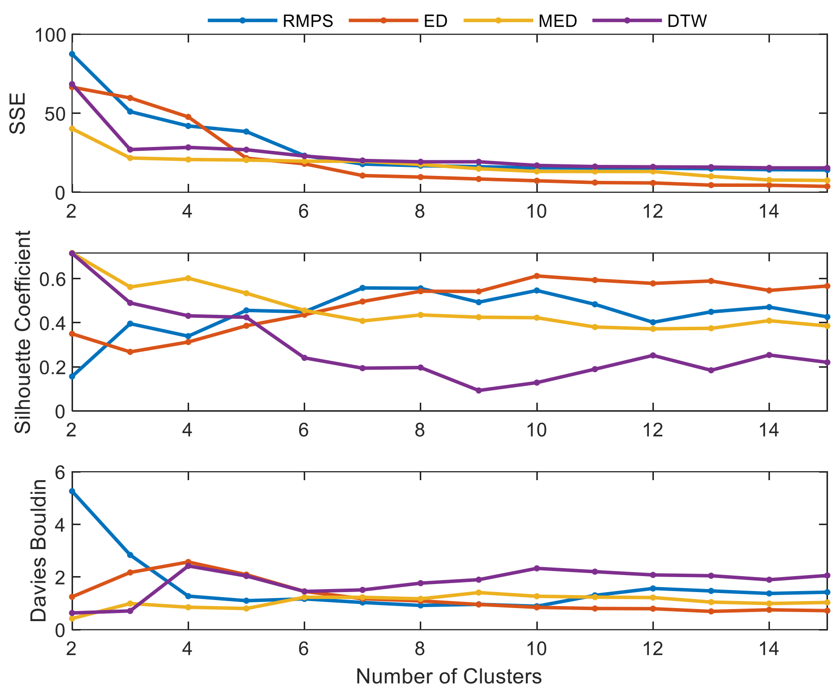


Figure 3.7: Classification performance measured by SBD compariosn between different distance measure: upper) SSE index; middle) SC index; bottom) DB index.

Besides considering the SBD error measure as the shape-wise similarity, we introduce the Eculideace Sequential distance ($EDSeq$) in (3.14). The $EDSeq$ measures the global "structure level" similarity by comparing each subsequence of $\boldsymbol{tts}_i$ and $\boldsymbol{tts}_j$.

$$EDSeq(\vec{x}, \vec{y}) = \frac{1}{L} \sum_{L} mp_{\boldsymbol{ABBA}}\left(\boldsymbol{tts}_i, \boldsymbol{tts}_j\right), L = \left|\boldsymbol{tts}_j\right| \leq \left|\boldsymbol{tts}_i\right| \tag{3.14}$$

The ED takes a value between $[0, +\infty)$. Here, we use the Ball-Hall index, which is the mean SSE of each cluster, $\frac{1}{K}\sum_{i=1}^{K}\overline{WCD}_k^{EDSeq}(i, m_k)^2$. Instead of SSE to better visually compare the difference of the cluster dispersion between different distance measures. The evaluation of the three indices is shown in Figure 3.8. Similar to the SBD error measure, the three indices of $RMPS$, ED, MED can converge and achieve the optimal when $K_{RMPdist} = 7$, $K_{ED} = 10$, $K_{MED} = 4$, respectively. DTW, on the other hand, can achieve the optimal of three indices at $K_{DTW} = 6$. The three indices values for the optimal K_{opt} are shown in Table 3.1.

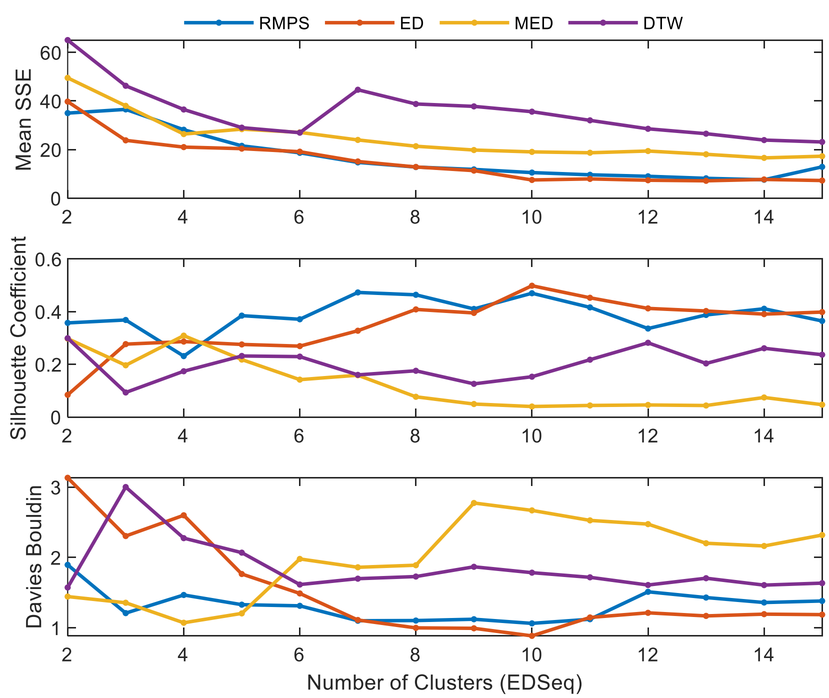


Figure 3.8: Classification performance measured by $EDSeq$ compariosn between different distance measure: upper) SSE index; middle) SC index; bottom) DB index.

Error Measure		$RMPS$	ED	MED	DTW
SBD	SSE	17.85	7.27	20.73	27.04
	SC	0.56	0.61	0.6	0.49
	DB	1.03	0.84	0.85	0.71
	K_{opt}	7	10	4	3
$EDSeq$	mean SSE	14.71	7.48	26.37	26.93
	SC	0.47	0.49	0.19	0.23
	DB	1.10	0.88	1.07	1.61
	K_{opt}	7	10	4	6

Table 3.1: Evaluation results for different distance measures.

(III) Correctly Clustered Ratio

For the same *userID*, the percentage of the user owing two EVs with distinct tails is low [76]. Hence, the majority of the tails from the same *userID* should be within the same cluster, and the number of tails from the same user consistently classified into one cluster can be utilized to evaluate the classification performance, defined as the Correctly Clustered Ratio (CCR). However, tails from the same *userID* equipped with multiple unique tails may happen in the database, such as an example shown in Figure 3.9 for *userID*_124.

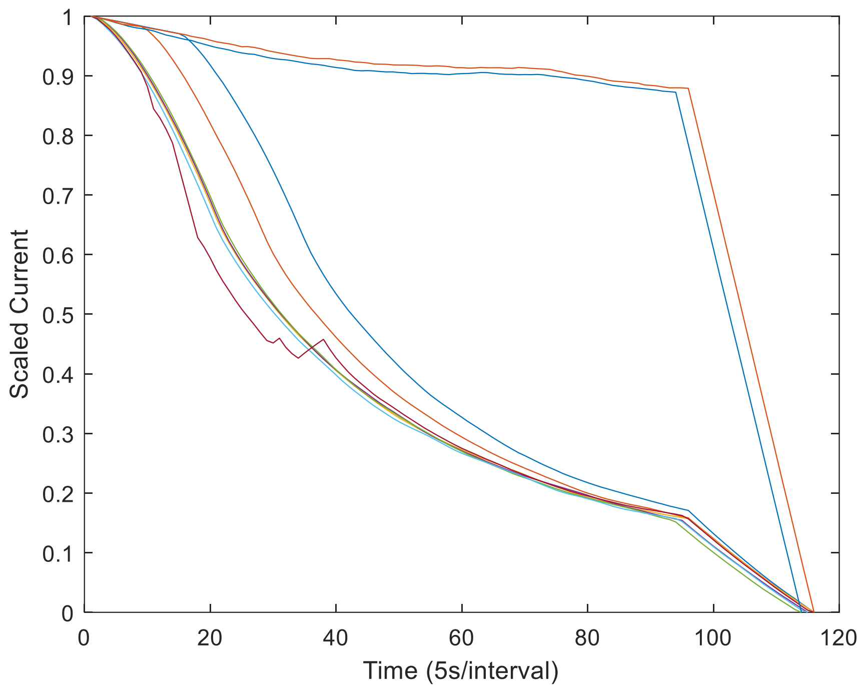


Figure 3.9: Tails from *userID*_124.

To evaluate CCR with the consideration of multiple distinct tails for the same *userID*, the other correctly clustered tails $\boldsymbol{tts}_{CC}$ is defined in (3.15) as follows: for $user_i$, the tails are $\boldsymbol{tts}^{user_i}$, and those tails are clustered into $\mathcal{M}_1, \ldots, \mathcal{M}_k$: where $|\mathcal{M}_1| \geq \cdots \geq |\mathcal{M}_k|, k \leq K$. The mean time series of $\mathcal{M}_1$ is $\boldsymbol{m}_{\mathcal{M}_1}$, and $\varepsilon \coloneqq \max WCD^{ED}_{i,j\in\mathcal{M}_1}(i,j)$.

$$\boldsymbol{tts}_{CC} \coloneqq \{\boldsymbol{tts}_i | i \in \mathcal{M}_{\boldsymbol{j\neq 1}}, ED\left(i, \boldsymbol{m}_{\mathcal{M}_1}\right) > \varepsilon\} \tag{3.15}$$

$$CCR = \frac{\sum_i |\mathcal{M}_1^{user_i}| + |\boldsymbol{tts}_{CC}^{user_i}|}{\sum_i |\boldsymbol{tts}_{user_i}|} \tag{3.16}$$

The results of the CCR with different clusters are featured in Figure 3.10. For $RMPS$ and ED, a higher CCR can be found at $K = 7$, while for MED and DTW, the CCR reduces when K reaches to 4. The above discussion implies that statically, the $RMPS$ and

ED perform comparatively at a higher $K \in [5,8]$, and outperform over the MED and DTW.

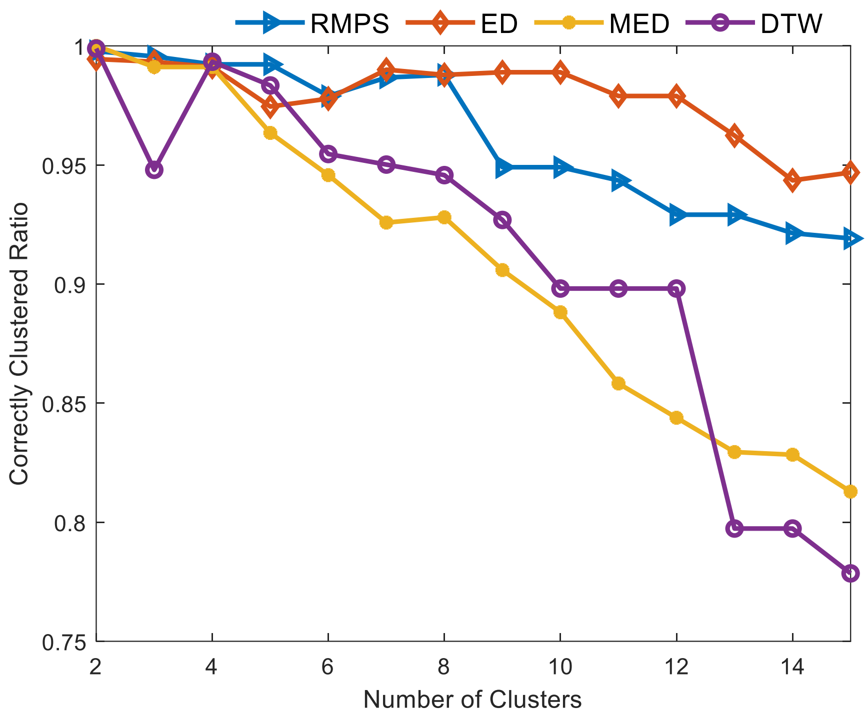


Figure 3.10: Correctly clustered ratio.

Meanwhile, the tail identification for individual users is illustrated in Algorithm 3.2.

Algorithm 3.2: Tail identification	
Input: learned tail templates $\boldsymbol{m}_K$, $\boldsymbol{cts}^{user_i}$ Output: tail templates $\boldsymbol{m}_{user_i}$ for $user_i$	
1	Extract tails of $\boldsymbol{cts}^{user_i}$ via Algorithm 3.1, and retrieve $\boldsymbol{tts}^{user_i}$.
2	Compute RMPS between $\boldsymbol{tts}^{user_i}$ and $\boldsymbol{m}_K$, and find it corresponding clusters $\mathcal{M}_1, \dots, \mathcal{M}_k$, and add $\boldsymbol{m}_{\mathcal{M}_1}$ to $\boldsymbol{m}_{user_i}$
3	Retrieve $\boldsymbol{tts}_{CC}$ from (3.15) and the related $\boldsymbol{m}_{\mathcal{M}_j}$, and add $\boldsymbol{m}_{\mathcal{M}_j}$ to $\boldsymbol{m}_{user_i}$
4	**return** $\boldsymbol{m}_{user_i}$

(IV) Classification Performance from Different Distance Measures

In this section, the classification results from different distance measures are plotted from Figure 3.11 - Figure 3.13. Each graph displays the tails within the clusters with the corresponding optimal number of clusters illustrated in Table 3.1. It is found that the longer tails with up-downs at the end of time series are consistently classified into one cluster by $RMPS$ in Figure 3.5, regardless of the tail length, up-downs frequency and duration of such up-downs. While for ED and other distance measures, those tails are consistently clustered into more than two clusters. Due to the calculation of subsequence length between time series, some short tails with up-down noise are clustered in Figure 3.5 (a2), resulting in incorrectly clustered tails. In Figure 3.11, although tails in each cluster are close to each other, the medoids of clusters are also close to each other, such as medoids in (a5) and (a10), (a4) and (a8). DTW and MED in Figure 3.12 - Figure 3.13 can only cluster minor features with their optimal K. Overall, the tail medoids by $RMPS$ in Figure 3.5 are more representative of each cluster comparing to other distance measures with fewer cluster numbers.

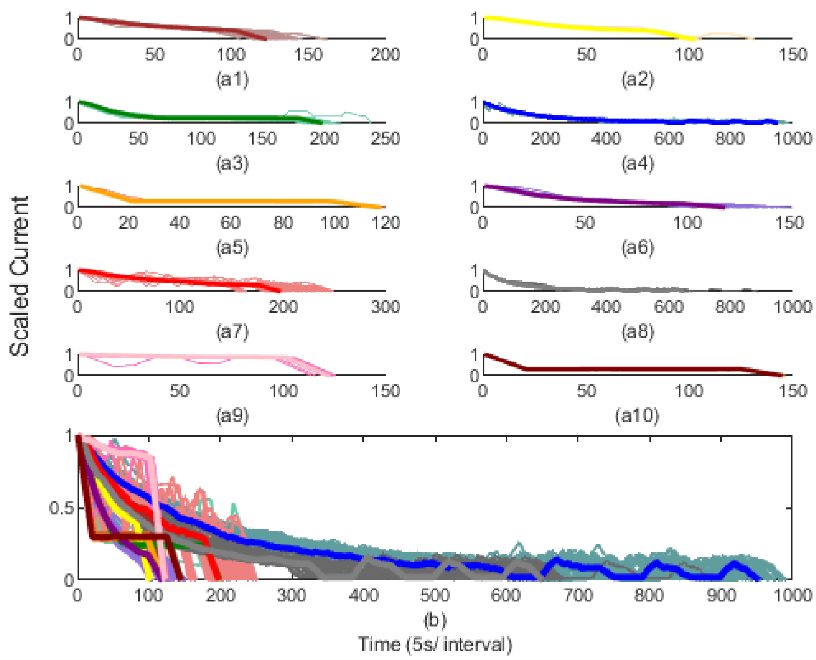


Figure 3.11: Classification via ED. (a1) – (a10) are 10 clusters with the medoids are highlighted; (b) is the overview of the entire tails.

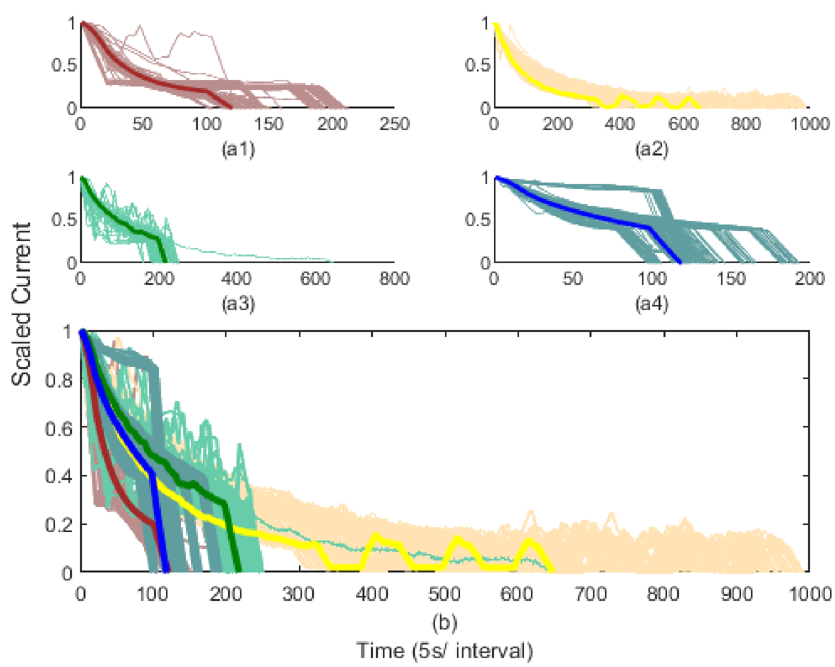


Figure 3.12: Classification via DTW. (a1) – (4) are 4 clusters with the medoids are highlighted; (b) is the overview of the entire tails.

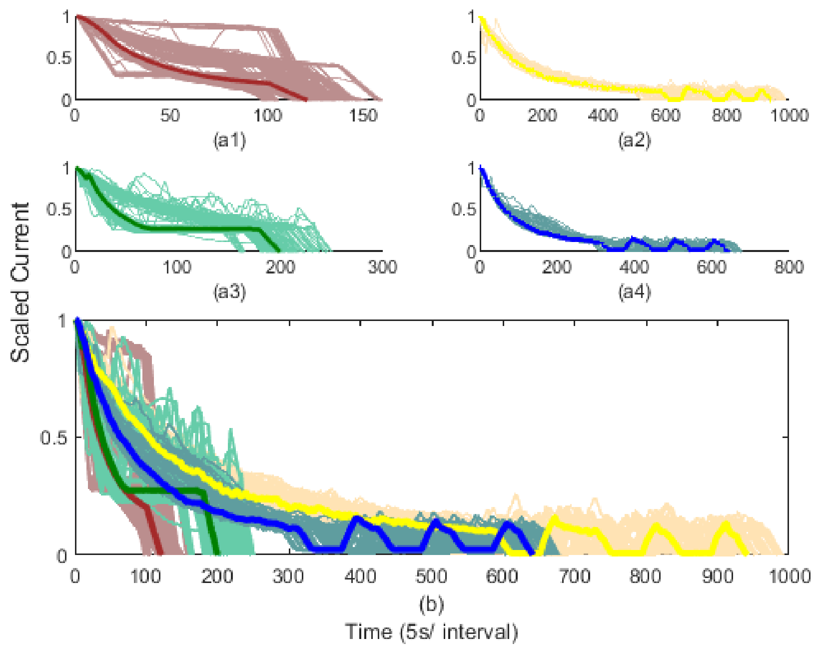


Figure 3.13: Classification via MED. (a1) – (a4) are 4 clusters with the medoids are highlighted; (b) is the overview of the entire tails.

3.2.6 Applications

(I) Future EV Charging Demand Projection

In order to demonstrate the potential EV charging demand difference with consideration of charging tails, this section utilizes the California future EV number projection of 350,000 at present, 1.5 million in 2025, and 5 million in 2030 [21]. For this case study, the complete charging sessions information of the Caltech site from ACN-data for September 2018 – September 2019 is utilized to formulate the sessions distribution. Each charging session is represented by four parameters: arrival time (a_i), parking duration (d_i), charging energy (e_i), and the tail representative ($\boldsymbol{m}_i$). Gaussian Mixture Model (GMM) is used to fit the distribution of $\{a_i, d_i, e_i\}$ with cluster numbers k and the cluster's covariance structure Σ: $\text{GMM} \sim \mathcal{N}(\mu_k, \Sigma_k)$. The GMM is tuned by Expectation-Maximization (EM) algorithm with exhausting all k from 1 – 30 and Σ with diagonal and full structures [97]. The results of different k are shown in Figure 3.14. The optimal GMM is chosen as $k = 24$ and full-unshared covariance structure. $\boldsymbol{m}_i$ is derived from (3.7) with the marginal probability of $\gamma \cdot \frac{|\mathcal{M}_i|}{|\boldsymbol{tts}|}$, where γ is the probability of whether this charging session enters the tail stage.

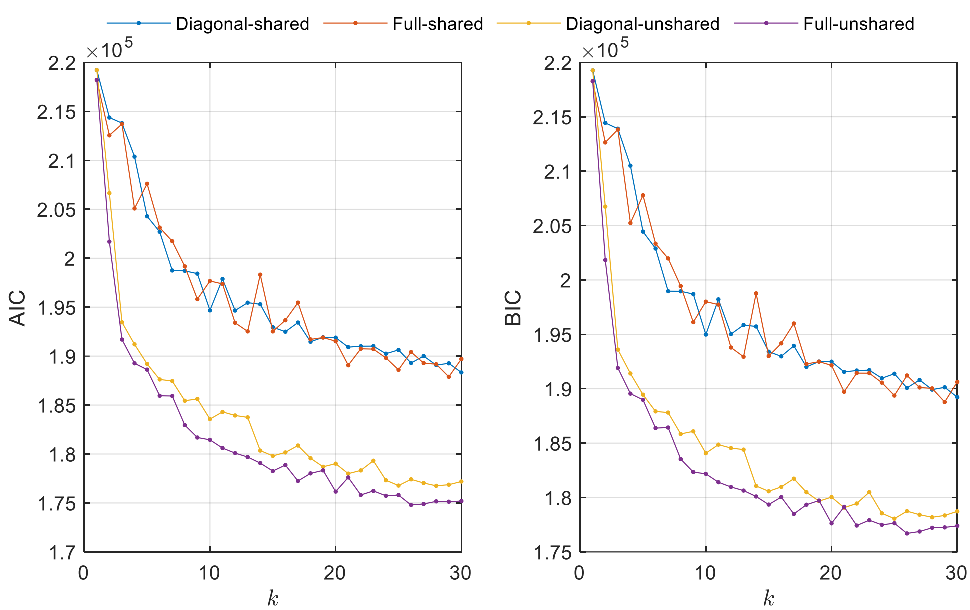


Figure 3.14: Fit GMM by Expectation-Maximization algorithm.

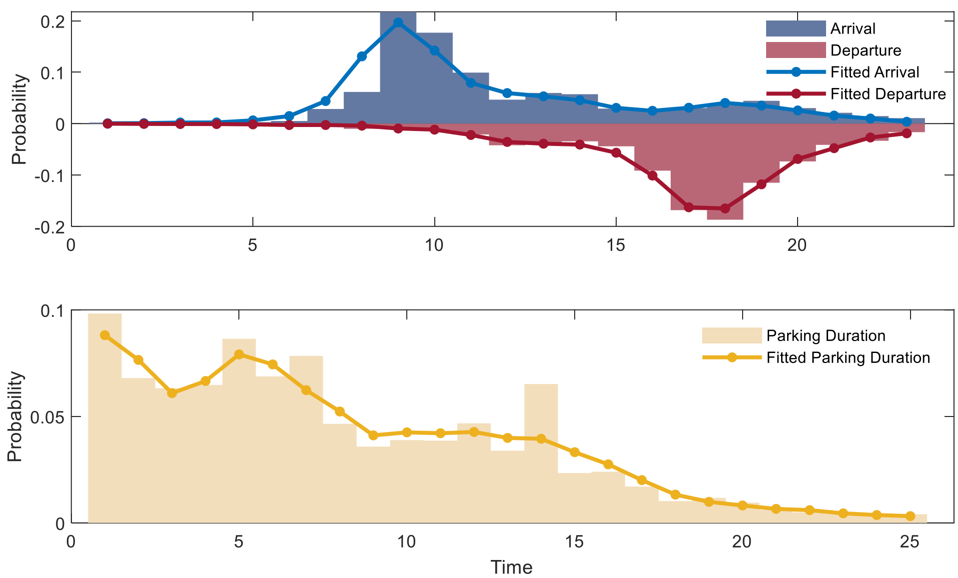


Figure 3.15: GMM fitted distributions for ACN-Data: upper) arrival and departure time; and bottom) parking duration.

The future EV demand projection and the difference in considering the charging tails are shown in Figure 3.16. The learned GMM model first populates the same number of charging sessions $\boldsymbol{cs_0}$ as the future population of EVs. The charging demand of those sessions is combined as the 'Unknown Tails'. K tail representatives with their marginal probability then added to $\boldsymbol{cs_0}$ to acquire the updated charging demand shown as 'Known Tails'. The upper and lower bounds of 'Known Tails' represent the probability of charging sessions containing the tail stage, where $\gamma = 0.9$ and 0.1 are chosen for the upper and lower bounds, respectively. With the more significant users who choose to charge their vehicle full at the charging station, an average of 5% of the charging demand decreasing can be found for the future charging demand prediction when considering tail features.

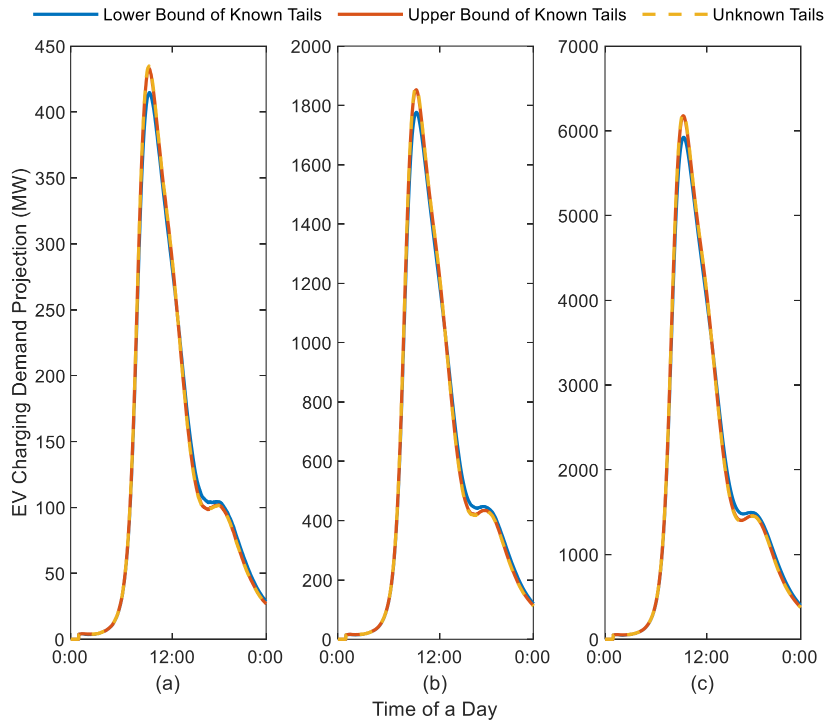


Figure 3.16: Projection of the future EV charging demand with different population of EVs: a) 350,000, b) 1.5 million and c) 5 million.

(II) EV Battery Capacity Estimation

For users' requested energy, the prediction accuracy of the charging duration for such a session can be increased by the learned tail features. For any session without user-provided information, the charging duration can also be online updated whenever the tail stage has been identified. The tail stage identification can be modeled as a binary decision problem [76] whenever the ED between the $\boldsymbol{tts}_i$. and any $\boldsymbol{m}_{k\in K}$ are small enough. Thus, the real-time energy scheduling of EVs can be more efficient with the consideration of charging duration and tail features. Another potential application is to quantify the battery capacity distribution within each cluster. Figure 3.17 quantifies the battery capacity for all 903 tails, where the battery capacity is chosen as the largest recorded energy within the same cluster. Each box represents the group of the battery capacity sets in the same cluster with the approximate same charging rate. The outliers of the lower bounds of blue boxes for $k = 1$ and 6 are for a lower charging current at about 10A. The battery capacity distribution in most clusters is in a small range. The battery capacity estimation provides the charging infrastructure for quickly learning and serving the plugged-in vehicles.

