An IoT Solution for Optimal Energy Storage Operation

Julian de Hoog
IBM Research
Australia

Bhagya Amarasekara
The University of Melbourne
Australia

Fatemeh Jalali
IBM Research
Australia

Abstract

Greater availability and affordability of distributed energy generation and storage has made home energy management an increasingly complex task. To get the most value out of the system as a whole, forecasting of generation and demand (and sometimes price) is becoming more important. At the same time, cloud-based computational resources, data sources, and weather forecasts are now easier to access and apply to home energy management. This paper presents a testbed that integrates a number of different components within a single solution towards solving a single goal: minimising the cost of home energy consumption. Machine learning-based forecasts are fed into a dynamic programming approach to optimal energy storage scheduling, leading to increased savings for the customer.

1 Introduction

Projections regarding the expected growth of the Internet-of-Things (IoT) vary widely, but the consensus across many industry analysts is that there will be tens of billions of devices interconnected by 2020 [Nordrum, 2016]. These will provide benefits across a wide range of sectors, including home energy monitoring and management. Indeed, a wide range of IoT devices has already been applied towards this purpose for many years, with home energy monitor use widespread. A large number of products are now available that aim to help homeowners understand their energy consumption, predict and identify individual loads, control these in centralised or distributed ways, and provide cloud-based interactivity for better management of the system as a whole [Beaudin and Zareipour, 2015; Lobaccaro *et al.*, 2016].

The home energy management scenario has become more interesting in recent years with the increasing availability and affordability of distributed energy resources, such as rooftop solar photovoltaic (PV) generation, and even the prospect of widespread distributed home energy storage. In Australia, for example, more than 20% of homes now have solar PV systems, and recent projections suggest that proliferation of home energy storage will follow a similar steep uptake [Jacobs, 2016]. This introduces significant *flexibility* and *controllability* in the management of home energy use.

However, challenges remain. The optimal scheduling of distributed energy resources is highly dependent on the availability and accuracy of *forecasts* – of demand, generation, and even possibly price. An intuitive example of this is provided by a typical homeowner with a solar PV system and an energy storage system, paying a two-part (peak and off-peak) tariff. On a sunny day, this homeowner would want their energy storage system to be empty in the morning so that its full capacity can be used to store excess solar energy generated throughout the day. On a cloudy day, this homeowner would want their energy storage system to by full in the morning, in order to have taken advantage of low energy prices overnight. Good forecasts are therefore essential to get the maximum possible value out of the system as a whole and must be integrated into system control in an ongoing and iterative manner.

An additional challenge is posed by the availability of computational resources. For some energy management problems, the optimal solutions, possibly calculated over lengthy future horizons, may be fairly complex to solve. A promising approach is to offload intensive computation to external (*e.g.* cloud-based) resources, and then apply the solutions to the local system comprised of small devices with less compute capacity. This, however, brings with its connectivity and communication vulnerabilities: solutions computed in the cloud are only useful if they can be returned to the querying device without delays or dropouts. In addition, some applications may require very fast closed-loop feedback control to account for changing conditions (such as the passing of clouds over solar panels, which can change available generation in a matter of seconds). When centrally computed solutions may take many seconds or minutes to complete, the changes in necessary behaviour locally may not happen fast enough. It is, therefore, necessary to investigate in greater detail the opportunities for applying methods from the artificial intelligence and machine learning communities to the control of distributed home energy management.

In this work, we introduce an integrated IoT testbed that uses both local hardware and control, as well as cloud-based forecasts and control recommendations, to optimally control a small energy storage system. We describe the hardware setup, which includes a small solar panel, battery, and Raspberry Pi for system control, and explain the communication framework that enables interaction both between local com-

