Application of Deep Reinforcement Learning to Wide Area Power System and Big Data Analysis to Smart Meter Status Monitoring

by

Gyoungjae Kim

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2021 by the
Graduate Supervisory Committee:

Yang Weng, Chair
Meng Wu
Yunpeng Zhao

ARIZONA STATE UNIVERSITY

August 2021

ABSTRACT

Due to the large scale of power systems, latency uncertainty in communication can cause severe problems in wide-area measurement systems. To resolve the issue, a significant amount of past work focuses on using emerging technology which is machine learning methods such as Q-learning to address latency issues in modern controls. Although such a method can deal with the stochastic characteristics of communication latency in the long run, the Q-learning methods tend to overestimate Q-values, leading to high bias. To solve the overestimation bias issue, the learning structure is redesigned with a twin-delayed deep deterministic policy gradient algorithm to handle the damping control issue under unknown latency in the power network. Meanwhile, a new reward function is proposed, taking into account the machine speed deviation, the episode termination prevention, and the feedback from action space. In this way, the system optimally damps down frequency oscillations while maintaining the system's stability and reliable operation within defined limits. The simulation results verify the proposed algorithm in various perspectives including the latency sensitivity analysis under high renewable energy penetration and the comparison with other machine learning algorithms. For example, if the proposed twin-delayed deep deterministic policy gradient algorithm is applied, the low-frequency oscillation significantly improved compared to existing algorithms.

Furthermore, under the mentorship of Dr. Yang Weng, the development of a big data analysis software project has been collaborating with the Salt River Project (SRP), a major power utility in Arizona. After a thorough examination of data for the project, it is examined that SRP is suffering from a lot of smart meters data issues. An important goal of the project is to design big data software to monitor SRP smart meter data and to present indicators of abnormalities and special events. Currently, the big data software interface has been developed for SRP, which has already been successfully adopted by other utilities, research institutes, and laboratories as well.

DEDICATION

I would like to thank my parents, Kungsook Lim and Youngsoo Kim for their affection throughout my life and support from my brother SeoungJae Kim that stood by me. Also, many thanks to my wife, Ariel Kim. Thanks for walking with me during the past five years of school life. Thanks for her support and encouragement in both study and life. Special thanks for a new family member, chewie, for coming to my life and bringing me unprecedented feeling of happiness.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my advisor, Dr. Yang Weng, for offering great support on this thesis research. This thesis would not have been possible without the support of my advisor. I sincerely thank him for his invaluable guidance, patience and encouragement throughout my master's studies. He sets a great example of a researcher to the group by his sharp engineering insight and great enthusiasm for research.

Furthermore, I would like to express my sincere thanks to my committee members, Dr. Meng Wu and Dr. Yunpeng Zhao for taking the time to give their invaluable direct and indirect feedback. I would like to thank Dr. Qiushi Cui who has given valuable suggestions and feedback for the study.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES viii

CHAPTER

1 INTRODUCTION 1

2 BACKGROUND 5

2.1 Reinforcement Learning and Policy Gradient Method 5

2.2 Deep Deterministic Policy Gradient Algorithm 7

3 REINFORCEMENT LEARNING BASED CONTROLLER 10

3.1 State of the Controller 11

3.2 Action of the Controller 13

3.3 Reward Design for Enhanced Control Results 13

3.4 Twin-Delayed Deep Deterministic Policy Gradient Method 15

3.5 Avoid the Overestimation Bias 19

4 NUMERICAL VALIDATION 21

4.1 Benchmark System 21

4.2 TD3 Control Agent: Fast Learning Curve 22

4.3 Control Agent: Robust to Communication Latency 25

4.4 Performance Comparison with DDPG 27

5 BIG DATA ANALYSIS TO SMART METER STATUS MONITORING . . 29

CHAPTER Page

6 DATA VISUALIZATIONS 31
7 DISTRIBUTION VISUALIZATIONS 32
8 BAD DATA INDICATOR 36
8.1 Incorrect Battery Configuration from Database 36
8.2 Duplicated Battery Meter Records 37
8.3 Voltage Violation 38
8.4 Battery Kwh Violation 41
9 BEHAVIOR ANALYSIS 44
10 CONCLUSION 49
REFERENCES 50

APPENDIX

A DELAYS DUE TO DIFFERENT COMMUNICATION LINKS 54
B THE PARAMETERS OF FOUR TESTING SCENARIOS 56

LIST OF TABLES

Table Page

4.1 Parameters of the Benchmark System under Study 23

4.2 Parameters of TD3 and DDPG Agents 28

8.1 Inconsistent Battery Configurations in the Database File 37

8.2 Duplicated Battery Meter Readings Table Information 38

8.3 Voltage Violations Table Information 40

8.4 Low Usage Readings Table Information 42

A.1 Delays Due to Different Communication Links 55

B.1 Mean and Variance Communication Delay Using the Gaussian Distributed Random Signal 57

LIST OF FIGURES

Figure Page

2.1 The Agent-environment Interaction in Reinforcement Learning. At the Time t, the Agent Perceives State S_t and Reward R_t from the Environment Then Takes an Action A_t, Which Will Affect the State of the Environment. The Agent Receives a Reward r_{t+1} and a New State s_{t+1} from the Resulting Environment. 6

3.1 Framework of the Overall Scheme. The Dotted Lines Indicate the Communication Lines with Delays. The Inputs to the Controller Are the States and Reward, Whereas the Output is the Action. 11

3.2 Zoomed-in Illustration of the Proposed Algorithm. The Dotted Lines Represent the Inputs and Output of the Controller. 12

4.1 A Modified Kundur Four-machine Two-area Power System. In This System, to Represent the Deep Renewable Generation, Two Aggregated Residential PV Generation – PV1 and PV2 Are Added. 22

4.2 A Learning Curve That Shows the Individual Episode Reward and Average Reward. The X-axis and the Y-axis Are the Number of Episodes and the Reward Respectively. The Blue Line Means Episode Reward (the Reward for Each Episode). The Red Line Is an Average Reward (a Running Average Reward Value). And, the Yellow Line Is Episode Q0 (the Critic's Estimate of the Discounted Long-term Reward at the Start of Each Episode). The Setup Behind This Figure Is in Table 4.2. 24

4.3 Result Comparison Using the Proposed TD3 Algorithm under 3 terms and 5 terms Reward Function, Where 3 Terms Refer to the First Three Terms in Equation (3.4), and 5 Terms Refer to the Entire Equation (3.4). The Reward Here Refers to the Running Average Reward Value. 25

4.4 The Control Performance Based on the Variation of Average Communication Delay. Due to the Stochastic Nature of the Proposed RL Algorithm, the Results under Different Time Delays Are Not Always the Same. This Figure Is a Representative Result Selected Among Extensive Simulations. However, the Largest Speed Deviation Is Always Well Contained in All Test Cases. 26

Figure Page

4.5 Result Comparison Between DDPG and TD3 Agents. The Parameters of Both Agents Are Presented in Table 4.2. The Reward Here Refers to the Reward for Each Episode. 28

6.1 The Location Number at 231731005 in Time Series 31

7.1 Histogram of Kwh - Delivered Smart Meter Data Distribution 33

7.2 Boxplot of Kwh - Delivered Smart Meter Data Distribution 34

8.1 Voltage Violation Visualization of Smart Meter Data in Time Series .. 40

8.2 Battery Kwh Violation Visualization of Smart Meter Data in Time Series 42

9.1 The First Most Typical Behavior Pattern Analysis 45

9.2 The Second Most Typical Behavior Pattern Analysis 45

9.3 The Third Most Typical Behavior Pattern Analysis 45

9.4 The First Abnormal Customer Analysis in the First Behavior Pattern 46

9.5 The Second Abnormal Customer Analysis in the First Behavior Pattern 47

9.6 The Third Abnormal Customer Analysis in the First Behavior Pattern 47

Chapter 1

INTRODUCTION

The inter-area low-frequency oscillations cause significant challenges to reliable control and economic operation in a typical cyber-physical system such as transmission networks. For example, if the inter-area oscillation has a poor damping, it will cause catastrophic disturbances, such as forming multiple outages, leading to widespread oscillation [1]. The failure to control frequency oscillation can cause severe damage to the stability and reliability of the power system. In the worst case, it can cause large-scale power outages, in other words, blackouts. There have been several incidents of low-frequency inter-area oscillation. The most notable incident occurred on August 14, 2003 at the Eastern Interconnection located in the United States [2]. This incident caused 45 million people to lose their power supply for periods of up to three hours. This was caused by poor damping of low-frequency oscillations. Also, the another most notable incident took place in the southern region, where the power system broke down on September 15, 2011. The incident was a result of poor frequency oscillation damping of the power system. Such events are harder to avoid with traditional solutions because the maximum available transfer capability is limited [3, 4, 5]. For example, traditionally power engineering approaches damp oscillations with power system stabilizers (PSSs), dependent on local measurements. However, such inter-area modes are neither always controllable nor observable from local measurement signals [6]. Fortunately, the observability of inter-area modes improves due to the advent of wide-area measurement systems (WAMS) and the implementation of phasor measurement units (PMUs) [7, 8, 9, 10, 11]. Due to the communication delay in modern Information and Communication Technologies (ICT), the associated issue, therefore, becomes the battle with its uncertainty [12].

Since the communication delay significantly affects the damping control performance, researchers have proposed various methods to solve this issue. The work in [13] designs a fuzzy logic wide-area damping controller to damp the inter-area oscillations compensating for the continuous latency. By selecting suitable stabilizing devices and input signal and taking variable latency into account, [14] presents an inter-area oscillation damping controller design considering the impact of variable latency. Meanwhile, [15] incorporates a series of integral methods, including average assignment, phase tracking, and magnitude attenuation, to overcome the limitations of the adaptive phasor power oscillation damping method operating in varying-latency situations.

Moreover, a variable loop gain controller was proposed based on the excessive regeneration for system stability, limiting the delay range up to 250 ms [16]. Also, [17] identifies a low-order transfer function model using a multi-input multi-output autoregressive moving average exogenous model. From the device perspective, the static VAR compensators are adopted under different operating conditions and renewable energy sources [18, 19, 20]. Although these methods have merits from a different perspective, some of them neglect communication delay and some assume correct network topology and system parameters. Unfortunately, such assumptions are hard to achieve in reality due to their accessibility, the network growth, and the instantaneous communication congestion condition, even though in 5G networks.

Targeting these issues, learning methods that are physically model-free are proposed. For example, [21] uses a deep learning WADC, but such a method relies extensively on the past data and was unable to adapt to the changes in transmission networks. One observation from such work is that the exploration of the system does help in capturing such transformations. Therefore, reinforcement learning (RL) provides a platform to explore the environment and learn the control strategy accordingly. Among different RL methods, Q-learning can handle problems with stochastic transitions and rewards using a value function. [22] leverages this capability of learning a stochastic control through exploring the network by using Q-learning. However, Q-learning fails where there is a large state space [23, 24, 25, 26].

To overcome such challenges of large state space, we can combine the advantage of RL and deep learning to provide a stochastic and robust control through WAMS, so that both the uncertainties and time-varying delays can be taken into account through interactive learning. The contribution of this thesis is the novel design of a policy-based RL method to address the damping control due to unknown latency in inter-area oscillation. Specifically, we build a power system testbed for RL's interactive environment. Then, we define the state and action that are suitable for damping control. In the end, a reward function that considers the physical measurements and the sustainability of RL is proposed.