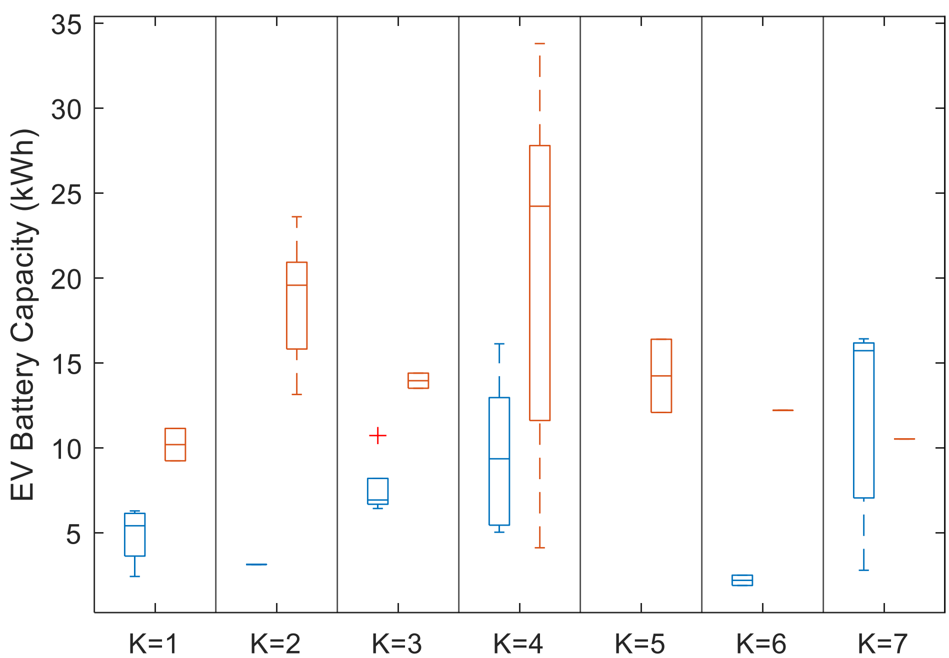


Figure 3.17: Boxplot for battery energy capacity distribution for each cluster.

The blue box is for maximum charging current less than 18 A (approximately 15 – 17 A), and the orange box is for current higher than 18 A (approximately 29 – 32 A).

3.3 EV Charging Management

In this section, a centralized EV charging schedule algorithm considering the tail features is proposed. Different case studies are carried out to showcase the importance of considering charging tails in conjunction with smart charging algorithms.

The raw EV charging data is noise caused by three reasons: 1) the pilot signal inference, 2) some recorded energy not aligned with the time-series data, and 3) multiple tails for some userID. To align time-series data with energy and tails more precisely, a charging event $CE(Request_o, Park_o)$ with requested charging energy $Request_o$ and parking duration $Park_o$ are reconstructed in Algorithm 3.3 as follows: in line 1, the constant charging stage is reconstructed by maintaining the stable charging current around the maximum value of the charging current. t_0 is the first time index when BAU reaches mc. Line 2 reconstructs the tail stage by the tail template. In such a case, only one tail template for each uID is considered. The tail template is located by first clustering all tails from uID to the learned templates $\boldsymbol{m}_{\mathcal{M}} \in \boldsymbol{m}_K$. Then $\boldsymbol{m}_{\mathcal{M}_1}$ is chosen, which includes the majority of tails as the only tail template. Line 3 revises the charging energy and parking duration for the reconstructed $\boldsymbol{tts}$.

Algorithm 3.3: Reconstruct charging events	
Input: $CE(Request_o, Park_o)$, $\boldsymbol{tts}$ from userID uID	
Output: $CE(Request, Park)$, $\boldsymbol{tts}$	
1	Replace piloted points Γ from (3.2) – (3.3) with the average current around maximum charging current $mc = \overline{\boldsymbol{tts}}(\|\boldsymbol{tts} - \max \boldsymbol{tts}\| \leq \epsilon)$, $\boldsymbol{tts}(\Gamma) = mc$, $\Gamma \geq t_0$.
2	For uID, find the tail template $\boldsymbol{m}_{\mathcal{M}_1}$ from the Tail Identification in Algorithm 3.2, and update $\boldsymbol{tts}(\tau) = \boldsymbol{m}_{\mathcal{M}_1} \cdot mc$, where $\tau = \begin{cases} \text{points for tail stage} \\ 0, \text{no tail exists} \end{cases}$.
3	Revise $Request = \boldsymbol{tts} \cdot 0.208 \cdot \Delta T$, $Park = \max(Park_o, \|\boldsymbol{tts}\|)$.
4	**return** $CE(Request, Park)$, $\boldsymbol{tts}$.

Figure 3.18 displays two reconstruction examples from different uIDs. $\epsilon = 2$ is utilized. The left graph shows an example of the reconstruction of the constant charging stage. The business as usual (BAU) is a noisy charging profile during the constant current stage. The reconstructed current remains a constant charging current 30.87 A. The right graph shows an example of the reconstruction tail stage. The BAU profile is controlled by the pilot, which results in the CE entering the tail stage at a lower current around 8 A. To reconstruct the tail stage without the inference of pilot signal, the tail template $\boldsymbol{m}_{k=5}$ is identified for this user, $\boldsymbol{tts}(\tau) = \boldsymbol{m}_{k=5} \cdot mc$. The constant charging stage is maintained for the period of $\left[t_0 \quad t_0 + \frac{Request_o - \sum_\tau \boldsymbol{tts}(\tau)\cdot 0.208\cdot\Delta t - \sum_t \boldsymbol{tts}(1:t_0)}{mc\cdot 0.208\cdot\Delta t}\right]$.

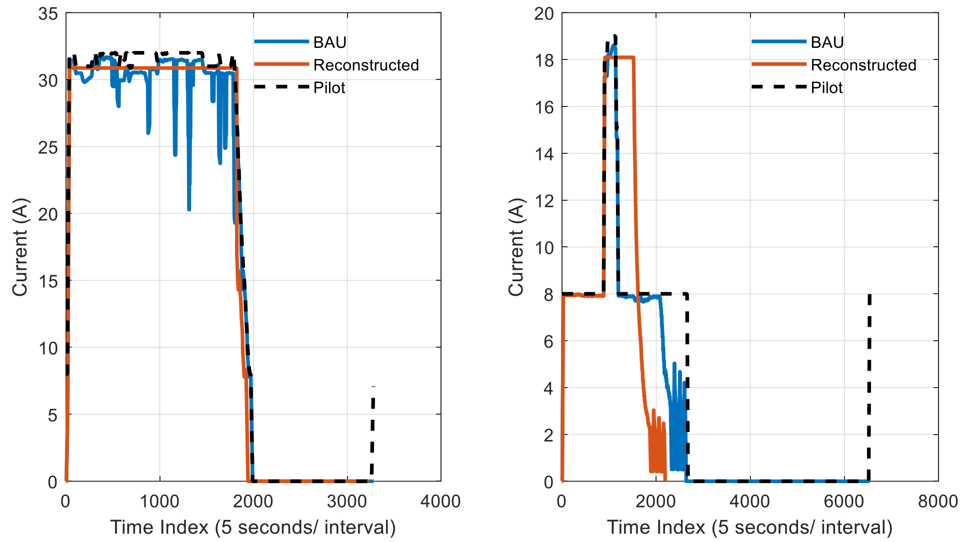


Figure 3.18: Reconstruction of charging events.

3.3.1 Centralized Control

For the EV smart charging applications, it is assumed that each user submits his/ her charging energy, parking time, arrival time, charging station 24-hour ahead of the time, and the EV central controller schedule the charging power of each EVSE for this day. The objectives of the central controller are minimizing the total charging (electricity) cost of the facility and the emission cost. The emission cost data is retrieved from the marginal emission cost in the Southern California Edison region for January 2018 [98], and the electricity cost data is from the SCE TOU-EV-8 winter rate schedule [99], as shown in Figure 3.19.

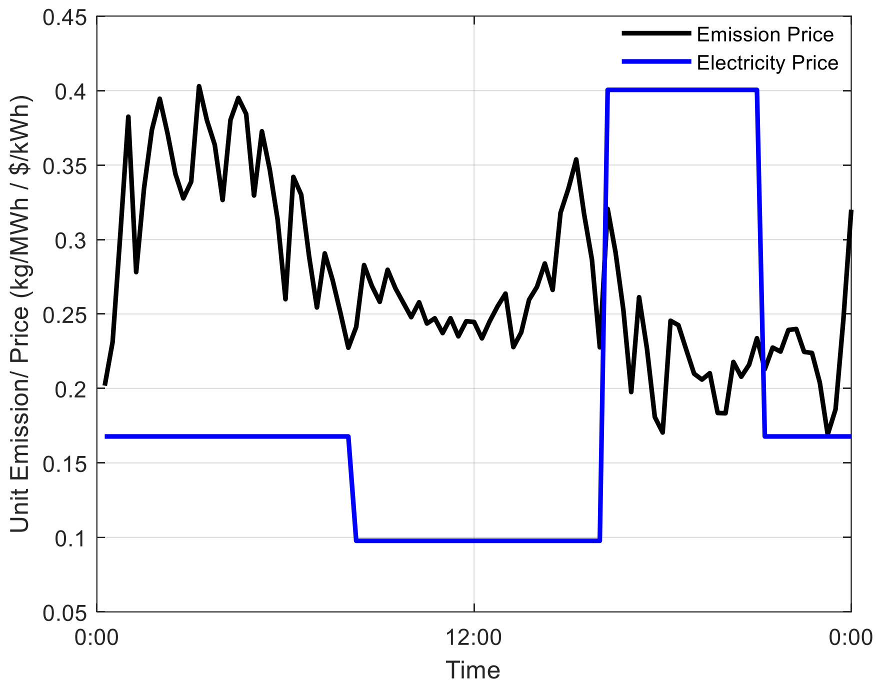


Figure 3.19: Unit marginal emission production and charging electricity cost.

(I) Control Objectives and Constraints

The objective function of the central controller is depicted in (3.30) with the constraints from (3.18) – (3.20). Each charging event can also be represented as $CE(t, m, n) = [Request_{m,n}, Park_{m,n}]$, where t is the time index $t_i \leq t \leq t_i + Park_{m,n}$ and $t \leq \frac{24}{\Delta t}$, t_i is the event initial time index, Δt is the unit control step duration; m is the m^{th} charging event for EVSE n; and n is the EVSE id, $n \leq N$. The total charging loads $\boldsymbol{p}_{EV}$ is equal to the sum of all scheduled charging power as shown in (3.18). For each individual charging event $CE(t, m, n)$, the sum of the scheduled charging energy should be equal to the request energy in (3.19), and the scheduled charging power during the parking duration should be no larger than the maximal charging power $cmax_{m,n}$.

$$\min_t (1-\omega)\cdot(price_{ele}\times \boldsymbol{p}_{EV}\cdot \Delta t+\gamma\cdot price_{dem}\cdot \max_t \boldsymbol{p}_{EV})+\omega \cdot price_{emi}\times \boldsymbol{p}_{EV}\cdot \Delta t \tag{3.17}$$

$$\boldsymbol{p}_{EV}=\sum_n\sum_m\sum_t \boldsymbol{x}(t,m,n) \tag{3.18}$$

$$\sum_t \boldsymbol{x}(t,m,n)\cdot \Delta t = Request_{m,n}, \forall m,n \tag{3.19}$$

$$0\le x(t,m,n)\le cmax_{m,n}, , t_i\le t\le t_i+Park_{m,n}, , \forall m,n \tag{3.20}$$

$\boldsymbol{x}(t,m,n)$: the optimized charging power at time index t for m^{th} charging events of EVSE n;

ω: the weight balancing the importance of minimizing electricity cost and emission cost;

$\boldsymbol{p}_{EV}$: the total charging power vector, $\boldsymbol{p}_{EV}=[\boldsymbol{p}_{EV}(t)]^T$;

$price$: price matrices for energy, where $price_{ele}$ is the unit electricity cost ($/ kWh), $price_{emi}$ is the unit emission production (kg/ MWh), and $price_{dem}$ is the unit demand cost ($/ kW).

γ: the coefficient of the importance of demand cost, $\gamma\le 1$.

(II) Simulation Results and Discussions

A. Different Choices of ω and γ

Figure 3.20 displays the relationship between different choices of ω, γ and peak to average ratio (PAR), total electricity cost, and the total emissions. Charging events from the caltech site on 5/8/2019 are utilized for simulation. ω increases, the electricity cost increases while the emission decreases. When $\omega = 0.5$, equal weights are given to the electricity and emission.

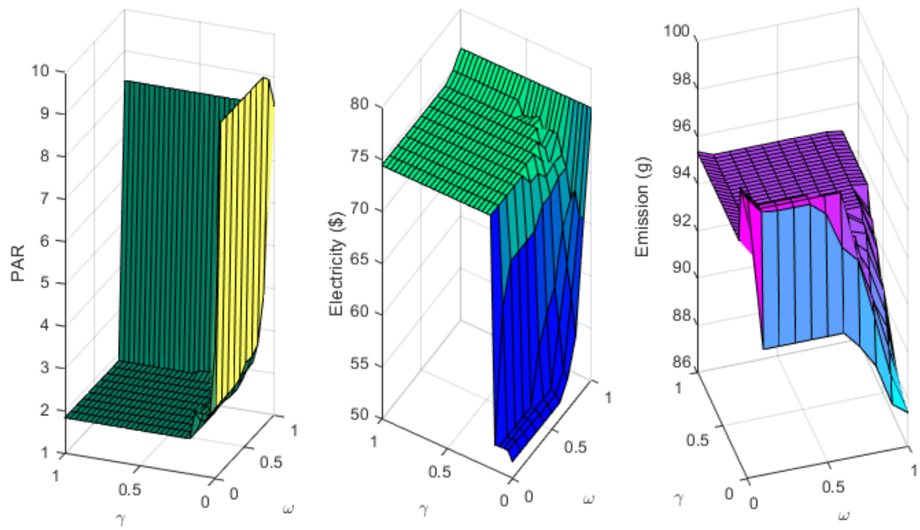


Figure 3.20: 3-D plots for different choices of ω and γ: left) PAR, middle) electricity, and right) emission.

In Figure 3.21, a higher γ evitably reduces the PAR. However, the electricity cost and emission increase. Meanwhile, the increment of emission is relatively more minor when compared to electricity cost.

Figure 3.22 displays different optimized charging power profiles with variable γ. When $\gamma \geq 0.2$, the optimized charging power profiles remain the same. The dropout during 4 PM – 5 PM when $\gamma < 0.2$ caused by the increased electricity and emission unit price/generation. It is noted that when γ is high, the charging electricity cost is higher in return for a more stable charging profile with fewer charging stops.

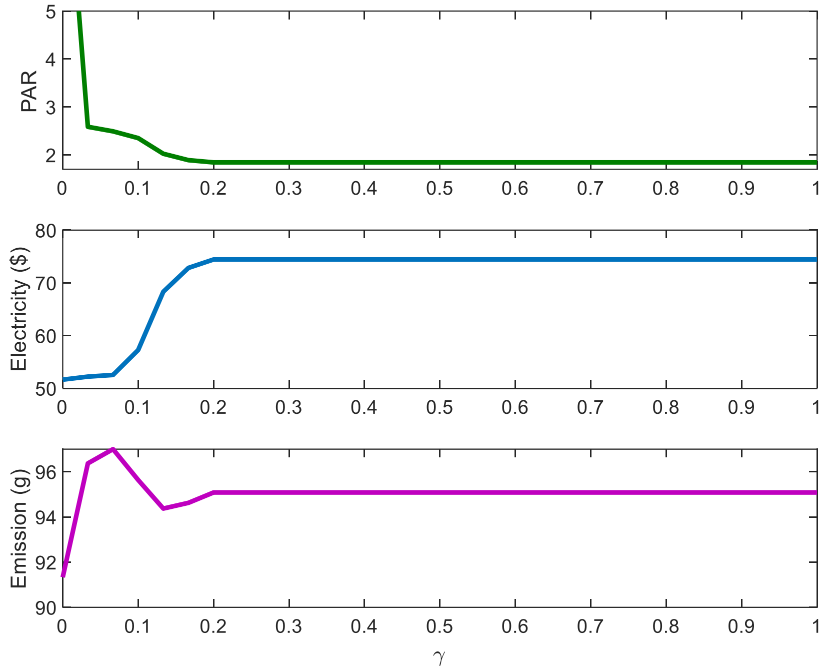


Figure 3.21: The relationship between γ and PAR, electricity cost and emission, respectively.

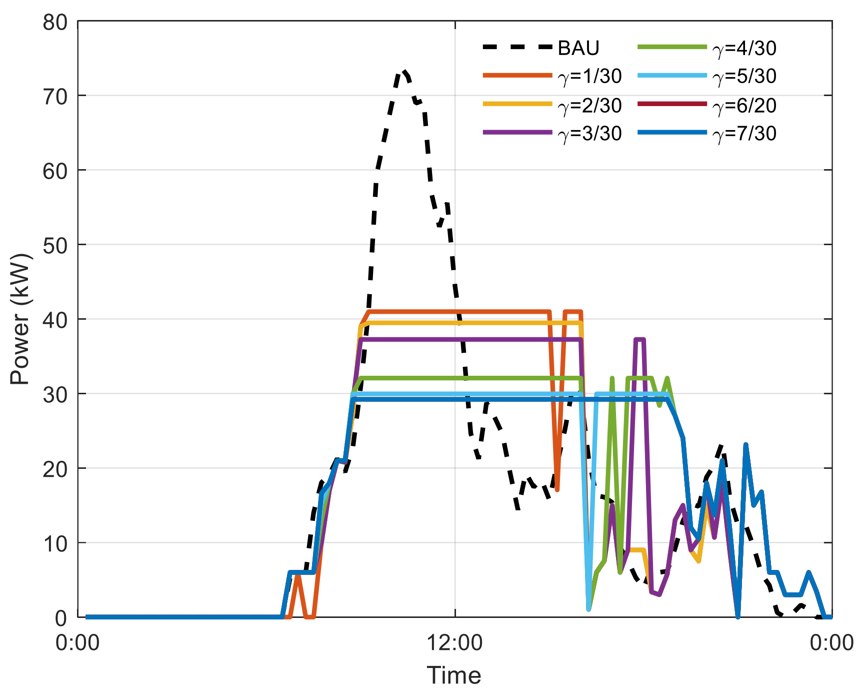


Figure 3.22: Charging power optimization for different choices of γ when $\omega = 0.5$.

B. Considering Tail Features

- Flexibility Analysis (FA)

The flexibility analysis aims at determining the difference in the EV charging power between charging schedules considering tails or not. The flexible energy refers to energy within the tail stages. The flexible energy is defined as (3.21).

$$fe = \sum_{t} (\boldsymbol{p}_{nt} - \boldsymbol{p}_{wt}) \cdot \Delta t \tag{3.21}$$

$\boldsymbol{p}_{nt}$ is the optimized $\boldsymbol{p}_{EV}$ without considering tails, and $\boldsymbol{p}_{wt}$ is the optimized $\boldsymbol{p}_{EV}$ considering tails.

The constraints of (3.19) – (3.20) can be further revised as (3.30) – (3.23). Due to inefficient charging during the tail stage, by constraining the scheduled charging energy no less than the charging energy minus the tail energy and delaying the tail stage to the end of the entire parking period for $CE(t,m,n)$, less priority is given to $\mathrm{EVSE_n}$ when the charging event enters the tail stage.

$$\sum_{t} \boldsymbol{x}(t,m,n) \cdot \Delta t \geq Request_{m,n} - \eta \cdot \sum_{\tau_k} \boldsymbol{tail}_{m,n,k} \cdot \Delta t\,, \forall m,n \tag{3.22}$$

$$\begin{gathered} 0 \leq x(t,m,n) \leq cmax_{m,n}, t < \Delta\tau, \forall m,n \\ 0 \leq x(t,m,n) \leq \ \boldsymbol{tail}_{m,n,k}, t \geq \Delta\tau, \eta = 1 \\ \Delta\tau = Park_{m,n} - \tau_k \end{gathered} \tag{3.23}$$

η: the indicator of whether $CE(t, m, n)$ contains the tail stage, 1 – tail stage exists, and 0 – no tail;

$\boldsymbol{tail}_{m,n,k}$: tail charging power profile from the learned tail template $\boldsymbol{m}_k$, where $\boldsymbol{tail}_{m,n,k}$ is the maximum charging power for every Δt steps in the tail stage formulated in (3.30);

τ_k: time duration for $\boldsymbol{m}_k$; $\Delta\tau$: time duration when the charging event not entering tail stage.

$$\boldsymbol{tail}_{m,n,k} = \max \boldsymbol{m}_k(\Delta\tau + j \cdot \Delta t : \Delta\tau + (j+1) \cdot \Delta t - 1) \cdot 0.208 \qquad (3.24)$$

Three different case studies are considered for FA: 1) actual EV charging events from the caltech site on 5/8/2019 with different unit control steps Δt, 2) generated/ projected EV charging events from Section 3.2.6 with a various number of charging events, and 3) most significant/longest charging tail in Figure 3.5 (a5).

Case study 1 analysis is shown in Figure 3.23, and the curve is the representation of fe difference between $\boldsymbol{p}_{wt|\Delta t}$ and $\boldsymbol{p}_{nt|\Delta t}$, where $\Delta t = [1,2,3,5,6,10,12,15]^T$ minutes. $\omega = 0.5$ and $\gamma = \frac{1}{15}$ are chosen for FA simulations to assign equal weights for each cost. A more frequent control step gives more flexible energy when considering tails. Most duration of tails lies around 10 minutes, when $\Delta t > 10$, the scheduling influence from tails is negligible for most tails. Thus, the flexible energy reduces significantly when $\Delta t \geq 10$.

Figure 3.24 illustrates two simulations with $\Delta t = 1$ and $\Delta t = 15$ minutes, respectively. $\boldsymbol{p}_{nt}$ maintains the same for different unit control steps. However, $\boldsymbol{p}_{wt}$ can achieve a lower PAR with lower peak charging power when $\Delta t = 1$ when compared to $\Delta t = 15$. With a fast charging scheduling operation such as under real-time circumstances, the importance of considering the tail stage for charging events increases and evitably improves the flexible energy due to more precise scheduled charging power for the tail stage as depicted in (3.23).

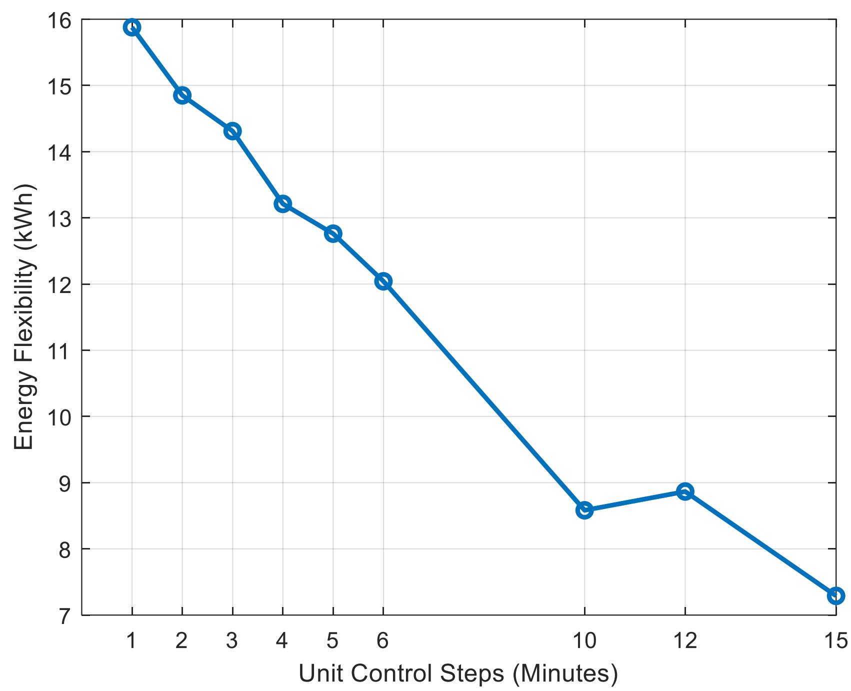


Figure 3.23: Case study 1 – the flexible energy difference between scheduling with and without considering tails.

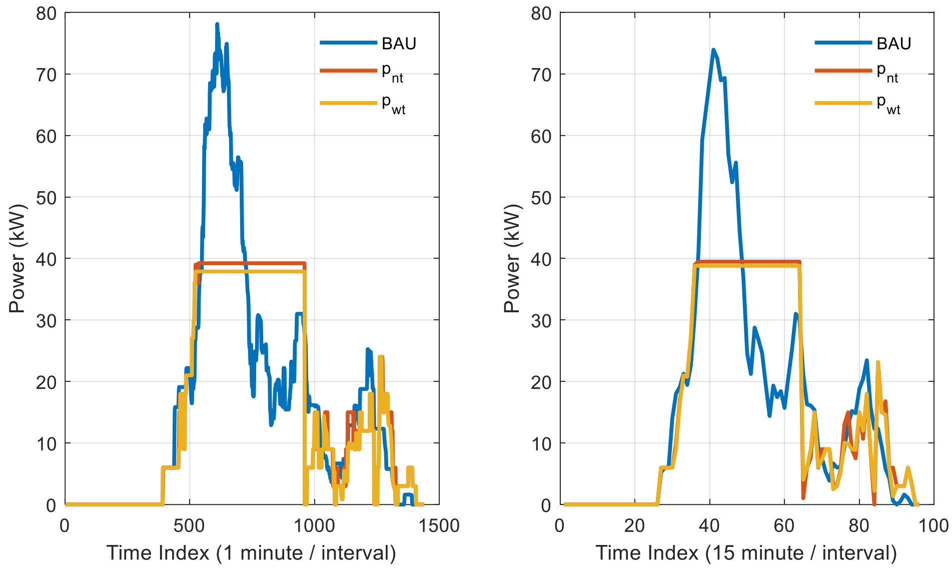


Figure 3.24: Case study 1 – scheduled charging power with different unit control step: left) $\Delta t = 1$, and right) $\Delta t = 15$.

For case study 2 analysis, the same GMM model illustrated in Section 3.2.6 (I) is utilized to generate more charging events. The probability of charging events, including the tail stage, is chosen as 0.9. It is assumed that enough EVSEs exist, and EVSEs are assigned in sequential order. For any CE_i, assuming the prior charging event is assigned to EVSE n_{i-1}: $CE_{i-1}(t, m_{i-1|n_{i-1}}, n_{i-1})$. If all EVSEs from 1 to n_{i-1} are occupied, $CE_i = CE_i(t_i, m_{i|n_i} = 1, n_i = n_{i-1} + 1)$; otherwise, $CE_i = CE_i(t, m_{i|n_i} = m_{i-1|j} + 1, n_i = j)$, where j is the available EVSE, $j \leq n_{i-1}$.

Figure 3.25 displays the fe simulation with various number of charging events, where the flexibility rate $fr := \frac{fe}{\sum_t \boldsymbol{p}_{nt}} \times 100\%$. In this simulation, $\Delta t = 3$, $\gamma = \frac{1}{15}$ and $\omega = 0.5$. When more events are added, it is evident that the fe increases due to the increased tail events. The fr lies in the range between 4% ~ 5%, which means that when tail characteristics are added to the EV smart charging operation, a 4% - 5% incremental energy flexibility can be preserved.

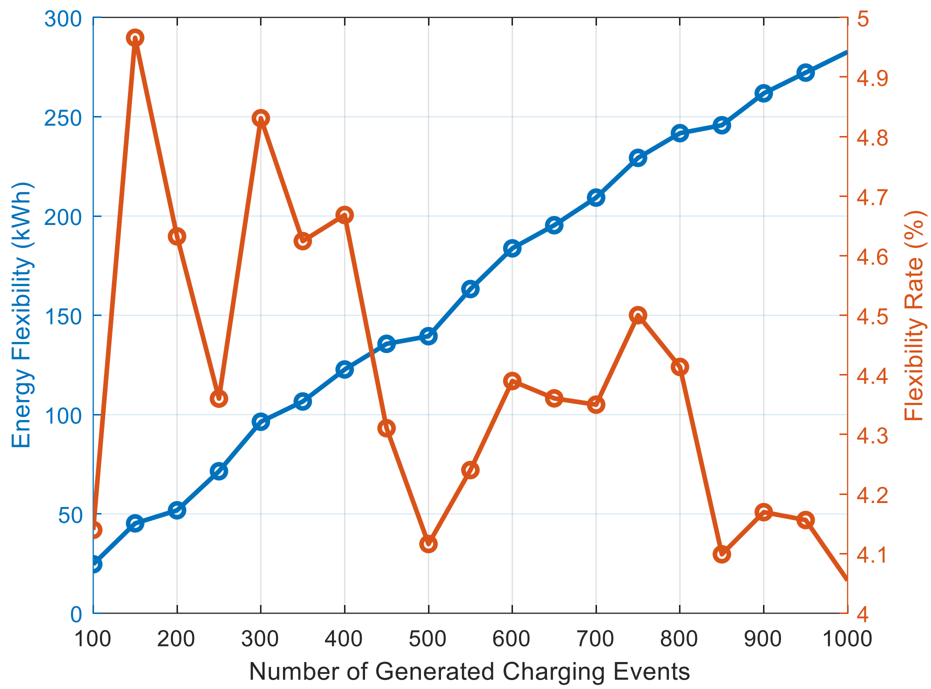


Figure 3.25: Case study 2 – the flexible energy and flexibility rate with a different number of charging events.