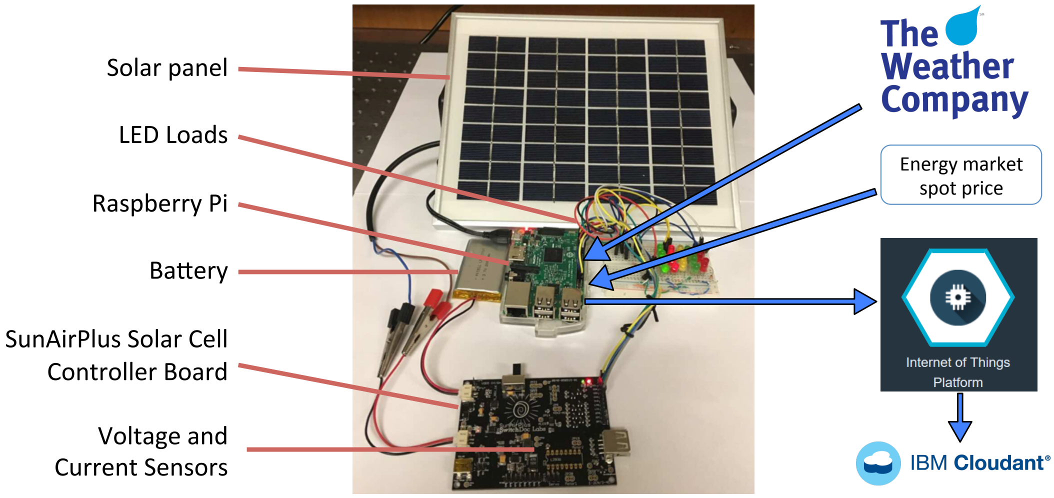

Figure 1: A testbed for home energy management via an IoT solution

ponents and with external data sources. We introduce the (partly machine learning based) approaches to forecasting of solar generation and electricity price, and describe how a dynamic programming approach can be used for optimal control of the battery. Finally, we present the results of a small study demonstrating the successful interaction between all of the above components in a manner that reduces the energy costs of the system as a whole.

2 System Architecture and Solution

Managing home energy use via an IoT architecture requires three essential elements. The first one is an appropriate set of sensors to measure the power consumption of the home (including perhaps individual appliances), the state of charge of the energy storage system, and any available distributed generation (such as solar PV). The second one is the set of controlling devices which can make control decisions to achieve objectives such as minimising cost. The third one is the communications network that allows the different components to communicate and interact. In this paper, we present a design of such an architecture with a simple test bed that closely replicates the real life problem of minimising the cost of energy in a home environment.

Our test bed is presented in Figure 1, and consists of:

- a 5W solar panel
- five independently controlled LED arrays
- a Raspberry Pi
- a 2000mAh, 3.7V rechargeable LiPo battery
- a SunAirPlus Solar Cell controller board

The LED arrays are used to represent system loads.

Full system control is conducted using the Raspberry Pi, which is a powerful, yet low-cost device. Through its general purpose input output (GPIO) pins, we control the number of LED arrays to be on, and the source of the supply for these loads, i.e. the solar panels, the battery, or mains power. The Raspberry Pi further interacts with external services and data sources.

The SunAirPlus Solar Cell controller board takes care of maximum power point tracking (MPPT) for the solar panel and contains several sets of voltage and current sensors that measure the available solar generation, battery power, and load power, respectively. When there is excess solar generation (i.e., more generation than demand), this is used to charge the battery. The controller board also makes it possible to start and stop battery discharge, as required.

For communication, we have used the I^2C two wire protocol for connecting the sensors to the Raspberry Pi, and WiFi for connecting the Raspberry Pi to the cloud-based IBM Watson IoT™ platform. Connecting the Raspberry Pi to Watson IoT offers several advantages: it enables straightforward connection to a large number of additional services, such as an IBM Cloudant® database, weather data and forecasts, and visualisation and control tools such as NodeRed. The connection to the Watson IoT platform uses the MQTT protocol, which is lightweight and robust. We have used the Python programming language to implement these key connections and for taking intelligent control decisions. All sensor data is logged both locally, and remotely (in the Cloudant database).

3 Forecasting

Forecasting (of generation, demand, electricity price, etc.) is receiving increasing attention in the energy community in recent years, due to greater availability of fine-grained data, and due to an increased need to control distributed systems more intelligently. This section describes the various forecasts made available to our IoT solution.