This thesis is organized as follows: Section 2 explains the background knowledge of the RL and policy gradient method. Section 3 elaborates on the specific design of the RL-based controller, including the design of state, action, and reward to maximize the control benefits and merge them into the power system concept. Further simulation results and shortcomings of the proposed method are described in Section 4, followed by the conclusion in Section 10.

Chapter 2

BACKGROUND

2.1 Reinforcement Learning and Policy Gradient Method

Power systems damping control has to deal with uncertainties and ambiguities in the entire system. RL is a perfect tool to solve such issues. RL is an area of machine learning concerned with how agents ought to take actions in an environment in order to maximize the notion of cumulative reward. Essentially, Markov decision process (MDP) is formally used to describe RL problems [27]. It is modelled with a tuple (S, A, E, r) which consists of a state space S; an action space A; a transition function

$$E[S_{t+1}|s_t, a_t]$$

that predicts the next state s_{t+1}, given a current state-action pair (s_t, a_t). Each of this pair (s_t, a_t) is corresponding to the immediate reward achieved at each state-action pair. In power systems, the state-action pair means the control action taken under present operating conditions, whereas the reward means the score obtained after a control action. The policy of an RL problem is a function π which maps the state space to the action space. An RL algorithm tries to find an optimal policy π to maximize the expected total reward. To achieve the optimal policy, we typically use gradient-based methods, where the policy is parameterized by a parameter vector θ. Such a vector is updated along the gradient direction of the expected total rewards. The corresponding RL operational principle diagram is shown in Fig. 2.1, which clearly displays the relationship between environment and agent. The agent observes the state of the environment through measurement and communication devices. Based on the observations, the agent takes the corresponding action to control the state of the environment through the calculation of the reward. To improve control performance, the agent updates its policy that defines the learning agent's way of behaving at a given time.

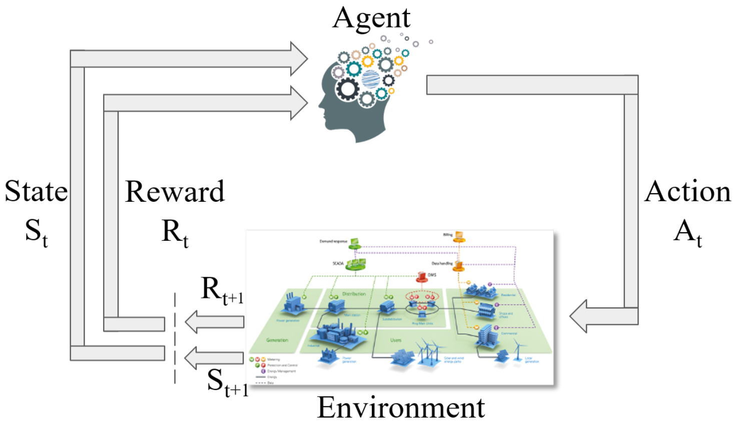

Figure 2.1: The Agent-environment Interaction in Reinforcement Learning. At the Time t, the Agent Perceives State S_t and Reward R_t from the Environment Then Takes an Action A_t, Which Will Affect the State of the Environment. The Agent Receives a Reward r_{t+1} and a New State s_{t+1} from the Resulting Environment.

2.2 Deep Deterministic Policy Gradient Algorithm

Lots of researches in the past focus on using machine learning methods such as deep Q-network (DQN) to address latency issues in modern controls. Although DQN has achieved huge success in higher dimensional problems, its action space is still discrete. However, many tasks of interest, especially in the task of damping control, the action space is continuous. Therefore, the Deep Deterministic Policy Gradient (DDPG) algorithm seems a promising solution. DDPG is very popular in RL control. It chooses an actor-critic model, where a critic-network might help to suppress the bad decisions made by an actor-network. The details of continuous control through the actor-critic model are presented in [28]. DDPG agent not only works with continuous action space, but also has a good accuracy in learning under complex environments, proved in [29]. Its value function is expressed as follows:

$$Q(s_t, a_t) = E[r(s_t, a_t) + \gamma \max_a E[Q(s_{t+1}, a_{t+1})]. \tag{2.1}$$

The function approximator Q for the critic network samples states from the WAMS and follows a specific distribution [28]. It estimates the effectiveness of the action taken. In power systems damping control, the action can be the reference value for the generator terminal voltage. An actor-network $\mu(s_{t+1})$ takes only the states as input features and directly estimates the actions. But, such estimation requires critical evaluation. So, we define the approximator y_i for critic network,

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}). \tag{2.2}$$

Since both of the function approximators are characterized as deep layers of neural networks, we parameterize them with θ^Q and θ^μ as presented in [28]. In power systems, they are the parameters for the control performance models. So, the loss function for M samples becomes

$$\text{Loss} = \frac{1}{M}\sum_{i=1}^{M}(y_i - Q(s_i, a_i))^2, \tag{2.3}$$

where $i = 1, \cdots, M$ is the number of samples in mini-batch. The parameters of critic-network are obtained by iteratively minimizing the above loss function [28]. The goal in RL is to learn a policy that maximizes the expected return from the start distribution J. To achieve the maximized expectation, we update the actor-network by applying the chain rule to the expected return from J concerning the actor parameters,

$$\nabla_{\theta_\mu} J \approx \frac{1}{M}\nabla_a Q(s_i, a_i|\theta^Q)\nabla_{\theta_\mu}\mu(s_i|\theta^\mu). \tag{2.4}$$

Then, we update the target actor Q' and target critic μ' parameters using a periodic approach, so that after each iteration the target actor becomes the initial actor and target critic becomes the initial critic as proposed in [28]

$$\theta^{Q'} = \theta^Q, \tag{2.5}$$

$$\theta^{\mu'} = \theta^\mu, \tag{2.6}$$

respectively. It is natural to expect policy-based methods are more useful in continuous space. Because there are an infinite number of actions and (or) states to estimate the values for and hence value-based approaches are way too expensive computationally in the continuous space. For example, in generalized policy iteration, the policy improvement step

$$arg \max_{a \in \mathcal{A}} Q^{\pi}(s,a) \tag{2.7}$$

requires a full scan of the action space, suffering from the curse of dimensionality. This is true power system stability control issues where the action space is continuous. However, using gradient ascent, we can move θ toward the direction suggested by the gradient

$$\nabla_{\theta} J(\theta)$$

to find the best θ for π_{θ} that produces the highest return.

Chapter 3

REINFORCEMENT LEARNING BASED CONTROLLER

In this section, we discuss the detailed design of the state, the action, and the reward. Importantly, based on the DDPG algorithm, we further realize the Twin-delayed deep deterministic policy gradient (TD3) algorithm to avoid the overestimation bias of DDPG. Fig. 3.1 gives a complete account of the mechanism, along with the inputs and outputs for the model. The proposed framework of the overall scheme is motivated by the wide-area monitoring system, where the phasor data concentrator (PDC) collects the data from PMUs. The PMUs connected to the remote buses send data (voltage and current phasors) to the PDC over different communication channels. The data helps in determining state and reward at every time step. The goal of this setup is to achieve good state observability of the whole system for RL control.

Furthermore, we design a controller that can produce accurate control action when the states and rewards are fed as input. Fig. 3.2 shows the zoomed-in structure of the controller, based on TD3. We utilize the neural networks for the representation of the actor and the critic. The backpropagation of the networks is realized by minimizing the loss function in Fig. 2.3. To mitigate the oscillations and prevent the overestimation of Q-values, we use a pair of actors and critics to

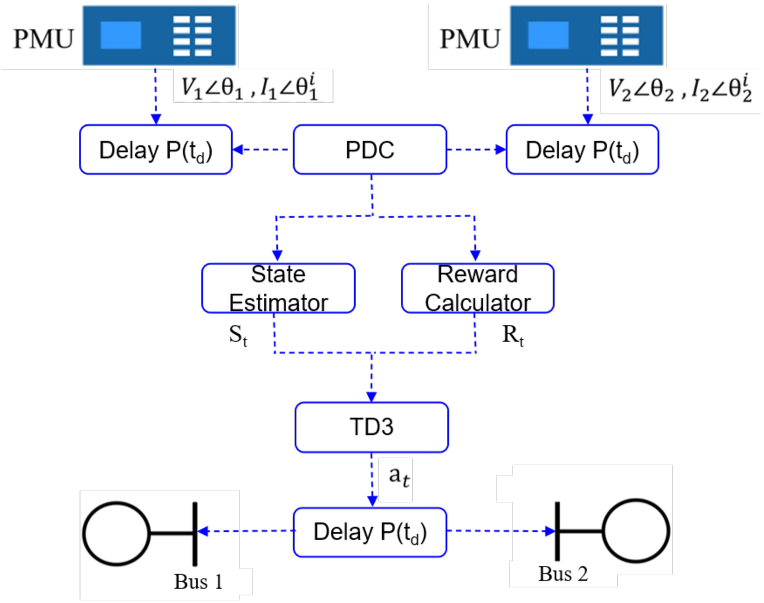

Figure 3.1: Framework of the Overall Scheme. The Dotted Lines Indicate the Communication Lines with Delays. The Inputs to the Controller Are the States and Reward, Whereas the Output is the Action.

form a TD3 algorithm. However, further work is needed for defining the states and the actions. In such a design, one of the challenges is the creation of the reward function that guides the agent to learn well. In the following subsections, we demonstrate how we design the RL controller.

3.1 State of the Controller

We start with the design of the states. The generator voltage, current, and phase angle are monitored through PMUs [30]. We define all the states s_t in the power system for all observable generators $g = 1, \cdots, G$ to be controlled. The generator speeds are represented as $\omega_{t,g}$, the deviations of generator speeds are $\Delta\omega_{t,g}$, the

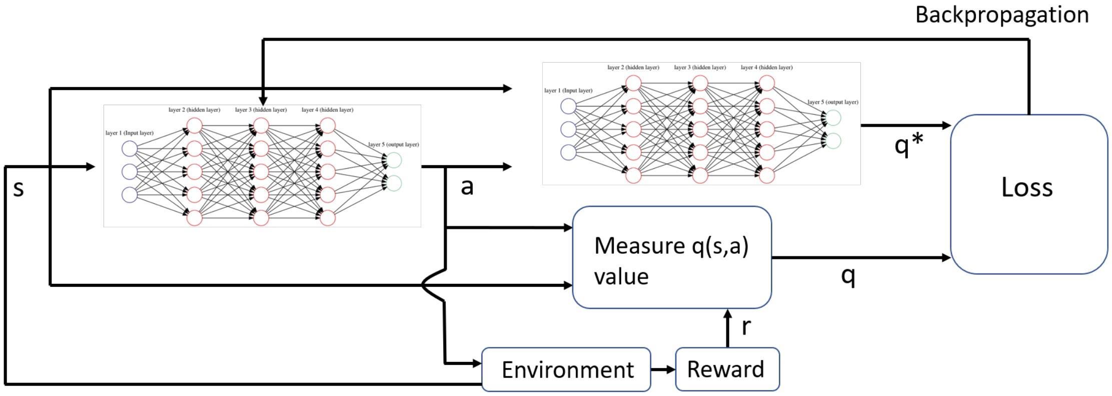

Figure 3.2: Zoomed-in Illustration of the Proposed Algorithm. The Dotted Lines Represent the Inputs and Output of the Controller.

phase angles are $\theta_{t,b}$ for the voltage of the bus $b = 1, \cdots, B$ at time $t = 1, \cdots, T$. As the speed of generators varies upon the occurrence of the disturbance, we define the speed deviations as

$$\Delta\omega_{t,g} = \omega_{t,g} - \omega_{t-1,g}. \tag{3.1}$$

The states are summarized as follows:

$$\begin{aligned} s_{t,1} &= \{\omega_{t,1}, \omega_{t,2}, \omega_{t,3}, \cdots, \omega_{t,G}\}, \\ s_{t,2} &= \{\Delta\omega_{t,1}, \Delta\omega_{t,2}, \Delta\omega_{t,3}, \cdots, \Delta\omega_{t,G}\}, \\ s_{t,3} &= \{\theta_{t,1}, \theta_{t,2}, \theta_{t,3}, \cdots, \theta_{t,B}\}, \\ s_T &= s_{t,1} \cup s_{t,2} \cup s_{t,3}. \end{aligned} \tag{3.2}$$

We design the state to directly capture the rotor speed, therefore, we include $s_{t,1}$. Meanwhile, we hope to be aware of the rotor speed deviation to monitor the direct results after an action. So, $s_{t,2}$ is designed. Since voltage angle is another significant factor that quantifies the state of the generators, we incorporate $s_{t,3}$ in the state.

