Real-Time Household Energy Prediction: Approaches and Applications for a Blockchain-Backed Smart Grid

by

Michelle Lauer

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2019

Author .....
Department of Electrical Engineering and Computer Science
February 1, 2019

Certified by .....
Marija Ilic
Senior Research Scientist
Thesis Supervisor

Accepted by .....
Katrina LaCurts
Chairman, Department Committee on Graduate Theses

Real-Time Household Energy Prediction: Approaches and Applications for a Blockchain-Backed Smart Grid

by

Michelle Lauer

Abstract

In the current era of Internet of Things (IoT) devices, household solar panels, and increasingly affordable local energy storage, energy grid systems are facing a new set of challenges that they were not originally designed to support. Energy systems of the near future must be capable of supporting these new technologies, but new technology can also be leveraged to improve reliability and efficiency overall. A major source of potential improvements comes from the increase of connected devices that are capable of dynamically adjusting their behavior, and offer new data that can be used for optimization and prediction. Energy predictions are used today at the bulk power system level to ensure demand is met through appropriate resource allocation. As energy systems become more responsive, prediction will be important at more granular system levels and timescales. Enabled by the rise in available data, existing research has shown some machine learning models to be superior to traditional statistical models in predicting long-term aggregate usage. However, these models tend to be computationally expensive; if machine learning prediction models are to be used at short timescales and performed close to the end nodes, there is a need for more efficient models. Additionally, most machine learning models today do not take advantage of the known and studied properties of the underlying energy data. This thesis explores the circumstances under which machine learning can be used to make predictions more accurately than existing methods, and how machine learning and statistical methods can serve to complement each other (specifically for short timescales at the household level). We find that basic machine learning models outperform other baseline and statistical models by using energy usage trends observed from statistical methods to better engineer the input features. For the increasingly distributed energy systems that these predictive models aim to support, the distributed nature of blockchain technology has been proposed as a good match for managing such systems. As an example of one possible distributed management implementation, this thesis presents a novel blockchain-enabled architecture that provides privacy for users, information security through improved household-level prediction, and takes into consideration the security vulnerabilities and computational constraints of the participants.

Thesis Supervisor: Marija Ilic
Title: Senior Research Scientist

Acknowledgments

First and foremost, I cannot express enough gratitude to Professor Marija Ilic for inspiring me throughout this entire process. I am incredibly grateful to have had the opportunity to work with a professor who has continuously been supportive of me exploring my own interests, while simultaneously guiding me towards work that is meaningful and relevant within the context of a field that was initially entirely unfamiliar to me. Huge thanks to Rupamathi Jaddivada for taking the time to keep up to date with what I was working on, sending me so many useful resources, and providing background and insight into my ideas. I always thoroughly enjoyed the mostly on-topic meetings the three of us had together, and I learned so much.

I would also like to thank Martin Wainstein, Eric Duong, and the MIT DCI for being a huge part of the reason that I ended up working on this project, and for supporting me through my circuitous path to settling on a research project.

As always, my friends and family deserve an enormous shoutout for listening to me through the many iterations of this project, and for making sure that I had a smile on my face while I was doing it.

Contents

1 Introduction **17**

1.1 Motivation 17

1.2 Unsolved Problems 19

1.3 Thesis Goals 20

1.3.1 Summary 20

1.3.2 Primary Objectives 20

1.3.3 Outline 20

2 Energy Grid Background **23**

2.1 Today's Energy Grid 23

2.1.1 US Energy History 23

2.1.2 Independent System Operators 24

2.1.3 Rise of Prosumer Behavior 25

2.2 Toward Smarter Grids 25

2.2.1 Demand Response 25

2.3 Previous Modeling Work 28

2.3.1 Low-Memory Statistical Models 28

2.3.2 Existing Machine Learning Approaches 29

3 Data-Driven Forecasting **31**

3.1 Baseline Statistical Models 31

3.1.1 Error Metrics 32

3.2 Stochastic Modeling 32

3.2.1 Preliminary Energy Usage Representation 32
3.2.2 Refining the Model 35
3.2.3 Clustering Loads 37
3.2.4 Predictions 38
3.2.5 Results and Discussion 40
3.3 Machine Learning Experiments 44
3.3.1 Feature Engineering 44
3.3.2 Selection of Meaningful Features 45
3.3.3 Prediction 46
3.3.4 Results 50
3.4 Takeaways 51

4 About Blockchain **55**
4.1 Motivation: Connection to Household-Level Prediction 55
4.2 Blockchain Background 56
4.3 Pairing Blockchain and Energy 56
4.4 Blockchain Energy Moves to Industry 57
4.4.1 Smart Contracts and Tokens 58
4.4.2 Energy Tracking 58
4.4.3 Demand Response 59
4.5 Consensus Mechanisms 60
4.5.1 Proof of Work 61
4.5.2 Proof of Stake (PoS) 62
4.5.3 Proof of Authority 62
4.5.4 Consensus Mechanism Considerations 63

5 Secure Blockchain-Enabled DyMonDS Design **65**
5.1 Design Goals 65
5.2 System Participants 66
5.3 DyMonDS Framework 67
5.3.1 Coordination through Device Controllers 67

5.3.2 Minimal Information Exchange 68
5.4 Secure Communication Protocol 69
5.4.1 IoT Security 70
5.4.2 Protecting Privacy and Data Integrity 71
5.5 Blockchain Utilization 73
5.5.1 Challenges of Blockchain for Energy 73
5.5.2 Blockchain in Secure DyMonDS Implementation 74
5.6 Validation through Learning 75

6 Conclusion **77**
6.1 Contributions 77
6.2 Future Work 78

A About the Data **81**

B Basic Machine Learning Models **85**
B.1 k-Nearest Neighbors 85
B.2 Random Forest Regression 86
B.3 Multilayer Perceptron 87
B.4 Support Vector Regression 88

C Accessed Pecan Street Data Tables **89**

List of Figures

2-1 US energy generation comes from a mix of different sources, both renewable and nonrenewable. 23

2-2 Example of a smart household setup, adopted by the Pecan Street households whose data is used for this thesis. Facilitated by the Network Optimized Distributed Energy System (NODES), smart homes communicate with the grid, and receive signals from the retail market. Within a household, individual devices communicate with a smart BluCube device, which is then responsible for communicating with the broader grid system. This figure shows one example of a household with two smart devices: an electric vehicle charger an HVAC system. 27

3-1 Mean daily use values, where the shaded range spans one standard deviation above and below the mean. Power is measured in kW. . . . 33

3-2 CDF of mean usages values for each 15-minute time segment (96 segments per day). 34

3-3 Transition matrix given 6 sorted states, demonstrating the trend that the transition counts are grouped along the diagonal of the matrix. This indicates that usage tends to stay in the same state for consecutive time periods. 34

3-4 Orange is the observed daily mean over the day, blue is simulated (with 6 states) 36

3-5 Sample simulation with 3, 12, and 24 states (left, center, right) 37

3-6 The inertia of the k-means clustering decreases as the value of k increases. Note that there is not a particularly distinct "elbow" in this graph, which would point to an optimal k value. 38

3-7 k-means clusters for k = 10 39

3-8 Load profile clusters as determined by daily k-means clustering for two different households. Each of the 6 clusters is assigned a different color. The clusters for the household on the left appear to correlate with seasonal variation, while the clusters on the right are not so readily interpretable. 40

3-9 Mean daily use values for Household 1, where the shaded range spans one standard deviation above and below the mean. Power is measured in kW. 42

3-10 Representative clusters for Household 1 data, found using k-Means clustering with 6 clusters 43

3-11 Snapshot of daily usage for Household 1, where the vertical red lines divide days. 43

3-12 The models built for predicting energy usage take in as inputs previous usage values and/or ambient conditions such as temperature, wind, precipitation, etc. For both statistical models and machine learning models, there are some intrinsic parameters $(i_1, i_2, ...)$ that determine the model architecture. The component outlined in red shows the difference between the machine learning and statistical models: machine learning models should be able to learn external physical parameters p_i (so the red boxed portion can be removed), while statistical models will pass these in explicitly. Alternatively, machine learning models may be able to learn the p_i values to be passed in to statistical predictive models. 53

5-1 DERs (black) directly communicate with their local neighborhood compute node (blue). Compute nodes are connected to each other via a blockchain network. 67
5-2 Interactive modeling and control design for DERs 69
5-3 Snapshot of the blockchain ledger that is replicated across all compute nodes. Each "block" in the chain stores data about the bids and optimally computed clearing results. In this figure, there are three neighborhoods, and all of the results at time t-2 and t-1 have already been published to the chain. At time t, the data from N3 (dotted lines) have been broadcast, but not yet synchronized across all devices. After synchronization, the new block can be appended to the chain (black arrow). 74

A-1 Snapshot of energy usage data collected for a single household. Vertical red lines separate days. 82
A-2 Snapshot of data collected for a household with solar generation. . . . 83

B-1 Example of k-Nearest Neighbors classification. On the left k=7, so the new test point X will have a blue predicted label. On the right, k=3, so X will have an orange predicted label 86
B-2 Random forest regression model. Each decision tree may be split on different attributes at each node, and the end result will be the average prediction across all models. 87
B-3 Multi-layer Perceptron architecture with 2 hidden layers. The different components of the vector input will be passed in as each node of the input layer, and the final energy consumption prediction is the result in the output layer. 88

List of Tables

3.1 Results of statistical modeling experiments relative to the PSS baseline, using 12 states and 6 clusters 41

3.2 Error metrics from Household 1 for each cluster, using dot prediction. 42

3.3 Relative importance of input features for the RF regression model. The feature that varies more importance than any of the others by an order of magnitude is "1_intervals_before", which stores the usage value from the previous 15 minute time period. 48

3.4 RF results for model with 10 estimators and maximum depth of 10 levels 49

3.5 Possible MLP hidden layer architectures, where "Layers" describes the hidden layer structure (i.e. (16, 16, 16) has three hidden layers between the input and output layers, and each hidden layer has 16 neurons) . 50

3.6 Household 1 results for the aforementioned ML models 50

C.1 Weather Available Data 89

C.2 Household Usage Available Data 90

Chapter 1

Introduction

1.1 Motivation

The landscape of household energy usage and interactions with the grid is undergoing rapid development. In today's energy systems, Independent System Operators (ISOs) — responsible for ensuring demand is met at all times given physical constraints — operate at the bulk transmission level. ISOs therefore have little reason to manage or model how this energy is then distributed at more granular levels closer to the ends of the grid. Short term, house-hold level prediction was neither possible nor useful until the recent rise in Internet of Things (IoT) devices and the potential for household-level participation in the grid. IoT device use, industry deregulation, increasingly cost-competitive household-level energy storage, and new types of generation sources are all putting pressure on the existing grid to be more flexible and responsive. With these changes, new opportunities have opened for technology to improve energy systems, but these opportunities come hand-in-hand with new challenges related to optimizing efficiency, coordinating interactions, and securing data. The increase in smart devices and participation by end nodes requires and enables more intelligent grid design.

Computationally efficient approaches for predicting energy usage are necessary for such designs, because more logic is moving away from aggregators towards the responsive end nodes of the system, which have less computational power. Economically incentivizing household-level participation in the dynamics of the electricity

grid can lower costs for consumers, reduce strain on the grid, and improve efficiency. Transactive Energy Management (TEM) is defined by NIST to refer to "techniques for managing the generation, consumption, or flow of electric power within the electric power system through the use of economic or market-based constructs while considering grid reliability constraints" [1]. Predictions of future energy usage are critical for establishing appropriate economic incentives in a TEM grid model, and for supporting real-time responsiveness by smart devices.

The field of low-memory energy prediction is currently dominated by statistical learning approaches. Existing and ongoing work demonstrates the effectiveness of such models to predict energy consumption for devices that are highly correlated with ambient conditions; for example, Section 2.3.1 shows one example of a model developed to perform computationally inexpensive, short-term wind forecasting. These statistical learning methods require model parameters to be set before data is passed in, such as determining how continuous variables should be discretized, or approximating the shape of noise. The success of these models is highly dependent upon the preset parameters.

Due to the challenge and importance of selecting appropriate model parameters, machine learning presents an alternative that does not require physical parameters to be known beforehand; these parameters can be learned by the model instead. Before the prevalence of IoT devices and accompanying "big data" that exists today at increasing volumes, machine learning models could only be adequately trained at longer timescales due to the absence of sufficient training data for shorter timescales. For this reason, there exist gaps in the types of approaches used to do short-term prediction with low computational cost. More specifically, most machine learning approaches that have been developed today demonstrate the effectiveness of complex deep learning models in learning long-term trends even with noisy data. The machine learning models most successful in learning these trends also generally have the most complex model architectures and are therefore the most computationally- and memory-intensive. Despite this downfall, there has not yet been adequate exploration of developing computationally efficient, short-term, household-level predictive

models. Machine learning methods are well-suited for this type of problem because these models can derive patterns directly from data which might be noisy or might not have clear physical intuition. The possible solution space will not be bounded by some fixed expected form, since the physical parameters necessary for most statistical models can be learned rather than preset. An important caveat to this proposed advantage is that machine learning models themselves are parameterized as well. However, these parameters are not typically on the form of the data, but instead on the learning architecture, and the process by which a model arrives at a solution.

1.2 Unsolved Problems

Machine learning strategies for energy forecasting have been designed to predict long-term aggregate energy usage, and have been highly successful at satisfying that goal. However, the success of these strategies is not directly transferable to implementation in the grid, because smart device-enabled end node participation requires computationally efficient models in terms of time and space. A largely unsolved question is whether machine learning is a more accurate forecasting tool at the short-term household level, or if statistical methods should continue to be used.

The motivation behind developing household-level prediction models is largely driven by the desire to incorporate them in responsive and dynamic grids. A major concern with such an implementation, where devices communicate directly with the grid, is that this two-way exchange exposes both parties to many potential vulnerabilities in terms of cybersecurity. Insecure demand response systems could lead to major problems such as privacy leakages, hardware failure, or power outages. For these reasons, as we move toward more connected and interactive grids, security must be at the forefront of design decisions.

1.3 Thesis Goals

1.3.1 Summary

The goal of this thesis is to evaluate existing options and explore new machine-learning backed options for short-term household-level energy forecasting with low computational cost. This goal is largely enabled by the increasing number of IoT devices that are tracking household energy usage, thus opening the door for more data-based energy forecasting. Some of the recently developed strategies discussed in Chapter 2 use machine learning to train complex models capable of using historic data (and sometimes other spatio-temporal factors) to learn usage patterns. However, a downside of these models is that they are computationally expensive, requiring high memory and storage. Making models that are computationally efficient is crucial for being able to make real-time predictions at high levels of granularity and for being able to distribute this computation. Looking ahead toward an energy future with more adaptation logic moving away from system operators and towards the end users, local computation will enable a scalable model where end users can be dynamic participants in a secure and distributed smart grid.

1.3.2 Primary Objectives

Objective 1: **Evaluate different machine learning energy forecasting models and compare their predictive capabilities with existing statistical models.**

Objective 2: **Design a system architecture for a blockchain-backed energy system, and assess the underlying blockchain architectures that align well with the goals of such a system.**

1.3.3 Outline

The remainder of this thesis is outlined as follows: Chapter 2 provides background on the current structure and state of the electricity grid, and introduces new directions

that grid management is moving towards. This chapter also includes a discussion of existing models for predicting energy usage. Chapter 3 demonstrates the effectiveness of various modeling experiments using both statistical and machine learning approaches for short-term household energy usage prediction. These predictions can be used by the various components of a smart grid system in order to generate optimal bids and schedules. An explanation of blockchain technologies relevant to energy is provided in Chapter 4, in order to provide background for the system design presented in Chapter 5. This design focuses on the interactions between distributed energy resources and local neighborhood-scale compute nodes, with a focus on system security given the types of components participating in the system. Chapter 6 concludes with the contributions of this thesis, and future work.

Chapter 2

Energy Grid Background

2.1 Today's Energy Grid

2.1.1 US Energy History

At a high level, the way that electricity travels from a power plant to a consumer can be broken into four stages: generation, transmission, distribution, and retail [2]. In the US, generation has typically come from a variety of renewable and nonrenewable sources, from coal and natural gas to solar and hydro power, shown in Figure 2-1.

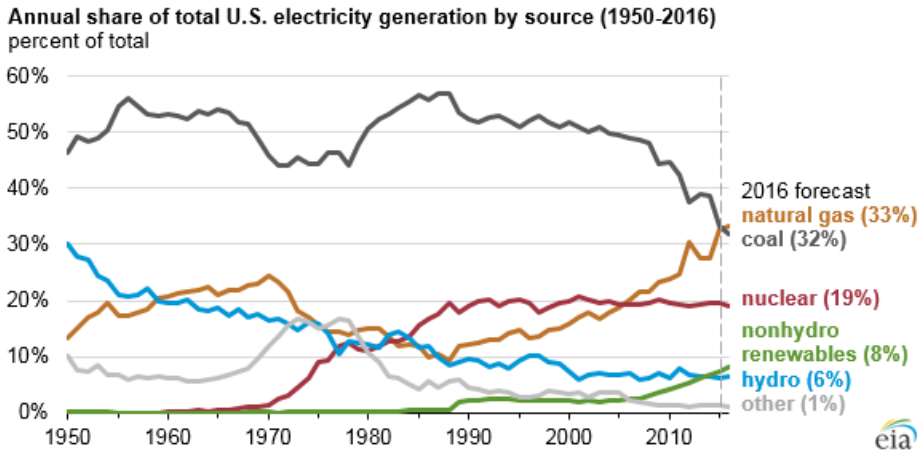

Figure 2-1: US energy generation comes from a mix of different sources, both renewable and nonrenewable.