3.1 Price forecasting via support vector machines

In most geographies, small scale energy consumers are typically not yet exposed to real-time fluctuations in energy prices, and many people have either a fixed or a static tiered tariff structure. However, real-time pricing is receiving in-

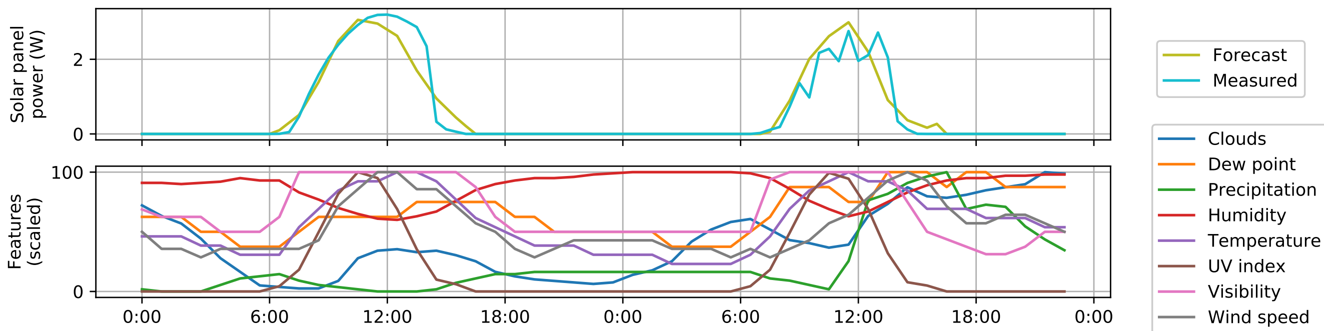

Figure 2: Example of a 48-hour solar forecast. Features (bottom) are provided by API calls to The Weather Company. These are fed into a multiple linear regression based model, trained on historical data, to provide the solar forecast (top). The shown forecast was made at 0:00 of the first day, while measured data represents actual generation recorded after the forecast was made.

creasing attention as a possible solution to distribute the costs of energy infrastructure to end users in a way that is more fair (i.e., that more accurately reflects individual users' contributions to network costs). In this study when we refer to "price" we are using the wholesale energy price as determined by the spot market in the National Electricity Market (NEM) of Australia. While end users would ultimately pay a higher price, we consider this to be a relevant proxy for the purposes of this study. The accuracy of price forecasts will depend significantly on the nature of the underlying energy market. In many cases, such forecasts can be challenging as they can be highly affected by singular events such as the outage of a plant. A good review of electricity price forecasting is available in [Weron, 2014].

In Australia, the national electricity market is controlled by the Australian Energy Market Operator (AEMO), which publishes demand and spot price data at 30-minute intervals for five states. We have used the historical data sets of load, price and temperature, and selected features that improve the forecast model accuracy. The data set is trained using support vector machines [Gao *et al.*, 2007], and the trained model provided price forecast values with 9.1% mean absolute percentage error (MAPE).

3.2 Solar generation forecasting using available weather data

Forecasting of solar generation over short horizons (such as the next 12 hours) has its own challenges but is typically less prone to singular events in the way that prices are. Data-driven forecasting techniques typically do not make that much sense over short horizons; in such cases, meteorological approaches are generally more relevant. A good review of solar irradiance forecasting techniques is available in [Diagne *et al.*, 2013], and an interesting review of specifically machine learning techniques applied to solar forecasting is available in [Voyant *et al.*, 2017].

Our solar generation forecasts use data provided by API access to The Weather Company®, one of the world's largest weather data gathering and forecasting organisations. Localised forecast data are retrieved using queries specific to a given latitude and longitude. Datasets that are available for use in our forecasting models include cloud cover, dew point, precipitation, humidity, temperature, UV index, visibility, and wind speed, among others. Since the existing API does not directly forecast solar irradiation, we developed for the purposes of this prototype a multiple linear regression model, trained using past forecasts and actual solar output measurements. The model can be repeatedly retrained online using the past month of available data.

Figure 2 shows an example of a 48-hour solar forecast. The bottom plot shows the features used for the forecast (all of which are provided by The Weather Company using their internal forecasting mechanisms). These are fed into the multiple linear regression model, which provides as an output the solar forecast shown in the top figure. The true (measured) panel generation, measured after the forecast was made, shows the reasonably good performance of the solar forecast.