3.2 Action of the Controller

After designing the controller's state, it is important to identify the control action. The power system stabilizer (PSS) is a device that measures improvements in system stability when added to a generator's automatic voltage regulator. The modern-day PSSs are responsible for damping down low-frequency oscillations by adjusting the voltage applied at the field windings $v_{t,g}$ of all the synchronous generators g. So, the output of the controller will essentially be an action vector a_t for all the generators g at time t. The action vector a_t, defined in equation (3.3), is a stabilizing voltage parameter that alters field voltage of synchronous generators. v_{t-1} in equation (3.4) is the variable where the action vector can get feedback from the previous time steps. This variable will help to have higher reward values and reduce system oscillation. Therefore, we identify the action space as follows:

$$a_t = \{v_{t,1}, v_{t,2}, v_{t,3}, \cdots, v_{t,G}\}. \tag{3.3}$$

By grouping the voltage signals, the learning agent can easily process and deliver the control signal to the automatic voltage regulator of each generator.

3.3 Reward Design for Enhanced Control Results

With states, actions, and policies clearly defined, we require a reward function, which helps in deciding the extent of the constancy of generated control action. We design a reward that can help maximize the information obtained from the wide-area-based observations from synchrophasors and local measurements from the generators. The goal is to minimize the frequency oscillations of wide-area systems. Therefore, we capture the variables like the rotor speed, its deviation, and phase angle variation between remote buses as well. To solve the high dimensionality and complexity of stability problems, we not only use variables for the power system but add more control effort from RL into the reward function to improve the performance. The form of reward function is shown in the following:

$$\begin{aligned} r_t = & - c_1 \sum_{g=1}^{G} (\omega_{t,g})^2 - c_2 \sum_{g=1}^{G} (\Delta\omega_{t,g})^2 \\ & - c_3 \sum_{\substack{i,j \in B \\ i \neq j, i < j}}^{B} (\theta_{t,i} - \theta_{t,j})^2 \\ & - c_4 \frac{T_s}{T_f} - c_5 \sum_{g=1}^{G} (v_{t-1,g})^2 . \end{aligned} \tag{3.4}$$

The five terms in equation (3.4) range from physical quantity associated reward (the first three terms) to the episode control (the fourth term) and the feedback of actions (the fifth term). The first term helps to bring the control ability of the speed of generators $\omega_{t,g}$. The second term overcomes the sustaining deviations in the speeds of the generators. The third term incorporates the difference between the phase angles of voltages at remote buses. We use the difference of the phase angles of remote buses to have better observation, because angle differences of remote buses were not observable without wide-area damping controls. We intend to reduce such a difference so that deviations of speeds of generators connected to remote buses are limited. The fourth term refers to the constant reward for preventing the termination of the episode due to zero reward. The fifth term in the equation (3.4) refers to the feedback for the action spaces from previous time steps. As we capture the major variables that impact the system stability, we add them into one reward function. A quadratic relationship among them is suitable since we want to have large enough differences for the learning agent to gain a reasonable reward and learn fast. The reward design is novel, since it quantifies the physical values, includes the episode control, and adds the feedback from the RL agent's action. Together, these innovations help to achieve superior control performance.

3.4 Twin-Delayed Deep Deterministic Policy Gradient Method

Conventional solutions for damping control are usually model-based. However, the parameter variation over time and the communication latency are two major issues for model-based solutions. Under this circumstance, RL algorithms are gaining popularity. But, some RL algorithms like Q-learning suffer from the overestimation issue in the Q-values. Therefore, based on the DDPG algorithm, we solve the overestimation issue by proposing a TD3 algorithm (refer to Algorithm 1), which is a model-free, off-policy RL method that uses deep neural networks to compute an optimal policy that maximizes the long-term reward.

In the RL-based controller design, we expect the RL-based controller to have some salient features. Firstly, the control should maintain a copy for each network, e.g., copies of the actor network and critic network. The copies keep on improving stability during the learning process. To achieve this, the algorithm inherits

Algorithm 1: TD3 [31]

```
Initialize critic networks $Q_{\theta_1}$, $Q_{\theta_2}$ and actor network $\pi_\phi$ with random parameters $\theta_1$, $\theta_2$, $\phi$ ;
Initialize target networks $\theta^{'}_1 \leftarrow \theta_1$, $\theta^{'}_2 \leftarrow \theta_2$, $\phi^{'} \leftarrow \phi$ ;
Initialize replay buffer $\mathcal{B}$ ;
for t = 1 to $\mathcal{T}$ do
    Select action with exploration noise $a \sim \pi_\phi(s)$ + $\epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward $r$ and new state $s^{'}$ ;
    Store transition tuple $(s, a, r, s^{'})$ in $\mathcal{B}$ ;
    Sample mini-batch of $\mathcal{N}$ transitions $(s, a, r, s^{'})$ from $\mathcal{B}$ ;
    $\tilde{a} \sim \pi_{\phi'}$(s) $+ \epsilon$, $\epsilon \sim$ clip ($\mathcal{N}$(0,$\tilde{\sigma}$), $-$c, c);
    $y \sim r$ + $\gamma$ $\min_{i=1,2} Q_{\theta'_i}$ ($s'$,$\tilde{a}$);
    Update critics $\theta_i \leftarrow min_{\theta_i} \frac{1}{M} \sum (y - Q_{\theta_i}(s, a))^2$ ;
    if t mod d then
        Update $\phi$ by the deterministic policy gradient ;
        $\nabla_\phi J(\phi) \approx \frac{1}{M} \nabla_a \, Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$ ;
        Update target networks;
        $\theta^{'}_i \leftarrow \tau \, \theta_i$ + $(1 - \tau) \, \theta^{'}_i$
        $\phi^{'} \leftarrow \tau \, \phi$ + $(1 - \tau) \, \phi^{'}$
    end
end
```

an actor-critic framework. This means that there are two components in the algorithm, the actor and the critic. The actor takes responsibility for a policy, which receives a state as the input and generates action. The critic estimates the action value function, which is used to assess the goodness of the actor.

Secondly, the controller should maintain a replay memory that stores all of the sample data during interaction with the environment. To store a certain amount of "experience", we randomly sample a batch of data from the replay memory and use them to train the networks at each time step. The replay memory removes the correlation in the sequence of the data sample. Using a deterministic policy is more stable than a stochastic policy where the actions are drawn from a distribution. Consequently, we design the controller using two deep neural networks, one each for the actor and the critic. Therefore, such a framework is powerful enough to work on tasks such as controlling communication delays in highly non-linear power systems.

Lastly, it is better for the controller to directly optimize the quantity of interest while remaining stable under function approximation (given a sufficiently small learning rate). This is easy to fulfill since we can use a deterministic policy gradient to train the actor network as follows:

$$\nabla_{\theta_\mu} J \approx \frac{1}{M} \nabla_a Q(s_i, a_i|\theta^Q) \nabla_{\theta_\mu} \mu(s_i|\theta^\mu). \tag{3.5}$$

Notably, the first feature maintains a copy for each network, e.g., copies of the actor network and critic network. The copies keep on improving the stability during the learning process. The second feature maintains a replay memory that stores all of the sample data during interaction with the environment. To store a certain amount of "experience", we randomly sample a batch of data from the replay memory and use them to train the networks at each time step. The replay memory removes the correlation in the sequence of the data sample. Using a deterministic policy is more stable than a stochastic policy where the actions are drawn from a distribution. The last feature indicates that TD3 directly optimizes the quantity of interest while remaining stable under function approximation (given a sufficiently small learning rate).

- The algorithm inherits an actor-critic framework. This means that there are two components in the algorithm, the actor and the critic. The actor takes responsibility for a policy, which receives a state as the input and generates action. The critic estimates the action value function, which is used to assess the goodness of the actor.

- The algorithm uses two deep neural networks, one each for the actor and the critic. Therefore, such a framework is powerful enough to work on tasks such as controlling communication delays in highly non-linear power systems.

- The algorithm uses a deterministic policy gradient to train the actor network as follows:

$$\nabla_{\theta_\mu} J \approx \frac{1}{M} \nabla_a Q(s_i, a_i|\theta^Q) \nabla_{\theta_\mu} \mu(s_i|\theta^\mu). \tag{3.6}$$

3.5 Avoid the Overestimation Bias

To make sure the control tasks are in a continuous action space, an actor-critic setting is adopted. In Double DQN, the authors propose using the target network as one of the value estimates, and obtain a policy by greedy maximization of the current value network rather than the target network. Unfortunately, if we borrow the above idea, the present and target value estimates are similar to each other since the policy in an actor-critic setting changes slowly, resulting in high bias. This should be avoided in the damping control. To address this issue, we utilize a clipped Double Q-learning variant which leverages the notion that a value estimate suffering from overestimation bias can be used as an approximate upper bound to the true value estimate [31]. This method is inspired by Double DQN [32], where the target network is used as one of the value estimates, and obtain a policy by greedy maximization of the current value network rather than the target network. When translated into the actor-critic environment, we update the present policy instead of the target policy with a pair of actors $(\pi_{\phi_1}, \pi_{\phi_2})$, critics $(Q_{\theta_1}, Q_{\theta_2})$, and the objective y:

$$y_1 = r + \gamma Q_{\theta'_2}(s', \pi_{\phi_1}(s')), \tag{3.7}$$

$$y_2 = r + \gamma Q_{\theta'_1}(s', \pi_{\phi_2}(s')). \tag{3.8}$$

To prevent the propagation of the overestimation when the smaller Q_θ has already overestimated the true value, the strategy to be taken is to use the biased Q_θ value as the upper bound of the less biased one [31]. This results in the clipped double Q-learning algorithm – the value function target is the sum of the experience reward r and the minimum discounted future reward from the critics:

$$y_1 = r + \gamma \min_{i=1,2} Q_{\theta'_i}(s', \pi_{\phi_1}(s')). \tag{3.9}$$

Chapter 4

NUMERICAL VALIDATION

In this section, we will firstly discuss the validation setup. Then, we demonstrate the performance of the proposed control agent with validation on the latency. In the end, a detailed comparison with the DDPG method is shown.