For case study 3 analysis, a particular assumption is made when vehicles are only exhibiting tails with the template set to be $\boldsymbol{m}_{K=5}$. This assumption is considered for the fleet charging scenario with the same type of EVs but different charging energy and parking duration. Similar to case study 2, $\Delta t = 3$, $\omega = 0.5$, and $\gamma = \frac{1}{15}$. It is anticipated that higher fe is available with a more significant number of charging events due to longer tail duration and energy. The fr can further increase over 10% and achieve 14.5% when the number is equal to 150.

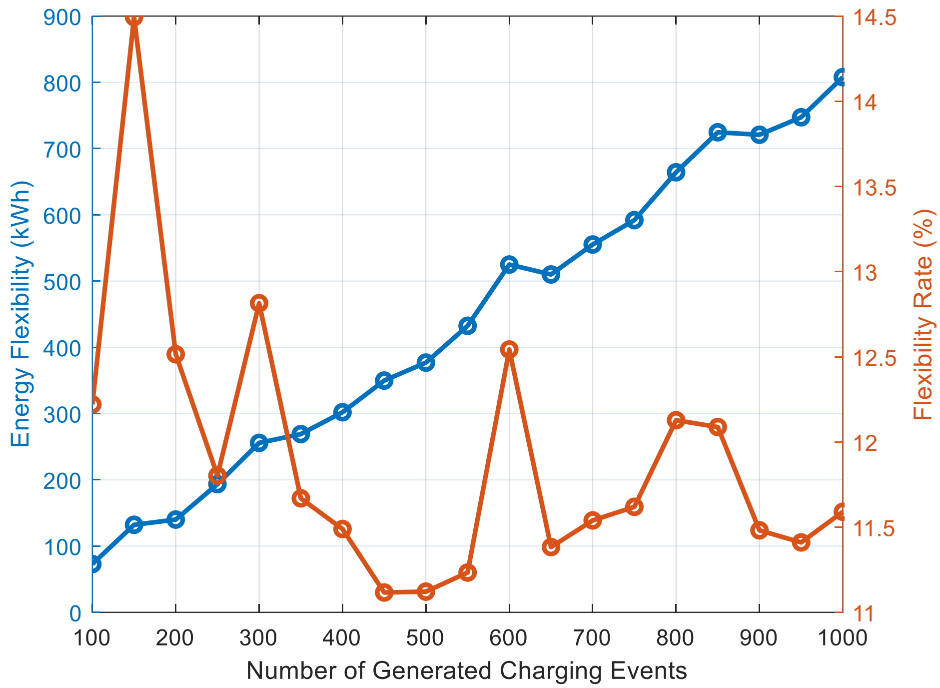


Figure 3.26: Case study 3 – the flexible energy and flexibility rate with a different number of charging events when considering only one tail.

- Satisfactory Analysis (SA)

The satisfactory analysis aims at determining whether the scheduled charging energy satisfies the user-requested energy. The satisfaction rate is defined as (3.30).

$$sr = \frac{\sum_t \boldsymbol{p}_{EV}}{Request_{m,n}} \tag{3.25}$$

For any $CE(t, m, n)$, although (3.19) is satisfied and constrained during optimization without considering tails, the actual charging power is reduced when $CE(t, m, n)$ enters the tail stage, which results in a non-fully charging event for $CE(t, m, n)$. In this section, the optimization function considering tails is formulated as (3.17) with constraints (3.18) - (3.19) and (3.23).

For any $CE(t, m, n)$, the calculated schedule vector is $x(t, m, n)$. For simplicity, m, n are omitted in the following description. The actual charging power $\overline{x}(t)$ is demonstrated in Algorithm 3.4: if $\eta = 0$, $\overline{x}(t) = x(t)$; otherwise, revise the charging power in the tail stage to limit the charging power less than $\boldsymbol{tail}_k$.

Algorithm 3.4: Actual scheduled charging power

Input: any $CE(t)$, $x(t)$, tail indictor η
Output: $\overline{x}(t)$

Retrieve the tail template $\boldsymbol{m}_k$ from Algorithm 3.2
if $\eta = 1$
  Find the first $\Delta\tau$ such that $\sum_{\Delta\tau} x(t) \cdot \Delta t > Request - e_{\boldsymbol{m}_k}$ and $x(\Delta\tau) > 0$.
  Find the remaining scheduled vector $x_r = x(\Delta\tau : t + Park) > 0$ and the allowed maximum charging power $\boldsymbol{tail}_k$ in (3.30)
  if $|x_r| > |\boldsymbol{tail}_k|$
    Calculate the cumulative sum energy of x_r and $\boldsymbol{tail}_k$, $\boldsymbol{e}_{x_r}$ and $\boldsymbol{e}_{tail}$
    for $t = 1 : |x_r|$ and $l = 1$
      if $\boldsymbol{e}_{x_r}(t) > \boldsymbol{e}_{tail}(l)$
        $l = l + 1$
        Jump out **for** when $l > |\boldsymbol{tail}_k|$
      end if
      $x_r(t) = \min(x_r(t), \boldsymbol{tail}_k(l))$
    end for
  else
    $x_r = \min(x_r, \boldsymbol{tail}_k(1 : |x_r|))$
  end if
  $\overline{x}(t) = [x(1 : \Delta\tau), x_r]$
else
  $\overline{x}(t) = x(t)$
end if
return $\overline{x}(t)$

Figure 3.27 displays the comparison between sr_{wt} and sr_{nt}, where $sr_{wt} = \frac{\overline{x}_{wt}}{Request}$ and $sr_{nt} = \frac{\overline{x}_{nt}}{Request}$. Charging events from 4/1/2019 – 5/10/2019 from the ACN-Data are utilized to run the simulations. The scheduling optimization only runs for days when the number of charging events for this day is larger than 30. There are a total of 1152 charging events, and most sr lies near 1. There are more charging events with $sr_{wt} = 1$ than $sr_{nt} = 1$.

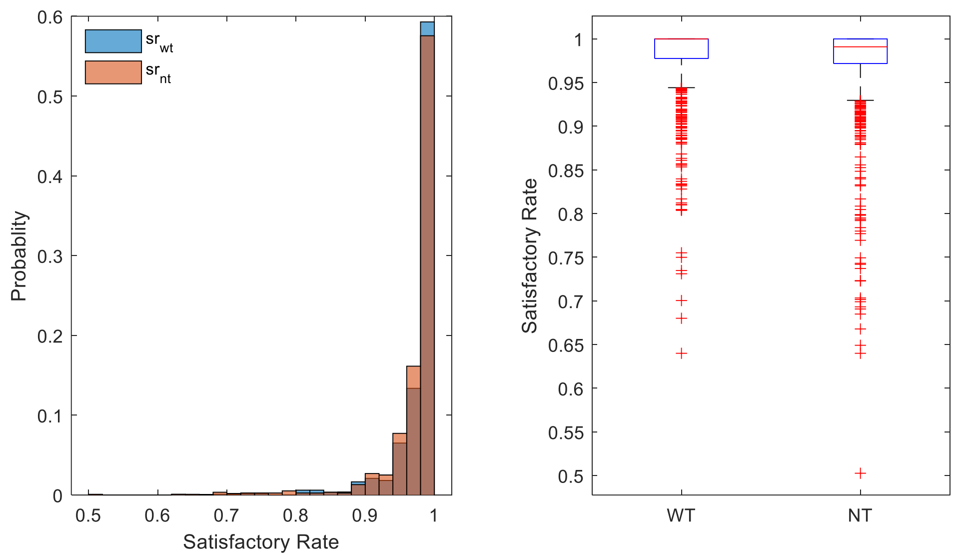


Figure 3.27: Satisfactory rate for ACN-Data: left) satisfactory rate distribution and right) boxplot for each charging event with and without considering tails during scheduling.

Additionally, ineffective charging events happen more frequently for NT scheduling as more outliers are shown in the right figure of Figure 3.27. An average of 0.69% sr improvement can be achieved by sr_{wt}.

3.3.2 Distributed Control

The centralized control requires future information for each EVSE, which is not available for most real-world applications. Furthermore, the distribution probability of charging events at the individual EVSE scale is more robust and random than the charging facility scale. In this section, a distributed smart charging algorithm is proposed.

(I) Control Objectives and Constraints

Suppose at time index $t-1$, the scheduling vector for all EVSEs is $\boldsymbol{y}_{T\times N}(t-1)$, where $\boldsymbol{y}_i$ is the scheduling charging power for EVSE i, and $\aleph_{t-1}$ is the EVSEs in use at $t-1$. At time index t, Δn new charging events are added and $\aleph_t$ EVSEs are in use. The newly in-use EVSEs are $\mathfrak{n}_t$, where $\mathfrak{n}_t \in \aleph_t$, $\mathfrak{n}_t \notin \aleph_{t-1}$. For each EVSE $n_i \in \aleph_t$, the objective of EVSE n_i for the new charging event $CE(t)$ is the same as the centralized scheduling, which can be formulated as $\min(1-\omega)\cdot\left(price_{ele}\times\left(\boldsymbol{y}_{-n_i}+\boldsymbol{y}_{n_i}\right)\cdot\Delta t+\gamma\cdot price_{dem}\cdot\max_t\left(\boldsymbol{y}_{-n_i}+\boldsymbol{y}_{n_i}\right)\right)+\omega\cdot price_{emi}\times\left(\boldsymbol{y}_{-n_i}+\boldsymbol{y}_{n_i}\right)\cdot\Delta t$, where $t<t+Park$, and $\boldsymbol{y}_{-n_i}$ is the scheduling vector for other EVSEs except n_i. However, the above scheduling problem only solves the scheduling vector for current known events and tends to constantly delay charging events to a late time when the price is low. To optimize the current scheduling vector with the considering of future possible charging events, instead of directly minimizing the electricity and emission cost function for EVSE n_i, the new

scheduling vector can be updated by minimizing the difference between $\widehat{\boldsymbol{y}}_{n_i}$ and $\boldsymbol{y}_{n_i}$, as shown from (3.26) – (3.29).

$$\min \omega_t \times \sum_{t=1}^{Park_{n_i}(t)} (\widehat{\boldsymbol{y}}_{n_i}(t) - \boldsymbol{y}_{n_i}(t))^2 \tag{3.26}$$

$$\widehat{\boldsymbol{y}}_{n_i} = \widehat{\boldsymbol{p}}_{wt|day_j} - \boldsymbol{y}_{-n_i} \tag{3.27}$$

$$\sum_{t=1}^{Park_{n_i}(t)} \boldsymbol{y}_{n_i}(t) \cdot \Delta t = Request_{n_i}(t) \tag{3.28}$$

$$0 \leq \boldsymbol{y}_{n_i} \leq c_{n_i} \tag{3.29}$$

ω_t: weight function of time $t \propto f(t)$, such that when the charging events number are small, or the first a few charging events of a day, the scheduling is not delayed to satisfy a lower difference for high $\widehat{\boldsymbol{p}}_{wt}$ at a later time.In this section, $\omega_t = \frac{1}{t}$.

$\widehat{\boldsymbol{p}}_{wt}$: the average of the optimized charging power with the consideration of tail feature from the past days of operation. $\widehat{\boldsymbol{p}}_{wt|day_j} = \frac{1}{days} \sum_{days=day_j} \boldsymbol{p}_{wt}$. In the simulation below, $\widehat{\boldsymbol{p}}_{wt}$ is further smoothed by the one-hour moving average filter.

c_{n_i}: the allowed charging power from (3.23).

(II) Scheduling Game

Each EVSE aims at updating its schedule vector and maximizing the benefit for its own purpose, and eventually, no EVSE will benefit when changing its scheduling vector. The scheduling problem can be formulated as a scheduling game, and the final schedule vector $\boldsymbol{y}(t)$ can be solved by the unique Nash Equilibrium (NE) [100]. The distributed

EV energy scheduling game is presented in Algorithm 3.5. The game is played whenever new charging events are submitted to EVSEs. Lines 1 – 3 initialize schedules for the new EVSEs $\mathfrak{n}_t$, and broadcast the new schedules to $\aleph_{t|-n_i}$. In lines 4 – 17, for $\forall n_i \in \aleph_t$, if it receives broadcast information from other EVSEs, it needs to update its schedule. $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.1$ are the converging tolereance. Once there are no changes for new $y_{\aleph_t}$, the unique NE is found, and the distributed schedules are optimized.

Algorithm 3.5: Distributed EV energy scheduling game

Input: new charging events $CE(t)$ at EVSEs $\mathfrak{n}_t$, $\widehat{\boldsymbol{p}}_{wt|day_j}$

Output: $\boldsymbol{y}_{\aleph_t}$

1 **for** $n_i \in \mathfrak{n}_t$

2 Update $\boldsymbol{y}_{n_i}$ by solving (3.26), and broadcast $\boldsymbol{y}_{n_i}$ to $\aleph_{t|-n_i}$, set $\boldsymbol{y}^{\boldsymbol{o}} = \boldsymbol{y}$

3 **end for**

4 **while** *true*

5 **for** $n_i \in \aleph_t$

6 **if** new schedule announcement received

7 $\widehat{\boldsymbol{y}}_{n_i} = \widehat{\boldsymbol{p}}_{wt|day_j} - \sum_{i \neq n_i, i \in \aleph_t} \boldsymbol{y}(i)$

8 Update $\boldsymbol{y}_{n_i}$ by solving (3.26)

if $\left|\boldsymbol{y}_{n_i} - \boldsymbol{y}^{o}_{n_i}\right| > \epsilon_1$

9 Broadcast $\boldsymbol{y}_{n_i}$ to $\aleph_{t|-n_i}$

10 **end if**

11 **end if**

12 Update $\boldsymbol{y}^{\boldsymbol{o}} = \boldsymbol{y}_{\aleph_t}$

13 **end for**

14 **if** $\left|\boldsymbol{y}_{\aleph_t} - \boldsymbol{y}^{\boldsymbol{o}}_{\aleph_t}\right| < \epsilon_2$ **or** $No\ broadcast\ for\ \forall n_i \in \aleph_t$

```
        jump out while
    end if
end while
return y_{ℵ_t}
```

The control principles for real-time distributed EV energy scheduling are as follows:

(i) Initialization: select $\widehat{\boldsymbol{p}}_{wt|day_j}$ for each day number, and initialize scheduling vector $\boldsymbol{y}(0)$ and charging events memory $\mathcal{M}_{N\times 3} = [t\ Request\ Park]$ to be zeros.

(ii) At the time index t, if new charging events are added to the system, $\mathcal{M}_{\daleth_t} = [t\ Request_{\daleth_t}\ Park_{\daleth_t}]$. Run Algorithm 3.5 and update $\boldsymbol{y}_{\aleph_t}(t)$; else, remain the same schedule: $\boldsymbol{y}_{\aleph_t}(t) = \boldsymbol{y}_{\aleph_t}(t-1)$. Charge EVs using $\boldsymbol{y}_{\aleph_t}(t)$.

(iii)Update $\mathcal{M}_{\aleph_t} = \left[t+1, Request_{\aleph_t} - \boldsymbol{y}_{\aleph_t} \cdot \Delta t, Park_{\aleph_t} - 1\right]$.

(iv)At time index $t \leftarrow t+1$, go back to (ii), until $t > T$.

(III) Simulation Results and Discussions

Figure 3.28 and Figure 3.29 display the BAU and the scheduled/ optimized EV charging power for April 2019, solved by (3.17) with the constraints (3.18) and (3.30) – (3.23). The black broken lines in each graph are the 1-hour moving average of the average power of this day type. Different curves are represented for the simulation results for different days. The shape of average power for weekdays shares a similar and more stable pattern than the BAU power. Hence, the $\widehat{\boldsymbol{p}}_{wt|day_j}$ can be chosen as the forecast load.

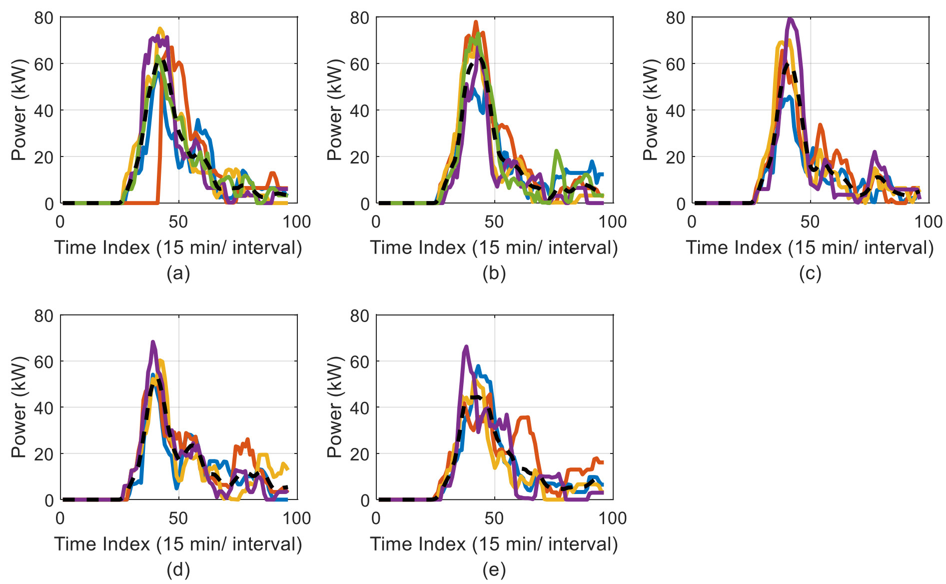


Figure 3.28: The BAU EV charging power for different weekdays: (a) Monday, (b) Tuesday, (c) Wednesday, (d) Thursday, and (e) Friday.

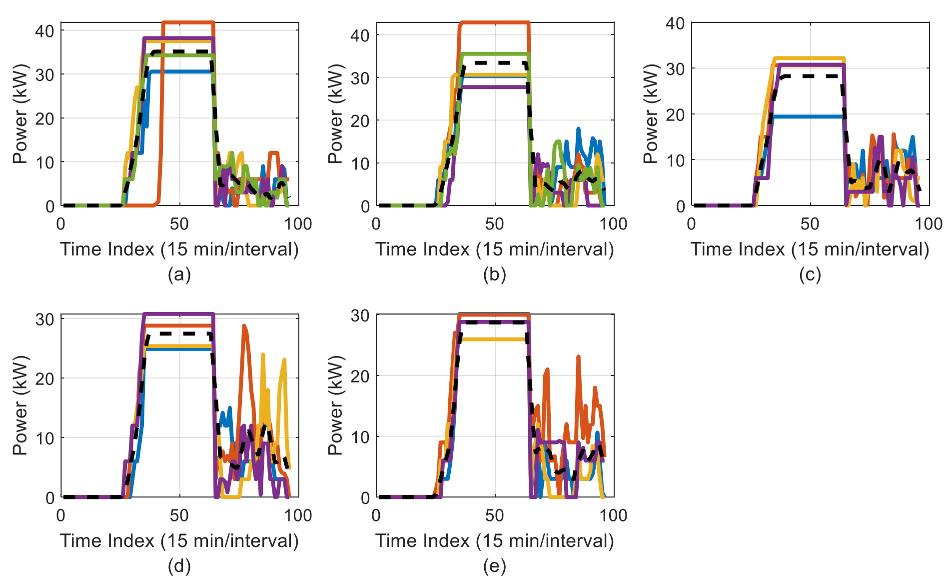


Figure 3.29: Optimized EV charging power for different weekdays: (a) Monday, (b) Tuesday, (c) Wednesday, (d) Thursday, and (e) Friday.

Figure 3.30 illustrates the distributed control simulation for days: May 6 – 9 and May 3, 2019 (Monday – Friday). In each simulation, the yellow ('Distributed') power curve can follow $\widehat{\boldsymbol{p}}_{wt|day_j}$ tightly and dramatically reduce the peak demand of charging power. The cost from different scheduling operations is shown in Figure 3.32 for the corresponding days.

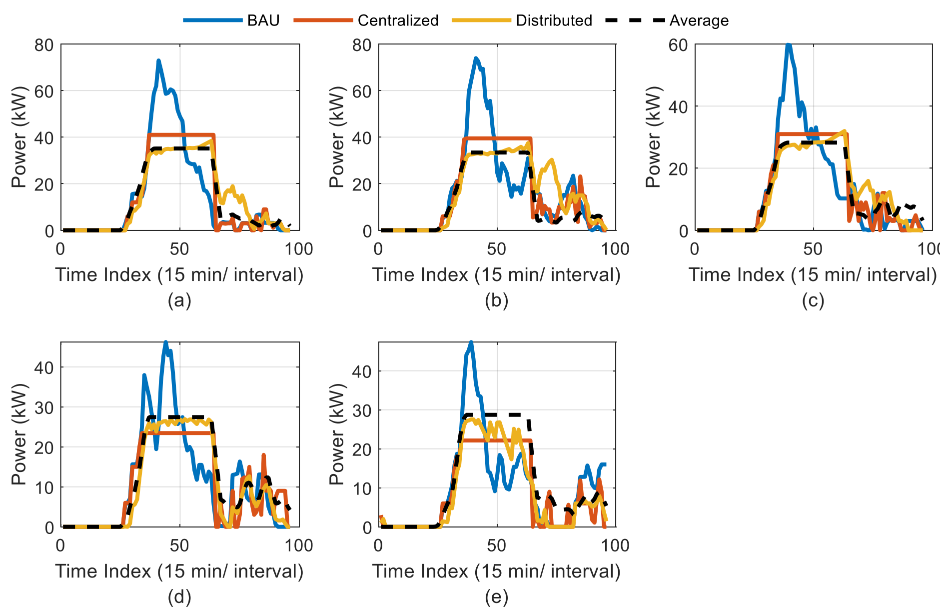


Figure 3.30: Distributed EV smart charging power for different weekdays via the average power baseline: (a) Monday, (b) Tuesday, (c) Wednesday, (d) Thursday, and (e) Friday.

One limitation is that the performance of the distributed control is heavily affected by the accuracy of the baseline – the average power from the historical data. When the charging events do not follow the historical patterns (e.g., more/less charging events), the distributed control fails to converge to the optimal solution. As shown in Figure 3.30 (a) – (c), the base average charging power is less than the centralized charging power during the time indexes from 35 – 64. Thus, the flexibility to charge EVs at a later time is reduced. Some charging events are delayed to the on-peak rate period, and as a result, the electricity cost may increase.

To improve the accuracy of the baseline power, instead of keeping $\widehat{\boldsymbol{p}}_{wt} = \widehat{\boldsymbol{p}}_{wt|day_j}$ unchanged, $\widehat{\boldsymbol{p}}_{wt}$ can be online adjusted as follows:

(i) Find the ratio between the average requested power and the actual request power at t, $\theta = \frac{\boldsymbol{rp}(t)}{\widehat{\boldsymbol{rp}}_{day_j}(t)}$, where the request power at time index t is defined as: $\boldsymbol{rp}(t) = \sum_{i\in\aleph_t} \frac{Request_{i,t}}{Park_{i,t}\cdot\Delta t}$. θ is chosen in the range [-0.8,1.2]. The average requested power $\widehat{\boldsymbol{rp}}_{day_j} = \frac{1}{days}\sum_{days=day_j} \boldsymbol{rp}$. Calculate the average power variation in the following ΔT steps: $\Delta p = \theta \cdot (\widehat{\boldsymbol{p}}_{wt|day_j}(t{:}\, t+\Delta T-1) - \widehat{\boldsymbol{p}}_{wt|day_j}(t-1))$.

(ii) Update $\widehat{\boldsymbol{p}}_{wt}(t{:}\, t+\Delta T-1)$ via (3.20) and record the updated time step of $\widehat{\boldsymbol{p}}_{wt}{:}\, \bar{t} = t + \Delta T$.

$$\widehat{\boldsymbol{p}}_{wt}(t{:}\, t+\Delta T-1) = \begin{cases} \theta + \widehat{\boldsymbol{p}}_{wt}(t{:}\, t+\Delta T-1), \Delta p(1) = 0, \theta > 1 \\ -\theta + \widehat{\boldsymbol{p}}_{wt}(t{:}\, t+\Delta T-1), \Delta p(1) = 0, \theta < 1 \\ \Delta p + \widehat{\boldsymbol{p}}_{wt|day_j}(t-1), otherwise \end{cases} \quad (3.30)$$

(iii)If no new event is added (no new schedules), when $t > \bar{t}$, update $\widehat{\boldsymbol{p}}_{wt}(t) = \Delta p(1) + \widehat{\boldsymbol{p}}_{wt}(t-1)$.

The distributed EV charging scheduling via the adjusted average power baseline is displayed in Figure 3.31. From Figure 3.31 (a) – (c), the ‘Adjusted Average’ curves dynamically update from time indexes 35 – 65 and move the baseline power closer to the

“Centralized” curve. The overall electricity cost ($) for the 5-day simulation is reduced by 3.55% compared to control via the average baseline. When compared to the BAU operation, the electricity cost for the five days is increased by $13.47, in exchange for an average of 26.28 kW daily peak reduction. The distributed control via both baselines shows the effectiveness of smart charging to smooth the charging power and reduce the charging peak.

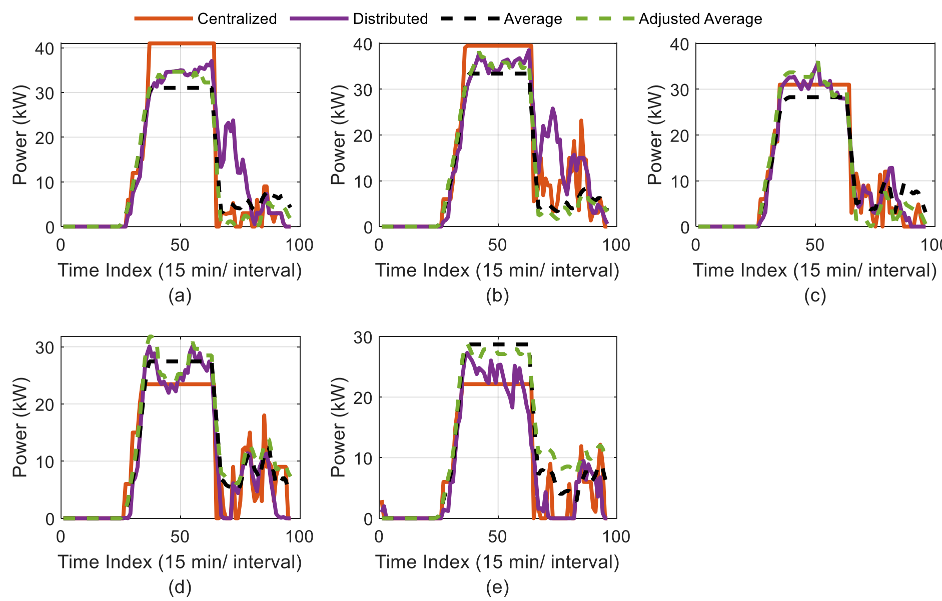


Figure 3.31: Distributed EV smart charging power for different weekdays via the adjusted average power baseline: (a) Monday, (b) Tuesday, (c) Wednesday, (d) Thursday, and (e) Friday.

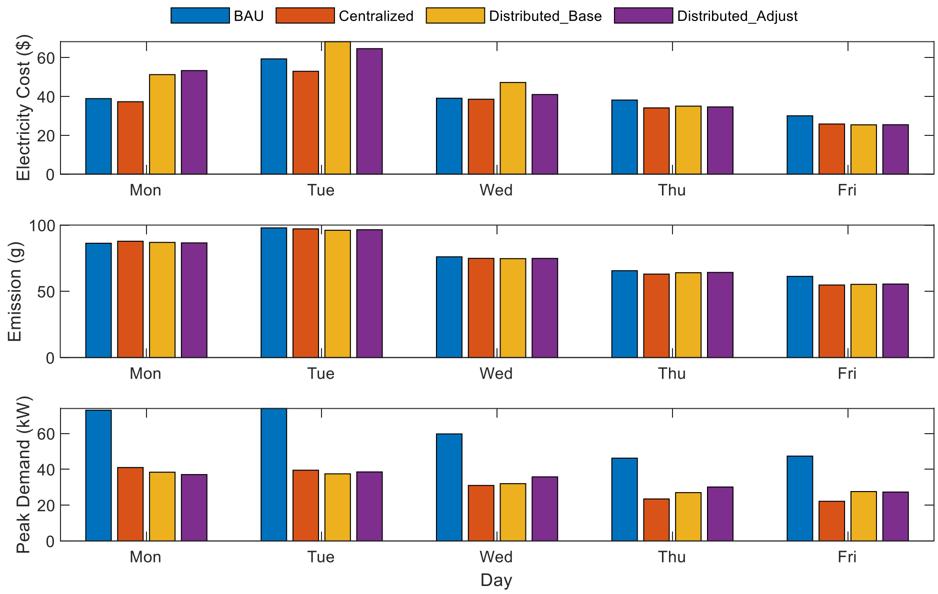


Figure 3.32: EV charging cost from different scheduling algorithms: top) electricity cost, middle) generated emission, and bottom) peak demand.