It should be noted that Figure 2 shows a 48-hour lookahead. In actual operation, the dynamic programming solution (described in Section 4) is applied in a receding horizon control style – meaning that the full horizon is repeatedly forecasted, and only the first interval of the solution is applied. As a result, forecasts for the intervals in the near future become the most important, and these will also be the most accurate. In other words, the real-time operation will use even more accurate forecasts than shown here.

The resulting solar forecast is made available via a cloud-based API solution that is available not only to the particular application described in this paper, but to any further use cases for which localised solar generation forecasts may be of value. One key advantage of this approach is that the model is trained on the actual output data of a specific location, and thereby inherently takes into account any location-specific factors such as time-dependent shading, angle of panels with respect to sun, etc.

3.3 Demand forecasting

For individual homes, demand forecasting is typically challenging. Trying to determine when a homeowner may turn on a particular appliance is usually inherently difficult. However, for many residential customers there are regular patterns – daily, weekly, seasonal, etc., – that can be leveraged using data-driven forecasting. Just as for solar forecasting, in many cases, it is sufficient to have a good forecast for the *aggregate* demand over longer time intervals (such as several hours), in which case the forecasting problem becomes easier. In recent years, the energy community has also increasingly considered application of probabilistic [Hong and Fan, 2016] and hierarchical forecasting [Hyndman *et al.*, 2011].

While weather (especially temperature) does have an impact on residential demand forecasts, to date we have used primarily data-driven time-series forecasts only (which in themselves can take weather into account indirectly). Figure 3 shows an example of a demand forecast for a real residential demand profile. When forecasting 24 hours ahead (or more), there can often be significant errors, particularly when the home in question does not exhibit highly regular daily load properties. In this case, where the forecast uses similar days in recent history to estimate future load, high demand in the evening peak is expected but is not ultimately exhibited on this particular day.

However, when the demand forecast is updated in a rolling manner – i.e. at each interval a new forecast is made for the next (24h) horizon, and the first forecast value is used, the forecasts become much more accurate (Figure 4). This again underscores the value of a receding horizon controller approach, which we propose in the next section.

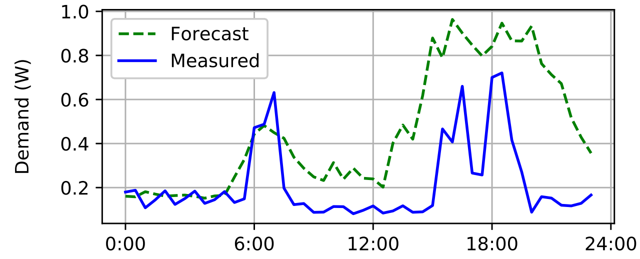

Figure 3: Example of a "one-shot" 24-hour demand forecast. The shown forecast was made at 0:00 (and not updated throughout the 24-hour horizon shown), while measured data represents actual demand recorded after the forecast was made.

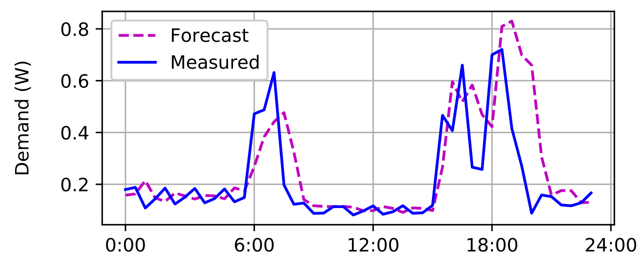

Figure 4: Example of a regularly updated 24-hour demand forecast. The shown forecast represents what is actually used by a receding-horizon controller. The full 24-hour demand forecast is calculated repeatedly in each interval, and only the first forecast value (one interval lookahead) is applied.

4 Dynamic Program

Given forecasts for expected solar generation, energy demand, and electricity price, the next question that arises is how to best leverage these to determine the optimal operational profile for the battery. This is an increasingly well studied problem, with existing approaches using *e.g.* different types of dynamic programming [Codemo *et al.*, 2013; Abdulla *et al.*, 2016; Kamyar and Peet, 2016], linear programming [Hoke *et al.*, 2013], quadratic programming [Ratnam *et al.*, 2015], or mixed integer programming [Khani and Zadeh, 2015].