4.1 Benchmark System

Various simulation studies were carried out in benchmark systems like the 2-area and 4-generator Kundur system and the IEEE 39-bus 10-generator system. The performances are similar. Due to the space limit, we use mainly the Kundur's system in Fig. 4.1 as illustration. The test system consists of two fully symmetrical areas linked together by two 230 kV lines of 220 km length. This benchmark system was extensively used for low-frequency electromechanical oscillations study in large interconnected power systems because it mimics very closely the behavior of typical systems in actual operation [33].

Table 4.1 summarizes the benchmark system parameters. Both area 1 and 2 share an identical generator except for the inertia value. The benchmark system presents a stressed operating condition, where 413 MW is exported from area 1 to area 2. Meanwhile, the surge impedance loading of a single line is 140 MW.

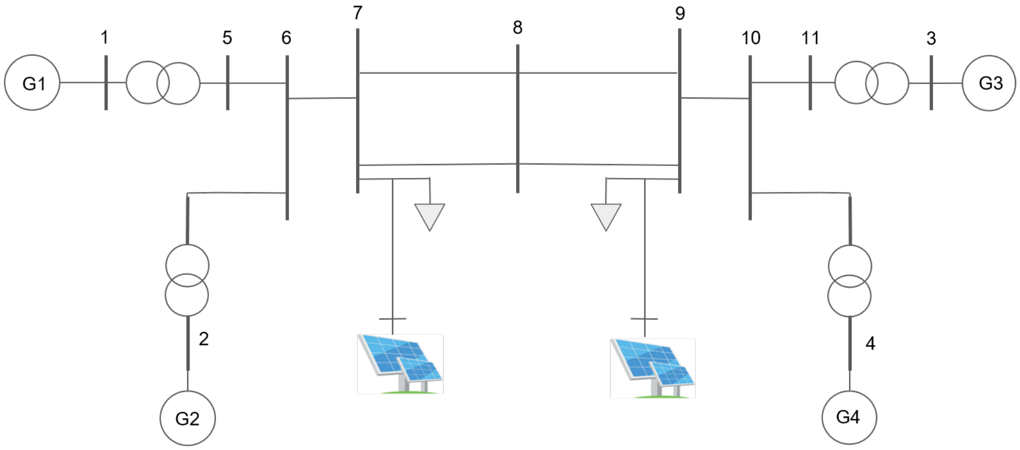

Figure 4.1: A Modified Kundur Four-machine Two-area Power System. In This System, to Represent the Deep Renewable Generation, Two Aggregated Residential PV Generation – PV1 and PV2 Are Added.

4.2 TD3 Control Agent: Fast Learning Curve

Since the reward functions are highly dependent on the existence of the oscillations, we aim to show the performance of a well-learned model when there are low-frequency oscillations, in case of different communication channel delays. With the proposed controller design in Section 3, we achieve the learning curve, as shown in Fig. 4.2. By looking at the average reward, it shows that the agent has good and bad attempts in the first five episodes. After that, the average reward is gradually increasing. As observed in the figure, after 100 episodes, the episode reward of the TD3 agent is observed to reach a value close to 0 – the highest value in the whole learning curve.

The high penalty is enforced in scenarios where the system loses synchroniza-

Table 4.1: Parameters of the Benchmark System under Study

Name	Value
Generator	20 kV/900 MVA
Synchronous machine inertia	6.5 s, 6.175 s
Thermal plant exciter gain	200
Solar capacity (PV1, PV2)	100 MW
Power exporting from area 1 to 2	413 MW
Area 2 power generation	700 MW

tion since such a case will be responsible for the collapse of the system. When the system loses synchronization in an episode, the agent learns associated parameters so that the system does not explore the outage of the system and learns what parameters to avoid next time. We consider such scenarios as game-over for the model and there is no further evaluation performed so that the training time can be curtailed. From episodes 0 to 40 in Fig. 4.2, we observe the process that the agent creates an appropriate policy that reflects proper damping of the low-frequency oscillations in the system. The model achieves a high average reward after sufficient exploration. After 40 episodes, the model converges to a very high reward. The effect of learning can also be observed from another perspective of implementing a control policy that maintains system synchronization for stable operation of the whole power system. Fig. 4.2 shows that after 100 episodes that high fidelity

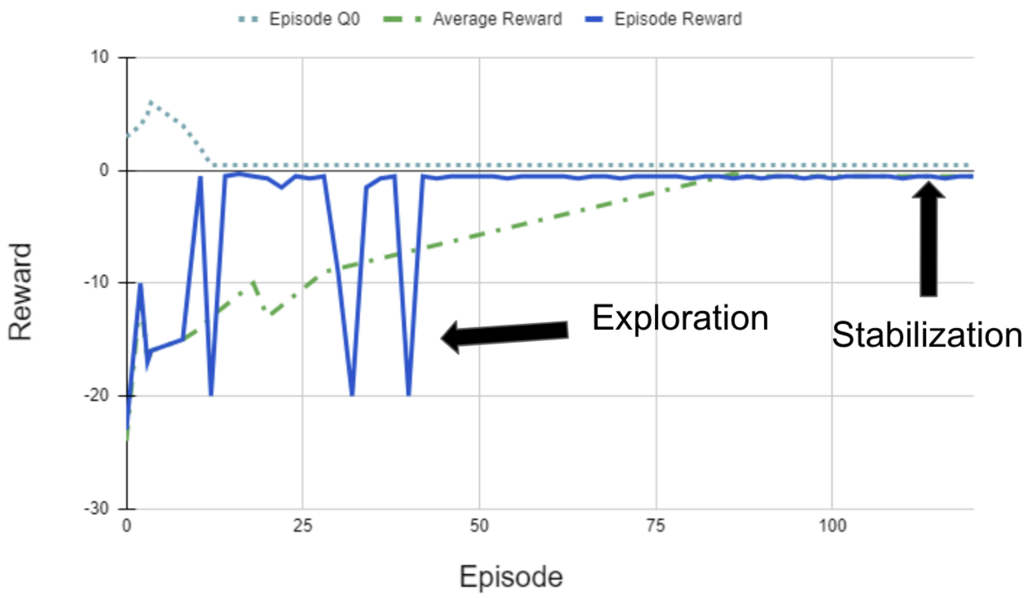

Figure 4.2: A Learning Curve That Shows the Individual Episode Reward and Average Reward. The X-axis and the Y-axis Are the Number of Episodes and the Reward Respectively. The Blue Line Means Episode Reward (the Reward for Each Episode). The Red Line Is an Average Reward (a Running Average Reward Value). And, the Yellow Line Is Episode Q0 (the Critic's Estimate of the Discounted Long-term Reward at the Start of Each Episode). The Setup Behind This Figure Is in Table 4.2.

control action enables the system to maintain stability.

The advantage of this reward function design is presented in Fig. 4.3, which clearly shows the necessity of the five terms in the reward equation. When there are less than three terms in the reward function design, the learning curve cannot be accomplished. With exactly the first three terms in equation (3.4), we can see that the learning curve in green in Fig. 4.3 is much worse than that in blue – with five terms. The proposed TD3 technique with five terms reward function gains higher reward values in general, which means the system gets into the stabilization condition faster than TD3 with three terms only.

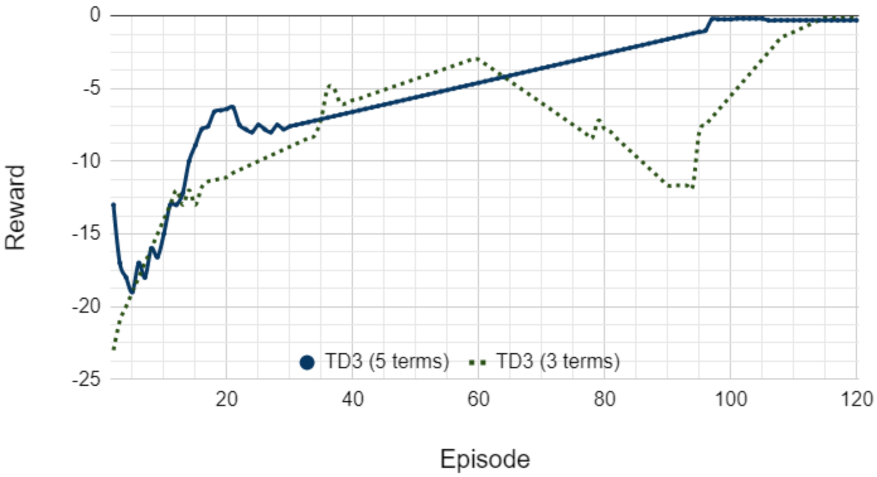

Figure 4.3: Result Comparison Using the Proposed TD3 Algorithm under 3 terms and 5 terms Reward Function, Where 3 Terms Refer to the First Three Terms in Equation (3.4), and 5 Terms Refer to the Entire Equation (3.4). The Reward Here Refers to the Running Average Reward Value.

4.3 Control Agent: Robust to Communication Latency

We analyze the performance based on the variation of average communication delay in the system. The communication delays in wide-area power systems can range from tens to several hundred milliseconds or more. Appendix A shows the delay range under different types of communication media. Based on this range, four scenarios that fully capture the delay range are created between $0.13 \sim 0.19$ seconds. The details of the four testing scenarios are shown in Appendix B.

If the mean and variance are changed inside the system and if a large number of signals are to be routed, then there is a potential to experience long delays and considerable variability (or uncertainty) in these delays. Fig. 4.4 shows that not only the TD3 agent with the proposed controller has achieved a high fidelity in controlling the generators, but also the performance based on the variation of average communication delay successfully damps down the oscillations.

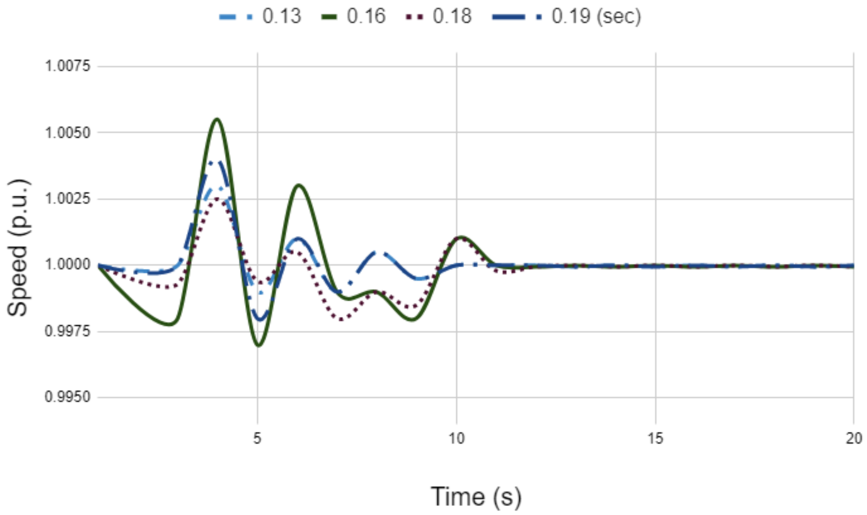

Figure 4.4: The Control Performance Based on the Variation of Average Communication Delay. Due to the Stochastic Nature of the Proposed RL Algorithm, the Results under Different Time Delays Are Not Always the Same. This Figure Is a Representative Result Selected Among Extensive Simulations. However, the Largest Speed Deviation Is Always Well Contained in All Test Cases.

