October 2018

Hybrid Black-box Solar Analytics and their Privacy Implications

Dong Chen
University of Massachusetts Amherst

Recommended Citation

Chen, Dong, "Hybrid Black-box Solar Analytics and their Privacy Implications" (2018). *Doctoral Dissertations*. 1331.
https://scholarworks.umass.edu/dissertations_2/1331

HYBRID BLACK-BOX SOLAR ANALYTICS AND THEIR PRIVACY IMPLICATIONS

A Dissertation Presented

by

DONG CHEN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

September 2018

Electrical and Computer Engineering

HYBRID BLACK-BOX SOLAR ANALYTICS AND THEIR PRIVACY IMPLICATIONS

A Dissertation Presented

by

DONG CHEN

Approved as to style and content by:

David Irwin, Chair

Prashant Shenoy, Member

Michael Zink, Member

Jay Taneja, Member

Christopher Hollot, Department Chair
Electrical and Computer Engineering

ACKNOWLEDGMENTS

I would not be able to complete this dissertation without the guidance of my advisor and dissertation committee members, and the support of my friends and family.

First, I would like to thank my advisor Professor David Irwin for his guidance and constant support during my Ph.D. study at UMass Amherst. He is always patient and willing to assist me with my courses and research. I could not have been as I am right now without his help every step of the way. Working with him, I learned how to find a research problem, solve it and present it to other people. Ph. D. is full of ups and downs, and he always encouraged me to be confident and enthusiastic about my research. Over the years, he helped me transit from a software engineer to an experimental system researcher to solve real problems using real data.

Second, I would sincerely express my thanks to my (unofficial) co-advisor Professor Prashant Shenoy. I visited his laboratory (LASS) in Summer 2012 as a student researcher. Prashant pointed me to the research area of energy data analytics and their privacy implications. He encouraged me to pursue and finish my Ph. D. timely. His invaluable advice and comments helped me to develop my research skills and make me ready for a new journey.

Third, I would like to thank my other dissertation committee members, Professor Michael Zink and Professor Jay Tanja. They have given me many valuable inputs on my research projects and dissertation. I also would like to thank my colleagues Sean Baker, Sandeep Karla, and Srini Iyengar. I am also profoundly grateful to my friends and labmates: Stephen Lee, Supreeth Shastri, Pradeep Ambati, Xue Ouyang, Noman Bashir, and Akansha Singh, who have stayed with me and discussed many research topics in the past five years.

Lastly and most importantly, I would like to thank my parents who taught me what I am. My wife Haiyan has been accompanying me since we were in graduate school in China. I am so happy for having two beautiful daughters Claire and Chloe in my harmonious family life. Their love and accompanying have been the source of my strength. I dedicate this dissertation to them.

This dissertation is supported by NSF grants IIP-1534080, CNS-1645952, CNS-1405826, CNS-1253063, CNS-1505422, Army Research Lab Award W911NF-17-2-0196, and the Massachusetts Department of Energy Resources.

ABSTRACT

HYBRID BLACK-BOX SOLAR ANALYTICS AND THEIR PRIVACY IMPLICATIONS

SEPTEMBER 2018

DONG CHEN

B.Sc., XI'AN COMMUNICATIONS INSTITUTE

M.Sc., NORTHEASTERN UNIVERSITY CHINA

Ph.D., NORTHEASTERN UNIVERSITY CHINA

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor David Irwin

The aggregate solar capacity in the U.S. is rising rapidly due to continuing decreases in the cost of solar modules. For example, the installed cost per Watt (W) for residential photovoltaics (PVs) decreased by ∼6X from 2009 to 2018 (from \$8/W to \$1.2/W), resulting in the installed aggregate solar capacity increasing ∼128X from 2009 to 2018 (from 435 megawatts to 55.9 gigawatts). This increasing solar capacity is imposing operational challenges on utilities in balancing electricity's real-time supply and demand, as solar generation is more stochastic and less predictable than aggregate demand.