In this work, we partially adapt the approach described in [Abdulla *et al.*, 2016], and use dynamic programming to determine the best operational profile for the energy storage system, since this provides fast solutions that are easy to adapt to different kinds of use cases and scenarios.

We assume we have forecasts for demand d_t, generation g_t, and price p_t over a full discrete future horizon, $t \in [0\ T]$. We refer to the *optimal operational profile* of the battery as its sequence of charge and discharge decisions b_t over this horizon. We further discretise the possible states of charge that the battery may be in into a set $s \in [0\ S]$. The full state space A over the full horizon, therefore, has $S \times T$ possible states.

It should be noted that discretisation of the battery's states of charge is an approximation of the state space (since state of charge is continuous). However, in practice it significantly reduces solution complexity and as long as the discretisation interval is kept small enough, still leads to near-optimal solutions (see [Abdulla *et al.*, 2016] for a more detailed discussion on this).

A key consideration in the set up of a dynamic program is the state transition cost, which can be easily adjusted to suit a particular scenario. The use case presented in the next section uses energy storage primarily for "solar self-consumption": ensuring that all solar energy, including excess energy generated at times of low demand, is used to minimise the need to import additional energy from the grid. We therefore do not charge the battery from the grid at any time – it is only charged by excess solar generation. In this case the state transition cost C from a state $A_{s_i,t}$ to a state $A_{s_j,t+1}$ is defined as:

$$C(A_{s_i,t}, A_{s_j,t+1}) = \begin{cases} 0, & \text{if } g_t > d_t \\ (s_j - s_i) \times p_t, & \text{otherwise} \end{cases}$$

The dynamic program is solved using backward iteration from the final interval T. All final states $A_{s_i,T}$ are initialised to zero. The only exception is if there is a preferred final state of charge s_x, in which case the particular state $A_{s_x,T}$ can be initialised to a very low number (meaning that the solution will finish in this state).

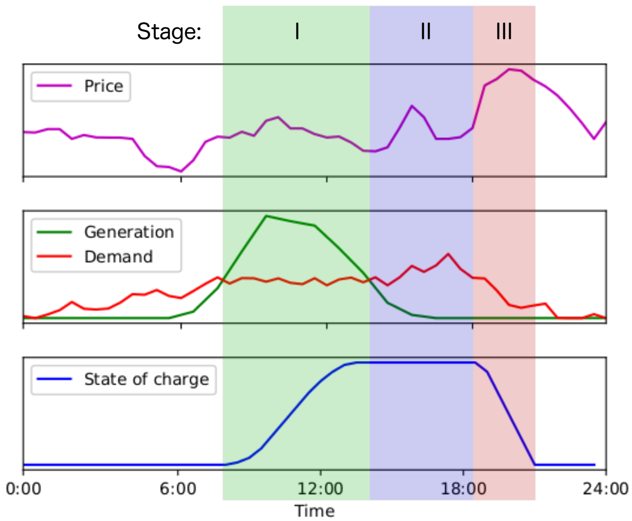

Figure 5: A demonstration of the dynamic programming solution. In Stage I, excess solar generation is used to charge the battery. In Stage II, the battery is fully charged and waits for the highest possible price to discharge. This occurs in Stage III, where discharging the battery avoids the high cost of importing energy from the grid at this time.

We determine the "cost to go" CTG for any possible state $A_{s,t}$ as the sum of the state transitions that get us there having the lowest joint total cost, using the recursive relationship:

$$CTG(A_{s_i,t}) = \min_{b_t}\{C(A_{s_i,t}, A_{s_j,t+1}) + CTG(A_{s_j,t+1})\} \tag{1}$$

The outcome is an optimal operational profile for our battery that minimises the total cost over the full horizon. We apply this using receding horizon control: the first decision b_0 is applied, and then the full operational profile is recalculated for the full horizon. Again the new first decision b_0 is applied, and so on. In this manner, updated information and improved forecasts are constantly being integrated into the ongoing operation of the battery.