As discussed in the literature, conventional methods suffer from the latency issue during their control efforts. Through multiple simulations, we observe that the performance of the proposed control agent is no longer dependent on the communication delay due to the fact that the largest time latency is not always associated with highest speed deviation. Fig. 4.4 shows one of the selected control results, where, unlike conventional methods, the shortest time delay does not result in the least amount of speed deviation. Here, the yellow line with 0.18 seconds of time delay presents the smallest speed deviation. The results show that the proposed RL agent decouples its performance with the time latency in communication. It is the hyper-parameters in the RL algorithm that affect the damping performance.

4.4 Performance Comparison with DDPG

The compared results between TD3 and DDPG agents are demonstrated in Fig. 4.5, including all the agent discount factors and batch sizes in Table 4.2. Evidently, the TD3 agent achieves the optimal policy faster than the DDPG agent has. DDPG agent could not converge into the stabilization condition in limited episodes. However, it has only several explorations before reaching out to the optimal policy. In other words, there could be the case that the agent might not learn the parameters.

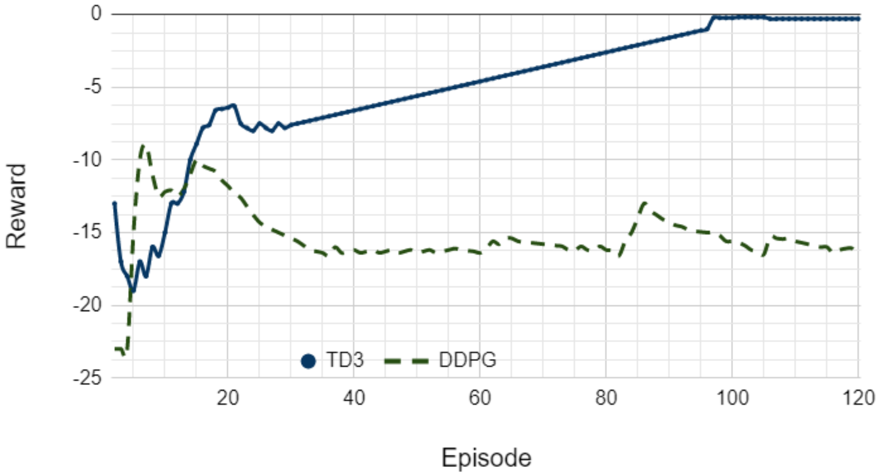

Figure 4.5: Result Comparison Between DDPG and TD3 Agents. The Parameters of Both Agents Are Presented in Table 4.2. The Reward Here Refers to the Reward for Each Episode.

Table 4.2: Parameters of TD3 and DDPG Agents

Name	Value
Input dimension	12 states
Output dimension	4 actions
Discount factor	0.75
Sample time	0.1 (sec)
Experience memory capacity	$500,000$
Batch size	64 samples

Chapter 5

BIG DATA ANALYSIS TO SMART METER STATUS MONITORING

The goal aims at developing a new tool that utilizes machine learning algorithms to visualize the big data, dig into the hidden values, and identify the performance trends. Specifically, utility is interested in developing software to give indications to mismatch records, meter communication issues, special (suspect) events and cluster the customers based on the load profiles and various sources of other information. For instance, some meters are suspected to be connected reversely, some customers are categorized in the wrong group, some meters lost their communication and some smart meters are suspected to be connected to the wrong place. The most severe and difficult to detect are the meters that are connected in a reverse way or the wrong place.

To solve this issue, supervised and unsupervised machine learning methods are integrated into the software for load forecasting, customer behavior analysis and event type differentiation. The other main contribution of the thesis is to integrate various cutting-edge machine learning methods into the software to help utility handle their data and understand the demands of their customers. Data integration techniques are aimed to sort out the data from different sources. Correlation analysis is used to reduce the redundancy of the data and reduce the size of the data. A rational check utilizing machine learning methods based on the constraints of professional knowledge applied to the abnormal data to provide indications of the possible reasons for the abnormality. For the missing values, the users can choose to fill the blanks by zero or most frequency constant value, the mean or median of the data, or the K-nearest neighbor (KNN) method. The pre-processed data can later be used to predict load, solar generation and cluster different types of customers. Support vector regression (SVR), polynomial regression, long-short term memory (LSTM) and other machine learning methods are selected to predict renewable generation and load forecasting which help utilities to better manage the grid and prepare for unexpected severe two-way power flows. K-means clustering and density-based spatial clustering of applications with noise (DBSCAN) are used on the load profile to give the utility a comprehensive understanding of current energy management and tariff structure. The regularity and irregularity of customer energy consumption were analyzed in depth.

Therefore, in this thesis, the four different main functions which are data visualizations, distribution visualization, bad data indicator and behavior analysis are developed in the software to visualize smart meter data in time series.

Chapter 6

DATA VISUALIZATIONS

The abundance of smart meters enables utilities to gain more visualizations of the smart meter readings. By monitoring the time series data, people have a high-level overview of the data and can quickly locate obvious problems. However, the smart meter data are mostly used for billing and there is a lack of an adaptable tool for utilities or customers to visualize the time series data. Therefore, in this section, we developed a section in the software to visualize the time series data from the smart meters. We show the times series visualization result.

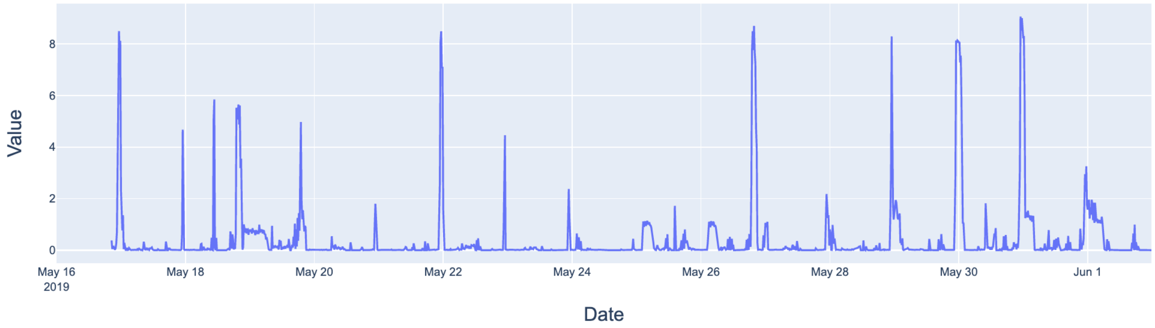

Figure 6.1: The Location Number at 231731005 in Time Series

In the data visualization section, the visualization of the smart meter data from user-selected properties is demonstrated. The user chose to visualize the solar meter ”kWh - Delivered” readings from the customer with location number ”231731005”.

Chapter 7

DISTRIBUTION VISUALIZATIONS

With the abundance of smart meters, it becomes more important for utilities to handle and analyze the information refined from the smart meter readings. Such information can be used to check the correctness of the model and quickly locate the suspicious events, which is critical for the utility to run a safer and more reliable grid. However, the raw data often contain a lot of bad data. Directly using the raw data for analysis will result in inaccurate results. Therefore, in this section, the insights into the existence of bad data from the statistical point of view are demonstrated. Specifically, the data distribution is analyzed.

The statistical analysis of the data distribution is not always consistent with our understanding. By examining the data distribution, we can determine if the data distribution obeys the normal distribution which determines the model we plan to use for other analyses. Histogram and boxplot are two types of graphical tools that are frequently used for analyzing the distribution of the data. The two types of the plot provide information of the median, quantile, upper limit, lower limit, overall data variability and outliers, etc. This information is of great importance to understand the data.

In order to show the overall distribution of the smart meter data, the user chose the ”CHNM” to be kWh - Delivered. Hence, the histogram in Fig. 7.1 and boxplot in Fig. 7.2 are showing the distribution information of all the ”kWh - Delivered” measurements from all the smart meters.

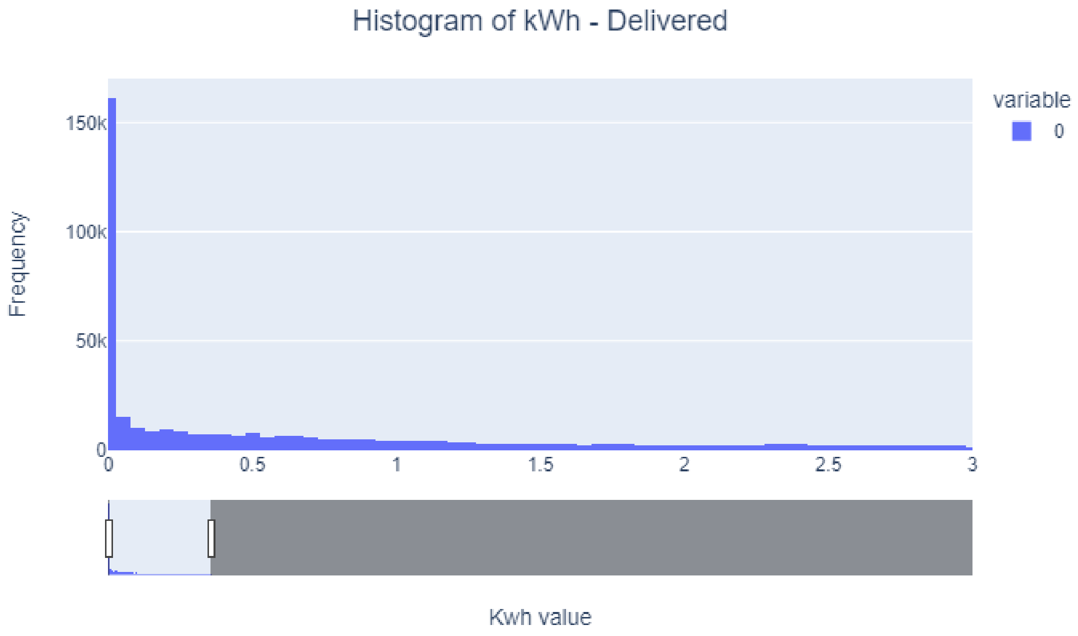

Figure 7.1: Histogram of Kwh - Delivered Smart Meter Data Distribution

As can be seen from Fig. 7.1, the range with the highest frequency is the most common value in the dataset, this information can help us to determine if most of the voltage values are lying in the specified range. We can also check if the data is skewed. When most of the data is located on the left side, we call the distribution left-skewed distribution, which means that the mean value of the voltage is smaller than the median value. This indicates that there is a portion of low voltage values we should pay attention to. When most of the data is located on the right side, we