To manage the electricity grid in the face of this array of generation sources, the

traditional model of US utilities was that of a fully vertically integrated utility, where a single entity had control over all four stages of the supply chain. This business model was supported by the fact that electricity generation was believed to be a natural monopoly, where high upfront costs created a cost-related barrier to entry. This meant that even without any prohibitive policies, there would be little economic incentive for other parties to enter the market, due to an inability to compete. The National Energy Policy Act of 1992 marked a shift in energy policy by legalizing and outlining a market for competitive wholesale electricity generation [3]. This allowed private entities, known as "merchant generators", to participate in the wholesale electricity market, bringing about competition in a previously monopolized system.

The rationale behind restructuring the market and moving away from a monopolized system was to provide consumer benefits due to better-aligned incentives. Deregulation could introduce competition that would in turn motivate market participants to drive down marginal costs. This could be done at the level of a single plant improving their operating costs, or at the dispatch level, since building a system to support a mix of plants could improve coordination between regions as well. While many of these positive changes were reflected as the number of deregulated systems increased, a deregulated market also resulted in a new set of technical challenges. In a deregulated, wholesale electricity market, generators submit bids for electricity production. These bids can come in many different forms depending on the market, and the details of when different generators will participate in production vary in complexity. Additionally, these bids can be placed in advance and over varying timescales. Initial attempts at deregulation demonstrated a need for improved market design to support this new system. One such change was adjustable electricity pricing, making demand more elastic to prevent the kinds of fluctuations seen in what is now known as the California Electricity Crisis [4].

2.1.2 Independent System Operators

To manage modern deregulated wholesale electricity markets, ISOs were established to maintain balance in grid operations. Since electricity has historically been costly

to store, the role of an ISO is to match supply and demand such that electricity needs are met at all times. This task is challenging, because there is little margin for error – mismatched supply and demand change the frequency of the grid's electricity, and large changes in frequency are damaging to the hardware [2]. ISOs rely upon predictions for a variety of purposes ranging from setting appropriate prices to deciding which power plants need to be in operation. Different use cases require different levels of granularity and access to different data. For example, prices might be set one day ahead, but demand needs to be satisfied every second.

2.1.3 Rise of Prosumer Behavior

The use of renewable energy is on this rise, particularly as concerns about climate change grow. Renewable energy sources such as solar and wind require more grid support than fossil fuels, because they are inherently intermittent sources. Since this inconsistent output does not necessarily align with demand for energy, a responsive grid system is necessary to ensure that demand can be met. The idea of end users being not only consumers, but instead "prosumers" who both consume and produce energy puts additional strain on the grid. Prosumer behavior is becoming increasingly economically viable as energy storage becomes more affordable, and household-level generation sources like rooftop solar gain traction. The adoption of these and other distributed energy resources (DERs) requires a strategy for integrating them into existing infrastructure. DERs also introduce the possibility of shifting more logic and energy exchange closer to the end users in local transactions.

2.2 Toward Smarter Grids

2.2.1 Demand Response

Unlike most other services, energy users are typically passive consumers who use energy without regards to the current price, then pay for that usage at the end of their payment period based on their energy provider's price. Contrast that with a model

where energy consumption is dependent upon transparent energy prices that are constantly changing in response to the market. This second type of model presents the opportunity to increase efficiency and save consumers money. Consumers could be encouraged to use energy at lower price times, and built into these economic incentives, peak loads would be leveled out and redistributed. Unpacking this a bit more, if we assume that when energy demand is high, prices will be higher, then there may be the opportunity to reduce strain on the grid by economically incentivizing consumers to decrease their consumption during peak periods and increase consumption during off-peak periods instead. This would offer the dual benefit of both reducing maximum load capacity, and lowering the rate of change. One option to provide this type of economic incentive would be to add tariffs or offer payments to users who adjust their usage to help optimize these loads.

Smart devices (i.e. smart thermostats, water heaters, etc.) enable this type of responsiveness without requiring a consumer to take any direct action, and potentially without having any impact on their realized usage. For example, imagine a scenario where a consumer has an electric vehicle that they charge overnight in their home, with the requirement that in the morning when they leave for work, their vehicle must be fully charged. A smart charging device could be programmed such that rather than charging the vehicle fully as soon as it gets plugged in, the device is charged during the 4 hour window over the course of the night where the predicted prices are the lowest.

Demand response motivated through economic incentives can be described as transactive energy management (TEM). These economic incentives can be used for load management on both the supply- and demand-sides. One example of a system designed to model and simulate transactive energy management is the novel Dynamic Monitoring and Decision Systems (DyMonDS) framework, discussed further in section 5.3 [5]. At a high level, DyMonDS supports scaling by operating under the proposition that each power system agent needs to share only minimal information with its neighbors (defined by some interface), and can still arrive at the same optimal energy allocation solution as a fully centralized method; treating each device

as its own black box with specified inputs and outputs reduces the complexity of an otherwise intractable high-dimensional optimization problem. Power system agents are thus abstracted to the level of detail specified by their interface variables. With this type of system, electric power systems can become active participants in the grid, where agents participate by placing bids based upon predicted knowledge and their own physical constraints [6].

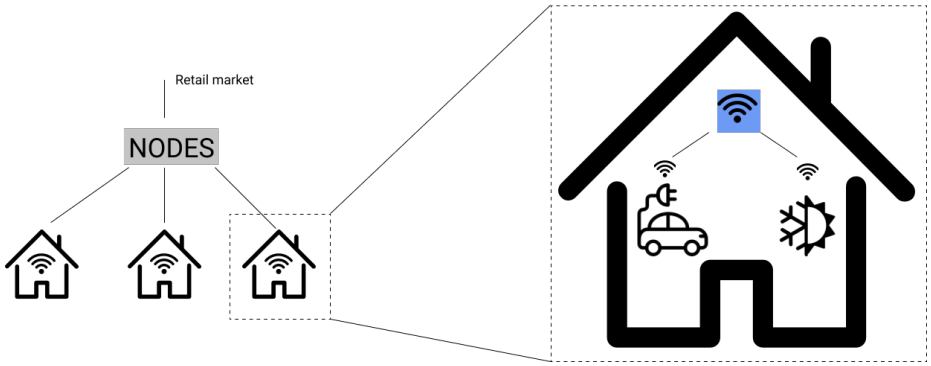

Figure 2-2: Example of a smart household setup, adopted by the Pecan Street households whose data is used for this thesis. Facilitated by the Network Optimized Distributed Energy System (NODES), smart homes communicate with the grid, and receive signals from the retail market. Within a household, individual devices communicate with a smart BluCube device, which is then responsible for communicating with the broader grid system. This figure shows one example of a household with two smart devices: an electric vehicle charger an HVAC system.

Each device has a unique set of physical constraints that define its ability to participate in such a system, but we can represent different devices with a common model that includes a device controller as well as the mathematical equations describing the physically defined parameters. The controller must also be aware of external disturbances that influence device behavior, and impact how the device should respond to various higher-level signals. For example, if there is a household with a smart HVAC system, we can imagine that if we know that the temperature outside is very hot, and have learned that at a particular time of day, people tend to frequently open doors (letting hot air into the household and requiring more energy to keep the household at a constant temperature), then the controller can use this information when determining the optimal device settings and communicating its needs to the grid. Figure 2-2

shows an example of a connected household setup with a smart home capable of communicating with the grid; this is the setup used in the Pecan Street homes whose data is used in this thesis.

2.3 Previous Modeling Work

2.3.1 Low-Memory Statistical Models

Increasing volumes of data and IoT devices do not guarantee that machine learning will be the best approach to energy forecasting. Despite the potential benefits of machine learning previously discussed, current research is also ongoing for exploring more traditional statistical methods within these new contexts, such as in [7]. These models were constructed with the explicit goal of achieving high accuracy with less data, while balancing the trade-off between accuracy lost when ignoring some features. They found their daily predictions at 30 minute intervals to have comparable results to Artificial Neural Network (ANN) models, due to the ability to better generalize than the highly parameterized ANN models.

In a wind-specific example, the primary goal is to develop a data-driven statistical model that is both interpretable, and capable of capturing the rapidly-changing dynamics of wind [8]. Three of the models explored were a persistent forecasting model, an autoregressive model, and a novel "spatio-temporal trigonometric diurnal model". The persistent forecasting model assumes that the wind reading for the next period will be the same as the current reading, which works well for very short term forecasting. The autoregressive model assumes future windspeed is a linear combination of past wind speeds. The final model is a space-time statistical model that takes into account spatio-temporal information as well as past data; it assumes variation in wind data follows a normal distribution (truncated to the non-negative, real domain), so building this model requires picking appropriate parameters to define this normal distribution.

These are just a few examples of an area of research dedicated to improving energy

modeling. In general, a commonality between these models is that there is an ongoing challenge to balance model complexity with generalizability and computational efficiency.

2.3.2 Existing Machine Learning Approaches

Some work has been done to use machine learning for energy forecasting, and to compare it to more traditional mathematical approaches. In [9], predictions are made for every minute of a 2 hour period. The baseline math model used is an Auto-Regressive Moving Average (ARMA) model. They compare this to a Support Vector Machine (SVM), and two different ANNs – a Nonlinear Auto-Regressing (NAR) recurrent ANN, and an Long Short-Term Memory (LSTM) network. These two ANN models are well-suited for timescale predictions because both maintain some concept of "memory". For each timestep, the NAR tracks the actual values from n previous timesteps, and also the most recent predicted value using the same network structure. For the LSTM, memory cells are used instead of neurons as the base unit for the machine learning architecture. These can store up to an unbounded amount of historic data, making it a good model for time-series data with potentially long time dependencies. For timesteps greater than 40 minutes, the two ANN approaches were the most successful, but for 0 to 40 minutes, the SVM performed the best. All these models performed better than the baseline ARMA model.

A more complex deep learning approach is taken in [10], where a combination Convolutional Neural Network (CNN) and LSTM model is used for electricity price forecasting. This paper also discusses many existing approaches that have been taken for price forecasting. The focus of this thesis is on predictions for usage rather than price, but usage is used to determine price, and price can be used as a driver for usage change, so the two are closely connected. Their findings determined that SVMs sometime fail to learn data trends, and thus lead to highly volatile results. While the other models were much less volatile, their hybrid architecture that combines findings from different types of models outperformed all other machine learning approaches for predicting hour-ahead data based on the previous 24 hours.

In [11], some previous ANN approaches for energy prediction are discussed that take ambient conditions into account. A challenge with ANNs (like any parameterized method) is that optimal parameter tuning can have a large impact on model accuracy, but can also be highly dataset dependent. This makes it challenging to use models designed for one dataset out-of-the-box and apply them to another, even if the form of the data is the same. For this reason, the approach to developing an accurate model is just as valuable as the architecture of the model itself, such that the same reasoning can be extended to different datasets if it is the case that the original model was overfitted to the given data. This paper found that the most promising methods were those that combined neural networks with other algorithms (e.g. using machine learning for model parameter estimation).

Chapter 3

Data-Driven Forecasting

All of the data used in this chapter is drawn from the Pecan Street Dataport [12]. The analyzed usage data is collected per device every 15 minutes. See Appendix A for more information about the data.

3.1 Baseline Statistical Models

Robust statistical and machine learning approaches like those discussed in Chapter 2 demonstrate the ability of existing models to well-encapsulate energy usage while taking advantage of spatial or temporal aggregation to smooth out fluctuating real time usage patterns. To see if these types of models translate well to household-level energy forecasting, we use as baseline a "persistent" model (PSS) that predicts that the energy usage in period t will be the same as the usage in period $t-1$. PSS modeling tends to perform well for short-term forecasting like the 15-minute-ahead modeling done here. Coupled with the fact that the energy data being modeled is linked with physical constraints and usage behaviors that prevent large fluctuations, any models that outperform PSS can be considered quite successful [8].

3.1.1 Error Metrics

To evaluate model performance and compare the baseline, statistical, and machine learning approaches, we use mean absolute error (MAE) and mean squared error (MSE). Let y_j be the true energy consumption value at certain time, and $\widehat{y}_j$ be the predicted value, across n total samples.

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \widehat{y}_j|$$

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \widehat{y}_j)^2$$

For models that outperform the baseline, we indicate the percent improvement of each of these metrics relative to the baseline PSS model.

3.2 Stochastic Modeling

The primary statistical model that we will analyze in this thesis is a model recently developed for simulating usage based on data collected at 15-minute intervals, like the Pecan Street data [13]. We will reconstruct this model, developed for simulating load at the substation level, and apply it to the household level. The model combines learning from many prior modeling attempts by leveraging a Markov approximation for modeling transitions from one time period to the next, but adds an additional layer of complexity by using historical data to take daily periodicity into account.

3.2.1 Preliminary Energy Usage Representation

We begin with the assumption that we have one typical daily load "profile" type, meaning that we evaluate all daily observations as if they all belong in a single cluster. This assumption is relaxed in Section 3.2.3. In this model, load usage is represented as a stochastic process, so the inputs to the model are individual observations of load usage. To demonstrate the construction of this model, we will first walk through a toy example, only using observations for days within one particular month (these results

are for June 2016). Data is collected every 15 minutes and summed across all devices within this individual household, termed "use" in the given dataset. For each of the 96 time periods per day, we take the mean across all of the different days, shown in Figure 3-1.

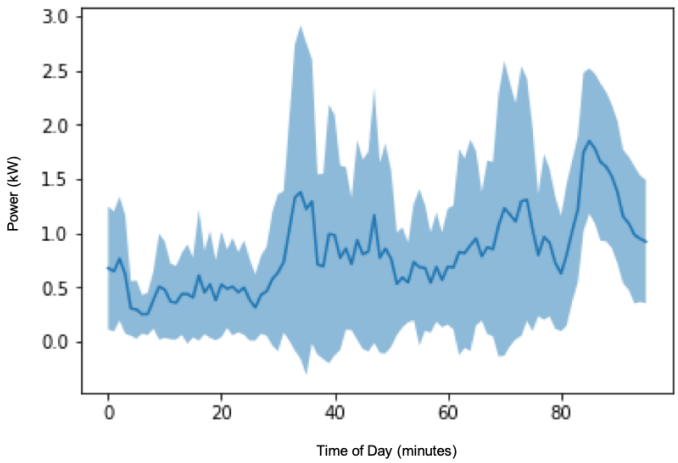

Figure 3-1: Mean daily use values, where the shaded range spans one standard deviation above and below the mean. Power is measured in kW.

The mean values for each of the 96 time periods are then sorted in increasing order, and scaled such that they represent a cumulative distribution function (CDF), as shown in Figure 3-2. This CDF can be used to divide the space of possible usage values into discrete states. This example is divided into `NUM_STATES = 6` equally sized states. Note that the number of states can be increased such that the mean for the state more closely represents the encapsulated data, but highly granular states will involve jumps between states for small fluctuations. This division into discrete states gives us an initial probability of being in each state, denoted as `pi_s`, a vector of length `NUM_STATES` whose values sum to 1; for equally sized states, the probability of being in any given state is simply 1/`NUM_STATES`.

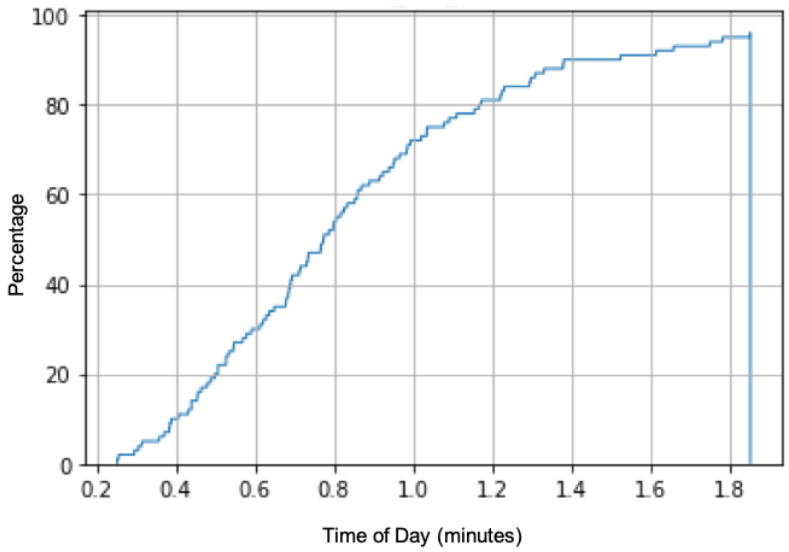

Figure 3-2: CDF of mean usages values for each 15-minute time segment (96 segments per day).