3.4 Conclusion

This chapter presents a direct tail extraction algorithm and novel distance measure $RMPS$ for tail time series classification. The proposed tail extraction finds the tail time series without the prior knowledge of the EV. However, those tails are noisy and commonly share different lengths. The proposed $RMPS$ as the shape-wise distance measure finds the global similarity of two tails by comparing each subsequence of the two time series. Section 3.2.5 compares the $RMPS$ with other distance measures. $RMPS$ surpasses all other distance measures when optimal K is chosen, and distinct cluster representatives can be extracted via k-medoids classification. Different applications of the learned tail features are discussed in Section 3.2.6: 1) reducing the future EV charging demand and 2) efficient real-time prediction and scheduling. With the tail representatives (medoids) learned from the tails, it is found that the EV charging demand planning for future EV population projection can reduce by 4%, resulting in a 17.4 MW, 74.5 MW, and 248.4 demand MW reduction. In most clusters, the battery capacity distribution ranges are small, which may provide the EV charging station to identify the possible battery capacity for the plugged-in vehicles, thus better serving their charging energy needs.

This chapter also presents centralized and distributed EV smart charging algorithms with the consideration of the tail features. As illustrated in Section 3.3.1, with proper ω and γ are chosen, the centralized EV charging power scheduling can continuously optimize the PAR, electricity cost, and emission generation. The FA and SA simulations under different case scenarios indicate the tail features can improve the scheduling efficiency and accuracy, with a 4% - 5% flexibility rate and 0.69% satisfactory rate improvement, respectively. However, the centralized optimization requires all information to be acquired at the stage of optimization, which is infeasible for most real-world applications. The distributed smart charging algorithm presented in Section 3.3.2 forms an energy scheduling game, and the optimal schedule is the unique Nash equilibrium that for any new charging event at $CE(t, m, n)$, the $n_i \in \aleph_t \backslash \aleph_{t-1}$ EVSE first optimizes for its own interest, and the rest $-n_i$ EVSEs adjust their own schedules until no EVSE would benefit by deviating from the scheduling vector. By applying the average charging power for each day type as the prediction model, and online adjusting the prediction by FL-AELM, the distributed scheduling can smooth out the load profile of an average of 26.28 kW daily peak reduction.

Chapter 4 Real-world Microgrid Applications with Solar + Battery Architecture

Based on the microgrid definition by the U.S. Department of Energy Microgrid Exchange Group [101], a microgrid should have the ability to generate and distribute energy within its internal architecture, as well as be able to control different load components as a single intelligent network entity. The on-site energy generated from the microgrid system can be from renewable energy sources such as solar and wind. From 2000 – 2020, renewable energy consumption increased from 6,104 Trillion Btu to 11,689 Trillion Btu, accounting for 6.2% - 12.6% of the total energy consumption in the U.S. [102]. One of the majority changes from the investment and increment of the solar energy generation. The solar energy consumption portion of the total renewable energy has increased from 1.0% - 10.7% in the recent 20 years.

Due to the intermittent nature of renewable sources such as solar, energy storage systems have been integrated into the architecture of microgrid systems to make them more efficient and robust. The integration of a BESS within a microgrid architecture that includes renewable generation provides a dual benefit: 1) it makes the microgrid system more controllable, reliable, and robust by managing the intermittent nature of renewable sources, and 2) adds economic value to the operation of the microgrid by implementing a variety of energy management strategies, including the reduction of demand charges under a time-of-use (TOU) rate schedule.

4.1 Background of Microgrid

4.1.1 Microgrid Definition

The Microgrid Exchange Group defines the microgrid power system as:

> *"A Microgrid is a group of interconnected loads and distributed energy resources within clearly defined electrical boundaries that acts as a single controllable entity with respect to the grid. A Microgrid can connect and disconnect from the grid to enable it to operate in both grid-connected or island-mode."* [101]

Based on the above definition, a microgrid should have the ability to generate and distribute energy within its internal architecture and control different load components as a single intelligent network entity [103]. Three essential components should be included in a microgrid based on CIGRÉ C6.22 Definition Qualifiers: (1) generators (most are renewable energy sources (RESs)), (2) storage devices, and (3) controllable loads. Microgrids are controlled by their own microgrid central controller, which has the ability to 1) coordinate self-distributed generation units, distributed energy storage units, and controllable loads; 2) provide support to critical loads; 3) be capable of being connected to or isolated from the distribution system [104] and, 4) plug and play operation [105].

4.1.2 Background of the Microgrid Energy Management

As an increasing number of microgrids have been developed and connected to the power grid distribution network, interconnecting multiple microgrids as a single microgrid cluster (MGC) provides an effective way to improve the operational performance and reliability of large-scale DERs integration [106]. Each microgrid has onsite renewable distributed generation (RDG), controlled distributed generation (CDG), battery energy storage system (BESS), and electrical loads. The RDG refers to non-controllable and non-dispatchable resources, such as PV and wind. Each individual microgrid is controlled by its own EMS and can communicate with the EMS of neighboring microgrid and the power grid. The microgrids are clustered and controlled by the upper-level community EMS.

Besides offering economic and environmental benefits, microgrids also improve the whole system's reliability. The microgrid systems have been demonstrated to minimize the power output of conventional generator sets and maximize power output from RESs [107]. Additionally, by implementing power sharing between microgrids with the main grid, the microgrids can reduce the total operational cost of the future distribution network [108], [109]. The additional avoided CO_2 emissions from the reduced active power loss along the generation and distribution from RESs using microgrids technologies have also been evaluated [106], [110]. Power systems with interconnected microgrids have shown to be more stable and resilient than individual islanded microgrids to deal with blackouts when intermittent DERs provide most of the generation [111]. Lu et al. investigated a wind turbine-solar photovoltaic (PV)-battery MGC system that reduced the distribution grid's power loss through power exchange and saved responsive reserve from dispatchable resources, demonstrating that both aspects can improve the system reliability [112].

In the literature, the microgrids are divided into two categories: AC microgrids and DC microgrids. AC microgrids are most studied in the current utility grid environment, connected with the utility grid at the point of common coupling [113]. Studies focus on the optimal power scheduling [107], [114], [115], voltage and frequency regulation [116], and islanding protection [117], [118].

Due to the higher efficiency associated with less DC-AC power conversion of the DC characteristic of RESs, energy storage systems, and electric vehicles, as well as circumventing the drawbacks of synchronization, reactive control, and voltage stability of the AC microgrids, significant research efforts have focused interconnecting microgrids through a common DC-link [119]. A novel simultaneous control interface and protection scheme for DC microgrids has been reported in [120]. A DC MGC testbed is operated in a real-world scenario by connecting 19 inhabited houses with a DC power bus line [121].

Hybrid microgrid systems with a combination of AC and DC microgrids taking advantage of both systems are also reported in the literature [122], [123].

There are two primary control schemes for the microgrid EMS: centralized and decentralized (distributed). A centralized EMS control scheme observes the real-time data within the microgrids and makes the decision for the whole microgrid system. A centralized EMS typically follows the hierarchical way with the two-level optimization to minimize the operational cost of the whole system and the individual components [124]. The lower-level EMS runs the optimization for the single microgrid component in a scheduled period for its minimal operational cost and shares the optimal exchange power with the upper-level EMS. After the upper-level EMS receives the whole information of the system, it determines the connection of the microgrids and the central MGC system. By sharing market transaction information, centralized control can achieve maximum economic benefit.

Due to lack of scalability, the burden of communication, and privacy issues with centralized controllers, decentralized microgrid EMS controllers have gained much attention recently [115] [125]. The multi-agent system (MAS) based distributed control architecture for microgrids is widely discussed in the literature. Han et al. carried out a comprehensive review of the MAS-based controller for microgrids [126]. The topology, mathematic models of the MAS system, communication delay, energy coordination, and economic dispatch strategies are reviewed. The RESs, controllable generators, energy storage system, load, and grid are modeled as different agents to monitor and/ or control the related components [115]. Global optimization is carried out by minimizing the usage of diesel generators on a daily basis. Non-cooperative game theory is used to coordinate each agent to optimize the energy dispatch and costs. Hierarchical bi-level control schemes are applied for a real-time interactive EMS framework [127]. The virtual load and grid are aggregated in the discussion. The primary controller of distributed EMS for each microgrid aims at maximizing the use of non-dispatchable resources and increasing the energy stored in the energy storage. After all EMSs are executed, the shortage or surplus of the microgrid power is sent to the secondary level controller to minimize the mismatch between the feed power by microgrids and the load demand. The stability of the tie-line, which connects different microgrids, shows the autonomy of each microgrid. Liu et al. focused on fair energy trading among microgrids [128]. The distributed robust optimal energy scheduling for microgrids is achieved by minimizing the total cost including fuel, degradation cost of BESSs, net energy trading cost with the main grid and energy transfer cost among the microgrids under the worst-case scenario of parameter uncertainties (real-time energy market, forecast of renewable generation and low consumption).

4.2 Microgrid I: Rancho Cucamonga City Hall (RCCH) Microgrid

This work designs and performs a real-time battery control strategy for a microgrid system located at Rancho Cucamonga, CA, which is located in Southern California and has an abundant solar energy resources. The microgrid integrated a Battery Energy Storage System (BESS), and carport photovoltaic (PV) system provides a dual benefit: 1) it makes the microgrid system more controllable, reliable, and robust by managing the intermittent nature of renewable sources, and 2) adds economic value to the operation of the microgrid by implementing a variety of energy management strategies.

4.2.1 System Architecture

Figure 4.1 illustrates the architecture of the RCCH microgrid system, which includes 125 kW of solar generation, 100 kW/87 kWh stationary BESS, and a high electricity consumption building. The existing microgrid system is displayed in Figure 4.1.

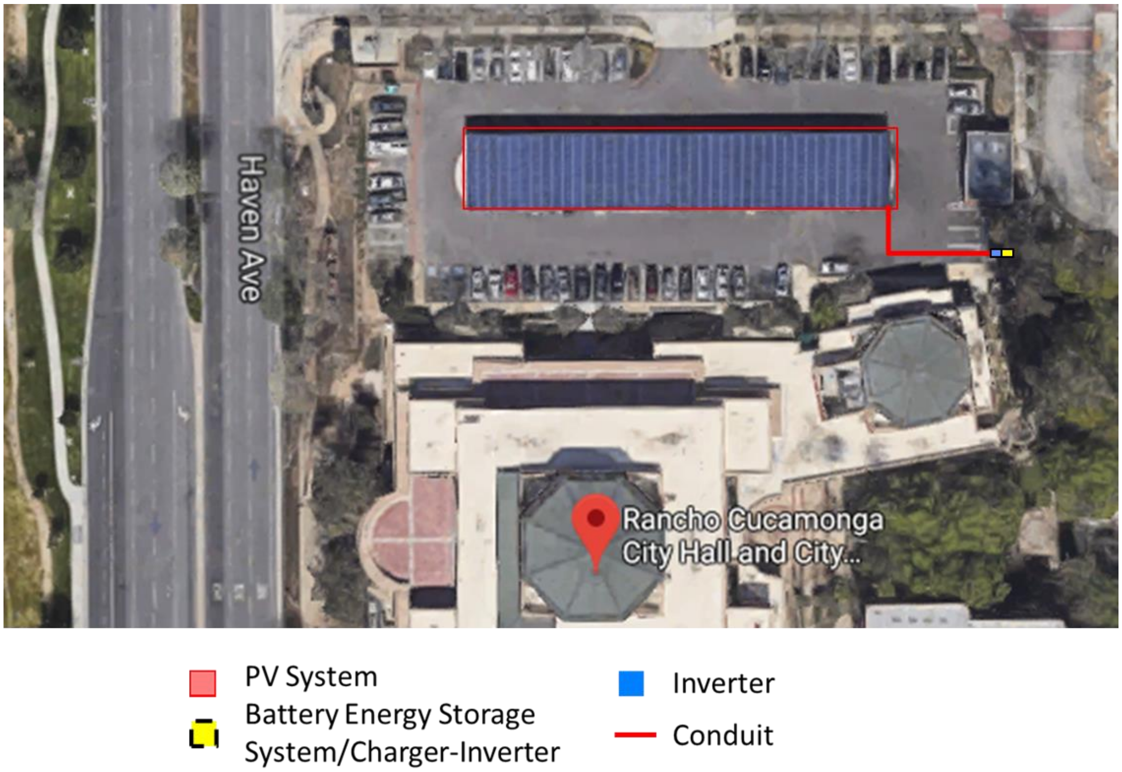


Figure 4.1: Photograph of the RCCH microgrid.

The data flow and architecture of the RCCH microgrid are shown in Figure 4.2. On the top left of Figure 4.2, the control system consists of LabVIEW and Matlab software for real-time battery operation computing and the microcontroller as the data bridge and control switch between the battery system and the software. On the top right of Figure 4.2, the physical battery system consists of 83 kWh battery banks, DC - AC inverter, and BMS for battery cell balancing and calibrating. The battery bank charger is directly connected to the building's power system that permits the BESS to directly charge (discharge) from(to) the utility grid. The solar PV system includes a data logger AcquiSuite which reads solar power from the solar meter (Veris) and sends solar power to the local database. Similar to solar, building load information is metered and sent to the database.

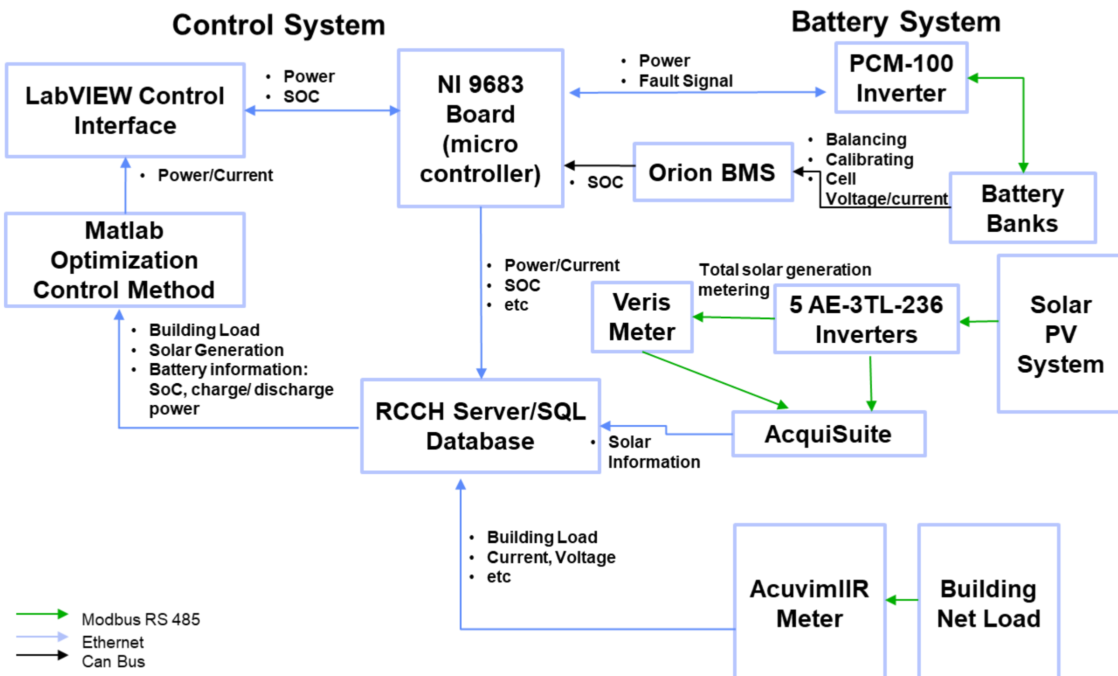


Figure 4.2: Overall system architecture and data transmission.

The electric tariff for the building is Riverside Public Utility's (RPU's) large general and industrial service, Time-of-Use (TOU) rate schedule. In addition to fix charges, the monthly TOU rate schedule charges consist of two components: demand charges ($/kW) and energy use charges ($/kWh). There are three TOU periods: off-peak, mid-peak, and on-peak, which change between the summer and winter seasons [129] as shown in Table 4.1, where off-peak has the lowest, and the on-peak has the highest energy rate. The demand charges consist of on-peak (summer) and facility (summer and winter) demand charges. Facility demand refers to the maximum average kilowatt input indicated or recorded by instruments during any 15-minute metered interval in a month.

Rate Period	Summer	Winter
	June to September	October to May
Off-Peak	11 PM – 8 AM	9 PM – 8 AM
Mid-Peak	8 AM –12 PM 6 PM – 11 PM	8 AM – 9 PM
On-Peak	12 PM – 6 PM	---

Table 4.1: Time-of-use rate periods for SIGI microgrid.

4.2.2 Solar Generation and Building Load Consumption Characteristics

Figure 4.3 displays the monthly and yearly comparison between the simulated solar generation and average building electricity consumption for RCCH. The solar generation is simulated via System Advisor Model (SAM) [130]. The parameters of SAM modeling are listed in Table 4.2. The building electricity consumption is from the facility's electrical bill from January 2014 – August 2015, and the monthly consumption is averaged for January – August. The on-site solar generation only contributes to slight energy reduction since the yearly solar generation is 3.56% of the building electricity consumption. As shown in Figure 4.4, the demand charge accounts for more than 27.5% of total electricity cost for all the months and over 45% for the summer rate periods. Thus, decreasing the building's peak demand, especially during the on-peak rate period, could significantly reduce the demand charges and electrical bills.

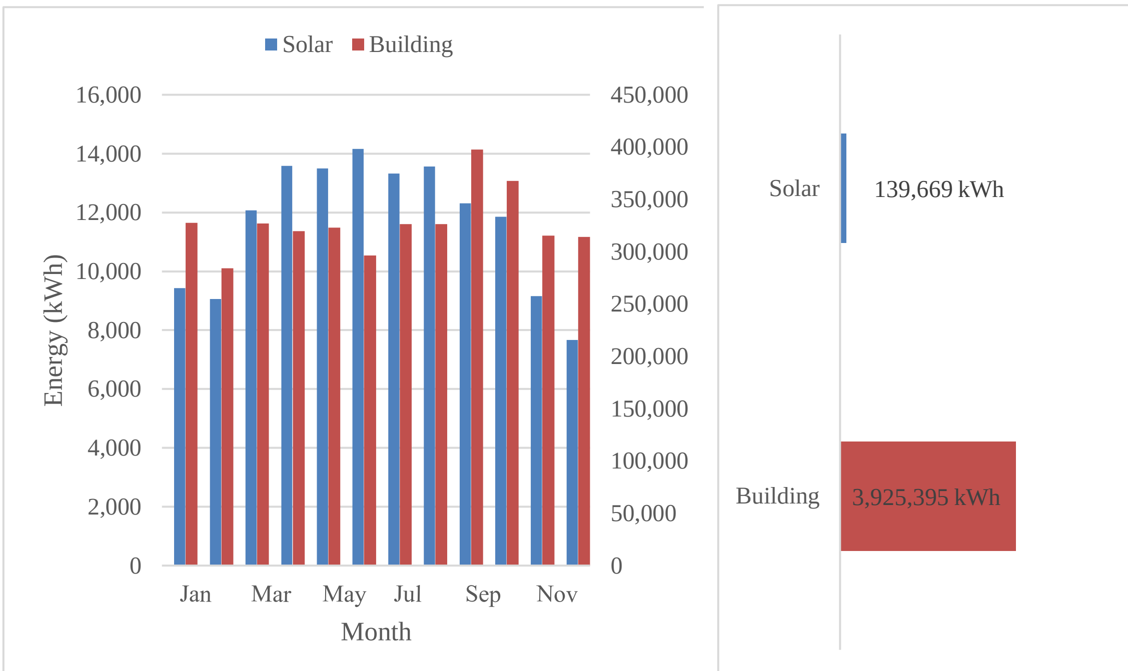


Figure 4.3: RCCH solar generation vs. building load consumption: left) monthly comparison; and right) yearly comparison.

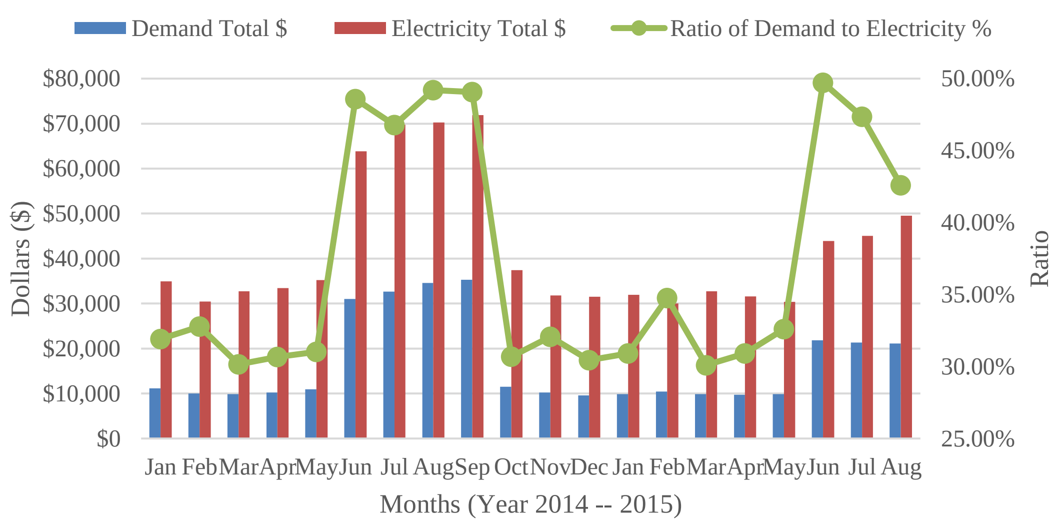


Figure 4.4: Demand charges and total electricity charges.

Location	Rancho Cucamonga, CA
Modules	Canadian Solar CS6P-260P SunPower SPR-P17-340-COM (2)
Inverter	5 AE 3TL-23-6
System Design	5 Subarrays Modules per string in subarray: 12 Strings in parallel in subarray: 6
System Power Rating	125 kW

Table 4.2: SAM parameters configuration for the RCCH solar carport system.

Figure 4.5 left) displays the average building load in different day patterns, such as workdays (Mondays – Thursdays) and non-workdays (holidays and Fridays – Sundays), as shown in the five lines except for the "Non-workdays" and "Non-workdays" lines, respectively. During regular workdays, the building load in summer is higher than in winter due to heating, ventilation, and air conditioning (HVAC) operations. In both seasons, the daily peak demand always appears between 1:00 PM – 5:00 PM, and the monthly peak load always appears around 1 PM – 3:30 PM, as shown in Figure 4.5 right). On non-working days, the electricity consumption is stable and much smaller than regular working days.

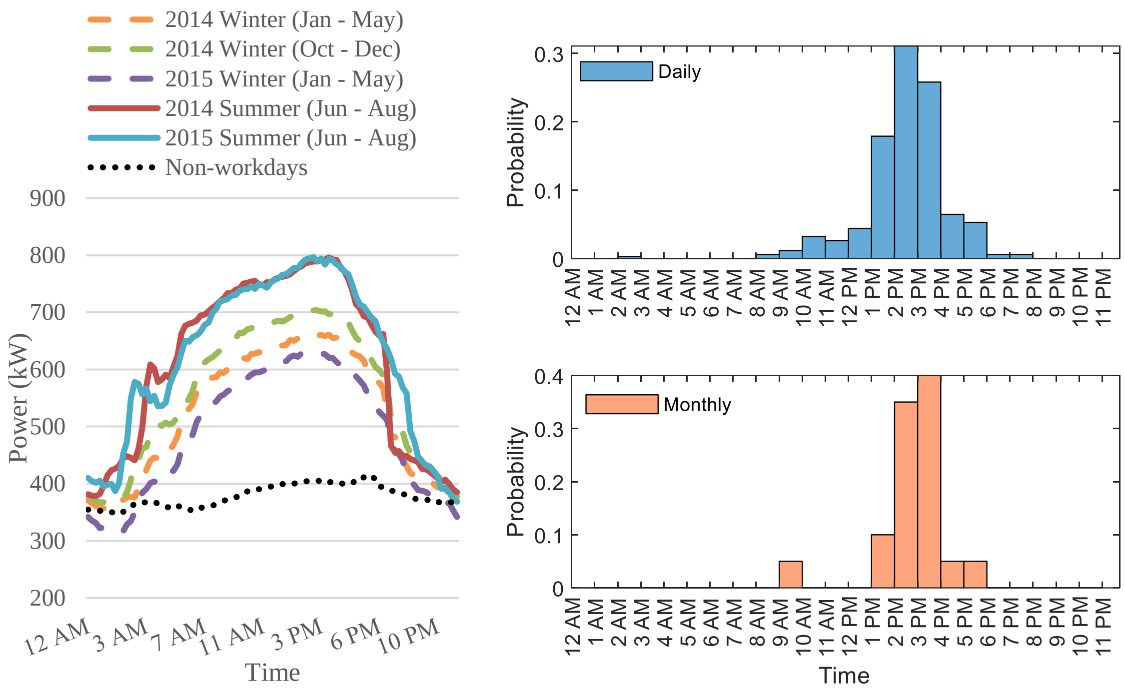


Figure 4.5: Building load characteristics: left) building load patterns for summer and winter rate periods, and workdays and non-workdays; and right) peak demand occurrence at the time of a day.

4.2.3 System Control Objectives and Constraints

Generally, well-documented meteorological information and long-term historical trends are required for establishing accurate prediction models for solar generation and building loads [77, 78]. In practice, building sophisticated prediction models is often hindered by the ease of access by end-users to comprehensive data sets. This challenge is augmented by the different demand charges during one billing cycle that requires further consideration for minimizing peak demand across multiple rate periods, which is an aspect of demand charge management that is regularly ignored in the literature.

The objective of the RCCH microgrid control is to design and perform a real-time battery control scheme for the RCCH microgrid where only basic information such as real-time building load, solar power generation, and short-term historical data are available. The proposed control scheme employing three different operations strategies, one for each rate period is to:

- minimize the costs associated with demand charges by optimizing the performance of a solar carport photovoltaic (PV) system coupled with the BESS;
- efficiently utilize the BESS to store and deliver energy.

The system's limitation lies in the limited battery storage and solar generation compared to the consumption, making the BESS impossible to cover the entire on-peak rate periods. As shown in Figure 4.5, the occurrence time of peak demand is consistent. Considering the limited discharging time of the BESS, an optimized battery discharging period is required for achieving lower monthly power demand for the building.

This advanced and easy implementation real-time battery control scheme can increase the system's economic benefit without adding the complexity of the microgrid system, which can achieve considerable electricity cost reduction for the RCCH microgrid.

4.2.4 Control Algorithm Description

To efficiently manage fluctuating peak demand under the TOU rate schedule, the developed battery control scheme consists of three battery operational modes for each of the three different rate periods. The battery charging and discharging process is modeled as a linear process from the field experiments. The charging and discharging process is constrained within 20% to 90% state of charge (SOC) to prolong the BESS's useful life span. The BMS stops discharging and only allows charging when the SOC reaches 20%.

As discussed above, for reducing the facilities demand throughout the day, 12 PM – 5 PM is the period of interest for the BESS operation.