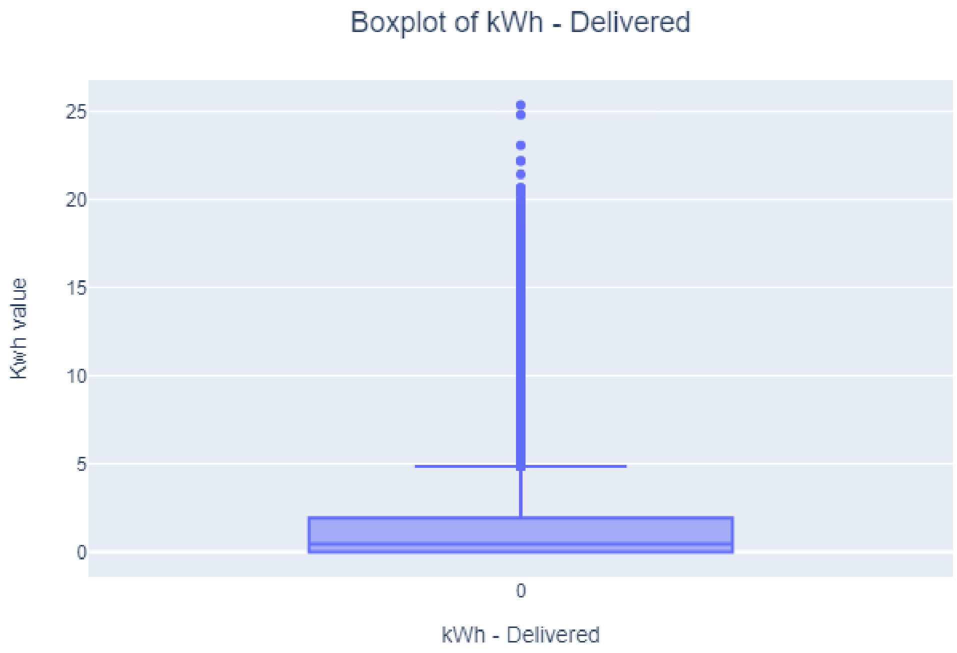

Figure 7.2: Boxplot of Kwh - Delivered Smart Meter Data Distribution

call the distribution right-skewed distribution, which means that the mean value of the voltage is larger than the median value. This indicates that there is a portion of high voltage values we should pay attention to. We can also roughly observe if there are abnormal values from the histogram. If we observe the boxplot there are any isolated bars at two ends that are far away from the peak, we can consider them as abnormal data. The abnormal data and the spread of the data can be better examined and visualized in Fig. 7.2. All the points that are marked with dots can be considered as abnormal data. For the small violations, we can see if we need to strengthen the regulation. For a large violation, we need to extract the case and figure out the reason. This distribution visualization section tries to provide the distribution information of the raw data. One can quickly know if there exists bad data by checking if there are any data that are far away from the mean or median value.

Chapter 8

BAD DATA INDICATOR

Data can be divided into two classes, one is descriptive data, the other is numerical data. The software tries to find the bad data separately for these two classes. Specifically, for the descriptive data, we consider two cases, data inconsistency and data duplication. Detailed explanations for the two cases can be found in Section 8.1 and Section 8.2. For the numerical data, we consider two different types of data, voltage and usage data. We tried to find the violations for the two types of data. Details are illustrated in Section 8.3 and Section 8.4.

8.1 Incorrect Battery Configuration from Database

In this section, we pulled out inconsistent information on the battery configuration from the database file. As we know, for battery configuration, ″1B″ is AC coupled system (DER only, with backup load panel), ″1C″ is AC coupled system (DER only, with no backup load panel), ″2A″ is AC coupled system (DER storage & DER generation, with no backup load panel), ″2B″ is AC coupled system (DER storage & DER generation, with backup load panel), ″3A″ is DC coupled system (DER storage & DER generation, with backup load panel), ″3C″ is DC coupled system (DER storage & DER generation, with no backup load panel). We compare the ″BatteryConfig″ in file ″Battery Info.xlsx″ with the definitions in the last sentence and pulled out all the records that do not match. The results are summarized in Table 8.1. We take the first row as an example to illustrate the results. As can be seen from the first row Table 8.1, the customer with location number ”659250008” and with battery configuration ”2B” is documented to be ”DC-Coupled”. However, it should be ”AC-Coupled” by definition.

Table 8.1: Inconsistent Battery Configurations in the Database File

LocationNumber	EssDerConfiguration	BatteryConfig
659250008	2B	DC-Coupled
291630003	4A	AC-Coupled

As can be seen from the table, there are two inconsistent battery configuration records.

8.2 Duplicated Battery Meter Records

In this section, we pulled out all the duplicated smart meter readings of the battery meter from the uploaded file. The duplicated smart meter readings mean that there are at least two records from the same customer, the same meter, the same parameter (”kWh - Delivered”, ”kWh - Received”, or ”Voltage Phase A”) on the same date, but with different values of the readings. We print out all the information about the duplicated readings except the data themselves in Table 8.2.

In Table 8.2, the first column of the table shows the location number of the customer being measured. The third column is an explanation for the code in

Table 8.2: Duplicated Battery Meter Readings Table Information

LOCATN	APCODE	METER	CHNM	DATE
40111006	B	DER STORAGE	kWh - Delivered	2019-05-11
40111006	B	DER STORAGE	kWh - Received	2019-05-11
40111006	B	DER STORAGE	Voltage Phase A	2019-05-11
40111006	B	DER STORAGE	kWh - Received	2019-05-12
40111006	B	DER STORAGE	kWh - Delivered	2019-05-12
40111006	B	DER STORAGE	Voltage Phase A	2019-05-12

the second column, which shows the name of the meter. There are three options for column ”CHNM” (”kWh - Delivered”, ”kWh - Received”, or ”Voltage Phase A”), which give the parameter that the meter measures. Finally, the last column shows the date that has duplicated smart meter readings. As can be seen from the table, there are 66 duplicated records. The customer who has the most frequent duplicated battery meter records is the customer with ”Location K”: 40111006.

8.3 Voltage Violation

The last two sections have discussed the bad data for descriptive data. Next, in the following two sections, we focus on indicating the bad data for numerical data. Specifically, we try to find voltage violations in this section. As we know, one of the main responsibilities of the everyday operation of utilities is to ensure voltages within regulations when supplied to customers. However, in the new regime with renewable and battery systems, the voltage limits are frequently violated. The voltage fluctuations will cause a life span shortage of most electrical and electronic equipment, which decreases the economic effects. A too low voltage will lead to a high current which increases the loss of the system. Therefore, it is of great importance for a utility to know when and where the voltage violates. To locate the voltage violations, we determine that the voltage should be within a certain range of its mean value. By preliminary studying the voltage data, we found the mean value of the voltage is around 980 (V). Hence, we set the voltage in the range [980 - threshold, 980 + threshold] to be normal, any value that exceeds the upper bound or the lower bound is considered to be abnormal. Currently, the threshold is set to be 160(V) by the user, therefore, the normal range is between [820, 1140] (V). We showed the results in two ways, one is a plot of the time series data shown in Fig. 7.1, and the other is a table of the necessary information other than the data shown in Fig. 7.1. The x-axis represents the time index of a day. There are 96 indices, which means that the data is measured with 15-minute-interval. The y-axis represents the voltage value and the unit is (V). The different curves show the daily smart meter readings that have at least one violation. There are also two dashed lines showing in the plot to indicate the area of the normal range. The legend contains the information of the smart meter reading, which is also given in Table 8.3.

The first column of the table shows the location number of the customer being

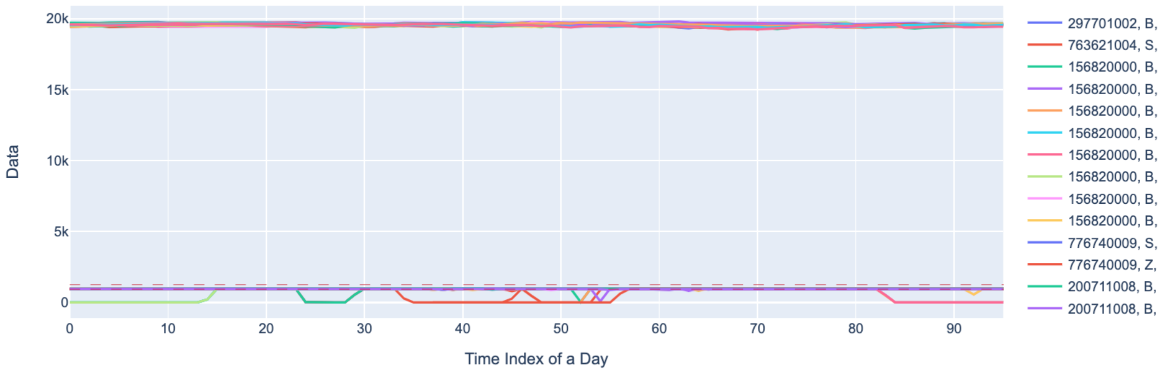

Figure 8.1: Voltage Violation Visualization of Smart Meter Data in Time Series

Table 8.3: Voltage Violations Table Information

LOCATN	APCODE	METER	CHNM	DATE
297701002	B	DER STORAGE	Voltage Phase A	2019-05-01
763621004	S	DER GEN	Voltage Phase A	2019-05-08
156820000	B	DER STORAGE	Voltage Phase A	2019-05-12
156820000	B	DER STORAGE	Voltage Phase A	2019-05-13
156820000	B	DER STORAGE	Voltage Phase A	2019-05-11
156820000	B	DER STORAGE	Voltage Phase A	2019-05-23

measured. The third column is an explanation for the code in the second column, which shows the name of the meter. The fourth column verifies that all the meters are measuring the voltage phase A. Finally, the last column shows the date the violation happens.

As can be seen from Table 8.3, there are 174 smart meter readings that violate the voltage threshold. Most of the violations happen on the customer with ”Location

K”: 521460009 and on meter:”BILLING”.

8.4 Battery Kwh Violation

In the last section, we check the violations of voltages that have potential risks to the power systems. In this section, we examine the battery usage value to check if there are meters that lose connection, are under maintenance, or are damaged, or if the batteries of the customers are damaged. Specifically, we pulled out all the daily smart meter readings whose readings never exceed the threshold. The threshold and the measurement type are set by the user and the threshold is set to be 0.06(kWh) and the measurement type is set to be ”kWh - Delivered”. The results are presented in two ways, one is a plot of the time series data shown in Fig. 7.2 the other is a table of the necessary information other than the data shown in Table 8.4 . In Fig. 7.2 the x-axis represents the time index of a day. There are 96 indices, which means that the data is measured with 15-minute-interval. The y-axis represents the usage value and the unit is (kWh). The different curves show the daily smart meter readings that are always under the threshold. There are dashed lines indicating the threshold in the figure. The legend contains the information of the smart meter reading, which is also given in Table 8.4.