To find the stationary transition probabilities between each of these states, we can walk through the mean values over the course of the day, and count the number of times we see a transition from one state to another. Note that this is counting the transitions across the day of the mean values shown in Figure 3-1, rather than counting the transitions for each individual daily observation. Let `stationary_P` be the matrix of dimension `NUM_STATES` $\times$ `NUM_STATES` counting these transitions, where `stationary_P[i][j]` is incremented for each transition from state `i` to state `j`, as shown in Figure 3-3.

$$\begin{bmatrix} 10 & 6 & 1 & 0 & 0 & 0 \\ 6 & 6 & 4 & 0 & 0 & 0 \\ 0 & 3 & 7 & 3 & 3 & 0 \\ 0 & 1 & 3 & 7 & 5 & 0 \\ 0 & 0 & 0 & 6 & 5 & 4 \\ 0 & 0 & 1 & 0 & 3 & 12 \end{bmatrix}$$

Figure 3-3: Transition matrix given 6 sorted states, demonstrating the trend that the transition counts are grouped along the diagonal of the matrix. This indicates that usage tends to stay in the same state for consecutive time periods.

Intuitively, the observation that the transition matrices have higher values along the main diagonal supports the generally accepted claim that PSS is an appropriate baseline model, since a value at the diagonal indicates that mean usage stayed within the same state between two consecutive periods.

3.2.2 Refining the Model

Notice that with the model so far, the probability of being in a particular state is fixed and equal to the value at the state's index in `pi_s`. However, energy usage behavior exhibits daily periodicity (in addition to being impacted by other periodic factors), which these `pi_s` state probabilities fail to take into account. To take daily time-dependency into account, the probability of being in each state will be assigned per time period. Each state has its own mean usage value, so state probabilities are assigned such that the expected value during a given time period is exactly equal to the observed mean. This is an under-determined problem, so the reweighting is done such that it deviates from the original `pi_s` as little as possible. The result is a new set of state probabilities, one for each time segment.

Next, the transition probabilities are adjusted such that they are compatible with the reweighted state probabilities. More specifically, the transition probabilities must satisfy the condition that the total probability of transitioning from any of the previous states into a particular state is equal to the adjusted state probability. This is again an under-determined problem, so we will select these values such that they deviate as little as possible from the original `stationary_P` values. The matrix equations used to generate these adjusted state and transition probabilities are shown explicitly in [13].

Once these adjusted matrices are successfully calculated, the original paper uses them to simulate daily loads. New simulated loads are generated through Markov chain Monte Carlo (MCMC) sampling. Each of the plots in Figure 3-4 is an independent simulated load, constructed assuming 6 states. Figure 3-5 shows simulated models for the same data, but with 3, 12, and 24 states.

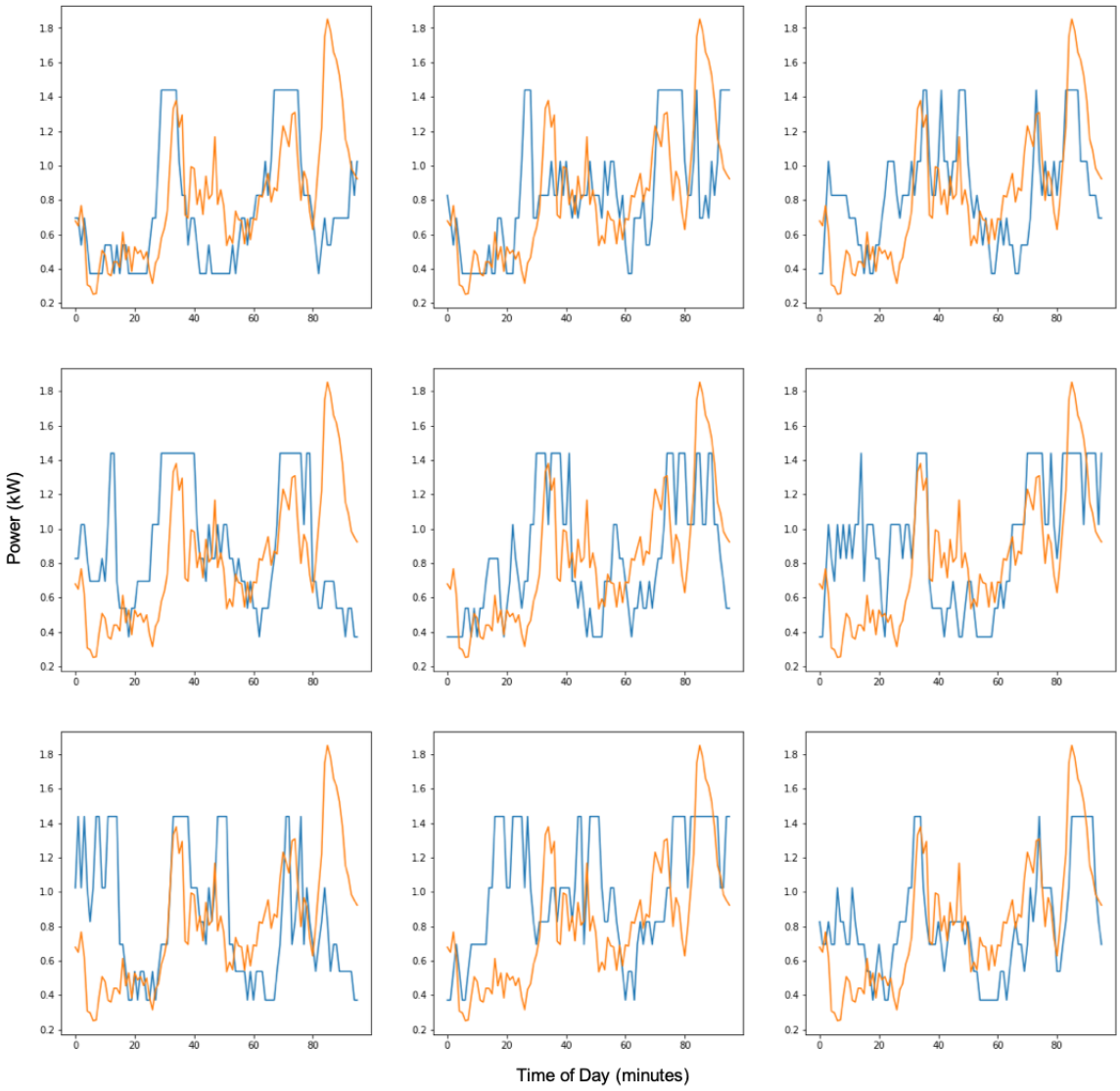

Figure 3-4: Orange is the observed daily mean over the day, blue is simulated (with 6 states)

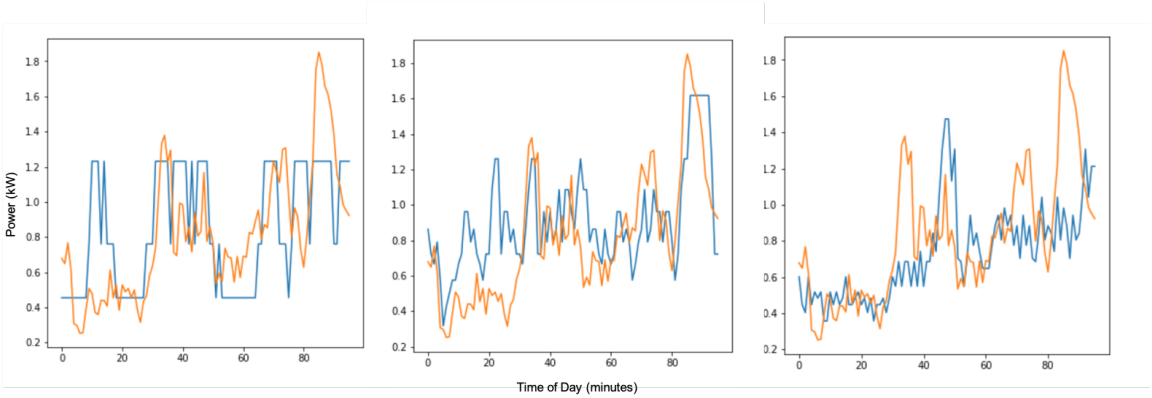

Figure 3-5: Sample simulation with 3, 12, and 24 states (left, center, right)

3.2.3 Clustering Loads

In this section, we relax the previous assumption that all load profiles fall into a single load profile type. Rather than gathering all the daily loads together, we instead cluster the loads into different profiles based off the shape of the load over the course of the day. The approach used here (suggested in the original paper as a possible clustering mechanism) is k-Means clustering of the load profiles. "Inertia" measures the sum of squared distances of samples to their closest cluster center. Inertia will decrease as the number of clusters k increases, but increasing k also increases the risk of overfitting.

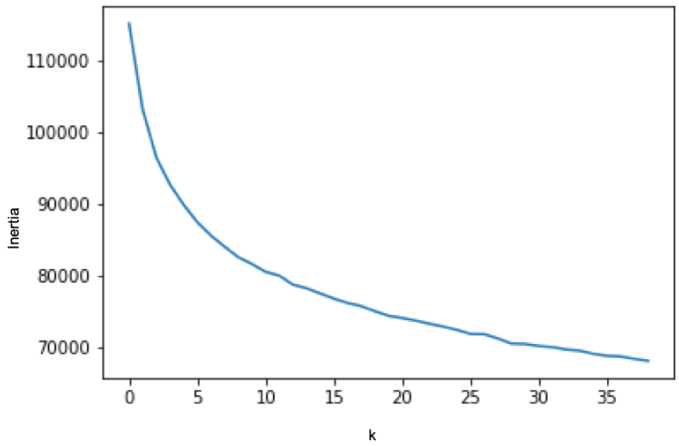

Figure 3-6: The inertia of the k-means clustering decreases as the value of k increases. Note that there is not a particularly distinct "elbow" in this graph, which would point to an optimal k value.

To determine an optimal value of k, we will look for an "elbow" in the graph of inertia vs k, and select the k value at that transition point. Ideally, the shape of this graph would be such that as k increases, inertia decreases steadily up to a point, then levels out. We pick k to be at that levelling out point (the "elbow"), since smaller k reduces the risk of overfitting, and increasing k beyond that point does little to improve inertia. The actual inertia graph for this data is shown in Figure 3-6, and the resulting clusters for k=10 are shown in Figure 3-7. The matrix modeling process previously discussed can be carried out independently for the data within each of the clustered load profiles.

3.2.4 Predictions

The transition matrices between states provide a mechanism for prediction as well as simulation. Transition matrices have the additional benefit of indicating not only the expected next step, but also show other possible next states with some probability. Two different approaches are taken to using the transition matrices for prediction. Given the time of day and the previous usage value, the predicted next value will be equal to 1) the mean value for the maximum likelihood next state or 2) a weighted av-

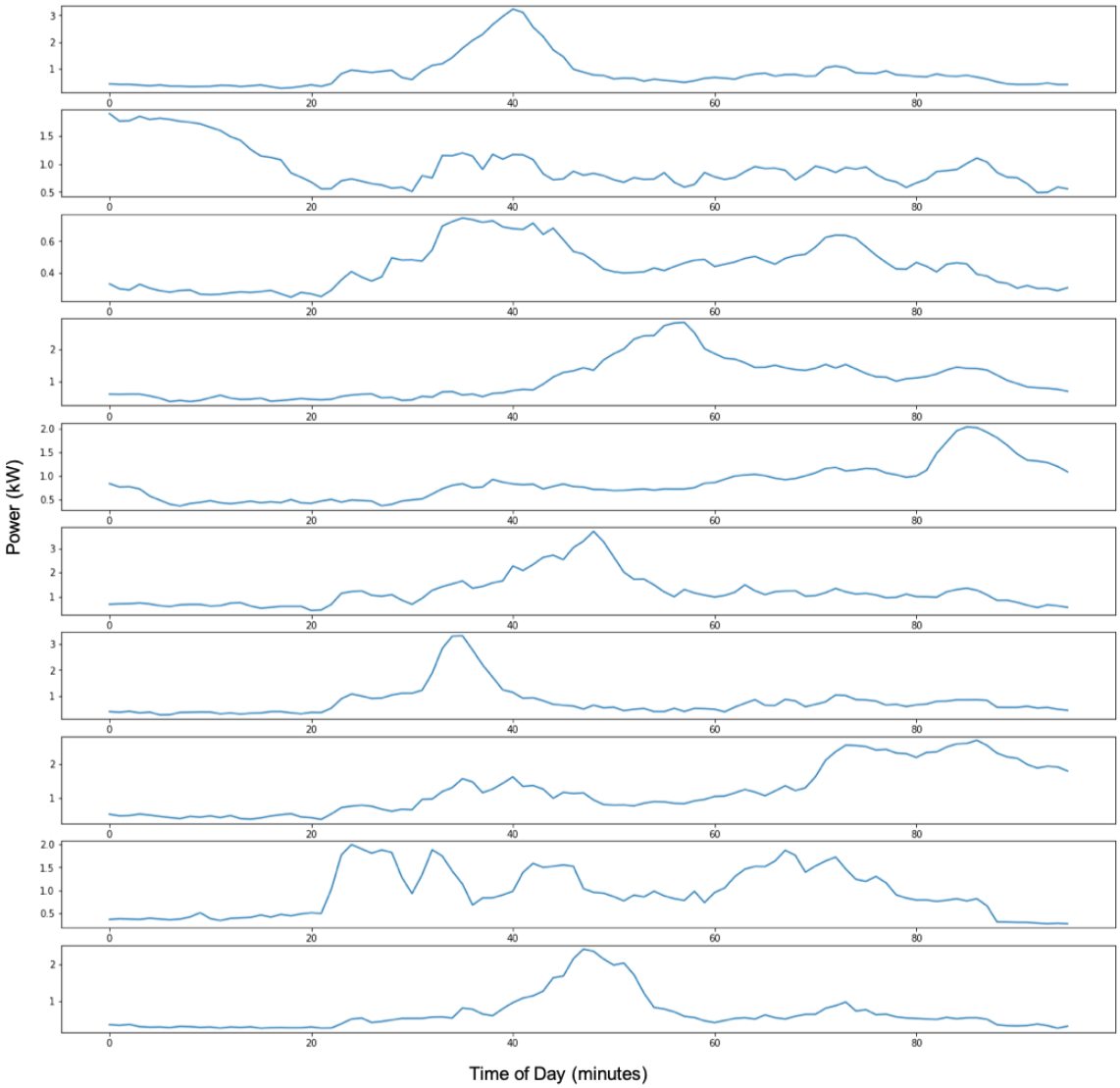

Figure 3-7: k-means clusters for k = 10

erage of the means of the next possible states, using the transition probability weights. These will be denoted as "max" and "dot" prediction, respectively. Notice that due to the Markov assumption, real-time energy prediction can be done using these methods by evaluating a single vector within one time-period's transition matrix.

For the statistical model, real time prediction requires one transition matrix for each time period across the day for each cluster. For the model selected here with 12 possible states, 96 time periods, and 6 clusters, the amount of memory required is 12*12*96*6*8 = 663552 bytes, assuming 8 byte matrix values.

Clustering is challenging for prediction problems because each day as a whole is assigned to a cluster, so at the beginning of the day, the cluster is not yet known. These prediction experiments assume that a day will belong to the same cluster as the previous day. Figure 3-8 shows that while this may be a reasonable assumption for some households where adjacent days are typically within the same cluster, the assumption may be insufficient for others, whose clusters appear randomly dispersed. Potential future work includes how to better predict clusters, or how to generate more human interpretable clusters.

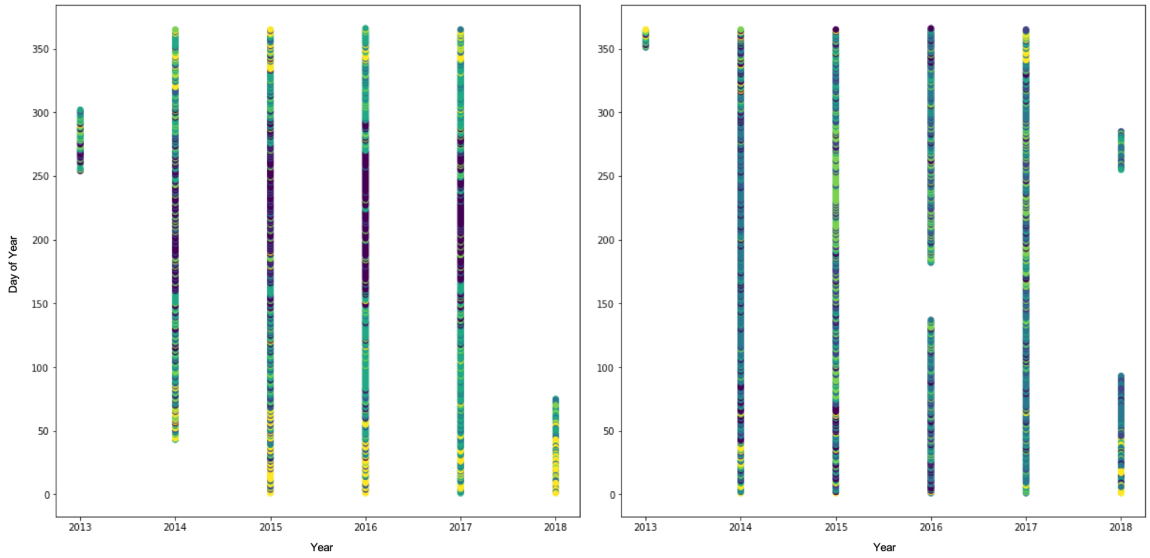

Figure 3-8: Load profile clusters as determined by daily k-means clustering for two different households. Each of the 6 clusters is assigned a different color. The clusters for the household on the left appear to correlate with seasonal variation, while the clusters on the right are not so readily interpretable.