(I) Off-Peak Mode

In this mode, the control algorithm determines the battery operation from 11 PM to 8 AM in the summer rate periods, 10 PM to 8 AM in the winter rate periods, and whole days on weekends and holidays. The control objectives during off-peak battery operation are: 1) charging the BESS to 90% SOC and 2) maintaining a low off-peak demand value. The off-peak threshold dem_{fac} is predetermined by the previous months' off-peak load demand. In most cases, the BESS maintains a stable charging power and does not exceed dem_{fac} due to low and stable electricity consumption. When the electricity load consumption is high, an autonomous adjustment of dem_{fac} is achieved by tracking the SOC difference between the desired and the actual charging power. The battery's charging power can be expressed in (4.1).

$$bp = \begin{cases} 0, \widetilde{nl}_{off} > 0 \\ -\widetilde{nl}_{off}, \widetilde{nl}_{off} + pca > 0 \cap \Delta soc < 0 \\ pca, else \end{cases} \tag{4.1}$$

bp: battery charging/discharging at the current time, where $bp > 0$ for discharging process and $bp < 0$ for charging process;

$\widetilde{nl}_{off}$: allowed off-peak net load, $\widetilde{nl}_{off} = bl - so - dem_{fac}$;

bl: the building load at the current time; nl: the net load at the current time;

so: the solar-generated power at the current time;

soc: the SOC at the current time;

pca: the average charging power at the current time, where

$$pca = -\min\left(\frac{(soc_{max} - soc) \cdot C}{Remaining\ Off\ Time \cdot \mu}, INV\right)$$

C is the BESS capacity, INV is the battery inverter size ($INV = 100$);

Δsoc: the difference between the preferred SOC and the actual SOC, where

$$\Delta soc = \mu \cdot (pca - bp) \cdot \Delta t + \Delta soc$$

μ is the charging and discharging loss/ efficiency: $\mu = \begin{cases} 1.1, discharging \\ 0.9, charging \end{cases}$. When $\Delta soc \leq -0.5$, dem_{fac} is increased by 2.5 kW and $\Delta soc = 0$ is reinitialized. In most cases, the BESS charges at pca due to the low electricity consumption.

(II) Mid-Peak Mode

In this mode, the control algorithm determines the battery operation between 8 AM – 12 PM and 6 PM – 11 PM. Since the net load is relatively low between 9 PM to 6 AM, only two time periods need to be monitored: 8 AM to 12 PM and 6 PM to 9 PM. During the first period, the BESS works in a schedule mode which only discharges when $nl > dem_{fac}$ and charges when $soc \leq 90\%$. During the second mid-peak period, the BESS charges up to 50% SOC to prepare for the unpredictable high demand for the rest of the time for this period.

(III) On-Peak Mode

In this mode, the control algorithm determines the battery operation between 12 PM – 5 PM. The control algorithm aims at reducing the peak demand. Besides the minimization in (3.30), the building net load NL (captical letters represent vectors of related parameters) is constrained by (4.7) to maintain the lowest value.

$$\min Pr_e^T \cdot NL \cdot \Delta t + \frac{1}{M - L + 1} \cdot \max (NL) \tag{4.2}$$

$$subject\ to\ BC = BC_0 - \mu \cdot \Lambda \cdot BP \cdot \Delta t \tag{4.3}$$

$$0 \leq BP \leq INV \tag{4.4}$$

$$BC_{min} \leq BC \leq BC_{max} \tag{4.5}$$

$$NL = BL - SO - \mu \cdot BP \tag{4.6}$$

$$NL \leq dem_{on} \tag{4.7}$$

Pr_e: energy price;

BC: the BESS battery capacity vector: $BC_{min} = 0.2C, BC_{max} = 0.9C$.

Λ: lower unit triangular matrix.

$\widetilde{nl}_{on}$: allowed on-peak net load, $\widetilde{nl}_{on} = bl - so - dem_{on}$

The Model Predictive Control (MPC) approach is utilized for decision-making at each control time interval. At time period k, the battery operation is achieved by optimized (4.2) with constraints (4.3) - (4.7). Due to the small capacity of the battery, the battery discharging power is always limited to make the net load no larger than 20 kW of the dem_{on} as shown in below (3.30) where $\overline{NL}_{max}$ is the maximal values of all previous net load during the battery operation, and $power_o = BP_{opt}(1)$ and recursively reduced by 2.5 kW until $load - solar - power_f \leq \overline{NL}_{max} - 20$.

$$\boldsymbol{bp}(k) = \begin{cases} 0, nl \leq \overline{NL}_{max} - 20 \\ power_f, load - solar - power_f \leq \overline{NL}_{max} - 20 \\ BP_{opt}(1), else \end{cases} \tag{4.8}$$

In (4.7), dem_{on} is adaptively adjusted by the deviation between actual SOC change and the predicted change ΔSOC. The online adjustment is expressed in (4.9).

$$\Delta SOC_i = \sum_{j=t}^{k} \Delta SOC(j) = [1 \quad \dots \quad 1]_{(k-t+1)\times 1} \cdot (\boldsymbol{bp}(t:k) - \Delta \boldsymbol{pd}(t:k)) \times \frac{\Delta t}{C} \tag{4.9}$$

$$dem_{on} = \max\left(\overline{NL}_{max}, dem_{on_o} - \frac{\Delta SOC_i \cdot C}{k-t+1}\right), if\ \Delta SOC_i \le \varepsilon$$

where t is the last time slot when dem_{on_i} changes, k is the current time, $\Delta \boldsymbol{pd} = BP_{opt}(2)$, and ε is a factor for the time and range of the deviation to change dem_{on_i}. A smaller initial dem_{on} makes dem_{on_i} adapting more quickly and vice versa. Meanwhile, a larger $|\varepsilon|$ increases the time to adjust to a new threshold which exploits more battery capacity to maintain the previous low load. The detailed control principles for RCCH are shown as follows:

(i) For each on-peak rate period of a day, time intervals can be divided into M slots (1 PM – 5 PM). The time slot is $\Delta t = 5$ minutes; within 1 hour, there are 12 time slots.

(ii) At the initial time, set time interval $i = 0$, and get the prediction model for solar generation building load, electricity price $\widehat{SO}_{M\times 1}$, $\widehat{BL}_{M\times 1}$, Pr_e respectively. Set BESS $disRate = 0.3$ and $\Delta SOC_i = 0$.

(iii) At time $i = k$, set optimization operation horizon $L_{des} = M - k + 1$. Retrieve the real time solar generation so_k, building load bl_k, battery capacity bc_k, battery operation power bp_k And the electricity price $Pr_e(k\!:\!M)$. Update prediction information $\widehat{SO}(k\!:\!M)$ and $\widehat{BL}(k\!:\!M)$.

(iv) If $soc(k) \le 85, bl_k - so_k < \max(dem_{on}, \overline{NL}_{max}) - 20$, allow charging process. $\boldsymbol{bp}_{charge}(k) = \max\left((\widetilde{nl}_{on} - 20), -50\right)$, $\Delta \boldsymbol{pd}(k) = 0$. Go to (vii).

(v) Else, calculate the average discharging power: $pda_k = \max(\frac{bc_k - disRate \cdot C}{M-k+1}, INV)$ and maximum average discharging power: $pdamax_k = \max(\frac{bc_k - 0.2 \cdot C}{M-k+1}, INV)$; when $pda_k \leq 0, set\ disRate = 0.2,\ pda_k = pdamax_k$.

(vi) Calculate the optimization problem (4.2) with constraints (4.3) – (4.7) and obtain the result BP_{opt} . Calculate $\boldsymbol{bp}(k)$ from (3.30) and $\Delta\boldsymbol{pd}(k) = \min\left(BP_{opt}(2), pdamax_k\right)$. Increase $flag$ by 1. If no result from optimization problem when the constraints cannot be satisfied, set $flag = 0, node = 1$,

$$\boldsymbol{bp}(k) =$$

$$\begin{cases} \min\left(bl_k - so_k - \max\left(\overline{NL}_{max}, dem_{on} - 10, dem_{on_o}\right), INV\right), \Delta SOC > 0 \\ \min\left(\widetilde{nl}_{on}, INV\right), else \end{cases}$$

Charging is allowed when $soc < 80\%$ and $\boldsymbol{bp}(k) < 0$. $\Delta\boldsymbol{pd}(k)$ is calculated following:

$$\Delta\boldsymbol{pd}(k) = \begin{cases} \min\left(\widehat{BL}(2) - \widehat{SO}(2) - dem_{on}, INV\right), \Delta SOC \leq 0 \\ \min\left(\widehat{BL}(2) - \widehat{SO}(2) - \overline{NL}_{max}, INV\right), \Delta SOC > 0 \end{cases}, \text{ and } \Delta\boldsymbol{pd}(k) \geq 0$$

(vii) Calculate ΔSOC_i by (4.9).

(viii) If $flag > 12$, update dem_{on} by (4.11).

$$dem_{on} = \max\left(dem_{on_o}, \overline{NL}_{max}\right) \tag{4.10}$$

Reset $flag = 12$. This step shows that if MPC can be successfully solved over an hour, on-peak threshold value should be chosen as the previous maximum net load or

dem_{on_o}. Future dem_{on} should be increased upon (4.11) rather than the previous calculated dem_{on_k}.

(ix) Calculate dem_{on_k} by (4.9), and record $t = k$. Reset $\Delta SOC = 0$; else go to (x).

(x) Set $i = k + 1, node = 0$; then go back to step (iv) until $i = M$.

4.2.5 Simulation Results

The solar and load forecast models in Figure 4.6 - Figure 4.7 are developed based on [133]: the solar generation and building load profile are predicted by first averaging the historical data as a baseline and then online updating proportionally between the actual values and baseline. The 'Building Load' curve represents the simulated building load by adding small Gaussian noise to the actual load. The difference between the 'Building Load' and 'Net w/o Battery' is the solar generation and the initial $dem_{on_o}$is retrieved by the average of monthly facilities demand. The two days are selected as the largest peak demand occurrence days for the related months. In Figure 4.6, dem_{on} is adjusted from the initial 789 kW to 860.73 kW. The peak demand is reduced from 968.05 kW to 887 kW with the help of solar generation, down further to 845.01 kW when combined with the proposed battery control algorithm. In Figure 4.7, dem_{on} is adjusted from the initial 710 kW to 790.96 kW. The peak demand is reduced from 837.27 kW to 810.4 kW, further to 787.83 kW.

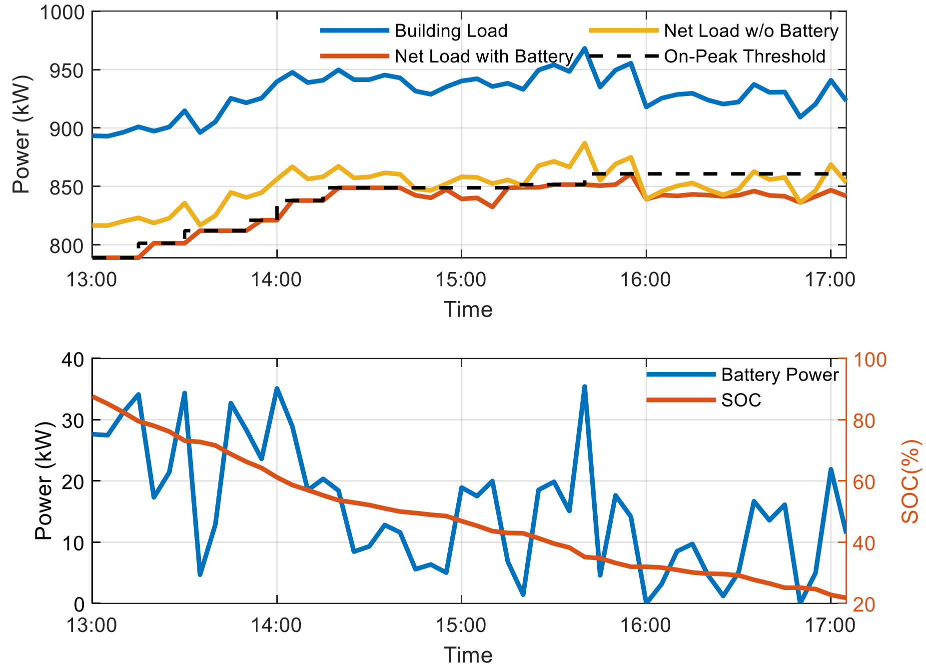


Figure 4.6: Battery operation for building load on 7/7/2014 during the summer rate period.

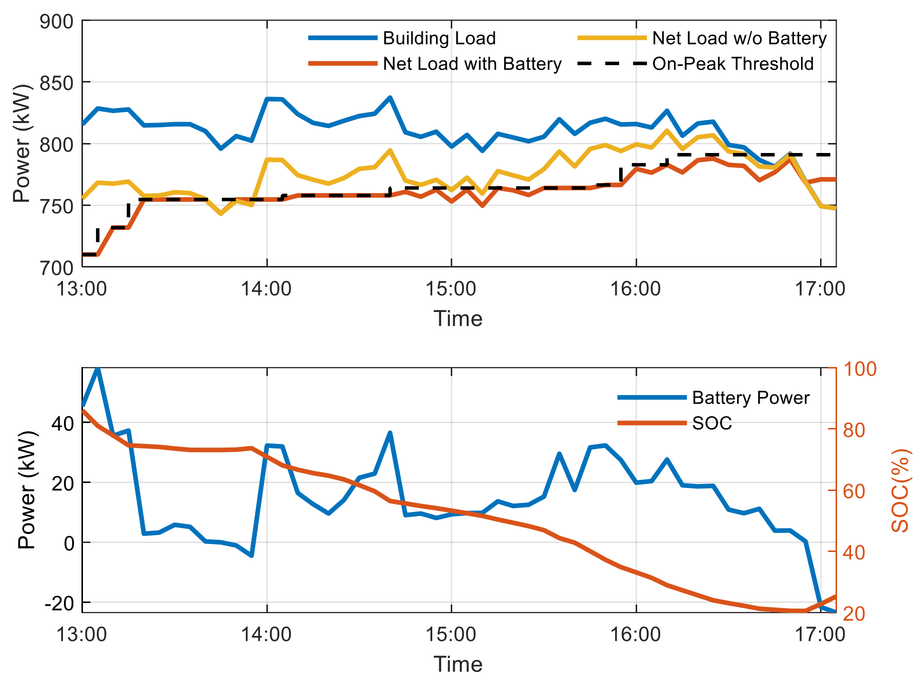


Figure 4.7: Battery operation for building load on 10/6/2014 during the winter rate period.

4.2.6 Cost Benefit Analysis

Figure 4.8 shows the simulated year-round peak demand reduction by solar and battery systems. The difference between 'Total Dollar Savings ($)' and 'Solar Savings ($)' is the dollar savings from the BESS operation. During the summer rate periods, besides the facilities demand reduction, on-peak demand is also reduced, which causes higher demand savings. The total demand savings are $19,448.01, and the BESS accounts for 37.78% of the total demand savings, up to $7346.47.

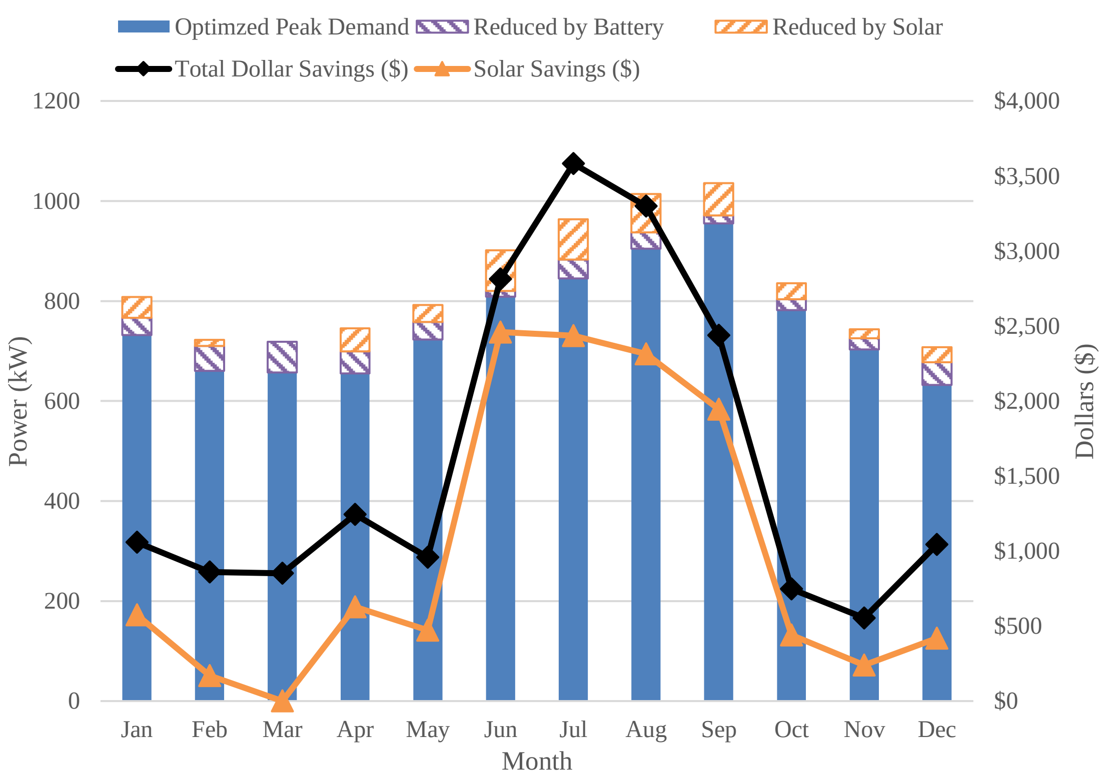


Figure 4.8: Peak demand kW and dollar savings by different components.

The energy reduction and savings are directly from the solar system, up to $13,003.22. Details of the monthly energy and demand savings are listed in Table 4.3.

Month	Energy Information			Demand Information		
	Building Energy consumption (kWh)	PV Energy Generation (kWh)	Energy Dollar Savings ($)	Building Peak Demand (kW)	Optimized Peak Demand (kW)	Demand Dollar Savings ($)
Jan	339,845	9426.87	877.64	808.24	731.7	1057.80
Feb	292,942	9056.19	843.13	722.19	660	859.47
Mar	327,595	12075.24	1124.20	718.58	657	851.04
Apr	328,830	13581.39	1264.43	745.06	655	1244.63
May	350,891	13499.4	1256.79	792.24	722.92	958.00
Jun	358,273	14155.65	1317.89	901.35	808.57	2813.09
Jul	399,201	13318.92	1239.99	963.16	845.01	3582.31
Aug	388,790	13564.89	1262.89	1013.74	904.91	3299.73
Sep	397,764	12308.28	1145.90	1035.57	955.14	2438.64
Oct	367,576	11855.97	1103.79	835.72	781.72	746.28
Nov	315,426	9154.83	852.31	743.15	703.06	554.04
Dec	314,197	7671.81	714.25	707.74	632.27	1043.00

Table 4.3: Simulated year-round electricity energy and peak demand savings.

4.3 Microgrid II: Chemehuevi Community Center (CCC) Microgrid

This work demonstrates the control strategy for a Flow Battery Energy Storage System (FBESS) in a microgrid system located at the Chemehuevi Indian Tribe in Havasu Lake, CA. Due to the intermittence of solar generation, energy storage systems have been integrated in microgrid systems for more efficiency and robustness. In late 2017, a microgrid system was installed at the Chemehuevi Community Center (CCC). The microgrid integrated a solar carport PV system and a zinc-bromine FBESS to manage energy use profiles while permitting flexibility in energy utilization by the energy management system for the community center.

4.3.1 System Architecture

The CCC microgrid consists of a 90.4 kW carport PV system, 25 kW/125 kWh flow battery system, three combiner boxes, one advanced inverter (Matrix unit), 1 AC disconnect switch and the community center building. The Matrix unit is an energy management platform, designed by EnSync Energy Systems, that integrates the Chemehuevi microgrid direct current (DC) sources (solar PV panels and flow battery) onto a common DC bus. Figure 4.9 is the actual system location, and Figure 4.10 is the overview of the system architecture and the power flow.

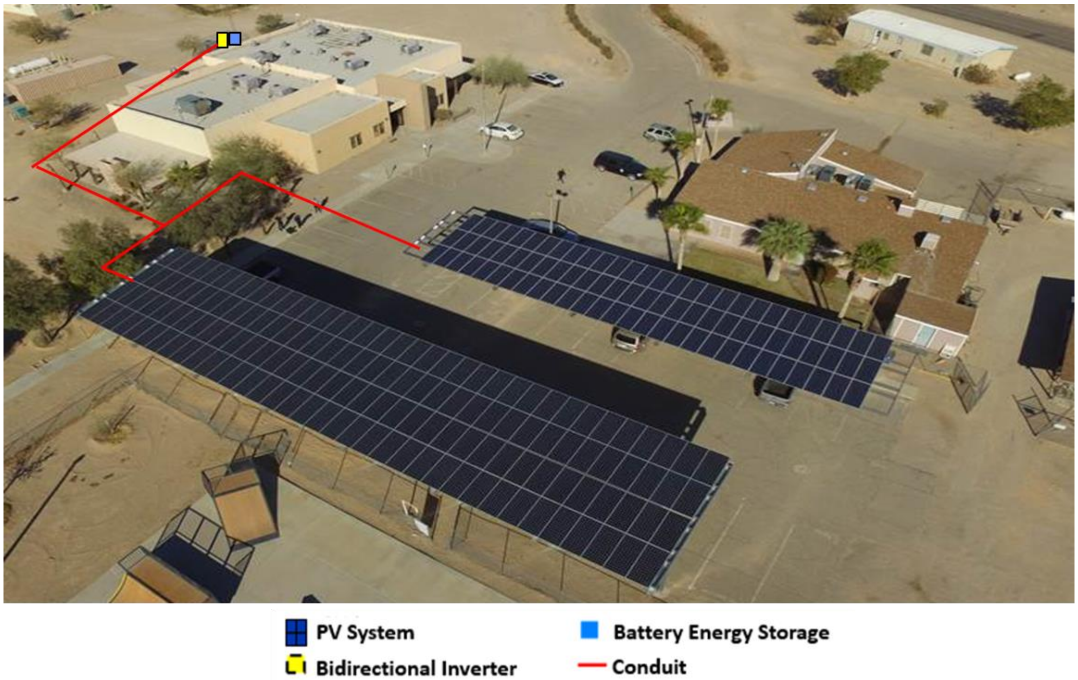


Figure 4.9: Photograph of the CCC microgrid.

As depicted in Figure 4.10, the CCC microgrid is a DC-coupled system with all DC sources (solar, battery) integrated into a single DC – DC power conversion unit at the Matrix. The Matrix unit has four DC-DC converter units. Three of the DC-DC units are connected to the solar PV system, and one is connected to the flow battery system. During the charging mode, the FBESS is directly charged from the solar PV system. Power generated from the solar PV system is divided into two parts: one directly feeds into the building loads and the Southern California Edison (SCE) grid, and the other flows to the FBESS when charging is needed. Power flowing from the solar PV system and the FBESS to the building is converted at a single DC-AC inverter. The system is connected to the main grid through the Grid Isolation Device (GID), which allows the system to operate in an island mode when there is a grid outage. When the system is operated in the island mode, the onsite 175 kW generator outputs electricity to power the lighting and critical loads in the building.

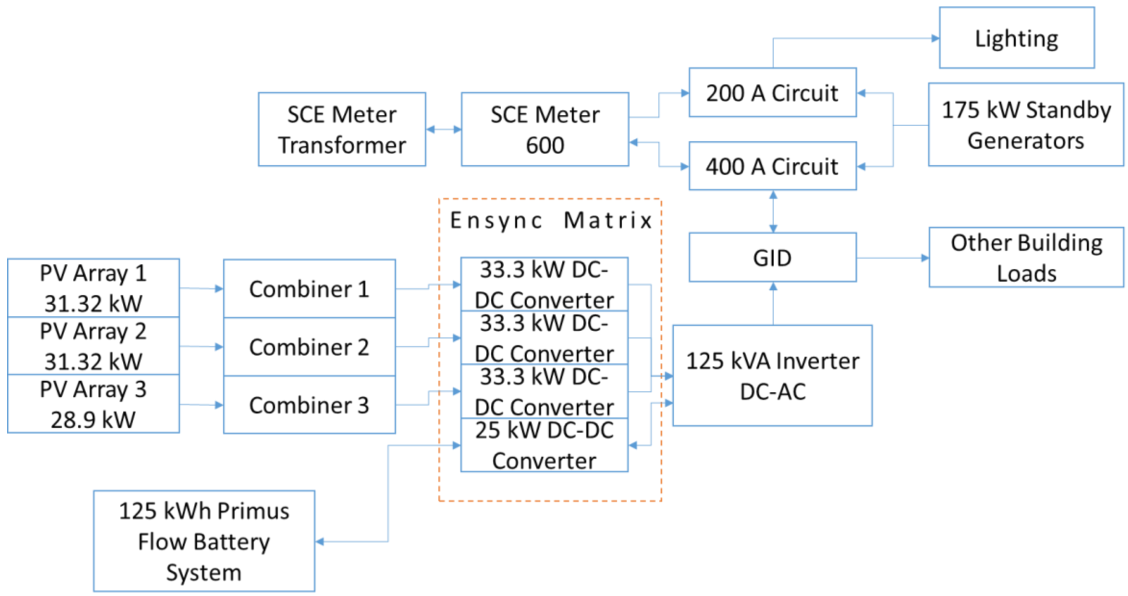


Figure 4.10: Architecture of the CCC microgrid.

The entire microgrid system information is collected and monitored by the local database, as shown in Figure 4.11. The EMS control center retrieves the historical and real-time system data such as solar, SOC, net load battery charging/discharging power, weather forecast, and demand response request signal from the local database. It then calculates the optimal battery operation power. When discharging is needed, the desired discharging power is directly sent to the FBESS from the DC – DC inverter. The desired discharging power is delivered to the building through DC – AC inverter. When charging is needed, instead of sending the charging power to the FBESS, the EMS sends the solar power minus charging power to the DC – DC inverter. In such a way, the desired DC charging power from the solar PV system can be converted at the DC-DC converter and flows to the FBESS. The total power from the FBESS and the rest of the solar PV system is converted at the DC-AC inverter and fed into the building.

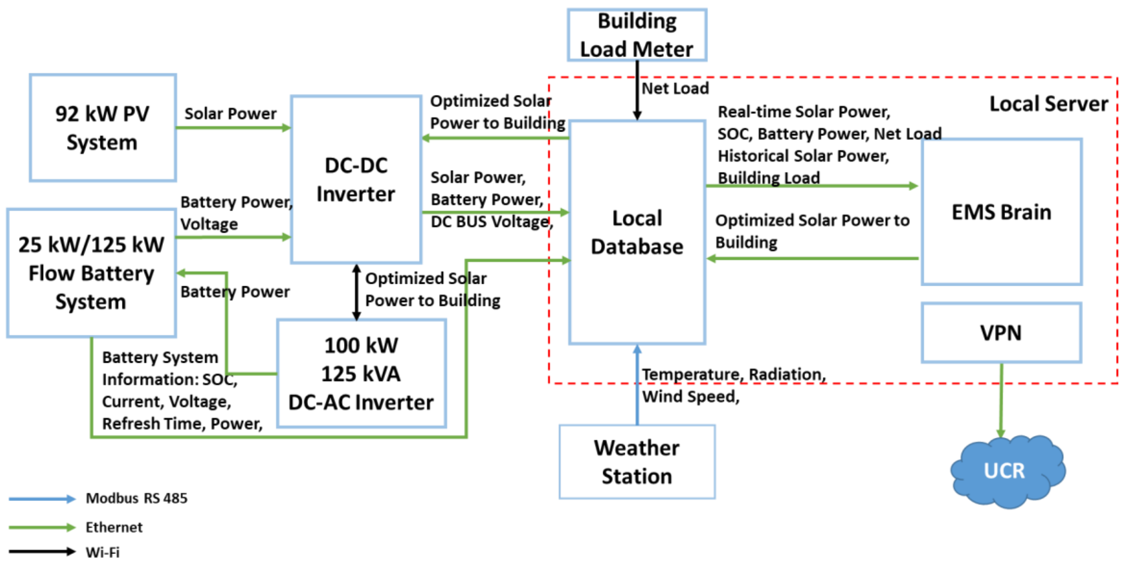


Figure 4.11: Diagram of EMS data flow for CCC microgrid.

4.3.2 Solar Generation and Building Load Consumption Characteristics

The overall monthly solar generation can be simulated with the System Advisor Model (SAM) [76] prediction model. The parameters selected in SAM are listed in Table 4.4. There are two different PV panels: SunPower SPR-E20-435-COM (144 modules) and SunPower SPR-P17-330-COM (84 modules). Since the DC-AC inverter in this project is not defined in the SAM database, parameters for the inverter datasheet are manually configured. Each type of PV modules can be considered as connected to different inverters with the same inverter parameters. PV modules at Combiner box 1 are SunPower SPR-P17-330-COM, and PV modules at Combiner boxes 2 and 3 are SunPower SPR-E20-435-COM.