Figure 5 illustrates the performance of the dynamic program. This figure shows typical price, generation, and demand profiles, based on real data collected as part of this study. The optimal operational profile for the battery is shown at the bottom. From about 8:00 - 14:00, generation exceeds demand, and the battery is charged. From ∼14:00 onwards, the best strategy for the battery is to discharge when the price is highest (to avoid the cost of importing energy from the grid). Even though there is a brief increase in price at ∼16:00, the dynamic program correctly decides to wait with discharging until ∼19:00, when the price is considerably higher still.

For the use case of an end user aiming to minimise their energy costs (as is presented in Section 5, it is typically sufficient to forecast and optimise over a 24-hour horizon, since this is the interval over which strategic decisions must be taken. Other use cases may require longer or shorter horizons. However, even for longer horizons, a simple dynamic programming approach is likely to scale reasonably well. Even on a Raspberry Pi, a one-week solution can be solved in a matter of seconds. The size of the problem does depend on the choice of discretisation interval of course, and higher levels of discretisation will lead to longer solving times.

5 Case Study

To demonstrate the system presented in Section 2, using the forecasts described in Section 3, and applying the dynamic program from Section 4, we now present the outcomes of a small case study conducted over the course of several days in May-June 2017. The solar panel and battery IoT solution was set up to continuously forecast price, generation, and demand, and to use the dynamic programming solution above to optimally schedule battery charge and discharge.

Figure 6 presents data collected from the set up over one day from 18.30 pm onwards for a 24 hour period. The figure shows the price per kWh, solar generation, and demand profiles, as well as the battery's state of charge (SOC). This particular 24-hour period illustrates a number of key behaviours. From 18:30 to 07:00, demand exceeds generation and the battery is used to meet this demand (rather than importing energy from the grid). After 07:00, generation exceeds demand and the battery uses any excess generation to charge. From about 17:30 onwards, the battery is fully charged. However, it does not discharge until there is a (forecasted) spike in the electricity price at 18:00.

On most systems installed today, such charging decisions are made in the inverter or charge controller connected to the energy storage system. Real-time, rolling forecasting and optimisation are typically not available, and most existing systems use some form of set-point control to manage the trade-offs between multiple value streams. In a recent study using real consumer data, we found that optimised charge control can offer significant benefits over set-point control – in some cases providing greater than double the returned value over the lifetime of the system [Abdulla *et al.*, 2016].

6 Conclusions

This work integrated a number of different components into a single, unified system that aims to optimally schedule distributed energy resources to minimise energy costs for the end user. The components include: an underlying hardware testbed, that includes distributed generation and energy storage; a control system, embedded in a Raspberry Pi, that interacts with local components and external components and services; the external data sources and services that enable forecasts of generation and price, and a dynamic programming solution to determine an optimal operational profile. Designing systems with so many interacting components is an ongoing challenge, and we hope that this testbed contributes to the ongoing discussions around best practice in the design and operation of such systems.

In future work, we intend to continually improve the various forecasts, deploy the next version of this solution at a number of real residential test sites, and explore the interesting questions around how to optimally trade off the costs and benefits of local versus cloud-based computation.

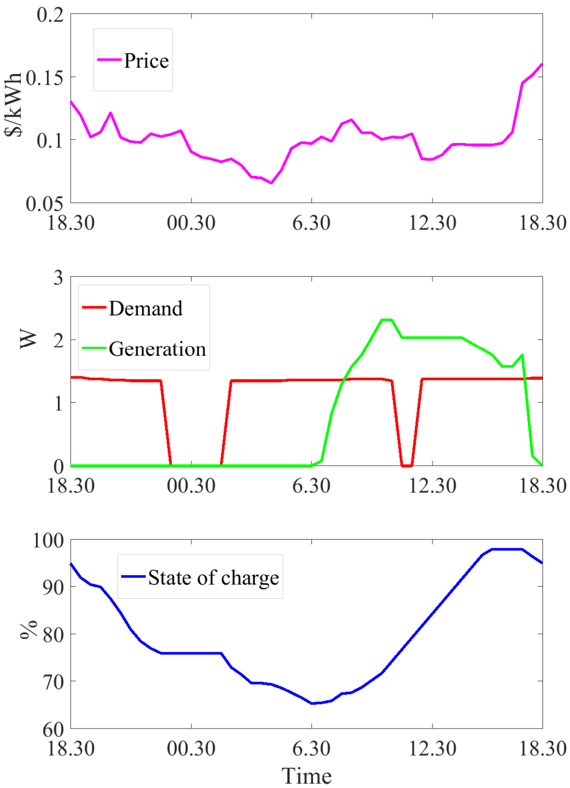

Figure 6: The data collected from a small scale set up with forecasting and dynamic programming methods implemented.