3.2.5 Results and Discussion

Performance varies household to household, but we will look at two households that encapsulate some of different modeling outcomes. These are the same households whose data is displayed in Figure A-1 and Figure A-2, and despite the differences between these households, the trends between models are consistent, and consistent with the dataset as a whole. All of the results shown in Table 3.1 are for the aforementioned statistical model, conducted with 12 states and 6 clusters.

Household 1	**MAE**	**MSE**	**Model Size (bytes)**
Baseline PSS	0.370	0.475	8
No cluster, dot	0.410	0.449	1.11e5
No cluster, max	0.405	0.447	1.11e5
Clustered, dot	0.409	0.458	6.64e5
Clustered, max	0.413	0.504	6.64e5

Household 2	**MAE**	**MSE**	**Model Size (bytes)**
Baseline PSS	0.448	0.587	8
No cluster, dot	0.514	0.605	1.11e5
No cluster, max	0.505	0.601	1.11e5
Clustered, dot	0.486	0.569	6.64e5
Clustered, max	0.505	0.842	6.64e5

Table 3.1: Results of statistical modeling experiments relative to the PSS baseline, using 12 states and 6 clusters

These results indicate that the baseline model performs better than the stochastic statistical model when applied to household level energy usage, but also that clustering seems to be a useful technique for the model. The particular results shown here seem to indicate that using the dot product prediction approach is better when using clustered data, and the max prediction approach is better for unclustered data, but across a wider dataset, there is no clear winner between them. One potential downside of both of these prediction strategies is that they are bounded by the mean value for the lowest-valued bucket the mean value for the highest-valued bucket. For this reason, the model always fails to accurately predict high spikes in usage because the prediction will never rise above the mean value for the highest bucket for either strategy.

For Household 1, the mean energy usage value across the day is 0.657, so the baseline error represents a 56.3% error rate, with even larger error for the statistical model. To try to understand why there is such a high error rate, we start by looking at the spread of the data. The mean standard deviation value across the 96 time periods is 0.724, with the mean and standard deviation across all 150,000 samples shown in Figure 3-9. Upon initial inspection, there is clearly a wide spread in the data. Ideally, clustering would help with this high spread of data, and for some clusters, the

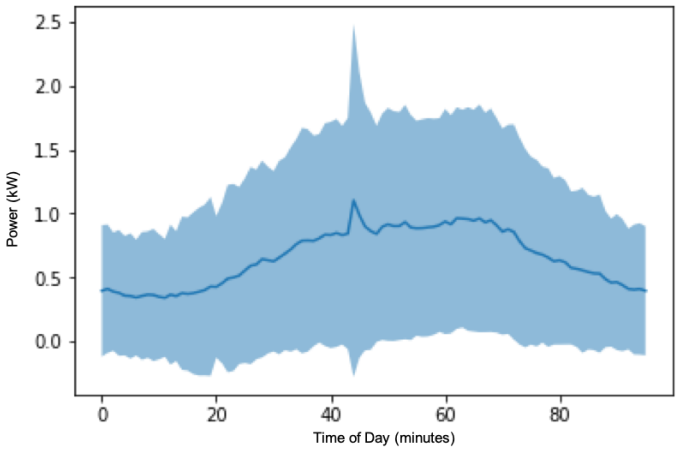

Figure 3-9: Mean daily use values for Household 1, where the shaded range spans one standard deviation above and below the mean. Power is measured in kW.

prediction accuracy is quite high. However, even clustering fails to capture a subset of the days whose predictions are highly inaccurate, making the overall clustered results comparable to the unclustered results. This can be seen in Table 3.2, where despite the fact that a plurality of results fall into Cluster 2, with the lowest error metrics, the results are greatly skewed by the high error predictions in the Cluster 5.

Household 1	**MAE**	**MSE**	**Cluster Size**
Cluster 0	0.419	0.529	2784
Cluster 1	0.423	0.453	5376
Cluster 2	0.348	0.406	12096
Cluster 3	0.411	0.458	3456
Cluster 4	0.441	0.387	4992
Cluster 5	0.812	1.18	1152
Overall	0.409	0.458	29856

Table 3.2: Error metrics from Household 1 for each cluster, using dot prediction.

However, even for the highest accuracy cluster, an MAE of 0.348 is still 53.0% error relative to the mean of 0.657. It is not entirely surprising that the 15-minute time periods are extremely challenging to predict due to high fluctuations that are typically smoothed out in predictive models that average over longer time periods. Therefore, the fact that this model relies heavily upon mean values over each day does not give the model the capacity to match the volatile nature of the actual usage values.

This can be clearly seen by comparing the shape of the representative cluster loads in Figure 3-10 to a random snapshot of real usage over the course of five consecutive days, shown in Figure 3-11.

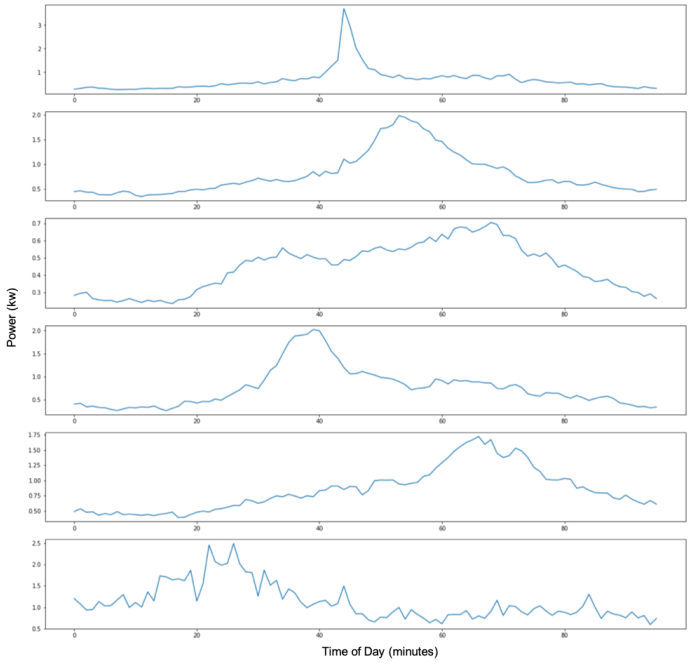

Figure 3-10: Representative clusters for Household 1 data, found using k-Means clustering with 6 clusters

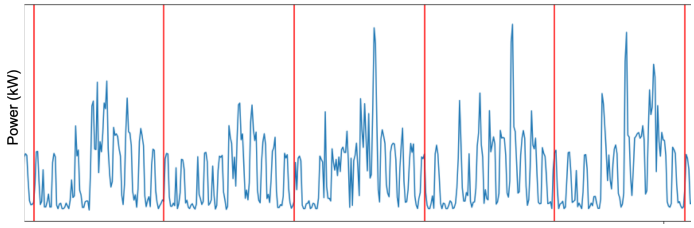

Figure 3-11: Snapshot of daily usage for Household 1, where the vertical red lines divide days.

3.3 Machine Learning Experiments

3.3.1 Feature Engineering

Overall Approach

Many machine learning models are capable of deriving meaning from relatively unstructured data. However, understanding the nature of the data being modeled is important for deciding what types of inputs should be passed in, and enables more sophisticated engineering of the features being passed into the model. The goal of these experiments is not only to see if any basic machine learning models demonstrate superior predictive capabilities relative to statistical models, but also to determine what type of features inform accurate predictions.

General Cleaning

Some households do not have (or do not have detection for) all of the possible devices listed in Table C.2, so columns referring to undetected devices were removed. Additionally, sparsely throughout the dataset, there are some entries where one particular device lacks data. For simplicity, those entries are removed from the dataset. Some of the machine learning models are sensitive to input scaling, so the weather data values which have highly varied units are adjusted by removing the mean and scaling to unit variance.

Handling Periodicity

A challenge of modeling highly time-dependent data is passing in time as an input while maintaining a measure of relative closeness. For example, looking at the hour of a day, a prediction model might learn that usage tends to be similar at times that are closer together. However, if we pass in hour of the day as a value between 0 and 23, then this model will see 0 and 23 as being the furthest apart and the least similar, whereas they are in fact adjacent times. To account for this, periodic features are split into two components: a sine component and a cosine component. In this case,

hour 4 is instead represented with `sin_hour` = sin(4/24) and `cos_hour` = cos(4/24). This both regularizes the data and maintains relative temporal distance.

Some known time-dependent trends are directly passed into the model as distinct features. For example, given the knowledge that PSS is an effective model for short-term forecasting, when predicting the usage at time t, the usage at $t-1$ is passed in directly as a parameter (along with the 7 periods before). Daily periodicity is another known trend; the entire statistical model from [13] is built off the assumption that daily periodicity exists. To take this into account, when generating the usage prediction at period t on day d, we will also pass in usage at period t on day $d-1$ (and the 7 days before).

3.3.2 Selection of Meaningful Features

Identifying the features that are helpful for prediction was done iteratively by incrementally adding in or removing features depending on their effect on model performance. Specific numerical values referenced in the preliminary evaluation in this section are for particular households, but the takeaways from feature selection and model comparison hold generally across households in the Pecan Street dataset without loss of generality.

kNN for Feature Selection

A simple kNN model was used for preliminary evaluation of relevant variables to include in the prediction models. kNN is very sensitive to scaling, and a refined model would require a distance metric optimized for the particular input data. For the generic kNN model here, weather data (whose range is highly varied across the different types of measurements) are scaled to unit variance with the mean removed. The default scoring mechanism used between kNN trials is the coefficient of determination R^2, which is sufficient to differentiate between including particular variables for this coarse initial pass through the data.

For a kNN model using only weather and temporal variables, excluding any data

on usage from previous time periods, the kNN model suggests that year can be excluded, and that day-of-year better encapsulates annual information than the combination of month-of-year and day-of-month. For this type of model, kNN with k values between 3 and 10 achieved peak R^2 scores; the highest score was 0.361. A prediction model that did not rely on usage from previous intervals would be an extremely powerful tool, because predictions could be made far in advance, and would not require much (if any) real-time computation.

However, as a separate experiment, *only* previous time period predictions were used, and this model achieved significantly higher R^2 scores. Depending on the number of previous intervals included, all of these models achieved peak R^2 score with on the order of hundreds of neighbors. All of the models performed similarly, but the best such model included the previous 5 intervals, for an R^2 of 0.472. This confirms that fluctuations of household level data at short time spans prove too challenging to model without taking real-time data into account.

Combining these two types of experiments, and using weather and temporal data in addition to previous usage values, R^2 for the kNN model decreased slightly to 0.455. For more robust models, adding inputs should not decrease prediction accuracy, only increase training time as the model learns to ignore superfluous features. The final list of variables included as inputs to the machine learning models is listed in Table 3.3.

3.3.3 Prediction

The models included in these experiments include RF, MLP, and SVR (see Appendix B). These models were selected because they can be constructed to have simple architectures while generally maintaining predictive power; they have also been used in previous work at aggregate levels, so they are used again here to determine whether their modeling capabilities as demonstrated at the aggregate level translates effectively to household-level prediction. Additionally, the architectures are varied across these models with the aim of gleaning more insights into what type of input data is useful for predictions and how those inputs can best be combined. The learning models explored here are built with the intention of being implemented in distributed

manner, enabling more local computation, so memory and processor-intensive deep learning approaches are not evaluated in this thesis. However, a better understanding of household-level usage can serve to guide the direction for more refined deep learning models, which have already proven to be powerful tools for long-term prediction at aggregate levels [11, 10].

Each model was tuned independently by splitting the data into a training set with 80% of the data and a test set with the remaining 20%, selected randomly across the dataset. The different models require particular intrinsic values to be set beforehand that parameterize the architecture of the model. To find the optimally tuned model, a grid search of the parameter space was performed to determine the best version of each type of model.

RF

The RF models validated the kNN conclusions about the types of useful features; namely that the excluding previous usage as an input results in much lower predictive accuracy than the results with only previous values, with R^2 values of 0.372 and 0.483, respectively. The combined model, with both types of features, had the highest score with $R^2 = 0.547$.

The multiple decision tree base of the RF structure enables the model to estimate the relative "importance" of the input features, defined as the features whose variance best characterize the variance of the dataset. The relative importance of the features included in the combined model are shown in Table 3.3. Additionally, since RF regression is robust to feature scaling, this type of modeling is a particularly useful tool for determining what to pass into the model without additional preprocessing.

The selected RF model uses 10 estimators with a maximum depth of 10 levels. Results for Household 1 and Household 2 are shown in Table 3.4. The size of this model is 8.36e5 bytes, comparable to the size of the statistical model described previously. Increasing the number of estimators improves predictive accuracy, but also increases model size. For example, for Household 1, using 1000 estimators and with no bounds on maximum tree depth, the RF model achieves MAE of 0.341 and MSE

feature	RF importance (H1)	RF importance (H2)
temperature	0.0148	0.0106
dew_point	0.00850	0.00683
humidity	0.00963	0.00573
apparent_temperature	0.0181	0.00916
pressure	0.00960	0.00602
wind_speed	0.00869	0.00621
cloud_cover	0.00469	0.00298
precip_probability	0.00171	0.00225
sin_hour	0.00559	0.0101
cos_hour	0.0224	0.0172
sin_minute	0.00139	0.000784
cos_minute	0.00160	0.000961
sin_dayofyear	0.00723	0.00882
cos_dayofyear	0.00856	0.00777
dayofweek_0	0.000335	0.000528
dayofweek_1	0.000756	0.000237
dayofweek_2	0.000495	0.000502
dayofweek_3	0.000478	0.000291
dayofweek_4	0.00114	0.000273
dayofweek_5	0.000411	0.000339
dayofweek_6	0.000506	0.000593
1_intervals_before	**0.713**	**0.740**
2_intervals_before	0.0318	0.0224
3_intervals_before	0.0234	0.0151
4_intervals_before	0.0255	0.0599
5_intervals_before	0.0204	0.0188
6_intervals_before	0.0151	0.0124
7_intervals_before	0.0256	0.0106
1_days_before	0.00172	0.00323
2_days_before	0.00236	0.00356
3_days_before	0.00285	0.00362
4_days_before	0.00219	0.00288
5_days_before	0.00346	0.00358
6_days_before	0.00305	0.00270
7_days_before	0.00267	0.00307

Table 3.3: Relative importance of input features for the RF regression model. The feature that varies more importance than any of the others by an order of magnitude is "1_intervals_before", which stores the usage value from the previous 15 minute time period.

of 0.313, but this model also requires 8.50e9 bytes of storage, and is significantly more time consuming to train for only marginal accuracy gains.

	MAE	**MSE**
Baseline (H1)	0.370	0.475
RF (H1)	0.344 (-7.03%)	0.315 (-33.7%)
Baseline (H2)	0.448	0.587
RF (H2)	0.400 (-10.7%)	0.404 (-31.2%)

Table 3.4: RF results for model with 10 estimators and maximum depth of 10 levels

MLP

As previously discussed, deep learning models that have received a lot of attention as a new tool for long term forecasting will not be considered due to the physical limitations of embedded smart devices. However, this does not mean excluding all layered neural networks, and basic MLP models are considered here. Different layer architectures and the resulting errors and size of model are shown in Table 3.5.

All of the architectures shown are sized comparably to the statistical model, and the error metrics indicate that the larger MLP models do not improve model performance. While there is no clear winner amongst the architectures in Table 3.5, we select the (32, 32) hidden layer architecture as the best model representative of the MLP approach to be used for further comparison.

SVR

Constructing an SVR model using all of the household data is computationally infeasible for most household IoT devices within the desired timeframe (order of minutes). With all of the data, this model achieves low error results with MAE of 0.328 and MSE of 0.350. Despite these relatively high accuracy results, in order for the calculation to be computationally feasible for the desired use case, the training dataset was reduced to 4% of the original size. Artificially restricting the possible learning is not an ideal way to reduce model size, but the predictions are still comparable to the

Layers	MAE	MSE	Model size (bytes)
(8)	0.364	0.339	1.77e4
(8,8)	0.360	0.336	2.04e4
(8,8,8)	0.351	0.334	2.31e4
(8,8,8,8)	0.354	0.336	2.54e4
(16)	0.364	0.337	2.75e4
(16,16)	0.353	0.331	3.56e4
(16,16,16)	0.361	0.336	4.45e4
(16,16,16,16)	0.350	0.336	5.25e4
(32)	0.363	0.336	4.55e4
(32,32)	0.350	0.333	7.89e4
(32,32,32)	0.351	0.335	1.13e5
(32,32,32,32)	0.348	0.340	1.48e5
(64)	0.370	0.337	8.38e4
(64,64)	0.355	0.332	2.15e5
(64,64,64)	0.357	0.334	3.49e5
(128)	0.359	0.334	1.58e5
(128,128)	0.361	0.332	6.86e5
(128,128,128)	0.355	0.333	1.21e6

Table 3.5: Possible MLP hidden layer architectures, where "Layers" describes the hidden layer structure (i.e. (16, 16, 16) has three hidden layers between the input and output layers, and each hidden layer has 16 neurons)

other ML models with MAE of 0.354 and MSE of 0.350. Results from all machine learning models are summarized in Table 3.6.