Location	Havasu Lake, CA
Modules	SunPower SPR-E20-435-COM (1) SunPower SPR-P17-340-COM (2)
Inverter	Power Ratings • Maximum AC Output Power: 125 kW • Weighted Efficiency: 95 Operating Ranges • Normal AC Voltage: 208 Vac • Minimum MPPT DC Voltage: 150 Vdc • Maximum DC Voltage: 300 Vdc • Nominal DC Voltage: 300 Vdc • Maximum DC Current: 150 Adc • Maximum MPPT DC Voltage: 300 Vdc
System Design	Modules per String: 4 for (1), 5 for (2) Strings in Parallel: 36 for (1), 17 for (2)
System Power Rating	(1): 62.64 kW (2): 28.56 kW

Table 4.4: SAM parameters configuration for CCC PV system.

The simulated monthly and yearly solar generation is displayed in Figure 4.12 compared to the building electricity consumption from 2016 – 2018. Figure 4.4 (left) shows that for most months, especially Spring and Fall seasons, the onsite solar PV system can generate more energy than the building's consumption when the temperature is not high compared to summer days. Figure 4.4 (right) shows that the yearly onsite PV production generation can provide sufficient energy for the community center and account for more than 100% of electricity consumption. The over-produced solar energy can not only satisfy the FBESS, but also help the facility reduce the electricity cost, efficiently respond to the Demand Response (DR) events, and perform storage-to-grid activities.

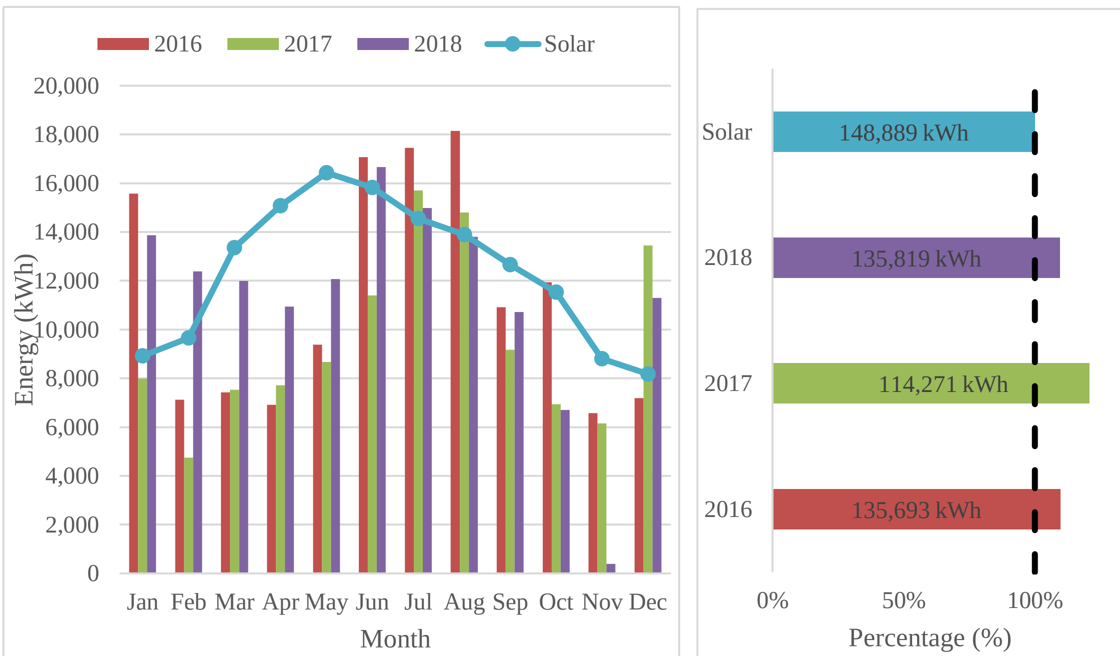


Figure 4.12: Comparison between simulated solar generation and building load consumption: left) monthly solar generation vs. building electricity consumption from 2016 – 2018; and right) yearly solar generation percentage of yearly building electricity consumption.

As shown in Figure 4.13, the impact of solar generation on the building's electricity consumption from SCE is compared, where the baseline is chosen as the average of the 3-year (2016 – 2018) building load. Since the solar PV system was operated in late January 2019, this month is replaced by data from January 2020 in Figure 4.13 to quantify full-year solar generation impacts and performance. The yearly electricity consumption with the solar PV system compensates in 2019 is net negative, with 23,547 kWh over generating by the PV system. The total consumption is reduced by 152,141kWh compared to the baseline. Consequently, the entire yearly billing amount savings is $11,042, reduced by 48.68% of the baseline billing amount.

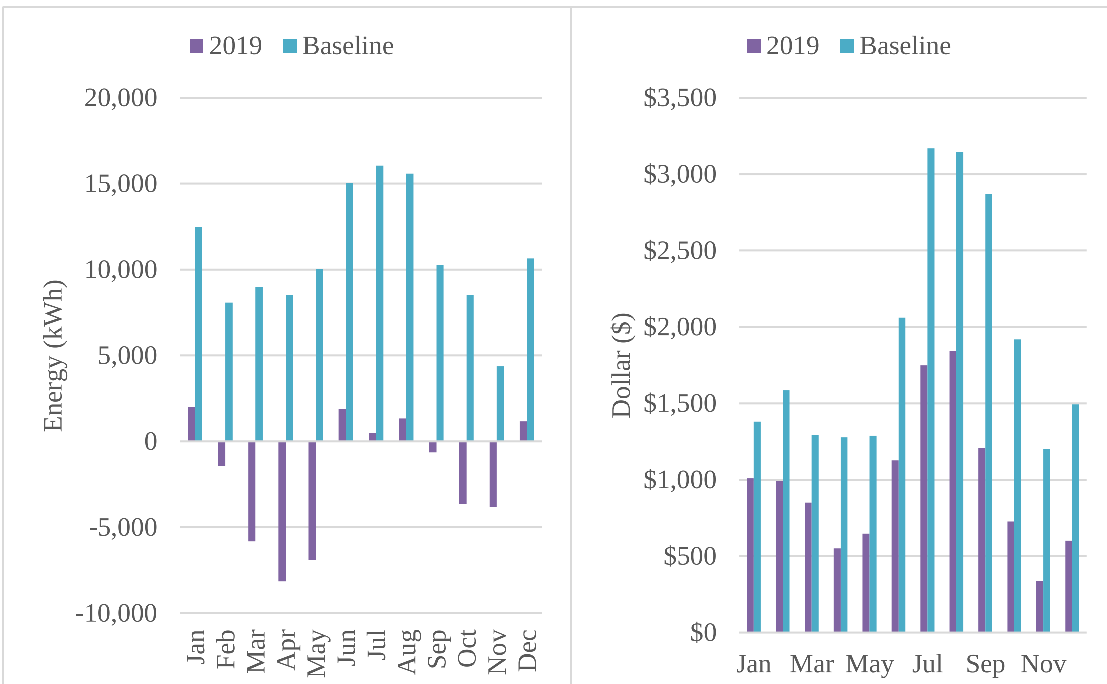


Figure 4.13: The impact of solar generation on the building's electricity consumption: left) monthly electricity consumption comparison between 2019 and baseline; right) monthly electricity billing comparison between 2019 and baseline.

4.3.3 System Control Objectives and Constraints

(I) Objectives

Over the last ten years, solar energy production has been increasing rapidly in the state of California. From 2009 to 2018, solar PV and solar thermal power plants generation have expanded from 0.41% to 13.97% of the state's total electricity production [134]. As California adopts more solar energy, energy is now less costly during midday and more costly in the late afternoon and evening [135]. Starting in March 2019 [136], the time-of-use (TOU) rate periods switched from a legacy TOU (L-TOU) shown in Figure 4.14 to an updated TOU (U-TOU) for SCE customers shown in Figure 4.15. It is found that the on-peak rate periods with the highest electricity cost shifted from 12 – 6 PM to 4 – 9 PM when the solar generation decreases.

Solar systems installed in California are eligible for the state's net metering (NEM) program [137]. NEM allows customers to get full retail rate credit for their onsite solar generation, and the credits are cumulatively added and applied to each electric bill on a monthly basis. The structure change of TOU periods inevitably decreases solar energy credits since solar is mostly credited during off-peak prices. Furthermore, changing the on-peak rate period also impacts the solar system's potential for on-peak demand reduction.

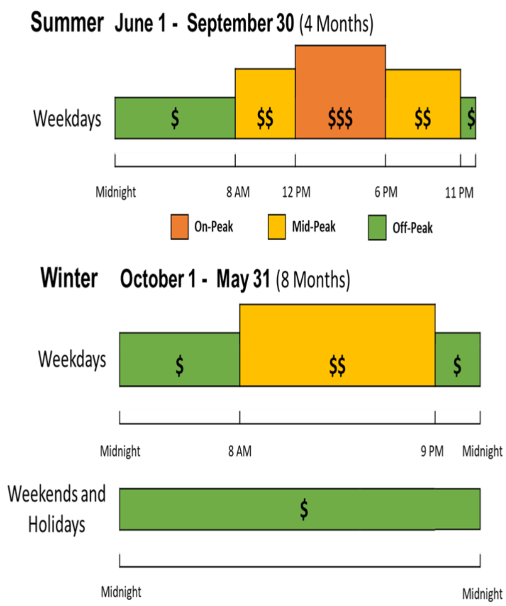


Figure 4.14: Legacy time-of-use periods.

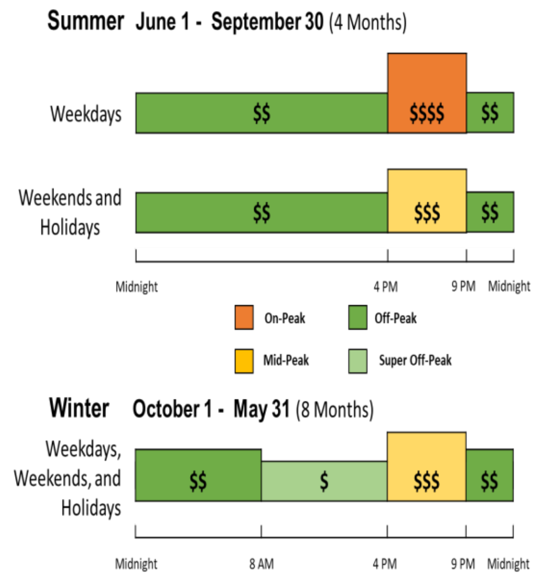


Figure 4.15: Updated time-of-use periods.

Various control algorithms have been carried out for microgrid systems consisting of solar and battery in the literature. They mainly focused on providing voltage and frequency support [138], mitigating power fluctuations [139], and reducing peak demand [133] under the L-TOU. Few papers consider the operational constraints of flow battery technology and economic benefit analysis under the U-TOU rate schedules. Thus, the objectives of the control algorithm are threefolds:

- Minimizing the electricity cost by considering the flow battery operational constraints;
- Decreasing the peak demand;
- Providing demand response support by the FBESS.

Besides, the large onsite solar generation and the characteristics of the FBESS provide a unique opportunity to perform NEM aggregation analysis for over-produced solar generation and the FBESS cost-effectiveness comparison under the U-TOU and L-TOU rate schedules.

(II) Constraints

Considering the characteristic of the specific flow battery used in this project, there are three primary battery constraints:

- The FBESS needs to be charged at a constant rate for a certain period of time during the charging process to ensure the uniform deposition of Zn on the battery electrodes.
- The FBESS needs to be fully discharged for most of the charge and discharge cycles for safety reasons and a long battery lifetime.
- The FBESS needs to perform a refresh cycle after each full deployment of discharging cycle (state of charge (SOC) = 0%). During the refresh cycle, the BMS performs maintenance on battery electrodes to ensure maximum energy storage capacity.

4.3.4 Control Algorithm Description

The flexibility of operating the flow battery is hindered by its characteristics and limitations. To take advantage of battery storage for reducing the peak demand during the on-peak/ demand response (DR) period with the battery limitations, the control strategy is separated into three modes: Before Charge, Charge, and Discharge. Each mode has its own tasks and requirements.

(I) Before Charge Mode

In this mode, the control algorithm determines the battery operation from 9 PM – 8 AM. Two different scenarios need to be addressed: (S1) the battery is fully discharged for sunny and non-DR days; and (S2) the battery maintains a high level of SOC to fulfill the requirements for DR or deficient solar generation on the following day. After the SOC drops to 0, a pre-set 2-hour duration is required for completing the refresh cycles from 8:00 to 10:00. The battery power flow can be expressed in (4.11).

$$bg = \begin{cases} \min\left(bg_{max}, \dfrac{bl - \mu_{DA} \cdot sg - dem_{sch}}{\mu_{DA}}\right), \\ \qquad\qquad if\ bl - \mu_{DA} \cdot sg - dem_{sch} > 0 \\ \text{S1: } \min(bg_{max}, \dfrac{soc_o \cdot \sqrt{\mu_{rt}} \cdot capacity}{Remain\ Time}) \\ \text{S2: } 0 \end{cases} \tag{4.11}$$

bg: power flow from the battery to the building/grid at the current time, bg_{max}: battery maximum allowed discharge power;

soc_o: the initial battery SOC;

μ_{rt}: the battery round-trip efficiency ($\mu_{rt} = 0.75$);

μ_{DA}: DC–AC inverter efficiency ($\mu_{DA} = 0.94$);

bl: the building load at the current time;

sg: power flow from the solar to the building/grid at the current time;

dem_{sch}: pre-set demand for corresponding rate period based on historical data.

(II) Charge Mode

In this mode, the control algorithm determines the battery operation from 8 AM to 4 PM, or any time beyond 4 PM, depending on the solar PV generation and DR event schedules. The control algorithm maintains the constant charge rate for a certain period (e.g., 2 hours). If solar power decreases significantly and is lower than the last charge power, a charge-must-change point occurs. dch memory is used to record the duration for keeping one constant charging power value. Only when $dch = 0 \; or \geq 2$ or charge-must-change points occur, the charge power can be recalculated by (4.12), and dch is reset to 0.

$$sb = -\min\left(0, \max\left(sb_{max}, \frac{bl - dem_{sch}}{\mu_{DA}} - so\right)\right) \tag{4.12}$$

sb: power flow from the solar to the battery at the current time;

sb_{max}: the maximum allowed charge power for the battery;

so: total solar production at the current time.

On rainy and cloudy days, if the solar generation and $sb = 0$, the control algorithm switches to Discharge Mode.

(III) Discharge Mode

In this mode, the control algorithm determines the operation for the remainder of the day after Charge Mode until midnight. The battery is optimally discharged using convex optimization by solving (4.13). During days prior to rainy/cloudy days or when DR participation is required, the battery is discharged only when the net load is higher than the pre-set value. For days during DR events, the control algorithm sets a constant discharging rate for the entire DR periods to discharge the battery to achieve the lowest energy consumption.

Equation (4.13) is developed to minimize the CCC's electricity cost while maintaining the most renewable energy usage drawn into the building. The first portion of (4.13) not only guarantees the minimal energy cost for the CCC but also limits the energy flow back to the utility grid. The second portion of (4.13) minimizes the on-peak demand cost for the entire discharging period. The battery constraints are similar to the SIGI controller and are listed in (4.14) – (4.16).

$$\min_{BG} NL^T \times Pr_e \times NL^T \cdot \Delta T + \omega \cdot Pr_d \cdot max(NL) \tag{4.13}$$

$$NL = BL - \mu_{DA} \cdot SG - BG \tag{4.14}$$

$$0 \leq BG \leq BG_{max} \tag{4.15}$$

$$soc_l \leq soc_o - \frac{1}{\sqrt{\mu_{rt}}} \cdot \Lambda \times BG \leq \mathrm{soc}_u \tag{4.16}$$

NL: net load vector for the entire Discharge Mode;

BL, SG, BG: related power vectors for the entire Discharge Mode;

Pr: rate price: Pr_e– energy price and Pr_d – demand price.

ΔT: time resolution, in our simulation, $\Delta T = 1$;

soc_l, soc_u: lower and upper bound of the operational soc during the discharge process, where $soc_l = \max{(50 - soc_o, 10)}$;

Λ: lower unit triangular matrix.

The battery control operation requires decision-making at each control time interval, which can be solved by model predictive control (MPC) [133]. At each control interval, convex optimization can be utilized to get the optimal battery discharge power vector BG_{opt} for the entire discharge period, while only the first index of BG_{opt} is taken for the discharge power bg of current time interval as (4.17).

$$bg = BG_{opt}(1) \tag{4.17}$$

The EMS receives weather forecasts and DR event notifications from external sources. For those days, the battery is only discharged when necessary where the building load is high.

$$bg = \min{(BG_{opt}(1), \max(0, BL - \mu_{DA} \cdot SG - dem_{sch}))} \tag{4.18}$$

(IV) Principles of the Control Algorithm

A 24-hour FBESS control algorithm is the combination of the charging strategies in the above three modes. The principles can be built as follows

(i) Initialization: at $k = 0$, retrieve the weather condition $rInd = \begin{cases} 0, rainy\ for\ the\ next\ 24\ hours \\ 1, sunny\ for\ the\ next\ 24\ hours \end{cases}$, demand response request signal $dr = \begin{cases} 0, no\ DR\ event\ for\ the\ next\ 24\ hours \\ 1, DR\ event\ for\ drp\ hours\ for\ the\ next\ 24\ hours \end{cases}$, and preset peak demand thresholds for on-peak, mid-peak, and facilities. The preset values are chosen as the previous peak demand for the corresponding rate periods.

(ii) At $k = j$, retrieve real-time building load bl_j, solar generation so_j, battery state of charge soc_j and demand threshold dem_{sch}. If $j > 21\ or\ j \leq 8$, "Before Charging" mode is identified, go to step (iii); else, if $j > 8 < j \leq 16$, the "Charging" mode is identified, go to step (iv); else go to step (v).

(iii) Retrieve power flow from solar to the building $sg_j = so_j$ and demand threshold $dem_{sch} = dem_{mid}$. If there is no DR event or rainy weather, the FBESS continues discharging under (S1) for the entire "Before Charging" period at $\min(bg_{max}, \frac{soc_j \cdot \sqrt{\mu_{rt}} \cdot capacity}{(9-j)\ or\ (33-j)})$ until fully discharged; if any of the two conditions are identified, the FBESS only discharges at $\max\left(0, \min\left(bg_{max}, \frac{bl_j - \mu_{DA} \cdot sg_j - dem_{sch}}{\mu_{DA}}\right)\right)$ to maintain the net load at dem_{sch}.

(iv) Set $dem_{sch} = dem_{fac}$. Charge at a constant charging rate $sb_j = sb_c$ kW for dch time. When $bl_j - so_j - \mu_{DA} \cdot sb_j > dem_{sch}$ (high net load occurs) and $dch > 2\ or\ dch = 0$, change sb_j based on (4.12), else maintain the charging rate at sb_{j-1}.

When the $so_j > -sb_{j-1}$, a charge-must-change point occurs, reduce charging power based on (4.12) or switch to 'Discharging' mode when the weather is rainy.

(v) Set $dem_{sch} = dem_{on}$. If DR is identified, discharge during DR period at the maximum discharging rate $\frac{(soc_{17}-soc_l)\cdot\sqrt{\mu_{rt}}\cdot C}{drp}$; else, predict the solar generation and building load for $\widehat{SO}(j:21)$ and $\widehat{BL}(j:21)$, and calculate the optimal discharging rate from (4.13) – (4.16).

(vi) Set $k = k + 1$, go back to (ii).

The overall flowchart of the FBESS control algorithm is shown in **Error! Reference source not found.**.

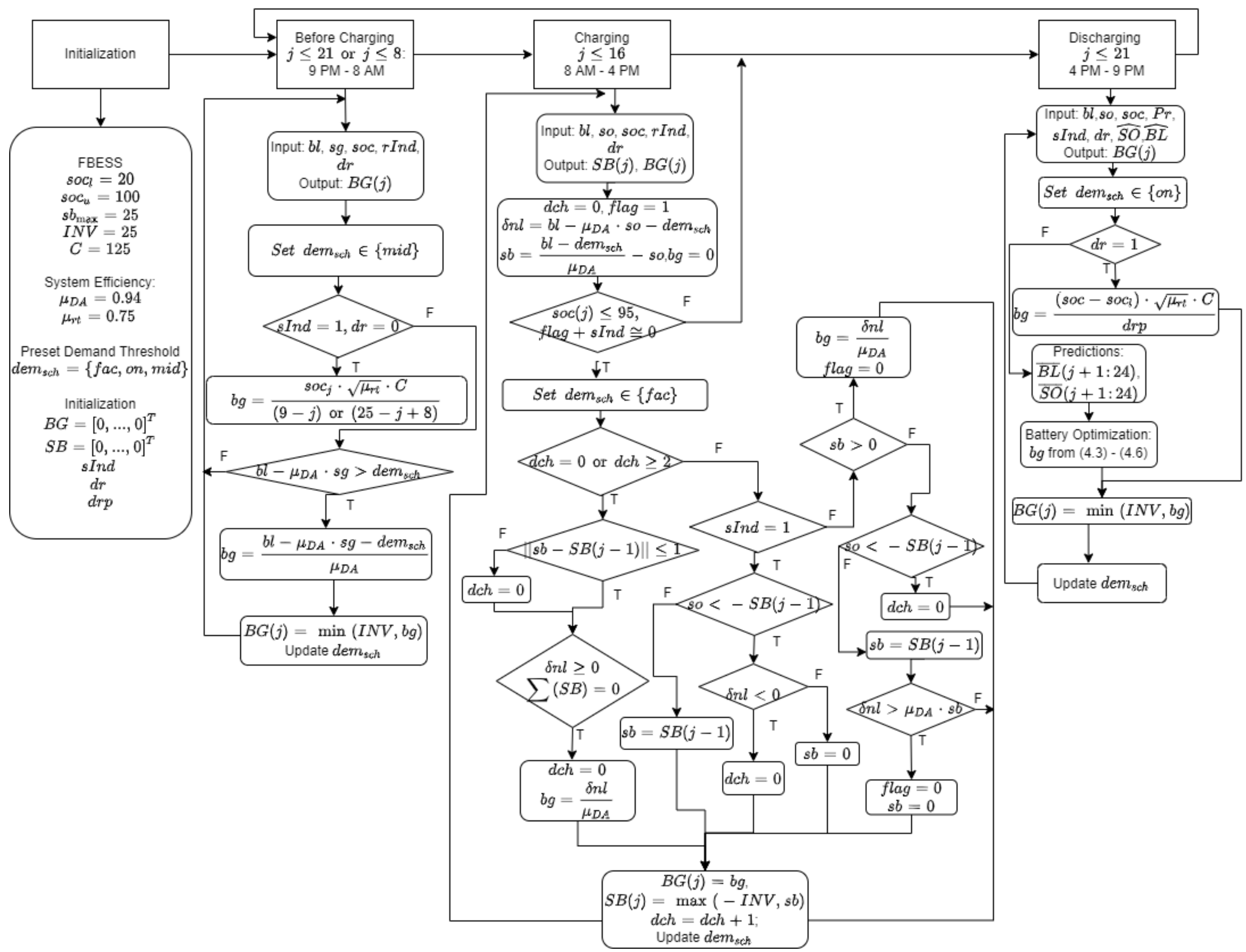


Figure 4.16: The flow chart of the battery control algorithm.

4.3.5 Simulation Results

The developed MPC algorithm consists of a predictive model for the building load and solar generation. In this simulation, the load forecast model is developed via neural networks. The features of the training sets include hourly time-series building load, day pattern (Sunday to Saturday are coded as 1 – 7 and national holidays as 0), and ambient temperature data from the CCC site. Solar production gradually decreases after 16:00, and the characteristics of solar power are relatively stable. The solar prediction model can be simplified based on the demonstration indicated in [133].

Figure 4.17 is the battery operation for a regular weekday. From 12 AM – 8 AM, the FBESS works in the "Before Charging" mode. After the battery is fully deployed around 8 AM, the FBESS goes into "Refresh" mode. Performance maintenance is scheduled to remove the full residual plating from the electrodes. Given sufficient solar production from 10 AM– 4 PM, the FBESS maintains the constant maximum charge rate at 20 kW, and the SOC reaches 83.14%. The FBESS switches to "Discharge" mode at 4 PM, and the discharge power is optimized following (4.13) – (4.16). After 9 PM, the FBESS returns to the "Before Charge" mode, and it uniformly distributes the power to the building and prevents any high peak following (4.11) during this period. The original on-peak demand for the building itself is 25.6 kW at 16:00. Although the solar PV system still provides sufficient generation simultaneously, the on-peak demand can only reduce to 22.4 kW at 7

PM after sunsets for the solar alone system. In contrast, the on-peak demand further reduces to 3.48 kW with the help of the FBESS.

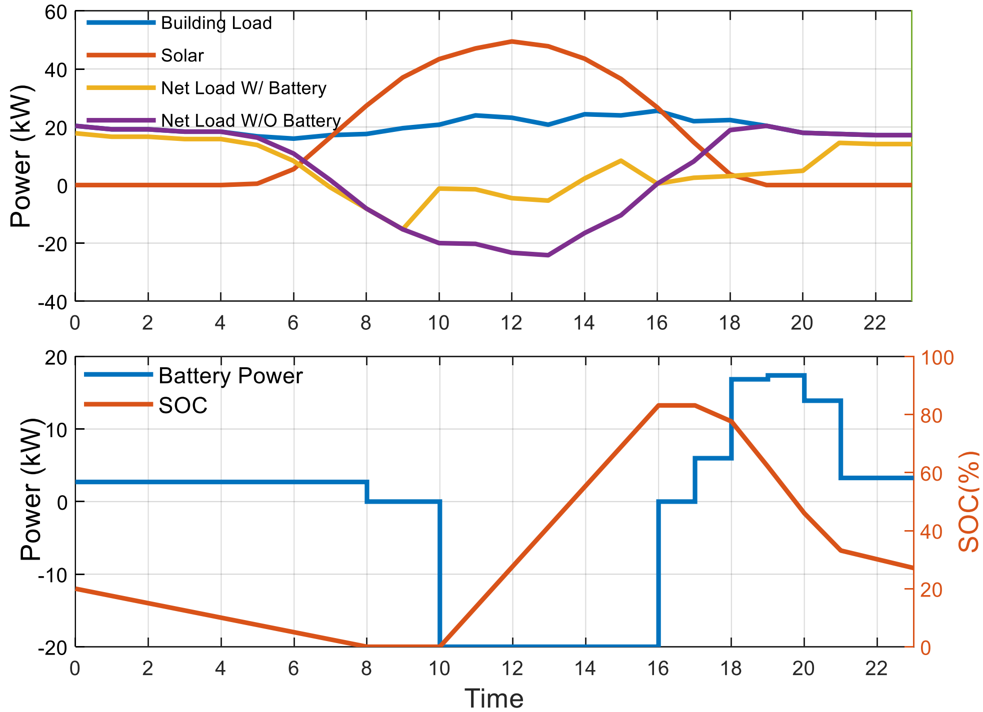


Figure 4.17: Simulation of battery operation on a regular weekday.

A single-day DR operation is simulated in Figure 4.18 when the DR period is from 4 PM – 9 PM on 7/6/2017. The FBESS is pre-charged to full capacity in advance (day-ahead) and continuously delivers 17.32 kW to the building during the DR periods.

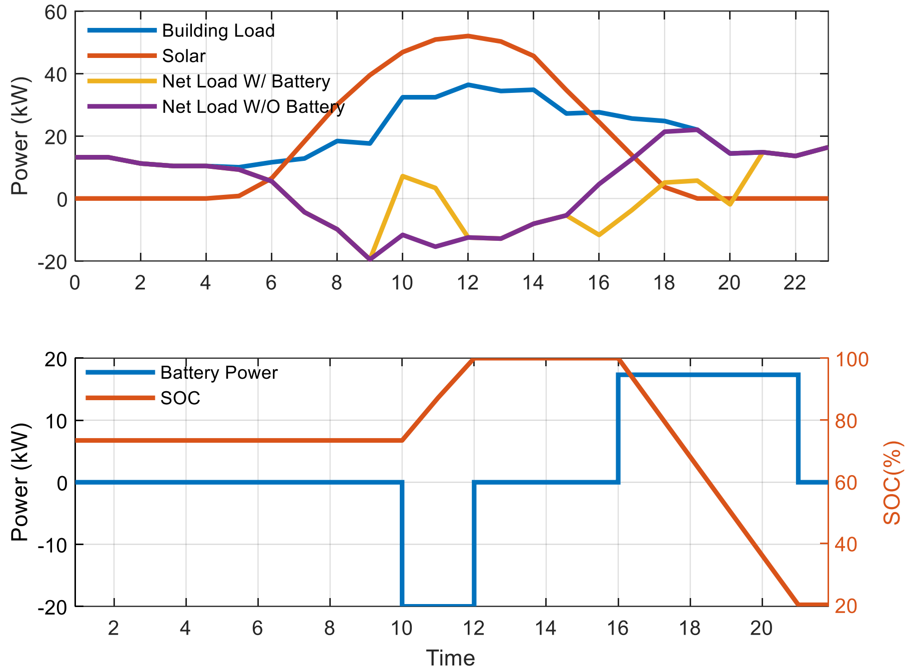


Figure 4.18: Simulation for single DR event from 4 PM – 9 PM.