References

[Abdulla *et al.*, 2016] K. Abdulla, J. De Hoog, V. Muenzel, F. Suits, K. Steer, A. Wirth, and S. Halgamuge. Optimal operation of energy storage systems considering forecasts and battery degradation. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2016.

[Beaudin and Zareipour, 2015] Marc Beaudin and Hamidreza Zareipour. Home energy management systems: A review of modelling and complexity. *Renewable and Sustainable Energy Reviews*, 45:318 – 335, 2015.

[Codemo *et al.*, 2013] C. G. Codemo, T. Erseghe, and A. Zanella. Energy storage optimization strategies for smart grids. In *2013 IEEE International Conference on Communications (ICC)*, pages 4089–4093, June 2013.

[Diagne *et al.*, 2013] Maimouna Diagne, Mathieu David, Philippe Lauret, John Boland, and Nicolas Schmutz. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. *Renewable and Sustainable Energy Reviews*, 27:65 – 76, 2013.

[Gao *et al.*, 2007] Ciwei Gao, Ettore Bompard, Roberto Napoli, and Haozhong Cheng. Price forecast in the competitive electricity market by support vector machine. *Physica A: Statistical Mechanics and its Applications*, 382(1):98 – 113, 2007.

[Hoke *et al.*, 2013] A. Hoke, A. Brissette, S. Chandler, A. Pratt, and D. Maksimovizc. Look-ahead economic dispatch of microgrids with energy storage, using linear programming. In *Technologies for Sustainability (SusTech), 2013 1st IEEE Conference on*, pages 154–161, Aug 2013.

[Hong and Fan, 2016] Tao Hong and Shu Fan. Probabilistic electric load forecasting: A tutorial review. *International Journal of Forecasting*, 32(3):914 – 938, 2016.

[Hyndman *et al.*, 2011] Rob J. Hyndman, Roman A. Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational Statistics & Data Analysis*, 55(9):2579 – 2589, 2011.

[Jacobs, 2016] Jacobs. Projections of uptake of small-scale systems, June 2016. Prepared for the Australian Energy Market Operator. Available online, accessed 4 May 2017.

[Kamyar and Peet, 2016] R. Kamyar and M. M. Peet. Multi-objective dynamic programming for constrained optimization of non-separable objective functions with application in energy storage. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 5348–5353, Dec 2016.

[Khani and Zadeh, 2015] H. Khani and M.R.D. Zadeh. Online adaptive real-time optimal dispatch of privately owned energy storage systems using public-domain electricity market prices. *Power Systems, IEEE Transactions on*, 30(2):930–938, March 2015.

[Lobaccaro *et al.*, 2016] G. Lobaccaro, S. Carlucci, and E Lfstrm. A review of systems and technologies for smart homes and smart grids. *Energies*, 9(5), 2016.

[Nordrum, 2016] Amy Nordrum. Popular internet of things forecast of 50 billion devices by 2020 is outdated. http://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated, August 2016. [Online; accessed 4 May 2017].

[Ratnam *et al.*, 2015] Elizabeth L. Ratnam, Steven R. Weller, and Christopher M. Kellett. An optimization-based approach to scheduling residential battery storage with solar PV: Assessing customer benefit. *Renewable Energy*, 75:123 – 134, 2015.

[Voyant *et al.*, 2017] Cyril Voyant, Gilles Notton, Soteris Kalogirou, Marie-Laure Nivet, Christophe Paoli, Fabrice Motte, and Alexis Fouilloy. Machine learning methods for solar radiation forecasting: A review. *Renewable Energy*, 105:569 – 582, 2017.

[Weron, 2014] Rafa Weron. Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30(4):1030 – 1081, 2014.