3.3.4 Results

The final selected results for the machine learning models discussed in Section 3.3.3 are shown in Table 3.6. All models demonstrate superior predictive performance over the baseline PSS model, with RF regressor being the best predictor.

	MAE	MSE	Model Size (bytes)
Baseline	0.370	0.475	8
RF	0.344 (-7.03%)	0.315 (-33.7%)	8.36e5
MLP	0.350 (-5.41%)	0.333 (-29.9%)	7.89e4
SVR	0.354 (-4.32%)	0.350 (-26.3%)	8.40e5

Table 3.6: Household 1 results for the aforementioned ML models

3.4 Takeaways

These results indicate that computationally efficient machine learning models perform better than the baseline persistent forecasting model, evaluating performance based on MAE and MSE for the 15-minute data in the Pecan Street dataset. The particular statistical modeling approach described in Section 3.2 performs worse than the baseline model.

Periodicity is a distinctive and important characteristic of energy usage that should be taken into account in modeling. However, the statistical modeling approach demonstrates that understanding periodic trends is insufficient for the challenging problem of predicting highly noisy data. Periodicity is still taken into account in the machine learning models, but is not explicitly represented in the model architecture. Instead, it is introduced through the input features.

Statistical models less heavily reliant upon periodic trends that use PSS as a foundation might outperform the baseline model. However, the machine learning models still have a major benefit that the structure and relationships between various features do not have to be set a priori, and can instead be learned. On the other hand, if these predictions are to be used to extrapolate and understand usage at a higher level, the fact that results may not be readily interpretable presents a barrier to understanding the reasoning behind the predictions and behavior. Another downside of the machine learning models is that bounds on predicted values are not as readily available as the statistical approach. For these models, error can be used as a proxy for variance.

An important quality to note about all of the methods described in this chapter is that training can be done in advance such that whatever model is being used can run in real time without retraining. The model can then be continually trained and improved in the background as new data come in, but it is not necessary to entirely retrain the models in real time given sufficient training data.

In summary, for short-term household-level prediction, machine learning models that incorporate knowledge derived from other types of modeling about usage patterns

demonstrate the best predictive performance. Thus, machine learning should be seen as a powerful tool that will not necessarily replace other types of modeling, but that can be used in conjunction with other models.

As another example, and one that could be pursued in future work, machine learning can also be incorporated into physics-based energy usage models where the shape of the load is known, but the particular parameters are unknown (see Figure 3-12). Existing research has indicated strong potential for machine learning to predict physical parameters that can then be used for more traditional models. A particular type of parameter estimation crucial for managing device-level grid interactions is being able to accurately model human behaviors within smart homes that cause devices to respond. For an HVAC system, this could be an estimation of behavioral patterns that cause disturbances (e.g. opening/closing doors, more people being in a space increasing the temperature). For an electric vehicle charging system, this could be estimating patterns of when people connect their vehicles to their charging devices. Current models assume constant parameters for these power disturbances. Future work could include exploration of parameter estimation to see if a learnable pattern can be derived. This is closely tied to modeling granular energy usage, since these disturbances are directly linked to usage.

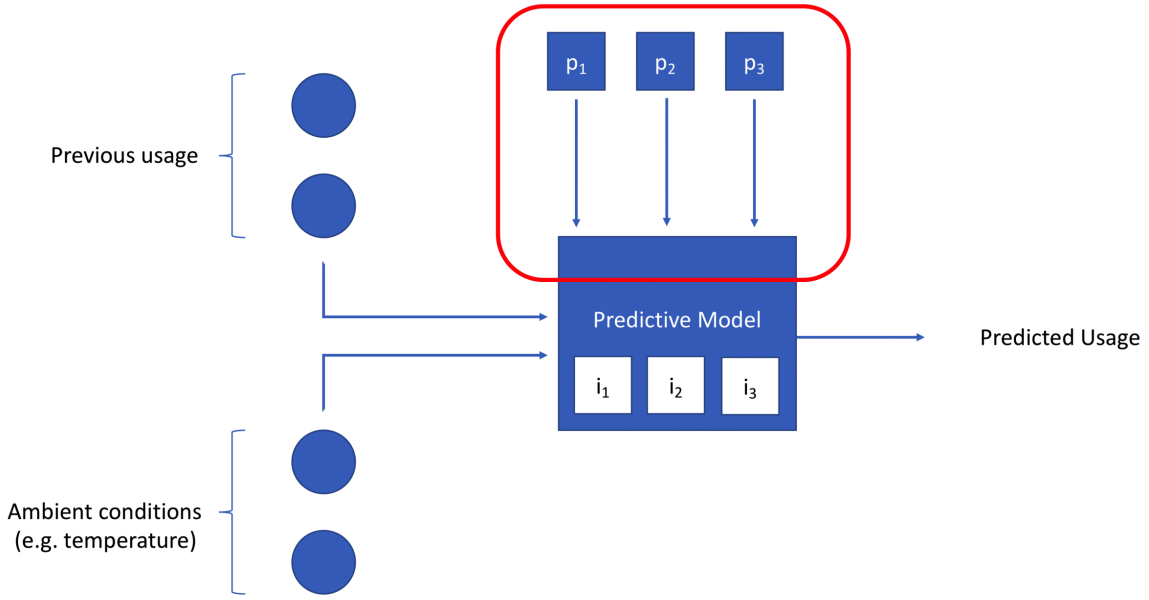

Figure 3-12: The models built for predicting energy usage take in as inputs previous usage values and/or ambient conditions such as temperature, wind, precipitation, etc. For both statistical models and machine learning models, there are some intrinsic parameters $(i_1, i_2, ...)$ that determine the model architecture. The component outlined in red shows the difference between the machine learning and statistical models: machine learning models should be able to learn external physical parameters p_i (so the red boxed portion can be removed), while statistical models will pass these in explicitly. Alternatively, machine learning models may be able to learn the p_i values to be passed in to statistical predictive models.

Chapter 4

About Blockchain

4.1 Motivation: Connection to Household-Level Prediction

In the previous chapter, we discussed methods for improving short-term household-level energy prediction. Having accurate predictions is incredibly valuable in smart grid systems to ensure optimal interactions between the distributed participants and system components. At the device level, improved prediction allows devices to better prepare for anticipated future usage, and ensure that they have access to the resources they need in order to satisfy those needs. At the aggregate level, predictions enable optimal scheduling and energy allocation to the lower level devices.

Given this improved predictive capacity, there is a need for a secure way to communicate this information between different participants, complying with computation and temporal constraints. Blockchain is currently being explored as a potential tool for facilitating these types of interactions, so the remainder of this chapter will focus on providing some background on blockchain technology and its use cases for energy systems. Next, in Chapter 5, we propose a novel system architecture that describes how smart energy devices can be connected to the grid in a secure and dynamic system.

4.2 Blockchain Background

Even if electricity consumers are not intimately familiar with the intricacies of predicting demand response and real-time electricity pricing, most people pay a utility bill to their local electricity company, and have some intuitive understanding about how electricity makes its way to their homes. In contrast, most public knowledge about blockchain is limited to having heard about Bitcoin or other cryptocurrencies. Bitcoin: the harbinger of popularized blockchain usage that acted as a tipping point for public awareness about blockchain technology. In 2008, Satoshi Nakamoto's paper described an implementation for a digital currency boasting the ability to support secure transactions without requiring a trusted third party [14].

This is partially achieved via a distributed ledger, where every transaction is completely transparent and publicly available; this can be thought of as a public and permanent database, where no data are ever erased. The fact that this database can be viewed by anyone in the system means that if Wallet 1 transfers all of its currency to Wallet 2, then if Wallet 1 later tries to purchase something, everyone is able to see that Wallet 1 lacks the necessary funds, without requiring any third party validation. All transactions, validation, and maintenance is done in a peer-to-peer (P2P) manner, rather than requiring a centralized authority. More details about how new data is written to this publicly maintained ledger will be discussed in Section 4.5.

4.3 Pairing Blockchain and Energy

The P2P nature of blockchain has the potential to be a good match for increasingly P2P electricity systems. One use case includes supporting direct P2P energy trading systems. Numerous companies are currently exploring test-cases for electricity microgrids, where microgrid participants can directly transact with each other (see Section 4.4.3). As generation and transactive capabilities move closer to the end nodes of the electricity grid, more usage data will enable better coordination between the different players. More data will give larger-scale system operators improved mod-

eling to predict consumption and market patterns to reduce strain on the grid. It will also eliminate the need to for extensive auditing or data validation. Supporting this conclusion, PricewaterhouseCoopers published an analytical report in 2017 that identifies six criteria for assessing the potential of blockchain solutions [15]:

1. Multiple parties share data - multiple participants need to view common information

2. Multiple parties update data - multiple participants take actions that need to be recorded and change the data

3. Requirement for verification - participants need to trust that the actions that are recorded are valid

4. Intermediates add cost and complexity - removal of central authority record keeper intermediaries has the potential to reduce cost and complexity

5. Interactions are time-sensitive - reducing delays has business benefits

6. Transaction interaction - transactions created by different participants depend on each other

Blockchain energy companies today are building solutions that satisfy the above criteria, with a large degree of variation in the types of approaches they are taking. Introducing blockchain to energy systems not only provides a new behind-the-scenes implementation of existing grid interactions, but also has the potential to provide improvements to areas that are not currently being addressed well, if at all. These areas include cybersecurity, privacy, single-point vulnerabilities, common-mode failures, and billing [16].

4.4 Blockchain Energy Moves to Industry

The potential for blockchain to disrupt the energy sector has been recognized by many groups, taking a variety of different approaches to solving energy problems

using blockchain. A majority of the companies can be categorized into two groups: energy tracking and demand response.

4.4.1 Smart Contracts and Tokens

Before looking into some examples of these two categories of companies, we first introduce some blockchain-related concepts relevant to the blockchain energy conversation: smart contracts and tokens. A smart contract is a piece of code that is automatically executed once an agreed-upon set of conditions is met, typically built on the Ethereum blockchain. The classic example that Nick Szabo used when first coining the term was comparing a smart contract to a vending machine: when coins are inserted into the machine (meeting the conditions), an item falls out of the machine (automatically executed outcome) back to the person who inserted the coins. On the blockchain, this asynchronous protocol provides a way for two parties to agree upon an exchange without needing to trust each other, because the code that will be executed when the contract conditions are met is publicly visible [17]. The payment used to execute a smart contract typically comes in the form of a token. However, tokens have other uses as well. For example, a token might represent a real-world asset (security token) or grant purchasing power for a particular good or service (utility token). It is very typical of blockchain energy companies to issue at least one new token to facilitate part of their services.

4.4.2 Energy Tracking

Many energy consumers today want the ability to choose and track where their energy originates. Additionally, being able to trace energy production is essential for properly issuing Renewable Energy Credits (RECs). In the United states, RECs are used as a mechanism to incentivize using renewable energy sources; RECs are legally recognized, and issued according to the amount and type of renewable electricity generated [18]. However, the existing process of buying and selling RECs is complicated and costly due to high overhead costs. RECs are poorly audited and often doubled-

counted due to the challenges of working with so many different organizations [19].

Launched out of MIT, SolarCoin aims to solve part of this problem by issuing 1 SLR token for each 1 MWh of solar electricity generation, similar to a REC. SLRs can be exchanged for other currencies, or redeemed directly with providers who have elected to participate in the SolarCoin ecosystem [20]. Swytch.io is taking a more complex approach by using their oracle technology to reward participants with Swtych tokens based on the marginal "impact" of a unit of energy. They use a new Proof of Production (PoP) protocol to validate the smart meter data they receive, comparing observed data to estimates, and flagging outliers to maintain system integrity [21]. Both of these companies take advantage of data sent directly from the source, providing a compelling alternative to the expensive and incomplete manual auditing that exists today. The data from these devices is also used for improving models and predictions of energy production and consumption.

4.4.3 Demand Response

Demand response resources are resources that are able to dynamically adjust to and supplement more centralized energy resources depending on how energy is being consumed. In the US, demand response (DR) capacity comprises around 9% of peak demand. Although this is much higher than average global adoption, DR resources suffer from structural issues. Most DR programs rely upon manual processes, and for utilities that do not have accurate load modeling and monitoring, this contributes to high costs and response errors [19].

At the time of writing, there are already many blockchain companies exploring solutions that handle demand response and optimize the sources of energy for consumers. Grid+ uses a hardware agent that can be easily installed into a home to learn typical energy usage patterns. The agent uses these patterns to decide where and when to purchase energy, selecting the most efficient sources. Grid+ has also developed an innovative blockchain-backed payment system, making it easy for users to pay for energy without requiring user knowledge about the underlying payment infrastructure [22]. Grid+ is one of the primary projects under the umbrella of Con-

senSys, a company whose mission is "building the infrastructure, applications, and practices" on the Ethereum platform [23]. Taking a slightly different approach, Drift also uses smart software to find the most efficient energy available, but focuses on working with local utilities to create more accurate forecasts for ISOs to use [24].

An even more direct solution to demand response is to support local microgrids, where participants can transact directly with each other. One notable example of such a system is the Brooklyn Microgrid, led by LO3 Energy. Like Swytch, they take more than just the watt-hours of energy produced into account; their "exergy" concept represents the portion of energy available for useful work, which includes where, how, and when the energy is produced. They have developed a platform to support a localized energy marketplace that is integrated together with existing grid infrastructure, where prosumers can autonomously transact. Distributed system operators use transaction data to manage energy use, load balancing, and demand response [25]. The blockchain energy space is moving forward internationally as well; Conjoule and Verv are two examples of companies that have built blockchain-backed P2P energy trading systems in Europe [26, 27].

4.5 Consensus Mechanisms

So far, blockchain has been discussed as a tool that might play a role in energy-related applications. However, the technology itself poses some energy-related concerns. In order to maintain a distributed ledger, there must be a mechanism to determine what can be added to this ever-growing chain of data. In most blockchain systems, new data to be added to the chain is broadcast and disseminated throughout the whole network; after enough data have accrued, a node selected by the consensus mechanism writes the new block to the chain, and typically receives a reward. Due to the decentralized nature of blockchain, different entities participating in a blockchain network might have conflicting ideas about what should go into the next block, either because they have an incomplete view, or because they are trying to manipulate the network to their own advantage. In order to ensure that the network is kept in sync

and maintained equitably, various consensus mechanisms have been proposed.

4.5.1 Proof of Work

Proof of Work (PoW) schemes rely on the fact that some computations are hard to solve, but easy to verify. A simple example is solving for the prime factors of a large number: finding the prime factors is computationally expensive, but once they have been found, simply taking the product will verify their correctness. Bitcoin uses a hashing-based PoW algorithm as this cryptographic puzzle, and the first entity ("miner") to solve this puzzle gets to add a new "block" to the chain, where each block contains contains a list of transactions. Once the block has been mined, it is added to the chain, and those transaction are executed. The miner also receives a reward in bitcoins.

In order to mine a block, a miner must find some number (the nonce) such that the hash of the nonce and block is less than some threshold value. Computing a hash value is computationally simple, but reversing a hash function is not, so the miner must repeatedly try different possible nonce values until they are able to find a suitable nonce. Once they have done so, they will broadcast their solution to the network, and it will be easy for other participants to verify their solution. In PoW schemes like this one, the probability of a certain node successfully mining a block is proportional to the hashing power of that node, which explains the enormous electricity usage dedicated to solving these PoW challenges [28]. A Bitcoin Energy Consumption Index estimates that at the time of this thesis, Bitcoin's current annual electricity consumption is estimated to be 47 TWh, emitting an additional 23,000 kt of carbon dioxide per year [29].

A major benefit of using PoW is that the blockchain can be permissionless, meaning that the participants in the system are not required to trust each other, and there is no need for authentication. This is because no single participant will be able to overpower the network unless they are able to control at least 51% of the total computing power. However, mining blocks is becoming increasingly difficult due to the high computing power already in the network. As a result, some miners now

participate in mining pools, where many miners work together and split the payout if they are successful [30]. This is exacerbated by the disproportionate gains from being able to invest upfront money in specialized mining machines. Thus, the system is becoming increasingly centralized [31].

4.5.2 Proof of Stake (PoS)

PoS is another category of consensus mechanisms that enables public, permissionless blockchains. In PoS, a node is given validating power that enables them to write a new block. Rather than requiring the solution to some sort of cryptographic challenge like with PoW, validating power is allocated based on the amount of that network's base currency they hold (known as their stake). The amount of stake that a node has is directly proportional to the probability that they will get validating power. Therefore, putting more currency into the network increases the stake of that node proportional to the value added, so there is less risk of centralization compared to PoW. Additionally, since there is no iterative hashing involved, a major benefit of PoS is that there is no need for huge amounts of electricity consumption [31].

4.5.3 Proof of Authority

PoW and PoS dominate the blockchain space, but the Energy Web Foundation (EWF) has developed a Proof of Authority mechanism called Aura. EWF is an open-source, scalable blockchain platform specifically designed for the energy sector's regulatory, operational, and market needs [32]. In their protocol, there are specialized "authority nodes", also known as "validators", and only these nodes are able to create new blocks. Block signing is conducted in rounds, where validators are designated a time slot during which they can create and sign a new block. Additional mechanisms are in place to ensure that even if a validator becomes disconnected from the network or attempts to act maliciously, the generally agreed upon and most recent block will be accepted [30]. Like PoS, Proof of Authority does not rely on high electricity consumption.