In Figure 4.19, the DR events happen on two consecutive days (7/17/2017 and 7/18/2017). On the first day, the FBESS is operated the same as Figure 4.18. The battery stays idle during the first "Before Charging" mode and maintains a high SOC. During the first "Charging" mode, the battery only takes one hour to be fully charged ($soc \geq 95\%$) and waits for the DR event. At 4 PM, the FBESS continuously discharges at 16.24 kW. SOC drops at 20% at the end of the first DR event. During the second "Before Charging" mode, two small discharging operations maintain the facility's demand below the preset demand threshold. During the second "Charging" mode, the FBESS charges at the maximal rate and takes more than four hours to full capacity. When the second DR event comes, the FBESS switches to "Discharge" mode and continuously delivers 17.32 kW to the building.

Thanks to sufficient on-site solar generation, the FBESS can be fully charged before each DR event.

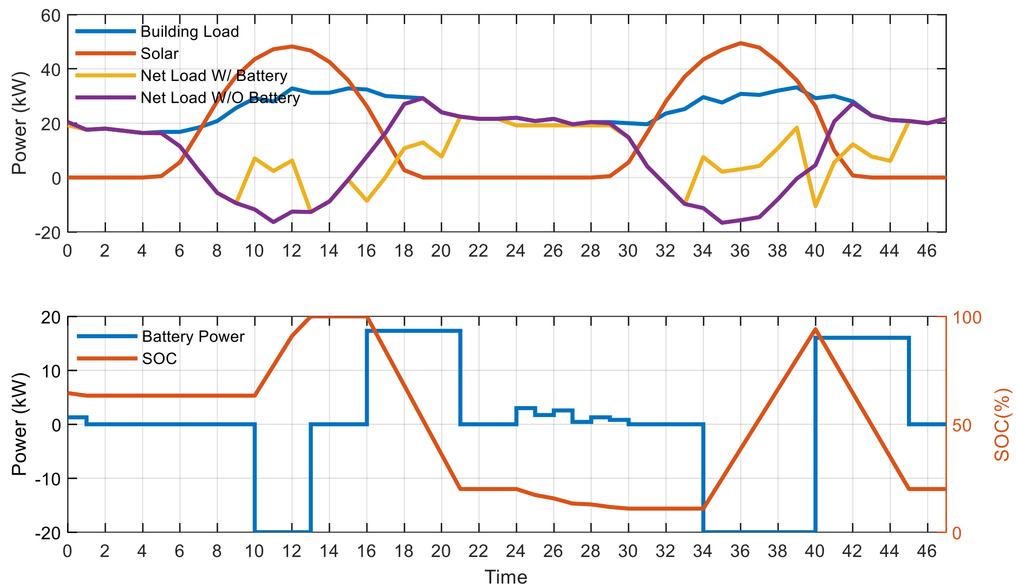


Figure 4.19: Battery operation for two consecutive DR events.

In Figure 4.20, one month of battery operations is simulated for July 2017. In this month, five DR events (the online database only records the DR events for the CCC after 2018. For simulation purposes, we incorporate the DR events in 2018 to our test data of 2017) are simulated based on the SCE DR history database [140]. Most days for this month are sunny. The FBESS performs full charge and discharge cycles for those days. When solar generation is insufficient (7/15) or DR events are called (marked in the dashed boxes), the FBESS maintains a higher SOC during previous day operation to be accessible for higher energy capacity during "Discharge" mode. The on-peak demand for the stand-alone community center of this month is 36.4 kW. After being integrated with the solar system, the on-peak demand only reduces to 30.9 kW. With implementing the proposed flow battery control strategy, the on-peak demand reduces further to 16 kW.

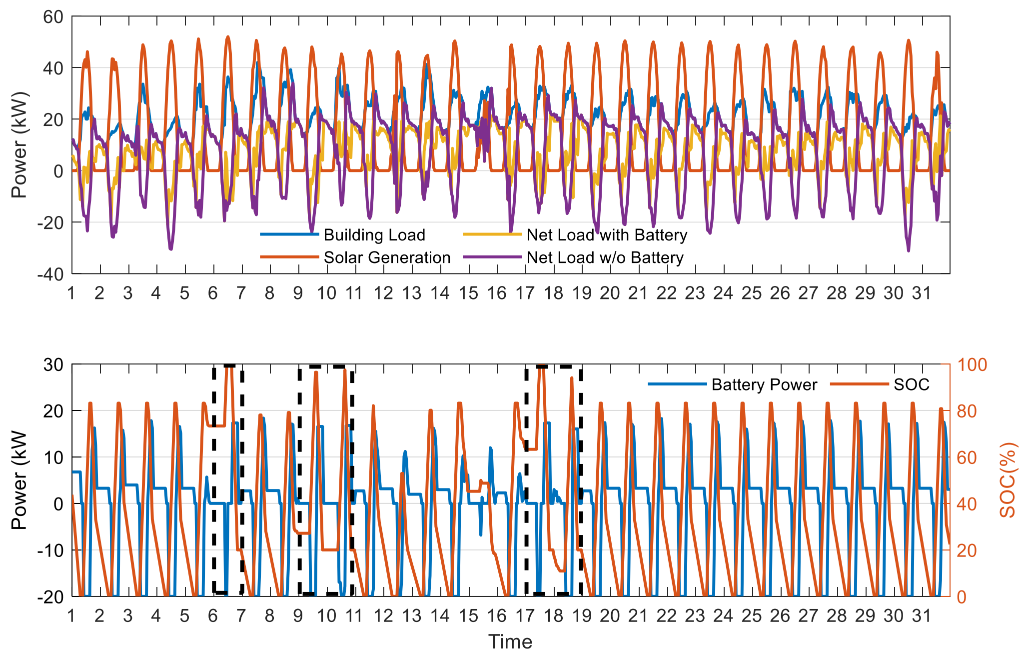


Figure 4.20: Simulation of battery operation for an entire month

4.3.6 Cost Benefit Analysis

To maximize the electric bill savings for the CCC, the system owner can choose the appropriate rate schedule and enroll in different programs provided by the utility company considering the microgrid structure. This section assesses the year-round cost savings for different rate plans and compares the economic benefit difference between the flow battery control operation under the U-TOU and L-TOU periods.

(I) Demand Response Program

DR programs offer incentives to reduce electricity use when the demand for the utility territory is high [141]. Critical peak pricing (CPP) is one of the SCE's DR programs. The CPP events are called year-round (total of 12 events) on a day-ahead basis. For rate plans of TOU-GS-2, CPP has a significantly higher energy price from 16:00 – 21:00, in exchange for a discount on summer on-peak demand charge. To maximize the program's incentives, the energy storage can contribute to the reduction of energy consumption from the utility during the 5-hour events. The on-peak electric bill charges for the summer months (June – September) of 2017 are listed in Table 4.5, calculated based on the CCC's actual rate schedule: TOU-GS-2-D [142].

Different Rate Schedule		**Different System Architectures**		
		With FBESS	***Without FBESS***	***Building Alone***
CPP	Energy	\$167.41	\$873.37	\$1,411.29
	Demand	\$1,323.14	\$2,746.93	\$3,246.86
	Total	\$1,490.55	\$3,620.30	\$4,658.15
Non CPP	Energy	\$187.86	\$637.41	\$1,058.47
	Demand	\$1,518.99	\$3,147.38	\$3,720.19
	Total	\$1,706.85	\$3,784.79	\$4,778.66

Table 4.5: Comparison of electric bill cost for summer on-peak period between different system architectures.

(II) Meter Aggregation Program

In 2012, NEM Aggregation (NEM-A) was approved, which allows an eligible customer-generator to aggregate the electrical load from multiple meters in the same property [137]. There are seven subaccounts for the Chemehuevi administrative electrical account, and each subaccount has one unique ID number and different rate schedules. The credits of solar production from the generating account (the community center) can be proportionally distributed to other benefiting accounts based on the generation credit allocation methodology [143].

Equation (4.19) finds the optimal set of allocated accounts $\mathcal{N}$ to maximize the energy savings for the property. There are total $\sum_{n=1}^{6} C(6,n) = 63$ different sets.

$$\min_{\mathcal{N}} \sum_{i\in\mathcal{N}} \sum_{j=1}^{4} \left(E_{i,j} - A_{i,j}\right) \cdot TOU_{i,j} - C \cdot n\,, 1 \in \mathcal{N} \tag{4.19}$$

i: account ID;

j: rate period: 1 – super off-peak, 2 – off-peak, 3 – mid-peak, 4 – on-peak;

$\mathcal{N}$: set of aggregated account IDs, which always includes account 1;

$E_{i,j}$: energy consumption of account i for rate period j;

$A_{i,j}$: allocated energy of account i for rate period j;

$TOU_{i,j}$: TOU electricity price of account i for rate period j;

C: fix charges for single meter aggregation;

n: number of accounts in $\mathcal{N}$.

Figure 4.21 demonstrates the energy savings comparison between systems with and without the FBESS for different aggregation accounts. Each bar represents the highest year-round savings of aggregated accounts through all combinations of $\mathcal{N}(n)$. Calculations are based on the actual rate schedules for those accounts. The system without the FBESS can allocate the entire excessive solar generation to other accounts. While the system with the battery requires the solar PV system to first achieve the highest savings for the generating account, with restoring a portion of the solar energy to the FBESS to offset on-peak energy consumption and demand reduction. The highest aggregation savings can be achieved when $\mathcal{N}_{opt} = \{1,2,3,4\}$ without the FBESS, and $\mathcal{N}_{opt} = \{1,2,4\}$ with the FBESS.

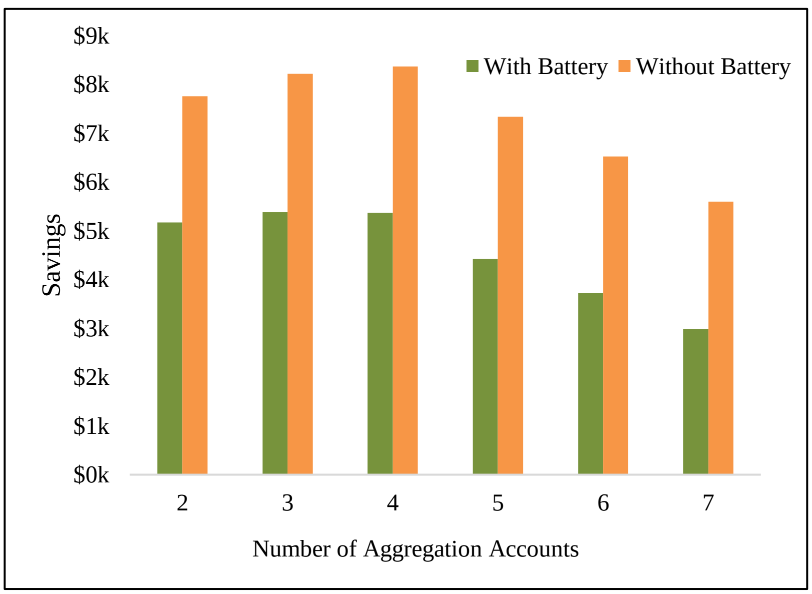


Figure 4.21: Maximum electricity cost savings for the different number of aggregated accounts.

Table 4.6 exhibits the details of the economic benefit before and after meter aggregation. The overproduced solar energy without NEM-A can be compensated by the Net Surplus Compensation Rate (NSCR) at a significantly low price [144].

	Without NEM-A		With NEM-A	
	With FBESS	***Without FBESS***	***With FBESS***	***Without FBESS***
Energy Savings for Account 1	$10,283.23	$10,545.00	$6,945.10	$4,342.45
Demand Savings	$3923.53	$1,276.81	$4,203.97	$1,349.69
Energy Savings for Accounts 2-7	0	0	$5,381.14	$8,371.66
Total Savings	$14,206.8	$11,821.8	$16,530.2	$14,045.8

Table 4.6: Cost savings comparison between NEM and NEM-A.

(III) Comparison between the Cost Savings of Legacy and Updated TOU Periods

This section evaluates the economic benefit under the different architectures of the microgrid system: varied solar PV system and the existence of the FBESS. Figure 4.22 quantifies the difference between different microgrid systems architectures: (a) the actual solar PV size and (b) 50% of the actual solar PV size. The calculation is based on the actual rate plan for the CCC, which is TOU-GS-2-B [142] for the L-TOU periods and TOU-GS-2-D for the U-TOU periods (to eliminate the deviation of the simulated DR events and be consistent with the grandfathered rate plan, TOU-GS-2-D is used to calculate the bill). Due to sufficient onsite solar generation, energy savings always account for a higher ratio than the demand reduction savings for all system architectures. Energy savings are always higher for the system without the FBESS when comparing to systems with the FBESS due to the FBESS's low round-trip efficiency.

In Figure 4.22 (a), the total savings are always higher with the FBESS. Furthermore, the FBESS can contribute to more than 67.7% of the total demand reduction savings and 22.2% of the total savings for the U-TOU periods compared to 50.5% and 13.2% for the L-TOU. The highest savings can be achieved for the system with the FBESS under the U-TOU periods. Figure 4.22 (b) shows that the demand savings portion increases with a smaller solar system. It is recognizable that the importance of the FBESS under the U-TOU periods raises significantly, where the savings ratio from the FBESS accounts for 34.9% of the total savings. For the L-TOU periods, although the FBESS can improve the demand reduction savings, the operational constraints and low round-trip efficiency of the FBESS diminish the total savings. From comparing different sizes of the solar PV system, the FBESS leads to increased cost savings under the U-TOU periods, while the benefit from the solar PV system decreases. In addition, the microgrid system integrating with the solar PV and the FBESS can consistently achieve the highest savings under the U-TOU periods among different system architectures or rate periods.

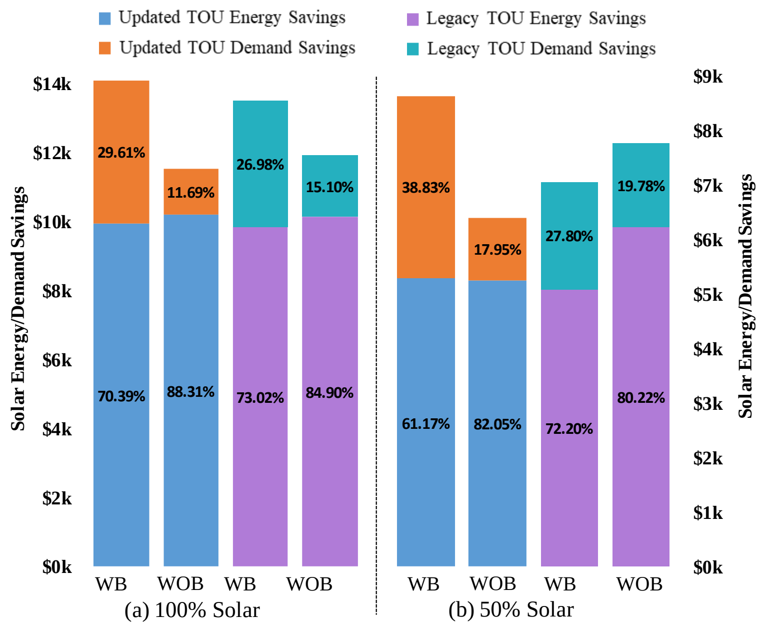


Figure 4.22: Comparison between the yearly energy and demand cost savings ratio of the U-TOU and L-TOU.

4.4 Conclusion

In this chapter, two different microgrids with various system architecture and constraints are discussed. In Section 4.2, a microgrid consists of a large building consumption and small battery storage and solar generation. In Section 4.3, a microgrid consists of a flow battery and sufficient solar PV generation.

For the RCCH microgrid, the optimal battery control algorithm is proposed for decreasing the building's peak demand with limited battery storage. Due to the small capacity of the BESS, it is impossible to deliver energy from the BESS during the entire on-peak rate period. By identifying the discharging period to 1 PM – 5 PM when the peak load happens the most, the BESS can achieve 37.78% of the total demand savings, around $7346.47 per year.

For the CCC microgrid, the optimal battery control strategy is proposed for the flow battery aiming at handling the unique battery operational constraints and reducing the peak demand under U-TOU periods. With the flexibility of the microgrid, the owner can achieve the highest electric bill savings by participating in different rate plans such as DR and NEM-A. The energy storage can contribute to the lowest energy consumption during the DR events. The NEM-A provides the opportunity to customers, who have an overproduction generating system and multiple accounts, to further reduce their electric bill by allocating the surplus energy to other aggregated accounts. Furthermore, we evaluate the economic benefits from the electric bill under the U-TOU and L-TOU periods. It is clear that if the system transitions to the U-TOU periods, the energy storage system can significantly contribute to more cost savings and take a more critical role in reducing the total electric bill than the L-TOU periods.

Chapter 5 Conclusions and Future Work

5.1 Conclusions

In this dissertation, comprehensive studies and algorithms, including prediction, time series analysis, and energy scheduling within a microgrid system, are carried out.

In Chapter 2, the FL-AELM algorithm is developed for solar power generation and electric vehicle charging load prediction. The FL-AELM can online update the model's weights with the newly available data, assign less importance to old training data (outdated) by the forgetting parameters, and tune the hidden layer inputs to a more linear regime by the affine transformation. The numeric prediction performance comparison with other state-of-the-art machine learning algorithms is carried out. The FL-AELM algorithm can increase short-term prediction accuracy for solar generation and EV charging power over 8% and 13%, respectively.

In Chapter 3, the unique tail feature from the EV charging time series is introduced. The tail feature can classify the diverse charging profiles into seven clusters via k-medoids with the novel RMPS distance. The RMPS is robust to time series noise and time series lengths yet computationally efficient. Comprehensive classification evaluations to compare and validate the accuracy of the proposed RMPS are carried out in a combination of the internal/external evaluation indices and physical-based error matrix CCR. The classification performance of the RMPS distance measure surpasses other state-of-art distance measures, offering a more reliable classification performance. Chapter 3 further proposes the centralized and distributed EV smart charging algorithms. Combining the learned tail templates into the centralized and distributed EV smart charging optimization, an average of 4% - 5% flexible rate or 0.7% satisfactory rate improvement can be achieved.

In Chapter 4, two real-world solar + battery microgrid systems are presented. The proposed battery control algorithm for the RCCH project aims at minimizing the demand charges costs with the limited BESS capacity. At the same time, the algorithm for the CCC project aims at minimizing the electricity cost by considering the operational constraints of the FBESS. With the help of the BESSs, the peak demand savings can reduce by 37.78%, 67.7%, and 34.9% for RCCH, CCC (100% solar), and CCC (50% solar) under the U-TOU, respectively.

5.2 Selected Publications

[1] **Y. Xue**, H. Nazaripouya, A. A. Alfredo Martinez-Morales, “Classification of Electric Vehicle Charging Tail Time Series by Refine Matrix Profile Similarity”, under preparation.

[2] **Y. Xue**, M. Penchev, H. Xin, and A. A. Martinez-Morales, "Flow Battery Control Strategy Implementation and Cost Benefit Analysis for Microgrid under Updated Time-of-Use Periods", 2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Washington, DC, USA, 2020, pp. 1-5, doi: 10.1109/ISGT45199.2020.9087673.

[3] **Y. Xue**, M. Todd, S. Ula, M. J. Barth, and A. A. Martinez-Morales, "Field Implementation of a Real-time Battery Control Scheme for a Microgrid at the University of California, Riverside", 2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia), Chengdu, China, 2019, pp. 2859-2864. doi: 10.1109/ISGT-Asia.2019.8881076

[4] **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", 2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC), Portland, OR, 2016, pp. 1875-1880. doi: 10.1109/PVSC.2016.7749947

5.3 Future work

As discussed in Chapter 2, the FL-AELM is implemented for one-step-ahead prediction and has achieved satisfying prediction performance. However, being able to implement multiple steps ahead prediction can further increase the microgrid operation stability. In future work, an elaborated multi-step-ahead prediction model can be developed. Besides, the real-time EV charging load prediction considering the tail features can also be developed to further increase prediction accuracy.

As discussed in Section 3.2, the purpose of classification is to cluster tails into small groups to find possible representatives. However, more detailed clusters can be generated from the base groups, for example, subgroups with different maximum charging power. In such a way, Figure 3.17 can be more accurate of different EV battery capacity distribution, thus possibly making tails one unique feature EV types (brands, capacity, etc.).

As discussed in Section 3.3.2, the adaptive adjusting of the average base power algorithm can be further improved by machine learning or reinforcement learning techniques. Instead of a fixed value of the change power Δp, Δp can be represented as $\Delta p \sim ANN(t, \theta)$.

BIBLIOGRAPHY

[1] J. BUZZ, “The Power2Swith Blog.” https://power2switch.com/blog/how-electricity-grew-up-a-brief-history-of-the-electrical-grid/ (accessed Jul. 14, 2021).

[2] EPA (United States Environmental Protection Agency), “Centralized Generation of Electricity and its Impacts on the Environment.” https://www.epa.gov/energy/centralized-generation-electricity-and-its-impacts-environment (accessed Jul. 14, 2021).

[3] ProCon.org, “Historical Timeline - History of Alternative Energy and Fossil Fuels,” *ProCon.org*, 2020. https://alternativeenergy.procon.org/historical-timeline/ (accessed Jul. 14, 2021).

[4] REN21, “Renewables 2017 Global Status Report,” May 2017. Accessed: Aug. 01, 2018. [Online]. Available: http://www.ren21.net/gsr-2017/chapters/chapter_03/chapter_03/.

[5] R. H. Lasseter and P. Paigi, “Microgrid: a conceptual solution,” in *2004 IEEE 35th Annual Power Electronics Specialists Conference (IEEE Cat. No.04CH37551)*, pp. 4285–4290, doi: 10.1109/PESC.2004.1354758.

[6] W. Feng *et al.*, “A review of microgrid development in the United States – A decade of progress on policies, demonstrations, controls, and software tools,” *Appl. Energy*, vol. 228, pp. 1656–1668, Oct. 2018, doi: 10.1016/j.apenergy.2018.06.096.

[7] L. Wen, K. Zhou, S. Yang, and X. Lu, “Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting,” *Energy*, vol. 171, pp. 1053–1065, Mar. 2019, doi: 10.1016/J.ENERGY.2019.01.075.

[8] R. Rahmani, I. Moser, and M. Seyedmahmoudian, “Multi-agent based operational cost and inconvenience optimization of PV-based microgrid,” *Sol. Energy*, vol. 150, pp. 177–191, Jul. 2017, doi: 10.1016/J.SOLENER.2017.04.019.

[9] S. Sobri, S. Koohi-Kamali, and N. A. Rahim, “Solar photovoltaic generation forecasting methods: A review,” *Energy Convers. Manag.*, vol. 156, pp. 459–497, Jan. 2018, doi: 10.1016/J.ENCONMAN.2017.11.019.

[10] S. Al-Dahidi, O. Ayadi, M. Alrbai, and J. Adeeb, “Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction,” *IEEE Access*, vol. 7, pp. 81741–81758, 2019, doi: 10.1109/ACCESS.2019.2923905.

[11] Z. Yang, M. Mourshed, K. Liu, X. Xu, and S. Feng, "A novel competitive swarm optimized RBF neural network model for short-term solar power generation forecasting," *Neurocomputing*, vol. 397, pp. 415–421, Jul. 2020, doi: 10.1016/J.NEUCOM.2019.09.110.

[12] Y. Wang, B. Feng, Q. S. Hua, and L. Sun, "Short-term solar power forecasting: A combined long short-term memory and gaussian process regression method," *Sustain.*, vol. 13, no. 7, p. 3665, Mar. 2021, doi: 10.3390/su13073665.

[13] A. Dahl and E. V. Bonilla, "Grouped Gaussian processes for solar power prediction," *Mach. Learn.*, vol. 108, no. 8–9, pp. 1287–1306, Sep. 2019, doi: 10.1007/S10994-019-05808-Z/FIGURES/4.

[14] A. Hammer *et al.*, "Solar energy assessment using remote sensing technologies," *Remote Sens. Environ.*, vol. 86, no. 3, pp. 423–432, Aug. 2003, doi: 10.1016/S0034-4257(03)00083-X.

[15] A. Catalina, C. M. Alaiz, and J. R. Dorronsoro, "Combining Numerical Weather Predictions and Satellite Data for PV Energy Nowcasting," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1930–1937, Jul. 2020, doi: 10.1109/TSTE.2019.2946621.

[16] C. Cui, Y. Zou, L. Wei, and Y. Wang, "Evaluating combination models of solar irradiance on inclined surfaces and forecasting photovoltaic power generation," *IET Smart Grid*, vol. 2, no. 1, pp. 123–130, Mar. 2019, doi: 10.1049/IET-STG.2018.0110.

[17] P. Lara-Benítez, M. Carranza-García, J. M. Luna-Romera, and J. C. Riquelme, "Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting," *Appl. Sci. 2020, Vol. 10, Page 2322*, vol. 10, no. 7, p. 2322, Mar. 2020, doi: 10.3390/APP10072322.

[18] S. Maldonado, A. González, and S. Crone, "Automatic time series analysis for electric load forecasting via support vector regression," *Appl. Soft Comput.*, vol. 83, p. 105616, Oct. 2019, doi: 10.1016/J.ASOC.2019.105616.

[19] J. Zhu, Z. Yang, Y. Guo, J. Zhang, and H. Yang, "Short-term load forecasting for electric vehicle charging stations based on deep learning approaches," *Appl. Sci.*, vol. 9, no. 9, 2019, doi: 10.3390/app9091723.

[20] M. S. Islam, N. Mithulananthan, and D. Q. Hung, "A Day-Ahead Forecasting Model for Probabilistic EV Charging Loads at Business Premises," *IEEE Trans. Sustain. Energy*, vol. 9, no. 2, pp. 741–753, Apr. 2018, doi: 10.1109/TSTE.2017.2759781.

[21] Z. J. Lee, T. Li, and S. H. Low, "ACN-Data: Analysis and applications of an open

EV charging dataset," *e-Energy 2019 - Proc. 10th ACM Int. Conf. Futur. Energy Syst.*, pp. 139–149, 2019, doi: 10.1145/3307772.3328313.

[22] X. Wang, Y. Nie, K. W. E. Cheng, and J. Mei, "Forecast of urban EV charging load and smart control concerning uncertainties," in *2016 International Symposium on Electrical Engineering (ISEE)*, Dec. 2016, pp. 1–7, doi: 10.1109/EENG.2016.7846358.

[23] G. Huang, Q. Zhu, and C. Siew, "Extreme Learning Machine : A New Learning Scheme of Feedforward Neural Networks," *IEEE Int. Jt. Conf. Neural Networks*, vol. 2, pp. 985–990, 2004, doi: 10.1109/IJCNN.2004.1380068.

[24] N. Liang, G. Huang, S. Member, P. Saratchandran, S. Member, and N. Sundararajan, "A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks," vol. 17, no. 6, pp. 1411–1423, 2006.

[25] K. Chen, Q. Lv, Y. Lu, and Y. Dou, "Robust regularized extreme learning machine for regression using iteratively reweighted least squares," vol. 230, no. June 2016, pp. 345–358, 2017, doi: 10.1016/j.neucom.2016.12.029.

[26] W. Deng, Q. Zheng, and L. Chen, "Regularized Extreme Learning Machine," *2009 IEEE Symp. Comput. Intell. Data Min.*, no. 60825202, pp. 389–395, 2009, doi: 10.1109/CIDM.2009.4938676.

[27] N. Liu and H. Wang, "Ensemble based extreme learning machine," *IEEE Signal Process. Lett.*, vol. 17, no. 8, pp. 754–757, 2010, doi: 10.1109/LSP.2010.2053356.

[28] G. Feng, G. Huang, Q. Lin, and R. Gay, "Error Minimized Extreme Learning Machine With Growth of Hidden Nodes and Incremental Learning," *IEEE Trans. neural networks*, vol. 20, no. 8, pp. 1352–1357, 2009, doi: 10.1109/TNN.2009.2024147.

[29] J. Cao *et al.*, "Extreme Learning Machine With Affine Transformation Inputs in an Activation Function," *IEEE Trans. Neural Networks Learn. Syst.*, vol. PP, pp. 1–15, 2018, doi: 10.1109/TNNLS.2018.2877468.

[30] T. Song, D. Li, Z. Liu, and W. Yang, "Online ADMM-Based extreme learning machine for sparse supervised learning," *IEEE Access*, vol. 7, pp. 64533–64544, 2019, doi: 10.1109/ACCESS.2019.2915970.