Table 8.4 provides the information of the daily smart meter readings that never exceed the threshold. The first column of the table shows the location number of the customer being measured. The third column is an explanation for the code in the second column, which shows the name of the meter. The second and third

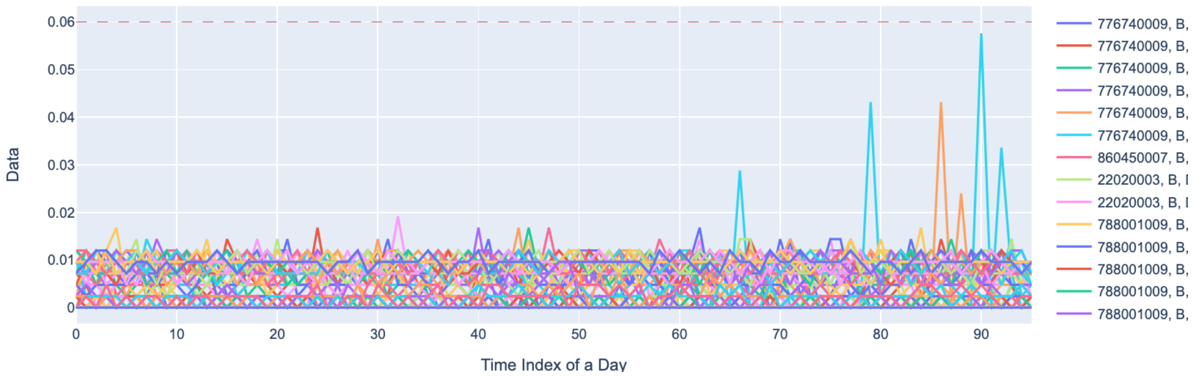

Figure 8.2: Battery Kwh Violation Visualization of Smart Meter Data in Time Series

Table 8.4: Low Usage Readings Table Information

LOCATN	APCODE	METER	CHNM	DATE
776740009	B	DER STORAGE	kWh - Delivered	2019-06-01
776740009	B	DER STORAGE	kWh - Delivered	2019-05-26
776740009	B	DER STORAGE	kWh - Delivered	2019-05-18
776740009	B	DER STORAGE	kWh - Delivered	2019-05-25
776740009	B	DER STORAGE	kWh - Delivered	2019-05-27
776740009	B	DER STORAGE	kWh - Delivered	2019-05-19

columns can also be used to verify that we are examining the value of the battery meter. The fourth column verifies that all the meters are measuring the ”kWh - Delivered”. Finally, the last column shows all the dates that the battery meter readings never exceed the threshold.

As can be seen from Table 8.4, there are 171 battery meter readings never exceed the threshold. Most of them happen on the customer with ”Location K”: 659250008

This report presents the bad data detected by the software. The bad data are determined from two aspects, one is from the descriptive information, the other is from the numerical information. For the descriptive information, we identified the inconsistent information from the database and found the duplicated readings with different values. For the numerical data, we found the battery voltage violations and battery kWh violations. The results show that there are 2 inconsistent battery configuration records and 66 duplicated records. The customer who has the most frequent duplicated battery meter records is the customer with ”Location K”: 40111006. Among all the readings, 174 smart meter readings violate the voltage threshold. Most of the violations happen on the customer with ”Location K”: 521460009 and on meter: ”BILLING”. There are 171 battery meter readings never exceed the threshold. Most of them happen on the customer with ”Location K”: 659250008.

Chapter 9

BEHAVIOR ANALYSIS

Understanding the customer behavior pattern can help utilities to determine the customers with abnormal behaviors and design demand response programs for targeted behavior patterns. The auxiliary work can help utilities to manage the demand peak and balance the dynamic supply and demand. Therefore, in this section, we tried to detect the customers with abnormal behavior patterns from the perspective of behavior analysis. The results are shown in the following sections

In this section, we showed the results of clustering the behavior patterns of customers with a specific battery configuration and rate plan in summer or winter season. Specifically, the user chose to analyze the behavior pattern in summer of customers with battery system configuration being ″2B″ and rate plan being ″E-27″. We grouped the behavior patterns into three clusters, which are shown in Fig 9.1, 9.2, and 9.3. The x-axis represents the time index of a day. There are 96 indices, which means that the data is measured with 15-minute-interval. The y-axis represents the usage value. The positive y-axis means the battery received kWh, the negative y-axis means the battery delivered kWh. The dashed curves summarize the typical pattern of the group.

The algorithm integrated into the software will automatically select the first

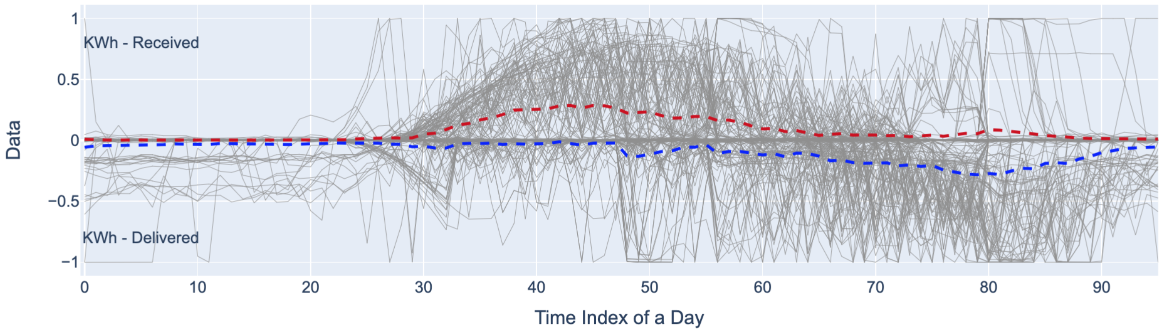

Figure 9.1: The First Most Typical Behavior Pattern Analysis

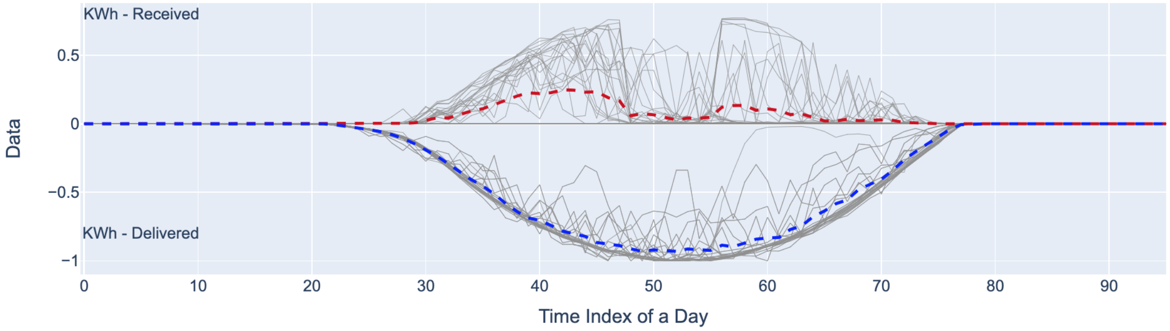

Figure 9.2: The Second Most Typical Behavior Pattern Analysis

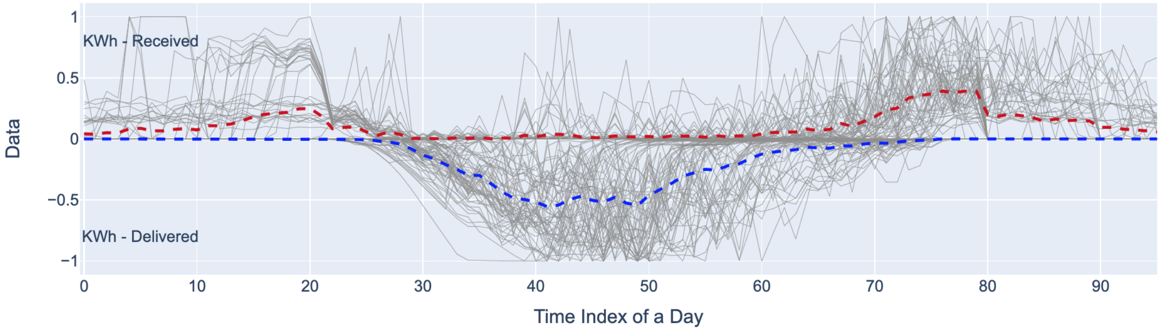

Figure 9.3: The Third Most Typical Behavior Pattern Analysis

three most abnormal customers out of each behavior pattern to visualize. The abnormal behaviors are determined by calculating the distance of the 10 percent quantile of the usage readings of each customer to the cluster center of the corresponding behavior pattern. If there are enough customers for us to conduct the analysis, in other words, the number of the customers in the behavior pattern is larger than 8, we choose the three with the largest distance to be the customers with abnormal behaviors. The daily readings from the three customers for each behavior pattern will be shown separately in the following figures. If there are not enough customers for analysis, we will print out the number of customers in the behavior pattern below. First, we show the results for the first behavior pattern. If there are figure results, the x-axis represents the time index of a day. There are 96 indices, which means that the data is measured with 15-minute-interval. The y-axis represents the usage value. The positive y-axis means the battery received kWh, the negative y-axis means the battery delivered kWh. The dashed curves summarize the typical pattern of the group.

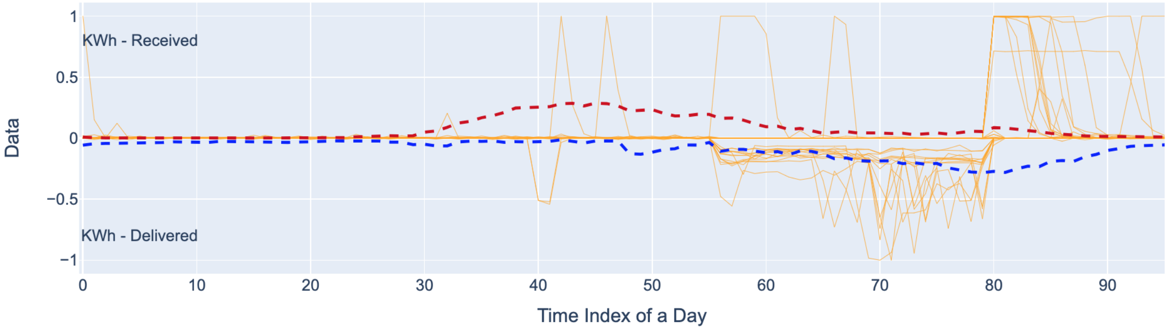

Figure 9.4: The First Abnormal Customer Analysis in the First Behavior Pattern

Next, we show the results for the second behavior pattern. If there are figure results, the x-axis represents the time index of a day. There are 96 indices,