4.5.4 Consensus Mechanism Considerations

Many other consensus mechanism options exist, fit for different use cases and network types. Two major points of consideration are 1) trust within the network and 2) network speed. These points are largely defined within the consensus mechanisms by who gets to validate new blocks and how they perform the validation, respectively. A tradeoff between trustlessness and speed typically exists because in a trustless or anonymous setting, there is more potential for malicious actors to manipulate the system to their own benefit, so the validation process must be more robust.

Chapter 5

Secure Blockchain-Enabled DyMonDS Design

5.1 Design Goals

In this section, we introduce the Secure Blockchain-Enabled DyMonDS design, with implementation details in the subsequent system design. The primary aim of this design is to support robust smart grid management in an architecture with a high focus on system security. This is done by proposing a practical implementation plan for the framework proposed in [5], referred to as the Dynamic Monitoring and Decisions Systems (DyMonDS). The DyMonDS framework operates in a setting where embedded smart energy devices participate in a minimal information exchange with coordinators, responsible for using that information to find optimal energy allocation and scheduling solutions to send back to the devices. This framework has been shown to arrive at the same optimized solution as a fully centralized system, despite its distributed nature and minimal information exchange.

Another complementary goal for this design is is to offer a functional blockchain use case for supporting energy systems, along with a more general discussion of some the challenges and practical use cases of blockchain for grid management. In the context of the discussion from Chapter 4, we can break down the blockchain discussion into three components: what data is written to the ledger, who is participating in

the exchange, and how the ledger is maintained. The ledger will include the data relevant for the DyMonDS calculations sent from the embedded IoT devices, and the optimal solutions. Only trusted nodes are permitted to participate in the network, and these nodes together represent a centralized body responsible for validating and maintaining the network. However, even these nodes that are considered "trusted" will be internally validated through physics-based and data-driven methods. This is partially achieved via the strategies discussed in Chapter 3 by generating real-time energy predictions, and identifying anomalous behavior that deviates substantially from the predicted values.

5.2 System Participants

Participants in this system design fall under two categories: distributed energy resources (DERs) and compute nodes. While the remainder of this architecture will be described in reference to a system with these two types of participants, it can be extended to any two-tiered architecture with participants of varying security levels and computational capabilities. For the particular setup described here, DERs include smart IoT energy devices, which can include both uncontrollable loads such as photovoltaic cells and controllable loads such as smart water heaters, thermostats, or small local storage. These household IoT devices have relatively limited local memory and processing power, so their primary responsibility is to communicate their local data to the compute nodes rather than performing much complex computation locally. The compute nodes are connected to each other via a meshed blockchain network, where compute nodes in different neighborhoods are synchronized in order to avoid single-point failures and allow for internal P2P validation. Each compute node is responsible for coordinating the DERs within its region and for interfacing between the DERs and higher system-level signals. This setup is shown in Figure 5-1, where household DERs communicate with their neighborhood compute node through a structured communication protocol described in Section 5.4.

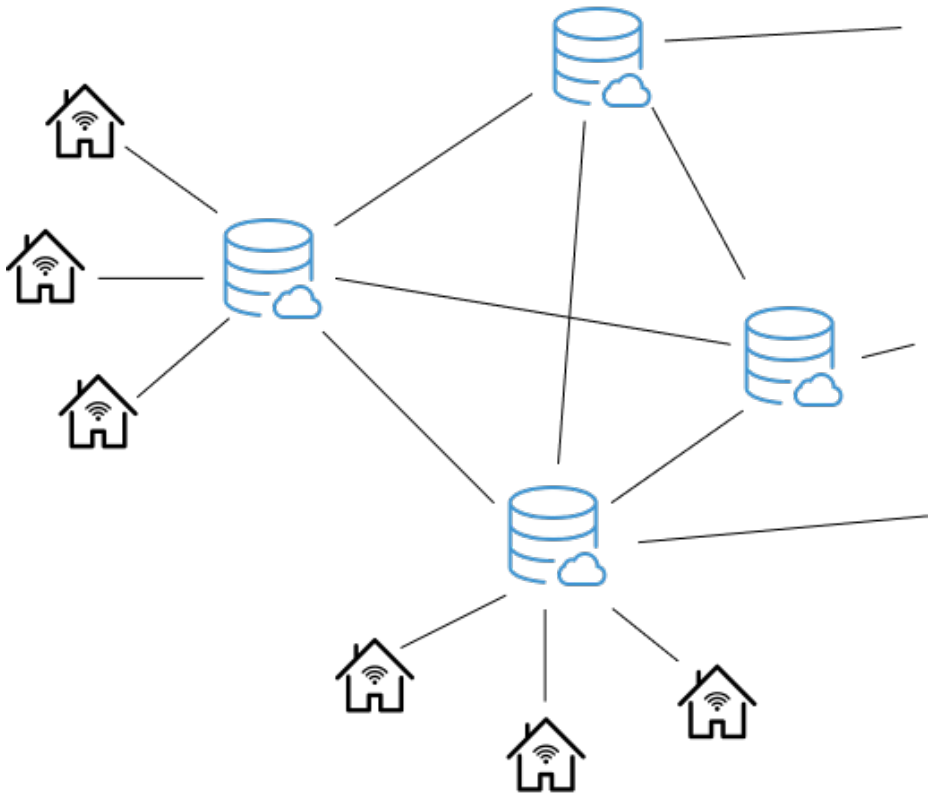

Figure 5-1: DERs (black) directly communicate with their local neighborhood compute node (blue). Compute nodes are connected to each other via a blockchain network.

5.3 DyMonDS Framework

5.3.1 Coordination through Device Controllers

Coordinating DER behavior within the context of the overall grid system is a challenging problem due to the many factors and potentially competing objectives that impact the effectiveness of such a system. At the device level, any individual device has its own set of local computations that determine how it will respond to various signals. These behaviors might be driven by direct user input, controllable cyber signals, or responses to exogenous drivers [33]. At the system level, controllable smart devices are powerful tools for optimizing system performance, but the constraints outlined by each device's physical characteristics or user preferences must be maintained in order to ensure that quality of service (QoS) specifications are met.

In today's grid, small DERs typically respond to signals that are generated locally like the state of the system (e.g. state of charge) and measurable exogenous variables (e.g. temperature). The responses are typically a basic ON/OFF operation, where the DERs draw or inject unconditional amounts of power when needed. However, as IoT devices become more intelligent and capable of more fine-grained control and real-time responsiveness, there arise more opportunities for adjusting behavior such that grid-level commitments are satisfied over slower time intervals, but fast device-specific actions are optimized for the user. Specifically, there is some flexibility in when energy is drawn or injected such that customers have all of their needs satisfied in practice, but where the actions are optimally timed such that their electricity bill is minimized. In a system where participating devices implement a minimal interface as defined by the DyMonDS framework, this seemingly intractable problem is reduced to a convex optimization problem with a unique solution, which the compute nodes can then communicate back to the respective devices [5].

5.3.2 Minimal Information Exchange

The optimizations within the proposed DyMonDS framework are economically driven, so DERs are responsible for generating bids to send to the compute nodes. The interface that each participating device implements in order to facilitate optimized interactions with the grid is defined by the triplet of stored energy (E), power (P), and rate of change of power ($\dot{P}$). The generated bids are a direct function of this triplet, shown in the upper layer of Figure 5-2 [34]. The intermediate generalized droop layer as described in [35] maps upper level energy and power data to lower level QoS specifications. For this generalized control figure, characteristics of the particular device are taken into account at the lower layer to ensure that the QoS specifications are met and that the various input signals are appropriately accounted for. The DyMonDS framework allows all such devices to participate in this connected system as long as the appropriate interface is implemented, regardless of the underlying design. The device-specific logic is not directly exposed outside of the interface, which helps protect the devices from cyber attack by abstracting away from the internal control

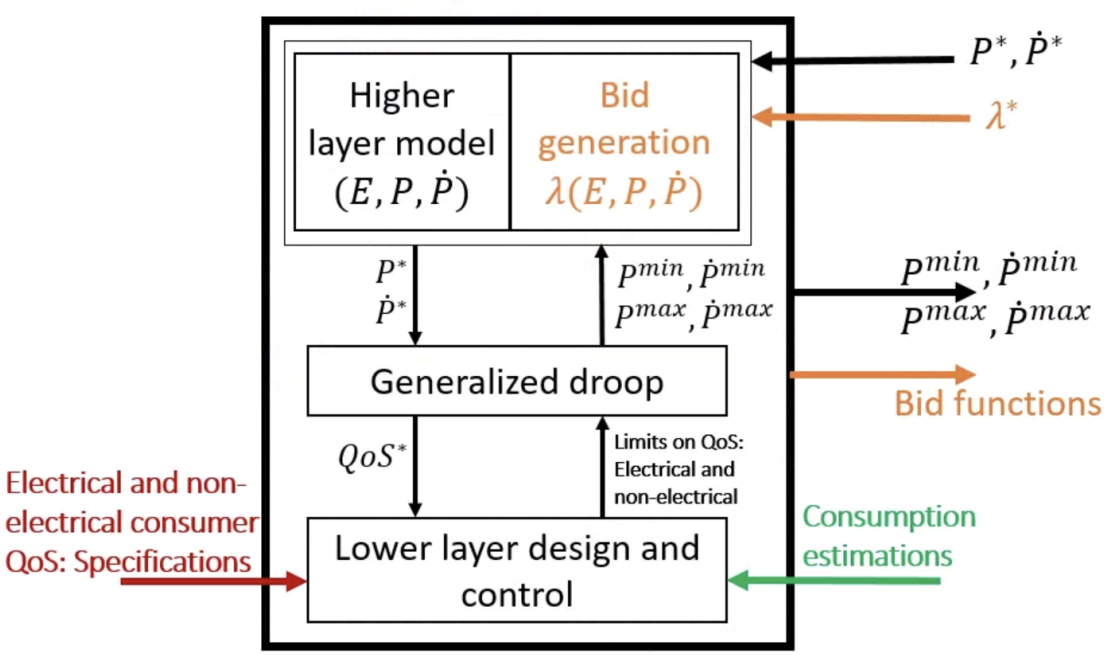

Figure 5-2: Interactive modeling and control design for DERs

system. When the compute node, acting as a coordinator for the system, receives this information from all of the neighborhood devices, it can then take the physical constraints and generated bid functions into account, and send back to each device a control signal for the next time interval. In [35], we see that for both real and reactive power, these signals exhibit linear grid interactions, enabling this distributed system to arrive at the same convex minimum solution as a fully centralized system.

The composite nature of this type of modeling enables this framework to scale well to various levels of the electricity grid. Likewise, the implementation for communicating this information as described in the following sections is not specific to DERs and compute nodes, but can be extended further up the layers of grid interactions, or potentially applied to different domains.

5.4 Secure Communication Protocol

The information that needs to be communicated in the DyMonDS framework as shown in Figure 5-2 will be exchanged between DERs and compute nodes via a

structured TCP-liked protocol. In this type of exchange, when a data packet is sent, a record of this interaction is stored; when a packet is received, the recipient sends an acknowledgement. If a communication failure occurs, the sender is aware that they did not receive the acknowledgement they were expecting from the recipient, and can resend the data. For memory-limited DERs, this means that after they send relevant data to the compute nodes, they can clear this data from their local memory once the compute node acknowledges the data receival.

Any type of information exchange across a public network exposes the system to potential security vulnerabilities, so the data must be stored and communicated in a secure manner. In this system, the two main security considerations are protecting participant privacy and maintaining data integrity. Participant privacy refers not only to ensuring that individual household energy data is kept confidential, but also keeping the internal mechanisms of the IoT devices and bid generation functions hidden [36]. Data integrity is crucial for the DyMonDS exchange because the solutions are based upon the aggregation of individual device characteristics and bids functions, so if the individual devices are compromised, the optimal solution may be skewed. In general, protecting against smart grid security breaches is crucial because a breach could expose personal data, increase costs, damage hardware, or cause power outages.

5.4.1 IoT Security

The rate of IoT adoption today in practice is outpacing the policies necessary to ensure secure implementation. Building a physical structure requires complying with a set of building codes, but there exists no parallel set of standards for building new software. A set of security policies is proposed in [37], and considered while evaluating the best ways to incorporate different components of this system design. Many IoT devices have been developed without a focus on security, so due to hardware and budget constraints, security mechanisms are typically quite limited [38, 39]. Additionally, the fact that these devices are typically directly connected to some public network means that they are a prime candidate for cyber attack. For the secure system design here, this means that even if the communication between DERs and compute nodes

is secured, the DERs still have the potential to become corrupted by external sources.

5.4.2 Protecting Privacy and Data Integrity

Data integrity is crucial for this system, because the data being broadcast by the DERs determines the optimized system-level solutions. To evaluate security, we consider potential avenues for an attacker to infiltrate the system. This includes attacks where the DERs themselves are corrupted, the stored data is corrupted, or the data is corrupted while being communicated.

To impede malicious DERs from entering the system, onboarding new devices requires going through the centralized management source to confirm device identity. This can be done by working directly with hardware companies to confirm existence of uniquely identifiable devices, or by developing a device's reputation over time based on its supposed identity [40]. This is useful for establishing new DER integrity at the onset, but as discussed above, IoT devices are particularly susceptible to cyber attack due to inadequate security mechanisms. For this reason, they will be subject to continual validation even after being incorporated into the system to ensure that if the device becomes corrupted after it is already actively participating in the system, this behavior can be detected. This validation is discussed further in Section 5.6.

Public-key encryption is used to prevent attackers from viewing or altering stored data or data that is being exchanged between the DERs and compute nodes. All stored and exchanged data is encrypted, such that only the intended recipient is able to decrypt the data and reveal the underlying message using their private key. Thus, any attacker who attempts to read this data will be unable to do so without access to the private key. At a high level, these encryption schemes rely on the fact that some problems are computationally infeasible to solve, but easy to validate.

Public-key cryptography will be used in two other ways to ensure that attackers cannot infiltrate the system. One potential attack could involve an actor who pretends to be an onboarded DER and sends a message to the local compute node. There is thus a needs for a method to guarantee that the parties exchanging data are who they claim to be. To achieve this, all exchanges will be signed with a digital signature; this

involves the sender using their own private key to "sign" each message such that the recipient can use the signer's public key to confirm the identity of the sender, but in such a way that the sender's private key is not exposed. In another type of attack, although unlikely, a malicious attacker who is unable to read or forge data (since we have already protected against that) may simply try to take down the system by altering the messages that are exchanged in what is known as a "man-in-the-middle" attack. In such an attack, the malicious actor would intercept the message, alter its contents, and forward it along to the intended recipient. Although this would most likely result in a meaningless decrypted message, if the attacker had managed to acquire some knowledge of the form of the data or the encryption scheme being used, it may be possible for them to alter the message in such a way that the result is realistic, but incorrect data. This type of data modification can be prevented by attaching a message authentication code (MAC) to every message that is sent across the network. A MAC is generated directly from the decrypted message, so given a MAC, the recipient can decrypt the message and easily verify that the MAC is valid for that message. If the message is altered, the MAC will no longer be valid.

Public-key cryptography is a recognized and established method of ensuring system security, but at its root, it relies upon the assumption that the private keys are entirely secured. In all of the aforementioned security schemes, if the private keys are exposed, then encrypted messages can be decrypted, and identities can be easily forged. Particularly given the discussion of the security concerns around IoT devices, extra care should be taken to protect these private keys, so the security of the system as a whole can be strengthened by using a private key management scheme. Such schemes designed for smart devices in particular already exist and include policies for key generation, distribution, and management that make acquiring these private keys even more challenging for attackers [41, 42, 43].

5.5 Blockchain Utilization

5.5.1 Challenges of Blockchain for Energy

Looking at blockchain in general, one of the major benefits of this tool is the ability to establish trustless networks where any party can participate, and malicious actors are implicitly prevented from overtaking the network due to computational barriers. However, trustlessness is challenging for energy systems because the data being exchanged is reliant upon physical devices, so there needs to be some sort of verifiable link between the reported data detected by the hardware and the hardware itself. Without a verifiable link, there would be no way to confirm that energy data being recorded was actually linked to a physical device. In order to establish this connection, there already needs to be some semblance of trust in the system, breaking pure trustlessness. While this might seem like a reversion away from some of the power of blockchain as tool, putting this system in context, current energy grid participants are already required to fully trust a centralized governing body with little information about its underlying logic. This system adds security and transparency over the existing setup.

Rather than utilizing a trustless consensus mechanism where anyone can participate, we instead propose using an underlying blockchain architecture like the Energy Web Foundation's Proof of Authority that allows only trusted validators to participate in writing to the chain [30]. Compute nodes thus act as validators, but in order to add an extra layer of system security, they will also be subject to P2P validation as described in Section 5.6. If new compute nodes want to join the system, they must first be identified as trusted by the other compute nodes or otherwise certified by a trusted source. An additional benefit and necessity of this type of consensus mechanism is that network speed can be maintained even as the system scales. In this setting of real time responsive bidding and behavioral adjustments, the network must be able to validate blocks quickly.