[31] J. M. Martínez-Martínez, P. Escandell-Montero, E. Soria-Olivas, J. D. Martín-Guerrero, R. Magdalena-Benedito, and J. Gómez-Sanchis, "Regularized extreme learning machine for regression problems," *Neurocomputing*, vol. 74, no. 17, pp.

3716–3721, 2011, doi: 10.1016/j.neucom.2011.06.013.

[32] X. Luo, X. Chang, and X. Ban, "Regression and classification using extreme learning machine based on L1-norm and L2-norm," *Neurocomputing*, vol. 174, pp. 179–186, Jan. 2016, doi: 10.1016/J.NEUCOM.2015.03.112.

[33] Y. Zhang, Q. Wu, and J. Hu, "An Adaptive Learning Algorithm for Regularized Extreme Learning Machine," *IEEE Access*, vol. 9, pp. 20736–20745, 2021, doi: 10.1109/ACCESS.2021.3054483.

[34] J. Cao, K. Zhang, M. Luo, C. Yin, and X. Lai, "Extreme learning machine and adaptive sparse representation for image classification," *Neural Networks*, vol. 81, pp. 91–102, Sep. 2016, doi: 10.1016/j.neunet.2016.06.001.

[35] M. van Heeswijk and Y. Miche, "Binary/ternary extreme learning machines," *Neurocomputing*, vol. 149, no. Part A, pp. 187–197, Feb. 2015, doi: 10.1016/J.NEUCOM.2014.01.072.

[36] Q. Zhu, A. K. K. Qin, P. N. N. Suganthan, and G.-B. Huang, "Evolutionary extreme learning machine," *Pattern Recognit.*, vol. 38, no. 0031–3203, pp. 1759–1763, Oct. 2005, doi: 10.1016/J.PATCOG.2005.03.028.

[37] Y. Xu and Y. Shu, "Evolutionary extreme learning machine - Based on particle swarm optimization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 3971 LNCS, no. 2, pp. 644–652, 2006, doi: 10.1007/11759966_95.

[38] F. Han, H. F. Yao, and Q. H. Ling, "An improved evolutionary extreme learning machine based on particle swarm optimization," *Neurocomputing*, vol. 116, pp. 87–93, 2013, doi: 10.1016/j.neucom.2011.12.062.

[39] Y. Lan, Y. C. Soh, and G. Bin Huang, "Ensemble of online sequential extreme learning machine," *Neurocomputing*, vol. 72, no. 13–15, pp. 3391–3395, 2009, doi: 10.1016/j.neucom.2009.02.013.

[40] Z. Xu, M. Yao, Z. Wu, and W. Dai, "Incremental regularized extreme learning machine and it ' s enhancement," *Neurocomputing*, vol. 174, pp. 134–142, 2016, doi: 10.1016/j.neucom.2015.01.097.

[41] R. Ma, J. Cao, T. Wang, and X. Lai, "Affine transformation based hierarchical extreme learning machine," in *Proceedings - IEEE International Symposium on Circuits and Systems*, Sep. 2020, vol. 2020-Octob, pp. 1–5, doi: 10.1109/iscas45731.2020.9181241.

[42] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016, Accessed: Jul. 28, 2021. [Online]. Available: https://arxiv.org/abs/1609.04747v2.

[43] L. Armijo, “Minimization of functions having lipschitz continuous first partial derivatives,” *Pacific J. Math.*, vol. 16, no. 1, pp. 1–3, Jan. 1966, doi: 10.2140/pjm.1966.16.1.

[44] O. Dekel, “From online to batch learning with cutoff-averaging,” in *Advances in Neural Information Processing Systems 21 - Proceedings of the 2008 Conference*, 2009, pp. 377–384.

[45] E. Bisong, “Batch vs. Online Learning,” *Build. Mach. Learn. Deep Learn. Model. Google Cloud Platf.*, pp. 199–201, 2019, doi: 10.1007/978-1-4842-4470-8_15.

[46] J. Kirkpatrick *et al.*, “Overcoming catastrophic forgetting in neural networks,” *Proc. Natl. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017, doi: 10.1073/PNAS.1611835114.

[47] J. Wang, R. Ran, and Y. Zhou, “A Short-Term Photovoltaic Power Prediction Model Based on an FOS-ELM Algorithm,” *Appl. Sci.*, vol. 7, no. 4, p. 423, 2017, doi: 10.3390/app7040423.

[48] J. Zhao, Z. Wang, and D. S. Park, “Online sequential extreme learning machine with forgetting mechanism,” *Neurocomputing*, vol. 87, pp. 79–89, 2012, doi: 10.1016/j.neucom.2012.02.003.

[49] “Woodbury matrix identity.” https://en.wikipedia.org/wiki/Woodbury_matrix_identity (accessed Aug. 05, 2021).

[50] W. Cao, Z. Ming, Z. Xu, J. Zhang, and Q. Wang, “Online Sequential Extreme Learning Machine with Dynamic Forgetting Factor,” *IEEE Access*, vol. 7, pp. 179746–179757, 2019, doi: 10.1109/ACCESS.2019.2959032.

[51] K. Xu, X. Tan, B. Fan, T. Xiao, X. Jin, and C. Zhu, “Dual Extreme Learning Machine Based Online Spatiotemporal Modeling with Adaptive Forgetting Factor,” *IEEE Access*, vol. 9, pp. 67379–67390, 2021, doi: 10.1109/ACCESS.2021.3075554.

[52] T. Watanabe, H. Takenaka, and D. Nohara, “Framework of Forecast Verification of Surface Solar Irradiance From a Numerical Weather Prediction Model Using Classification With a Gaussian Mixture Model,” *Earth Sp. Sci.*, vol. 7, no. 11, p. e2020EA001260, Nov. 2020, doi: 10.1029/2020EA001260.

[53] J. S. Stein, “The photovoltaic Performance Modeling Collaborative (PVPMC),” *Conf. Rec. IEEE Photovolt. Spec. Conf.*, pp. 3048–3052, 2012, doi:

10.1109/PVSC.2012.6318225.

[54] M. Mahooti, “Sunrise Sunset,” *MATLAB Central File Exchange*. .

[55] “Sunrise and sunset times in Riverside, California,” *Sunrise-Sunset*. https://sunrise-sunset.org/us/riverside-ca/2021/7 (accessed Aug. 10, 2021).

[56] “What Is a Linear Regression Model? - MATLAB & Simulink.” https://www.mathworks.com/help/stats/what-is-linear-regression.html (accessed Nov. 17, 2021).

[57] K. Kafadar, “The Efficiency of the Biweight as a Robust Estimator of Location,” *J. Res. Notional Bur. Stand.*, vol. 88, no. 2.

[58] N. R. Draper and H. Smith, *Applied regression analysis*, 3rd ed. wiley, 2014.

[59] “Understanding Support Vector Machine Regression - MATLAB & Simulink.” https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html (accessed Nov. 17, 2021).

[60] “Gaussian Process Regression Models.” https://www.mathworks.com/help/stats/gaussian-process-regression-models.html (accessed Nov. 17, 2021).

[61] “Kernel (Covariance) Function Options,” *MATLAB & Simulink*. https://www.mathworks.com/help/stats/kernel-covariance-function-options.html (accessed Nov. 17, 2021).

[62] A. Zimmermann, “Ensemble-Trees: Leveraging Ensemble Power Inside Decision Trees,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5255 LNAI, pp. 76–87, Oct. 2008, doi: 10.1007/978-3-540-88411-8_10.

[63] “Greenhouse Gas Inventory Data Explorer,” *United States Environmental Protection Agency (EPA)*. https://cfpub.epa.gov/ghgdata/inventoryexplorer/#allsectors/allsectors/allgas/econsect/all (accessed May 18, 2021).

[64] “Global EV Outlook 2020,” *International Energy Agency*, 2020. https://www.iea.org/reports/global-ev-outlook-2020.

[65] “Regression Learner App - MATLAB & Simulink.” https://www.mathworks.com/help/stats/regression-learner-app.html (accessed Aug. 23, 2021).

[66] A. Brown, S. Lommele, A. Schayowitz, and E. Klotz, "Electric Vehicle Charging Infrastructure Trends from the Alternative Fueling Station Locator: First Quarter 2020," 2020. [Online]. Available: www.nrel.gov/publications.

[67] B. Wang, Y. Wang, C. Qiu, C.-C. Chu, and R. Gadh, "Event-based electric vehicle scheduling considering random user behaviors," in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2015, pp. 313–318, doi: 10.1109/SmartGridComm.2015.7436319.

[68] N. Sadeghianpourhamami, J. Deleu, and C. Develder, "Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning," *IEEE Trans. Smart Grid*, vol. PP, no. c, pp. 1–1, 2019, doi: 10.1109/tsg.2019.2920320.

[69] B. Sun, T. Li, S. H. Low, and D. H. K. Tsang, "ORC: An Online Competitive Algorithm for Recommendation and Charging Schedule in Electric Vehicle Charging Network," *e-Energy 2020 - Proc. 11th ACM Int. Conf. Futur. Energy Syst.*, pp. 144–155, 2020, doi: 10.1145/3396851.3397727.

[70] S. Li, J. Li, C. Su, and Q. Yang, "Optimization of Bi-Directional V2G Behavior with Active Battery Anti-Aging Scheduling," *IEEE Access*, vol. 8, pp. 11186–11196, 2020, doi: 10.1109/ACCESS.2020.2964699.

[71] A. Y. Saber and G. K. Venayagamoorthy, "Plug-in vehicles and renewable energy sources for cost and emission reductions," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1229–1238, Apr. 2011, doi: 10.1109/TIE.2010.2047828.

[72] A. M. A. Haidar and K. M. Muttaqi, "Behavioral characterization of electric vehicle charging loads in a distribution power grid through modeling of battery chargers," *IEEE Trans. Ind. Appl.*, vol. 52, no. 1, pp. 483–492, Jan. 2016, doi: 10.1109/TIA.2015.2483705.

[73] K. Chaudhari, N. K. Kandasamy, A. Krishnan, A. Ukil, and H. B. Gooi, "Agent-based aggregated behavior modeling for electric vehicle charging load," *IEEE Trans. Ind. Informatics*, vol. 15, no. 2, pp. 856–868, Feb. 2019, doi: 10.1109/TII.2018.2823321.

[74] S. Yang, M. Wu, X. Yao, and J. Jiang, "Load Modeling and Identification Based on Ant Colony Algorithms for EV Charging Stations," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 1997–2003, Jul. 2015, doi: 10.1109/TPWRS.2014.2352263.

[75] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, "Data-driven smart charging for heterogeneous electric vehicle fleets," *Energy AI*, vol. 1, p. 100007,

2020, doi: 10.1016/j.egyai.2020.100007.

[76] C. Sun, T. Li, S. H. Low, and V. O. K. Li, "Classification of electric vehicle charging time series with selective clustering," *Electr. Power Syst. Res.*, vol. 189, no. August, p. 106695, 2020, doi: 10.1016/j.epsr.2020.106695.

[77] R. Mehta, D. Srinivasan, A. M. Khambadkone, J. Yang, and A. Trivedi, "Smart charging strategies for optimal integration of plug-in electric vehicles within existing distribution system infrastructure," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 299–312, Jan. 2018, doi: 10.1109/TSG.2016.2550559.

[78] T. Ma, A. Mohamed, and O. Mohammed, "Optimal charging of plug-in electric vehicles for a car park infrastructure," *Conf. Rec. - IAS Annu. Meet. (IEEE Ind. Appl. Soc.*, 2012, doi: 10.1109/IAS.2012.6374035.

[79] D. Dallinger and M. Wietschel, "Grid integration of intermittent renewable energy sources using price-responsive plug-in electric vehicles," *Renew. Sustain. Energy Rev.*, vol. 16, no. 5, pp. 3370–3382, Jun. 2012, doi: 10.1016/J.RSER.2012.02.019.

[80] L. Jian, H. Xue, G. Xu, X. Zhu, D. Zhao, and Z. Y. Shao, "Regulated charging of plug-in hybrid electric vehicles for minimizing load variance in household smart microgrid," *IEEE Trans. Ind. Electron.*, vol. 60, no. 8, pp. 3218–3226, 2013, doi: 10.1109/TIE.2012.2198037.

[81] D. Wang, X. Guan, J. Wu, P. Li, P. Zan, and H. Xu, "Integrated Energy Exchange Scheduling for Multimicrogrid System with Electric Vehicles," *IEEE Trans. Smart Grid*, vol. 7, no. 4, pp. 1762–1774, Jul. 2016, doi: 10.1109/TSG.2015.2438852.

[82] P. Makeen, S. Memon, M. A. Elkasrawy, S. O. Abdullatif, and H. A. Ghali, "Smart green charging scheme of centralized electric vehicle stations," *https://doi.org/10.1080/15435075.2021.1947822*, 2021, doi: 10.1080/15435075.2021.1947822.

[83] I. K. A. Aswantara, K. S. Ko, and D. K. Sung, "A centralized EV charging scheme based on user satisfaction fairness and cost," *2013 IEEE Innov. Smart Grid Technol. - Asia, ISGT Asia 2013*, 2013, doi: 10.1109/ISGT-ASIA.2013.6698730.

[84] L. Yu, T. Zhao, Q. Chen, and J. Zhang, "Centralized Bi-level Spatial-Temporal Coordination Charging Strategy for Area Electric Vehicles," *CSEE J. Power Energy Syst.*, vol. 1, no. 4, pp. 74–83, 2015, doi: 10.17775/CSEEJPES.2015.00050.

[85] J. Rivera, C. Goebel, and H. A. Jacobsen, "Distributed Convex Optimization for Electric Vehicle Aggregators," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1852–1863,

2017, doi: 10.1109/TSG.2015.2509030.

[86] F. L. Da Silva, C. E. H. Nishida, D. M. Roijers, and A. H. R. Costa, "Coordination of Electric Vehicle Charging through Multiagent Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2347–2356, 2020, doi: 10.1109/TSG.2019.2952331.

[87] "ACN-Data." https://ev.caltech.edu/dataset (accessed May 14, 2021).

[88] "Alternative Fueling Station Locator." https://afdc.energy.gov/stations/#/find/nearest?country=US&fuel=ELEC&ev_levels=all (accessed May 14, 2021).

[89] R. PRATT, "Vehicle Communications and Charging Control." https://www.energy.gov/sites/prod/files/2014/07/f18/vss142_pratt_2014_p.pdf (accessed May 19, 2021).

[90] C.-C. M. Yeh *et al.*, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets," in *2016 IEEE 16th International Conference on Data Mining (ICDM)*, Dec. 2016, pp. 1317–1322, doi: 10.1109/ICDM.2016.0179.

[91] S. Gharghabi, S. Imani, A. Bagnall, A. Darvishzadeh, and E. Keogh, "Matrix Profile XII: MPdist: A Novel Time Series Distance Measure to Allow Data Mining in More Challenging Scenarios," *Proc. - IEEE Int. Conf. Data Mining, ICDM*, vol. 2018-Novem, pp. 965–970, 2018, doi: 10.1109/ICDM.2018.00119.

[92] J. Shi, N. Yu, E. Keogh, H. K. Chen, and K. Yamashita, "Discovering and Labeling Power System Events in Synchrophasor Data with Matrix Profile," *iSPEC 2019 - 2019 IEEE Sustain. Power Energy Conf. Grid Mod. Energy Revolution, Proc.*, pp. 1827–1832, 2019, doi: 10.1109/iSPEC48194.2019.8975286.

[93] L. Kaufman and P. J. Rousseeuw, Eds., *Finding Groups in Data*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 1990.

[94] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035, doi: 10.5555/1283383.1283494.

[95] B. Desgraupes, "Clustering Indices," *CRAN Packag.*, no. April, pp. 1–10, 2013, [Online]. Available: cran.r-project.org/web/packages/clusterCrit%0Ahttps://cran.r-project.org/web/packages/clusterCrit/vignettes/clusterCrit.pdf.

[96] J. Paparrizos and L. Gravano, "Fast and accurate time-series clustering," *ACM Trans.*

Database Syst., vol. 42, no. 2, 2017, doi: 10.1145/3044711.

[97] "Tune Gaussian Mixture Models - MATLAB & Simulink." https://www.mathworks.com/help/stats/tune-gaussian-mixture-models.html (accessed Sep. 13, 2021).

[98] "California Self-Generation Incentive Program." https://sgipsignal.com/download-data (accessed Nov. 22, 2021).

[99] "Schedule TOU-EV-8 Sheet 1 GENERAL SERVICE TIME-OF-USE, ELECTRIC VEHIC," *Southern California Edison*. https://library.sce.com/content/dam/sce-doclib/public/regulatory/tariff/electric/schedules/general-service-&-industrial-rates/ELECTRIC_SCHEDULES_TOU-EV-8.pdf (accessed Nov. 22, 2021).

[100] A. H. Mohsenian-Rad, V. W. S. S. Wong, J. Jatskevich, and R. Schober, "Optimal and autonomous incentive-based energy consumption scheduling algorithm for smart grid," *Innov. Smart Grid Technol. Conf. ISGT 2010*, pp. 1–6, 2010, doi: 10.1109/ISGT.2010.5434752.

[101] D. T. Ton and M. A. Smith, "The U.S. Department of Energy's Microgrid Initiative," *Electr. J.*, vol. 25, no. 8, pp. 84–94, Oct. 2012, doi: 10.1016/j.tej.2012.09.013.

[102] "U.S. Energy Information Administration (EIA) - Data," *U.S. Energy Information Administration*. https://www.eia.gov/totalenergy/data/browser/index.php?tbl=T01.03#/?f=A&start=2000&end=2020&charted=1-2-3-5-12 (accessed Nov. 28, 2021).

[103] Y. Xue, "Real-time Battery Control Method for Microgrid Energy Management," University of California, Riverside, 2015.

[104] A. M. R. Lede, M. G. Molina, M. Martinez, and P. E. Mercado, "Microgrid architectures for distributed generation: A brief review," in *2017 IEEE PES Innovative Smart Grid Technologies Conference - Latin America (ISGT Latin America)*, Sep. 2017, pp. 1–6, doi: 10.1109/ISGT-LA.2017.8126746.

[105] L. Meng *et al.*, "Review on Control of DC Microgrids," *IEEE J. Emerg. Sel. Top. Power Electron.*, pp. 1–1, 2017, doi: 10.1109/JESTPE.2017.2690219.

[106] E. Bullich-Massagué, F. Díaz-González, M. Aragüés-Peñalba, F. Girbau-Llistuella, P. Olivella-Rosell, and A. Sumper, "Microgrid clustering architectures," *Appl. Energy*, vol. 212, pp. 340–361, Feb. 2018, doi: 10.1016/J.APENERGY.2017.12.048.

[107] Firdaus, Umar, and O. Penangsang, "Economic dispatch of multi microgrid systems with renewable energy sources using particle swarm optimization," *Proceeding -*

2016 Int. Semin. Intell. Technol. Its Appl. ISITIA 2016 Recent Trends Intell. Comput. Technol. Sustain. Energy, pp. 595–600, Jan. 2017, doi: 10.1109/ISITIA.2016.7828727.

[108] N. Nikmehr and S. Najafi Ravadanegh, "Optimal Power Dispatch of Multi-Microgrids at Future Smart Distribution Grids," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1648–1657, Jul. 2015, doi: 10.1109/TSG.2015.2396992.

[109] J. Vasiljevska, J. A. Peças Lopes, and M. A. Matos, "Evaluating the impacts of the multi-microgrid concept using multicriteria decision aid," *Electr. Power Syst. Res.*, vol. 91, pp. 44–51, Oct. 2012, doi: 10.1016/J.EPSR.2012.04.013.

[110] J. Vasiljevska, J. A. Peças Lopes, and M. A. Matos, "Multi-microgrid impact assessment using multi criteria decision aid methods," 2009, doi: 10.1109/PTC.2009.5282054.

[111] M. S. Saleh, A. Althaibani, Y. Esa, Y. Mhandi, and A. A. Mohamed, "Impact of clustering microgrids on their stability and resilience during blackouts," *Proc. - 2015 Int. Conf. Smart Grid Clean Energy Technol. ICSGCE 2015*, pp. 195–200, Apr. 2016, doi: 10.1109/ICSGCE.2015.7454295.

[112] T. Lu, Z. Wang, Q. Ai, and W. J. Lee, "Interactive Model for Energy Management of Clustered Microgrids," *IEEE Trans. Ind. Appl.*, vol. 53, no. 3, pp. 1739–1750, May 2017, doi: 10.1109/TIA.2017.2657628.

[113] L. Che, M. Shahidehpour, A. Alabdulwahab, and Y. Al-Turki, "Hierarchical coordination of a community microgrid with AC and DC microgrids," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3042–3051, Nov. 2015, doi: 10.1109/TSG.2015.2398853.

[114] E. Amicarelli, T. Q. Tran, and S. Bacha, "Optimization algorithm for microgrids day-ahead scheduling and aggregator proposal," *Conf. Proc. - 2017 17th IEEE Int. Conf. Environ. Electr. Eng. 2017 1st IEEE Ind. Commer. Power Syst. Eur. EEEIC / I CPS Eur. 2017*, Jul. 2017, doi: 10.1109/EEEIC.2017.7977487.

[115] M. R. Basir Khan, R. Jidin, and J. Pasupuleti, "Multi-agent based distributed control architecture for microgrid energy management and optimization," *Energy Convers. Manag.*, vol. 112, pp. 288–307, Mar. 2016, doi: 10.1016/J.ENCONMAN.2016.01.011.

[116] A. Mohammed, S. S. Refaat, S. Bayhan, and H. Abu-Rub, "AC Microgrid Control and Management Strategies: Evaluation and Review," *IEEE Power Electron. Mag.*, vol. 6, no. 2, pp. 18–31, Jun. 2019, doi: 10.1109/MPEL.2019.2910292.

[117] U. N. Ekanayake and U. S. Navaratne, "A Survey on Microgrid Control Techniques in Islanded Mode," *J. Electr. Comput. Eng.*, vol. 2020, 2020, doi: 10.1155/2020/6275460.

[118] D. P. e Silva, J. L. Félix Salles, J. F. Fardin, and M. M. Rocha Pereira, "Management of an island and grid-connected microgrid using hybrid economic model predictive control with weather data," *Appl. Energy*, vol. 278, p. 115581, Nov. 2020, doi: 10.1016/J.APENERGY.2020.115581.

[119] L. Meng *et al.*, "Review on Control of DC Microgrids and Multiple Microgrid Clusters," *IEEE J. Emerg. Sel. Top. Power Electron.*, vol. 5, no. 3, pp. 928–948, Sep. 2017, doi: 10.1109/JESTPE.2017.2690219.

[120] A. Abdali, R. Noroozian, and K. Mazlumi, "Simultaneous control and protection schemes for DC multi microgrids systems," *Int. J. Electr. Power Energy Syst.*, vol. 104, pp. 230–245, Jan. 2019, doi: 10.1016/J.IJEPES.2018.06.054.

[121] A. Werth, N. Kitamura, M. Tokoro, and K. Tanaka, "Evaluation model for multi-microgrid with autonomous DC energy exchange," *IEEJ Trans. Electr. Electron. Eng.*, vol. 12, no. 5, pp. 676–682, Sep. 2017, doi: 10.1002/TEE.22453.

[122] J. Zhou, H. Zhang, Q. Sun, D. Ma, and B. Huang, "Event-based distributed active power sharing control for interconnected AC and DC microgrids," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6815–6828, Nov. 2018, doi: 10.1109/TSG.2017.2724062.

[123] H. J. Yoo, T. T. Nguyen, and H. M. Kim, "Consensus-based distributed coordination control of hybrid AC/DC microgrids," *IEEE Trans. Sustain. Energy*, vol. 11, no. 2, pp. 629–639, Apr. 2020, doi: 10.1109/TSTE.2019.2899119.

[124] P. Tian, X. Xiao, K. Wang, and R. Ding, "A Hierarchical Energy Management System Based on Hierarchical Optimization for Microgrid Community Economic Operation," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2230–2241, Sep. 2016, doi: 10.1109/TSG.2015.2470551.

[125] Y. Yan, D. Shi, D. Bian, B. Huang, Z. Yi, and Z. Wang, "Small-signal Stability Analysis and Performance Evaluation of Microgrids under Distributed Control," *IEEE Trans. Smart Grid*, Sep. 2018, doi: 10.1109/TSG.2018.2869566.

[126] Y. Han, K. Zhang, H. Li, E. A. A. Coelho, and J. M. Guerrero, "MAS-Based Distributed Coordinated Control and Optimization in Microgrid and Microgrid Clusters: A Comprehensive Overview," *IEEE Trans. Power Electron.*, vol. 33, no. 8, pp. 6488–6508, Aug. 2018, doi: 10.1109/TPEL.2017.2761438.

[127] M. Marzband, N. Parhizi, M. Savaghebi, and J. M. Guerrero, “Distributed Smart Decision-Making for a Multimicrogrid System Based on a Hierarchical Interactive Architecture,” *IEEE Trans. Energy Convers.*, vol. 31, no. 2, pp. 637–648, 2016, doi: 10.1109/TEC.2015.2505358.

[128] Y. Liu *et al.*, “Distributed Robust Energy Management of a Multi-Microgrid System in the Real-Time Energy Market,” *IEEE Trans. Sustain. Energy*, vol. 3029, no. c, pp. 1–11, 2017, doi: 10.1109/TSTE.2017.2779827.

[129] CITY OF RANCHO CUCAMONGA MUNICIPAL UTILITY, “Electric Rates and Tariffs.” https://www.cityofrc.us/civicax/filebank/blobdload.aspx?BlobID=14441 (accessed Aug. 01, 2018).

[130] National Renewable Energy Lab and Sandia National Lab, “System Advisor Model,” *National Renewable Energy Laboratory*, 2010. https://www.nrel.gov/analysis/sam/.

[131] C. Chen, S. Duan, T. Cai, B. Liu, and G. Hu, “Smart energy management system for optimal microgrid economic operation,” *IET Renew. Power Gener.*, vol. 5, no. 3, pp. 258–267, May 2011, doi: 10.1049/IET-RPG.2010.0052.

[132] H. Kanchev, D. Lu, F. Colas, V. Lazarov, and B. Francois, “Energy management and operational planning of a microgrid with a PV-based active generator for smart grid applications,” *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4583–4592, 2011, doi: 10.1109/TIE.2011.2119451.

[133] 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 *2016 IEEE 43rd Photovoltaic Specialists Conference (PVSC)*, Jun. 2016, pp. 1875–1880, doi: 10.1109/PVSC.2016.7749947.

[134] California Energy Commission, “California Solar Energy Statistics and Data.” https://ww2.energy.ca.gov/almanac/renewables_data/solar/index_cms.php. (accessed Aug. 28, 2019).

[135] California Energy Commission, “What are TOU rates?” https://www.cpuc.ca.gov/General.aspx?id=12194 (accessed Aug. 28, 2019).

[136] “TIME-OF-USE (TOU) PEAK PERIODS ARE CHANGING.” https://pages.email.sce.com/TOU/en/ (accessed Aug. 20, 2019).

[137] California Energy Commission, “Net Energy Metering (NEM).” https://www.cpuc.ca.gov/general.aspx?id=3800 (accessed Aug. 28, 2019).

[138] S. Adhikari and F. Li, “Coordinated V-f and P-Q Control of Solar Photovoltaic

Generators With MPPT and Battery Storage in Microgrids," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1270–1281, May 2014, doi: 10.1109/TSG.2014.2301157.

[139] Z. Yi, W. Dong, and A. H. Etemadi, "A Unified Control and Power Management Scheme for PV-Battery-Based Hybrid Microgrids for Both Grid-Connected and Islanded Modes," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 5975–5985, Nov. 2018, doi: 10.1109/TSG.2017.2700332.

[140] "SCE DR Event Status." https://www.sce.openadr.com/dr.website/scepr-event-history.jsf (accessed Aug. 29, 2019).

[141] "Demand Response Programs for Business." https://www.sce.com/business/demand-response.

[142] "Schedule TOU-GS-2 TIME-OF-USE - GENERAL SERVICE - DEMAND METERED." https://www1.sce.com/NR/sc3/tm2/pdf/ce329.pdf (accessed Apr. 11, 2019).

[143] "Net Energy Metering Aggregation (NEM-A) WELCOME GUIDE." https://www.sce.com/sites/default/files/inline-files/NEM-A Welcome Guide 1018_WCAG %281%29.pdf (accessed Aug. 20, 2019).

[144] "Net Surplus Compensation Rate." https://www.sce.com/regulatory/tariff-books/rates-pricing-choices/net-surplus-compensation (accessed Apr. 09, 2019).