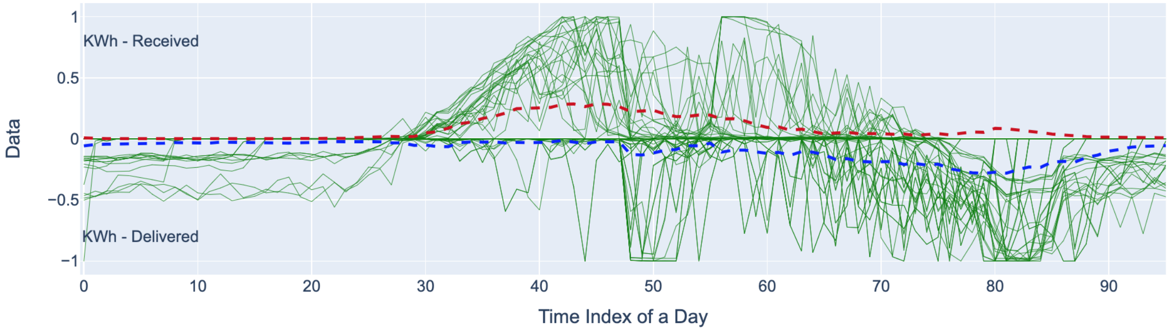

Figure 9.5: The Second Abnormal Customer Analysis in the First Behavior Pattern

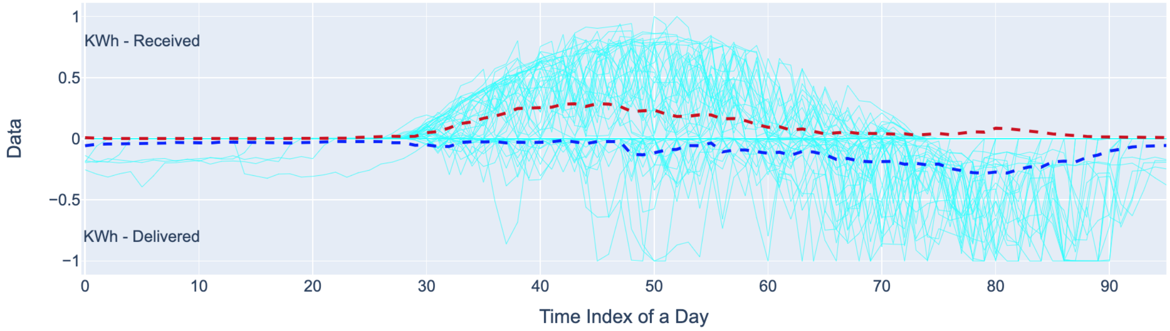

Figure 9.6: The Third Abnormal Customer Analysis in the First Behavior Pattern

which means that the data is measured with 15-minute-interval. The y-axis represents the usage value. The positive y-axis means the battery received kWh, the negative y-axis means the battery delivered kWh. The dashed curves summarize the typical pattern of the group. The results showed that the data is insufficient for analysis. The number of customers in the second behavior pattern is 1. The location number of the customers is: ['40111006,']. Finally, we show the results for the third behavior pattern. If there are figure results, the x-axis represents the time index of a day. There are 96 indices, which means that the data is measured with 15-minute-interval. The y-axis represents the usage value. The positive y-axis means the battery received kWh, the negative y-axis means the battery delivered kWh. The dashed curves summarize the typical pattern of the group. The results showed that the data is insufficient for analysis. The number of customers in the third behavior pattern is 5. The location number of the customers are: [’231731005,’ ’372150007,’ ’393550001,’ ’40111006,’ ’952721004,’].

This section first showed the behavior analysis results for the selected battery configuration, rate and season. Next, when data is sufficient for analysis, by using the information concluded in the behavior analysis, we detect the abnormal customers in each behavior and showed the daily battery meter readings

Chapter 10

CONCLUSION

There is a rising interest in the stability of power systems in order to balance the distribution model and offer reliable electricity to customers. The idea is to take a power system with uncertain communication latency and then create successful damping by optimally anticipating load imbalances in the power system. This is investigated through the insight of machine learning, which can solve problems of instability in large systems. This ability is studied by implementing a twin-delayed deep deterministic policy gradient algorithm which takes all the uncertainties of the unbalanced systems into account. In this way, the system is optimally explored to damping down frequency oscillations while keeping the system's balance within defined limits. We show that if a twin-delayed deep deterministic policy gradient algorithm is used then low-frequency oscillation can be significantly improved in comparison to existing algorithms. The simulation results are presented to verify the validity and effectiveness of the proposed control strategy.

Also, after conducting data visualization, data analytics and evaluation of all the cutting edge machine learning methods, we built the interface and structure of the software. The regularity and irregularity of customer energy consumption were analyzed in depth.

REFERENCES

[1] R. V. Yohanandhan and L. Srinivasan, “Decentralized wide-area neural network predictive damping controller for a large-scale power system,” in *IEEE International Conference on Power Electronics, Drives and Energy Systems*, 2018, pp. 1–6.

[2] K. Prasertwong, M. Nadarajah, and D. Thakur, “Understanding low-frequency oscillation in power systems,” *International Journal of Electrical Engineering Education*, vol. 47, 2010.

[3] S. P. Azad, R. Iravani, and J. E. Tate, “Damping inter-area oscillations based on a model predictive control HVDC supplementary controller,” *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 3174–3183, 2013.

[4] M. Klein, G. J. Rogers, and P. Kundur, “A fundamental study of inter-area oscillations in power systems,” *IEEE Transactions on Power systems*, vol. 6, no. 3, pp. 914–921, 1991.

[5] I. Zenelis and X. Wang, “Wide-area damping control for interarea oscillations in power grids based on pmu measurements,” *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 719–724, 2018.

[6] M. E. Aboul-Ela, A. A. Sallam, J. D. McCalley, and A. A. Fouad, “Damping controller design for power system oscillations using global signals,” *IEEE Transactions on Power Systems*, vol. 11, no. 2, pp. 767–773, 1996.

[7] S. Zhang and V. Vittal, “Design of wide-area power system damping controllers resilient to communication failures,” *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4292–4300, 2013.

[8] I. Kamwa, R. Grondin, and Y. Hebert, “Wide-area measurement based stabilizing control of large power systems-a decentralized/hierarchical approach,” *IEEE Transactions on Power Systems*, vol. 16, no. 1, pp. 136–153, 2001.

[9] J. Ma, T. Wang, Z. Wang, and J. S. Thorp, “Adaptive damping control of inter-area oscillations based on federated kalman filter using wide area signals,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1627–1635, 2013.

[10] I. Erlich, A. Hashmani, and F. Shewarega, “Selective damping of inter area oscillations using phasor measurement unit signals,” in *IEEE Trondheim PowerTech*, 2011, pp. 1–6.

[11] Y. Hashmy, Z. Yu, D. Shi, and Y. Weng, “Wide-area measurement system-based low frequency oscillation damping control through reinforcement learning,” *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5072–5083, 2020.

[12] T. L. Vu and K. Turitsyn, “Lyapunov functions family approach to transient stability assessment,” *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1269–1277, 2016.

[13] M. Mokhtari, F. Aminifar, D. Nazarpour, and S. Golshannavaz, “Wide-area power oscillation damping with a fuzzy controller compensating the continuous communication delays,” *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1997–2005, 2013.

[14] T. Surinkaew and I. Ngamroo, “Inter-area oscillation damping control design considering impact of variable latencies,” *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 481–493, 2019.

[15] S. S. Yu, T. K. Chau, T. Fernando, and H. H.-C. Iu, “An enhanced adaptive phasor power oscillation damping approach with latency compensation for modern power systems,” *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4285–4296, 2018.

[16] D. Roberson and J. F. O’Brien, “Variable loop gain using excessive regeneration detection for a delayed wide-area control system,” *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6623–6632, 2018.

[17] H. Liu, L. Zhu, Z. Pan, F. Bai, Y. Liu, Y. Liu, M. Patel, E. Farantatos, and N. Bhatt, “Armax-based transfer function model identification using wide-area measurement for adaptive and coordinated damping control,” *IEEE Transactions on Smart Grid*, vol. 8, no. 3, pp. 1105–1115, 2017.

[18] A. Vahidnia, G. Ledwich, and E. W. Palmer, “Transient stability improvement through wide-area controlled svcs,” *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3082–3089, 2016.

[19] X. Y. Bian, Y. Geng, K. L. Lo, Y. Fu, and Q. B. Zhou, “Coordination of psss and svc damping controller to improve probabilistic small-signal stability of power system with wind farm integration,” *IEEE Transactions on Power Systems*, vol. 31, no. 3, pp. 2371–2382, 2016.

[20] K. Zhang, Z. Shi, Y. Huang, C. Qiu, and S. Yang, “SVC damping controller design based on novel modified fruit fly optimisation algorithm,” *IET Renewable Power Generation*, vol. 12, no. 1, pp. 90–97, 2018.

[21] S. Jhang, H. Lee, C. Kim, C. Song, and W. Yu, “ANN control for damping low-frequency oscillation using deep learning,” in *IEEE Australasian Universities Power Engineering Conference*, 2018, pp. 1–4.

[22] J. Duan, H. Xu, and W. Liu, “Q-learning-based damping control of wide-area power systems under cyber uncertainties,” *IEEE Transactions on Smart Grid*, vol. 9, no. 6, pp. 6408–6418, 2018.

[23] S. Araghi, A. Khosravi, M. Johnstone, and D. C. Creighton, “A novel modular Q-learning architecture to improve performance under incomplete learning in a grid soccer game,” *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 2164–2171, 2013.

[24] P. Das, D. H. Behera, and B. Panigrahi, “Intelligent-based multi-robot path planning inspired by improved classical Q-learning and improved particle swarm optimization with perturbed velocity,” *Engineering Science and Technology, an International Journal*, vol. 19, pp. 651–669, 2016.

[25] S. Shinohara, T. Takano, H. Takase, H. Kawanaka, and S. Tsuruoka, “Search algorithm with learning ability for mario AI – combination A* algorithm and Q-learning,” in *Australasian Conference on Information Systems International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, 2012, pp. 341–344.

[26] Q. Cui, S. M. Y. Hashmy, Y. Weng, and M. Dyer, “Reinforcement learning based recloser control for distribution cables with degraded insulation level,” *IEEE Transactions on Power Delivery*, 2020.

[27] H. Yao, C. Szepesvari, R. S. Sutton, J. Modayil, and S. Bhatnagar, “Universal option models,” in *Advances in Neural Information Processing Systems*, 2014, pp. 990–998.

[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. M. O. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *International Conference on Learning Representations*, 2016.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, p. 529, 2015.

[30] L. Simon, K. S. Swarup, and J. Ravishankar, “Wide area oscillation damping controller for dfig using wams with delay compensation,” *IET Renewable Power Generation*, vol. 13, no. 1, pp. 128–137, 2019.

[31] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.

[32] H. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in *Association for the Advancement of Artificial Intelligence*, 2016, p. 2094–2100.

[33] I. Kamwa, G. Trudel, and L. Gerin-Lajoie, “Robust design and coordination of multiple damping controllers using nonlinear constrained optimization,” *IEEE Transactions on Power Systems*, vol. 15, no. 3, pp. 1084–1092, 2000.

APPENDIX A

DELAYS DUE TO DIFFERENT COMMUNICATION LINKS

Delays due to different communication links are provided in Table A.1. This table includes fiber-optic cables, microwave links, power line carriers, telephone lines, and satellite links. The one-way delay ranges from 100 ms to 700 ms.

Table A.1: Delays Due to Different Communication Links

Communication Link	One-way delay (ms)
Fiber-optic cables	$\approx$ 100-150
Microwave links	$\approx$ 100-150
Power line carriers	$\approx$ 150-350
Telephone lines	$\approx$ 200-300
Satellite links	$\approx$ 500-700

APPENDIX B

THE PARAMETERS OF FOUR TESTING SCENARIOS

Four testing scenarios that are used in the result section are listed here. They include four representative latency cases under different mean and variance of the signals.

Table B.1: Mean and Variance Communication Delay Using the Gaussian Distributed Random Signal

	Mean (s)	Variance
Scenario 1	0.13	0.195
Scenario 2	0.16	0.024
Scenario 3	0.18	0.027
Scenario 4	0.19	0.0285