DERs thus participate in the blockchain network only indirectly through their neighborhood compute node. Not only does this protect the integrity of this trusted

network due to the susceptibility of IoT devices to cyber-attack, but even if IoT devices could be trusted, typical smart energy devices lack the memory and processing capabilities necessary to participate in maintaining the complete ledger, replicated across all participating devices.

5.5.2 Blockchain in Secure DyMonDS Implementation

We have so far established that in this system, DERs communicate with compute nodes through a secure communication protocol, and compute nodes act as relatively trusted nodes in a private blockchain network. The actual data being recorded in the distributed ledger that makes up the "blocks" is exactly the data that is necessary for the DyMonDS exchange shown in Figure 5-2. An example of how the new data is written is shown in Figure 5-3.

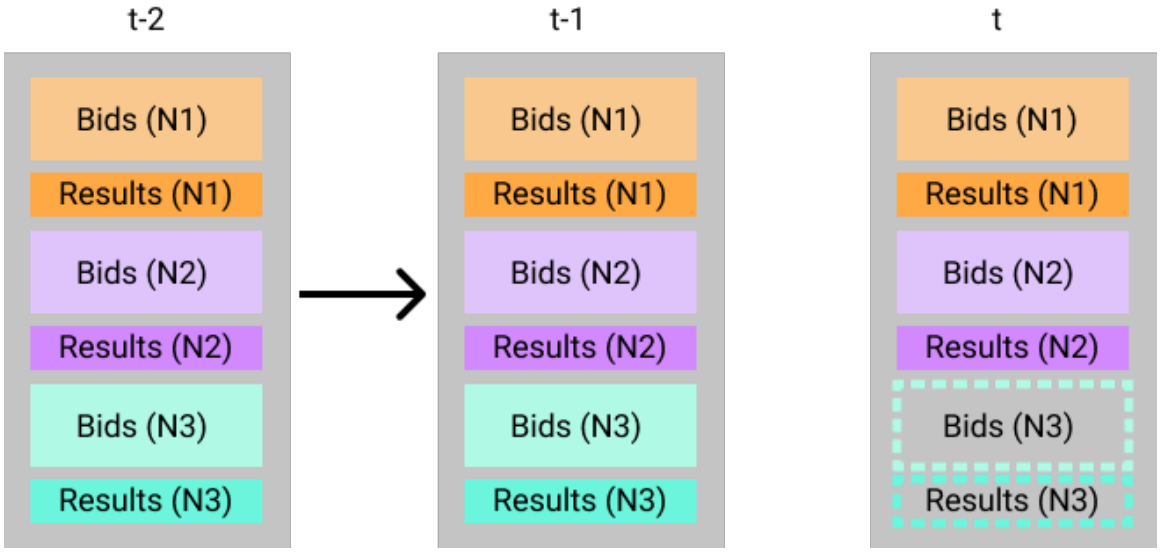

Figure 5-3: Snapshot of the blockchain ledger that is replicated across all compute nodes. Each "block" in the chain stores data about the bids and optimally computed clearing results. In this figure, there are three neighborhoods, and all of the results at time t-2 and t-1 have already been published to the chain. At time t, the data from N3 (dotted lines) have been broadcast, but not yet synchronized across all devices. After synchronization, the new block can be appended to the chain (black arrow).

In this example setup, there are three neighborhoods denoted as N1, N2, and N3. All of the DERs in N1 will communicate their bids to the N1 compute node, which then calculates an optimal solution for the system using an internal clearing

mechanism, taking into account other system-level objectives. Once it has found an optimal solution, it broadcasts this solution to the other compute nodes, and proposes adding this data to the chain. Each of the compute nodes is performing this same task, so once they have mutually exchanged the new data and synchronized the new block across compute nodes, the new block can be added to the chain. The results are then sent back to the DERs, which will adjust their behavior according to the optimized solution. Note that this setup assumes collaborative neighborhoods that are willing to exchange information about local bids and solutions; additional consensus logic would be needed if there was not mutual trust between neighborhood compute nodes.

In this system, blockchain is effectively used like a shared database with protected read and write capabilities. One major vulnerability of many existing energy systems is that with a traditional database model, the server storing information relevant for system management is not only at risk of single-point failure if the server goes down, but it also acts as a bottleneck if all information exchange must go through the same machine. Blockchain is designed with a P2P backbone, so it aligns very naturally with the distributed nature of the DyMonDS framework, and is setup well for real-time validation and information exchange. Blockchain is also well-suited to support the DyMonDS framework because the minimal-exchange infrastructure prevents the ledger from needing unbounded amounts of memory capacity.

5.6 Validation through Learning

As discussed previously, IoT devices are particularly vulnerable to cyber-attack due to limited security mechanisms and virtually no policy to regulate security requirements. Additionally, while the compute nodes together act as a centralized management body, the security of the system a whole is much stronger if these nodes are not blindly trusted. To address both of these concerns, the system will be continually validated in real time so that if previously trusted sources become corrupted, this can be identified in a P2P manner.

Using machine learning to identify anomalous energy behavior has been done at various levels of the grid as shown in [44, 45, 46]. Other approaches model threats with mathematical processes like in [47], and consider complications such as identifying corruption even in the face of many compromised components [48].

An advantage for detecting suspicious behavior in this setting is that the data being exchanged in a smart grid is quite constrained in the possible values that can be taken on. This is partially due to physical constraints of the system that cannot be violated, and partially dictated by known user preferences and responses to ambient conditions. This means that a compute node can detect whether the data it is receiving from a particular DER is possible using physics-based modeling given the properties of that particular device. This serves as a baseline for identifying corrupted devices. These physics-based models can be further developed by taking into consideration known periodic energy usage trends that can be extrapolated through decomposition modeling as shown in [49].

To take into account the fact that energy usage follows particular patterns at more granular levels depending on the type of device, user preferences, and ambient conditions, data-driven predictive modeling can be used to characterize "typical" usage behavior. The models and experiments explored in Chapter 3 focus on developing predictive models for household-level prediction at short time intervals. Refining these types of prediction mechanisms is useful for the devices to anticipate and optimize for expected future usage, but is also useful for protecting system integrity. If realized usage substantially deviates from predicted usage based off previously learned typical behavior, then this may be an indication that the device has been corrupted. However, note that due to the inherent noisiness of usage patterns at short time intervals, determining what constitutes a significant enough deviation to be flagged as potentially corrupt is a margin which needs to be learned as well.

Chapter 6

Conclusion

6.1 Contributions

We will evaluate the contributions of this thesis with respect to the two primary objectives introduced in Section 1.3.2, shown again here:

Objective 1: **Evaluate different machine learning energy forecasting models and compare their predictive capabilities with existing statistical models.** We have constructed a statistical prediction model based off of a successful stochastic simulation model, and found that due to the noisiness of household-level data, PSS outperforms this statistical model. Basic machine learning models were explored, and these models were able to outperform PSS, with RF Regressor as the lowest error model. However, these models are not blindly data-driven, but rather, the input features take into account periodic and ambient factors known to be relevant for energy. The results from this section more generally indicate that known periodic trends that are accurate at spatial or temporal aggregation levels largely fail to capture the noisiness of short-term prediction at the household level. For this reason, PSS is a good estimate of future usage, and an even better model will take this information into account directly, and supplement with other features for further accuracy gains.

Objective 2: **Design a system architecture for a blockchain-backed en-**

ergy system, and assess the underlying blockchain architectures that align well with the goals of such a system. We have presented a system design that leverages the optimal minimal-exchange DyMonDS framework in order to enable secure communication between between household-level DERs and neighborhood-level compute nodes. Using the mathematical formulations necessary to provide optimal grid-wide solutions and designing a system around them is a crucial step in order to change and improve grid management today; in doing so, making such systems cyber-secure at the outset rather than as an afterthought will be necessary for being able to support these increasingly interconnected systems. In the process of constructing this design, the challenges and potential use cases of blockchain as a tool for supporting highly interconnected energy systems are described.

While these contributions have primarily been posed within the context of deregulated industry environments, they are valuable for regulated settings as well, where there is still a need for accurate predictive mechanisms and secure information exchange.

6.2 Future Work

For household-level prediction, some avenues for future work include experimenting with using learning from one household to better understand another. The models shown in this thesis treat different households as entirely independent. Another direction is to focus on improving statistical methods. Many statistical models involve some form of clustering, but this is challenging to utilize for prediction because clusters are typically based on the load profile as a whole, which is unknown at the beginning of the day. Thus, using machine learning to predict which cluster a current load is likely to fall into would be a useful tool for better characterizing real-time data. Alternatively, rather than using standard k-means clustering or other purely mathematically-driven models, finding a way to cluster the data such that the clusters better-represent some interpretable situation (i.e. seasonal, rainy day, etc.) could be

helpful for ISOs to understand the different types of possible usage scenarios.

For the Secure Blockchain-Enabled DyMonDS design, the primary candidate for future work is to test the system and confirm that the time-requirements for bid generation and scheduling can be met. Additionally, the design presented in this paper describes the interactions between household DERs and local neighborhood compute nodes, but this same architecture can be extended to higher, more aggregated levels of the energy grid system, or to other domains entirely. Such an extension requires further work to develop a more formal process for onboarding new actors in a secure manner.

Appendix A

About the Data

Pecan Street is a research and development organization whose goal is to work towards technical and policy solutions for challenges related to energy and water [12]. The first step in accomplishing this goal is to develop a detailed understanding of energy and water use behavior. Their claim of providing the worlds best data on consumer energy and water consumption behavior is backed by a network of over 1000 active research participants primarily located in Texas, Colorado, and California. The database includes information about households participating in various usage-related experiments, since many research institutions have taken advantage of the Pecan Street households to test out energy solutions.

The participating homes have IoT devices streaming data down to second-by-second granularity from devices like water heaters, solar panels, lights, and more. This data is accessible to university researchers, and the analysis conducted throughout this thesis is all done using Pecan Street data, although the intention is to demonstrate and compare methodologies that can be applied independently of the datasource.

The primary sources of data used throughout this thesis come from two data tables: `weather` and `electricity_egauge_15min`. The weather table contains weather data each hour for Austin, Texas; Boulder, Colorado; and San Diego, California. Each participating household has a unique dataid, so using the Dataport Metadata file (accessed through the Spotlight section of the Pecan Street Dataport), weather can be linked to households through their location. A complete list of the columns

accessible from each of these tables is shown in Appendix C.

The dataport can be queried directly using SQL queries; The following query, for example, retrieves relevant weather data for the homes in San Diego: `SELECT localhour, temperature, dew_point, humidity, apparent_temperature, pressure, wind_speed, cloud_cover, precip_probability FROM university.weather WHERE latitude = 32.778033 AND longitude = -117.151885;`

The following figures show some examples of the shapes of various device loads collected at 15-minute intervals for two of the Pecan Street households.

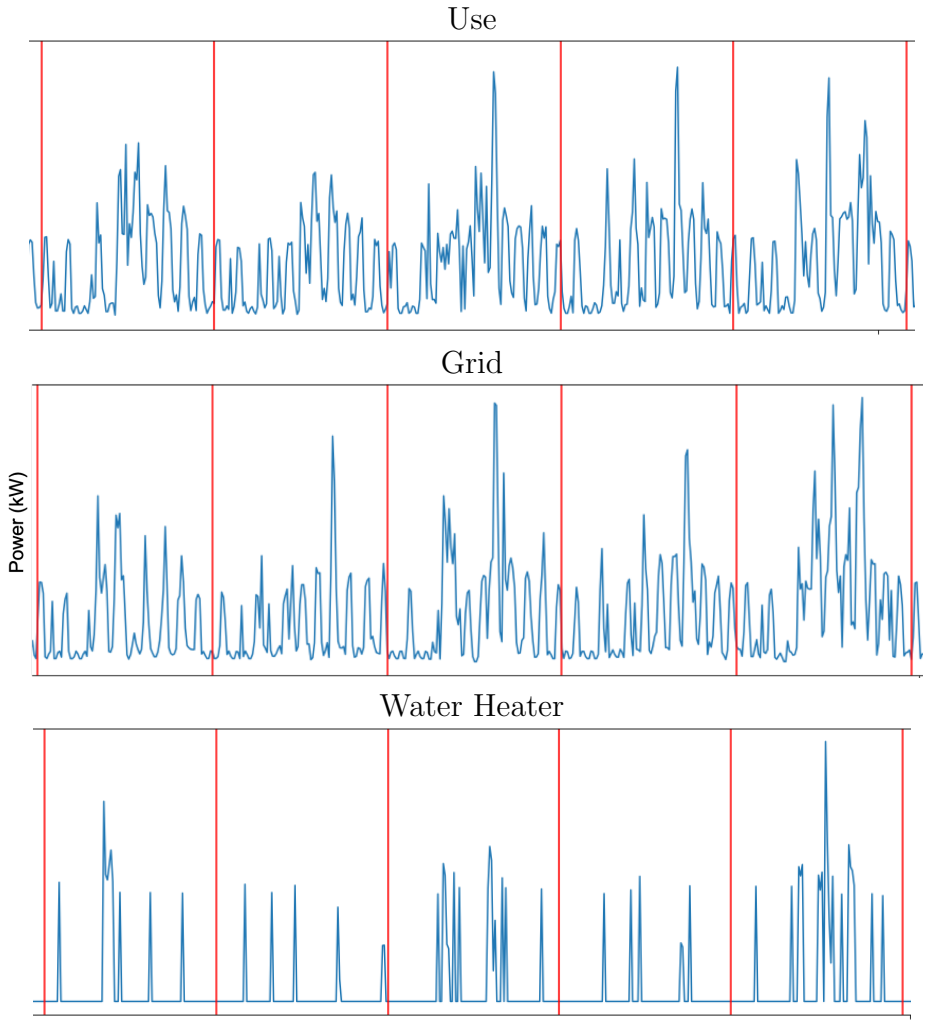

Figure A-1: Snapshot of energy usage data collected for a single household. Vertical red lines separate days.

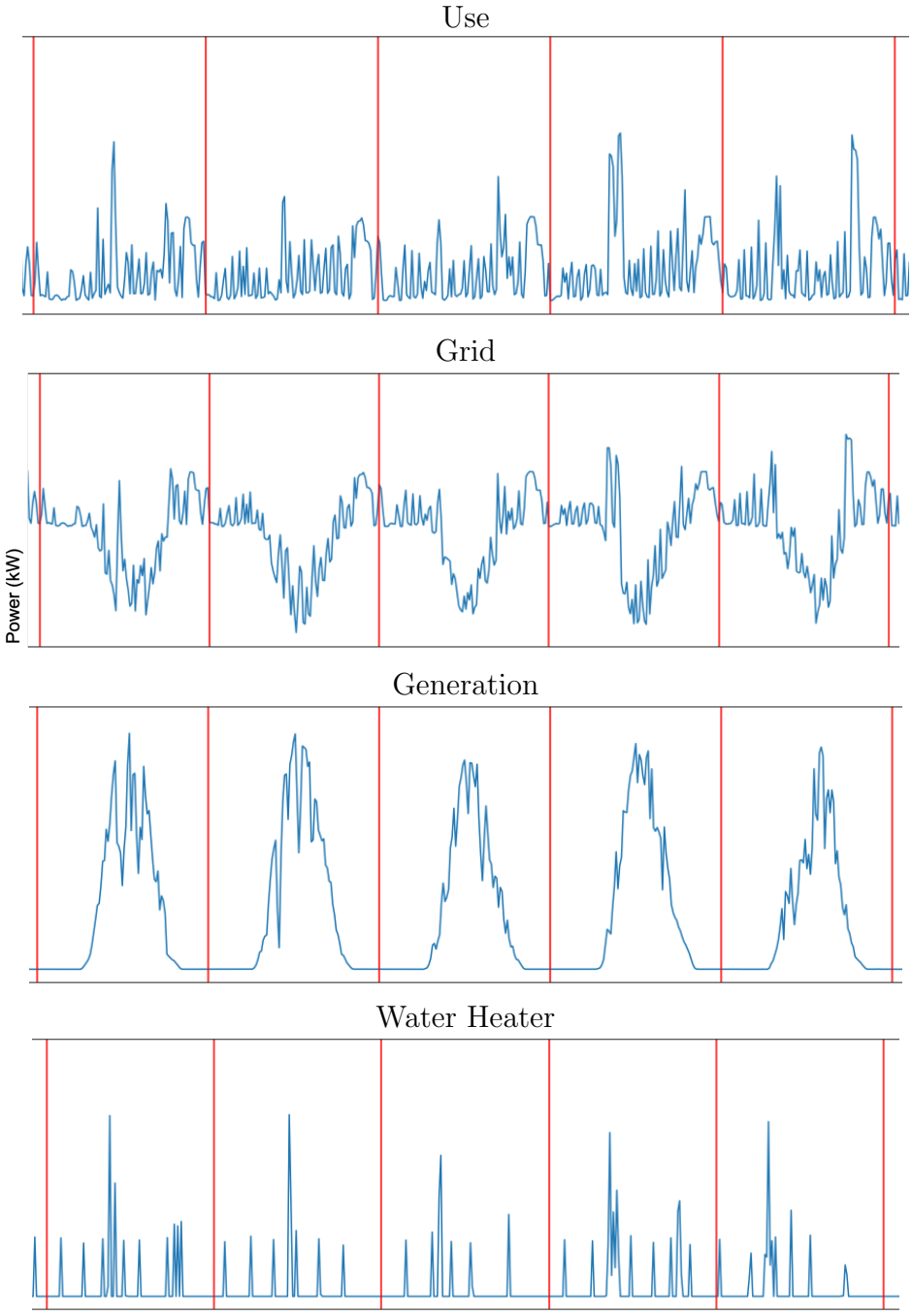

Figure A-2: Snapshot of data collected for a household with solar generation.

Appendix B

Basic Machine Learning Models

This appendix introduces some popular regression methods and how they can be used with the given Pecan Street data to predict household energy usage. We discuss at relatively high level the intuition behind some common regression methods, and what their role might be in energy modeling and prediction. Note that we discuss only regression methods (as opposed to classification methods), since we are trying to predict a particular energy output value. We will sometimes refer to real or predicted energy value as the "label". The inputted data can be represented by a vector of features, each of which has a numerical value.

B.1 k-Nearest Neighbors

For k-Nearest Neighbors (kNN) classification, the label of some new test point is determined by finding the most common label of the k points which are closest to the test point. Similarly, kNN regression finds the k points closest to the test point, and averages their labels, outputting this as the label for the test point. The two key factors that have a large impact on the performance of this algorithm are 1) selecting an suitable value of k, and 2) defining an appropriate distance metric. An example of kNN classification is shown in Figure B-1.

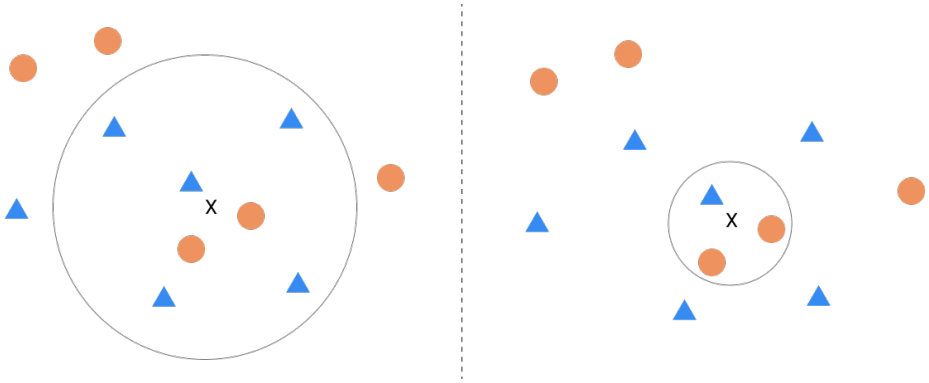

Figure B-1: Example of k-Nearest Neighbors classification. On the left k=7, so the new test point X will have a blue predicted label. On the right, k=3, so X will have an orange predicted label

B.2 Random Forest Regression

A random forest model is built using a collection of individual decision trees. Decision trees can be used to build regression models by splitting up the original data into decreasing sized subsets, with the goal that as the data is split, the subsets are similar according to a particular attribute. At each step of the decision tree process, the attribute and split are selected so as to reduce the standard deviation within each side of the split; each internal decision node thus has a particular attribute, as well as the criteria for splitting (i.e. the value is greater/less than some threshold). For regression, the value outputted at the leaf nodes at the bottom of the tree can be found by averaging the values of all of the training data that falls into that final leaf node.

A random forest regressor (RF Regressor) is called a "forest" because it is comprised of a set of decision trees, all built from the same data. The trees will not be identical due to the randomness added to the model: in the process of determining which attribute to next split on, instead of considering all of the possible attributes, the algorithm will only look at a random subset. After conducting this process independently to create multiple decision trees, the results of all of the trees are averaged to output the final result. This randomness makes RF Regressors quite robust against overfitting.

The tree structure used in RF Regression makes it possible to extract the relative importance of different attributes of the data in determining the final outcome. However, a disadvantage is that since the model does not generate aggregate trends or continuous visual outputs, it is not conducive to qualitative analysis.

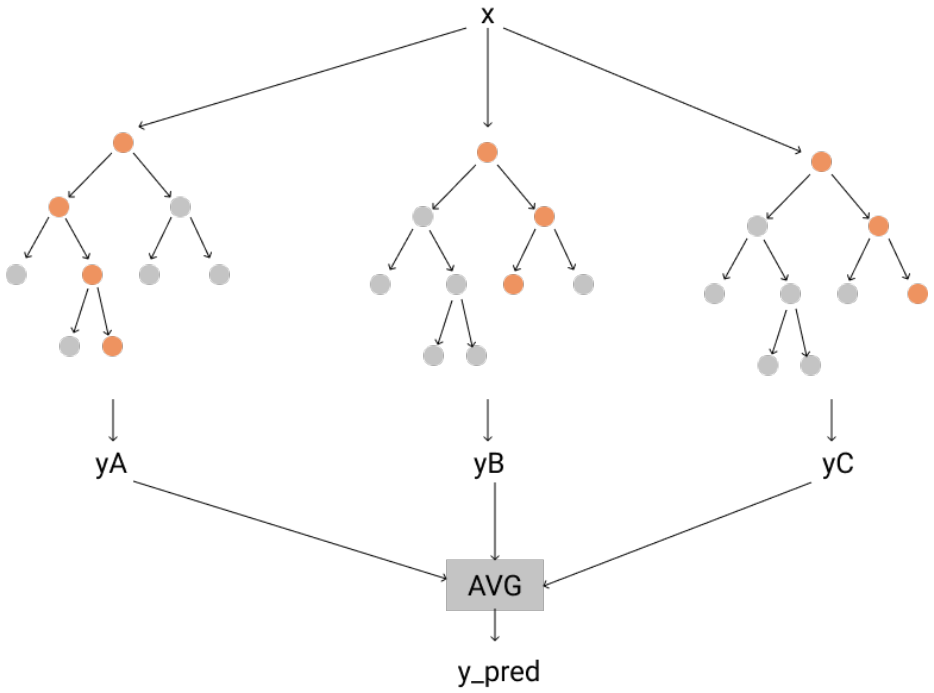

Figure B-2: Random forest regression model. Each decision tree may be split on different attributes at each node, and the end result will be the average prediction across all models.

B.3 Multilayer Perceptron

A multilayer perceptron (MLP) is a feedforward neural network, whose basic structure consists of an input layer, some number of hidden layers (≥ 1), and an output layer. For example, for energy usage prediction, the input layer will be the data taken from Pecan Street, and the output will be the expected energy usage. Each hidden layer consists of at least one neuron. Each neuron takes in a weighted sum of the values from the previous layer (and potentially a bias term) and passes this into a non-linear activation function, outputting the result of the function. An MLP can exactly reconstruct a linear regression model, but can also be used for data that is not

linearly structured. This type of model can thus learn patterns that are dependent upon particular combinations of input features.

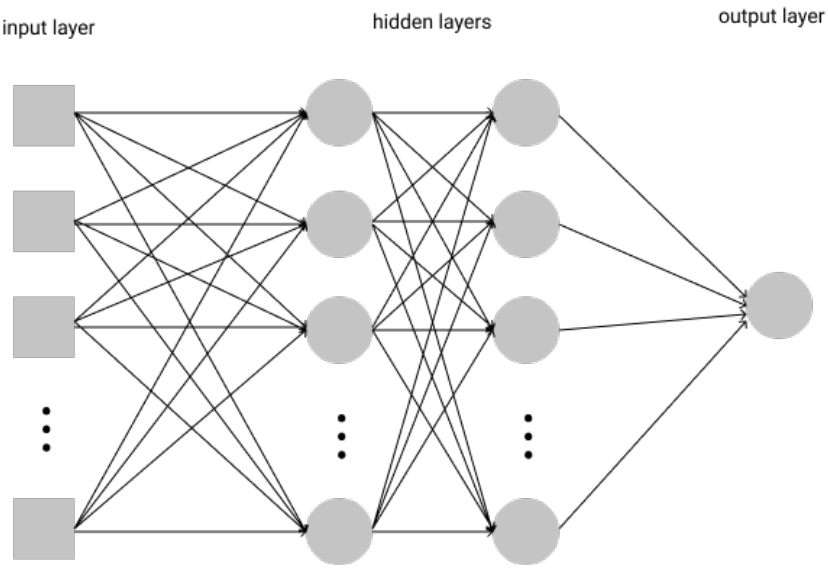

Figure B-3: Multi-layer Perceptron architecture with 2 hidden layers. The different components of the vector input will be passed in as each node of the input layer, and the final energy consumption prediction is the result in the output layer.

B.4 Support Vector Regression

For binary classification problems, support vector machines (SVMs) aim to find a hyperplane to separate this data with some margin. Breaking this statement down: some data is not linearly separable, so kernel functions are used to transform the data into a new space where it ideally becomes separable. Various model parameters determine the width of the margin around the hyperplane (a smaller margin will classify more points correctly, but a larger margin may generalize better) and whether points far from the hyperplane will be considered when tweaking the location of the plane. These are typically notated as C and γ, respectively. Similarly for regression, all predictions are restricted to be within some constant bound ϵ of the actual label, similar to the margin concept for classification. The ϵ bound introduces regularization to the problem as high ϵ will be less sensitive to misclassification and may generalize better, but low ϵ will more closely fit the data.

Appendix C

Accessed Pecan Street Data Tables

Table C.1: Weather Available Data

temperature
dew_point
humidity
apparent_temperature
pressure
wind_speed
cloud_cover
precip_probability

Table C.2: Household Usage Available Data

dataid	jacuzzi1
local_15min	kitchen1
use	kitchen2
air1	kitchenapp1
air2	kitchenapp2
air3	lights_plugs1
airwindowunit1	lights_plugs2
aquarium1	lights_plugs3
bathroom1	lights_plugs4
bathroom2	lights_plugs5
bedroom1	lights_plugs6
bedroom2	livingroom1
bedroom3	livingroom2
bedroom4	microwave1
bedroom5	office1
car1	outsidelights_plugs1
clotheswasher1	outsidelights_plugs2
clotheswasher_dryg1	oven1
diningroom1	oven2
diningroom2	pool1
dishwasher1	pool2
disposal1	poollight1
drye1	poolpump1
dryg1	pump1
freezer1	range1
furnace1	refrigerator1
furnace2	refrigerator2
garage1	security1
garage2	shed1
gen	sprinkler1
grid	utilityroom1
heater1	venthood1
housefan1	waterheater1
icemaker1	waterheater2
	winecooler1

Bibliography

[1] NIST Transactive Energy Challenge. `https://pages.nist.gov/TEChallenge/`.

[2] Christopher Knittel. Lecture notes in energy economics, 2016.

[3] Energy policy act of 1992, 1992.

[4] Timothy P Duane. Regulation's rationale: learning from the california energy crisis. *Yale J. on Reg.*, 19:471, 2002.

[5] Marija D Ilic. Dynamic monitoring and decision systems for enabling sustainable energy services. *Proceedings of the IEEE*, 99(1):58–79, 2011.

[6] Marija Ilic and Rupamathi Jaddivada. Transactive energy challenge: Nist phase ii report. Technical report, Massachusetts Institute of Technology, 2018.

[7] Muhammad Kumail Haider, Asad Khalid Ismail, and Ihsan Ayyub Qazi. Markovian models for electrical load prediction in smart buildings.

[8] Le Xie, Yingzhong Gu, Xinxin Zhu, and Marc G Genton. Short-term spatio-temporal wind power forecast in robust look-ahead power system dispatch. *IEEE Transactions on Smart Grid*, 5(1):511–520, 2014.

[9] Riccardo Bonetto and Michele Rossi. Machine learning approaches to energy consumption forecasting in households. 2017.

[10] Ping-Huan Kuo and Chiou-Jye Huang. An electricity price forecasting model by hybrid structured deep neural networks. *Sustainability*, 10(4):1280, 2018.

[11] Seyedeh Fallah, Ravinesh Deo, Mohammad Shojafar, Mauro Conti, and Shahaboddin Shamshirband. Computational intelligence approaches for energy load forecasting in smart energy management grids: state of the art, future challenges, and research directions. *Energies*, 11(3):596, 2018.

[12] Christopher Alan Smith. *The Pecan Street Project: developing the electric utility system of the future*. PhD thesis, Citeseer, 2009.

[13] Pedro MS Carvalho, Luís AFM Ferreira, Alexandre MF Dias, et al. Distribution grids of the future: Planning for flexibility to operate under growing uncertainty. *Foundations and Trends® in Electric Energy Systems*, 2(4):324–415, 2018.

[14] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[15] Gunther Dütsch and Neon Steinecke. Use cases for blockchain technology in energy and commodity trading. *Snapshot of current developments of blockchain in the energy and commidity sector. pwc*, 2017.

[16] Ashley Pilipiszyn and David P Chassin. Crypto-control: Why transactive control needs blockchain.". 2018.

[17] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[18] Nick Johnstone, Ivan Haščič, and David Popp. Renewable energy policies and technological innovation: evidence based on patent counts. *Environmental and resource economics*, 45(1):133–155, 2010.

[19] elab summit: blockchain and transactive energy.

[20] Solarcoin. `https://solarcoin.org/`.

[21] Swytch. `https://swytch.io/`.

[22] Grid+. `https://gridplus.io/`.

[23] Consensys. `https://consensys.net/`.

[24] Drift. `https://www.joindrift.com/`.

[25] Lo3. `https://lo3energy.com/`.

[26] Conjoule. `http://conjoule.de/de`.

[27] Verv. `https://vlux.io/us/`.

[28] Proof of work. `https://en.bitcoin.it/wiki/Proof_of_work`.

[29] Bitcoin energy consumption index. `https://digiconomist.net/bitcoin-energy-consumption`.

[30] Jonas Bentke. Proof of authority - ewf - energy web foundation, 2018.

[31] Proof of stake. `https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs`.

[32] Energy web foundation. `https://energyweb.org/`.

[33] Marija Ilić, Rupamathi Jaddivada, and Xia Miao. Modeling and analysis methods for assessing stability of microgrids. *IFAC-PapersOnLine*, 50(1):5448–5455, 2017.

[34] Marija D Ilić and Rupamathi Jaddivada. Multi-layered interactive energy space modeling for near-optimal electrification of terrestrial, shipboard and aircraft systems. *Annual Reviews in Control*, 2018.

[35] Marija Ilic and Rupamathi Jaddivada. New energy space modeling for optimization and control in electric energy systems. *Optimization and Engineering*, 2019, (*Under Review*).

[36] Mikhail A Lisovich, Deirdre K Mulligan, and Stephen B Wicker. Inferring personal information from demand-response systems. *IEEE Security & Privacy*, 8(1), 2010.

[37] Carl E Landwehr and Alfonso Valdes. Building code for power system software security. *Technical Report. IEEE Computer Society*, 2017.

[38] Frances M Cleveland. Cyber security issues for advanced metering infrasttructure (ami). In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*, pages 1–5. IEEE, 2008.

[39] David Grochocki, Jun Ho Huh, Robin Berthier, Rakesh Bobba, William H Sanders, Alvaro A Cárdenas, and Jorjeta G Jetcheva. Ami threats, intrusion detection requirements and deployment recommendations. In *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012.

[40] Atonomi. `https://atonomi.io/`.

[41] Nian Liu, Jinshan Chen, Lin Zhu, Jianhua Zhang, and Yanling He. A key management scheme for secure communications of advanced metering infrastructure in smart grid. *IEEE Transactions on Industrial electronics*, 60(10):4746–4756, 2013.

[42] Jia-Lun Tsai and Nai-Wei Lo. Secure anonymous key distribution scheme for smart grid. *IEEE transactions on smart grid*, 7(2):906–914, 2016.

[43] Jinyue Xia and Yongge Wang. Secure key distribution for the smart grid. *IEEE Transactions on Smart Grid*, 3(3):1437–1443, 2012.

[44] Anish Jindal, Amit Dua, Kuljeet Kaur, Mukesh Singh, Neeraj Kumar, and S Mishra. Decision tree and svm-based data analytics for theft detection in smart grid. *IEEE Transactions on Industrial Informatics*, 12(3):1005–1016, 2016.

[45] Paria Jokar, Nasim Arianpoo, Victor CM Leung, et al. Electricity theft detection in ami using customers' consumption patterns. *IEEE Trans. Smart Grid*, 2016.

[46] Yichi Zhang, Lingfeng Wang, Weiqing Sun, Robert C Green II, and Mansoor Alam. Distributed intrusion detection system in a multi-layer network architecture of smart grids. *IEEE Trans. Smart Grid*, 2(4):796–808, 2011.

[47] Zubair Md Fadlullah, Mostafa M Fouda, Nei Kato, Xuemin Shen, and Yousuke Nozaki. An early warning system against malicious activities for smart grid communications. *IEEE Network*, 25(5), 2011.

[48] Zhifeng Xiao, Yang Xiao, and David Hung-Chang Du. Exploring malicious meter inspection in neighborhood area smart grids. *IEEE Trans. Smart Grid*, 2013.

[49] Marija Ilic, Le Xie, and Qixing Liu. *Engineering IT-enabled sustainable electricity services: The tale of two low-cost green azores islands*, volume 30. Springer Science & Business Media, 2013.