To address this problem, both academia and utilities have raised strong interests in solar analytics to accurately monitor, predict and react to variations in intermittent solar power. Prior solar analytics are mostly "white-box" approaches that are based on site-specific information and require expert knowledge and thus do not scale, recent research focuses on "black-box" approaches that use training data to automatically learn a custom machine learning (ML) model. Unfortunately, this approach requires months-to-years of training data, and often does not incorporate well-known physical models of solar generation, which reduces its accuracy. Instead, in this dissertation, we present a hybrid “black box” approach that can achieve the best of both to solar analytics. Our hypothesis is that the hybrid “black-box” approach can enable a wide range of accurate solar analytics, including modeling, disaggregation, and localization, with limited training data and without knowledge of key system parameters by integrating “black-box” machine learning approaches with “white-box” physical models. In evaluating our hypothesis, we make the following contributions:

(Mostly) ML “black-box” Solar Modeling. To get benefits from both of ML and physical approaches, we present a configurable hybrid “black-box” ML approach that combines well-known relationships from physical models with unknown relationships learned via ML. Rather than manually determining values for physical model parameters, our approach automatically calibrates them by finding values that best to the data. This calibration requires much less data (as few as 2 datapoints) than training an ML model. And we show that our hybrid approach significantly improves solar modeling accuracy.

(Mostly) Physical “black-box” Solar Modeling. The physical model used in the hybrid model above performs significantly worse than other approaches. To determine the primary source of this inaccuracy, we conduct a large-scale data analysis and show that the only weather metrics that affect solar output are temperature and cloud cover, and then derive a new physical model that accurately quantify cloud cover’s effect on solar generation at all sites. We then enhance our physical model with a ML model that learns each site’s unique shading effect. And we show that the hybrid modeling yields higher accuracy than current state-of-the-art ML approaches. We also identify a universal weather-solar effect that has not been articulated before and is broadly applicable to other solar analytics.

Solar Disaggregation. Solar forecast models require historical solar generation data for training. Unfortunately, pure solar generation data is often not available, as the vast majority of small-scale residential solar deployments (<10kW) are “Behind the Meter (BTM)”, such that smart meter data exposed to utilities represents only the net of a building’s solar generation and its energy consumption. To address this problem, we design SunDance, a “black-box” system that leverages the clear sky maximum solar generation model, and the universal weather-solar effect from the hybrid “black-box” models above. We show that

SunDance can accurately disaggregate solar generation from net meter data without access to a building's pure solar generation data for training.

Solar-based Localization. The energy data produced by solar-powered homes is considered "anonymous" and usually publicly available if it is not associated with identifying account information, e.g., a name and address. Our key insight is that solar energy data is not anonymous: every location on Earth has a unique solar signature, and it embeds detailed location information. We then design SunSpot to localize the source of solar generation data and show that SunSpot is able to localize a solar-powered home within ∼500 meters and ∼28 kilometers radius for per-second and per-minute resolution.

Weather-based Localization. However, the above solar-based localization has a fundamental limit due to Earth's rotation. To further localize towards a specific home, we identify another key insight: every location on Earth has a distinct weather signature that uniquely identifies it. Interestingly, we find that localizing coarse (one-hour resolution) solar data using weather signature is more accurate than localizing solar data (one minute or one second resolution) using its solar signature. Both of "SunSpot" and "Weatherman" expose a new serious privacy threat from energy data, which has not been presented in the past.

TABLE OF CONTENTS

Page

CHAPTER

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

CHAPTER 1
INTRODUCTION

The aggregate solar capacity in the U.S. is rising rapidly due to continuing decreases in the cost of solar modules. This solar penetration is placing pressure on grid operations, which balance electricity's supply and demand in real-time, since solar generation is more stochastic and less predictable than aggregate demand. To address this issue, we present a hybrid "black box" approach to solar data analytics that can help utilities accurately monitor, predict and react to variations in intermittent solar power.

1.1 Solar Penetration Is Increasing

The penetration of intermittent solar in the U.S. is rising rapidly due to continuing decreases in the cost of solar modules. For example, the installed cost per watt(w) for residential photovoltaics (PVs) decreased by ∼3X from 2009 to 2016 (from \$8/w to \$2.93/w). As a result, the return on investment for "going solar" in many locations is now less than 5 years. Therefore, the installed aggregate solar capacity increased ∼3X from 2009 to 2016 (from 435 megawatts to 14,762 megawatts).

Nearly all the solar deployments are "grid-tied", such that they feed any solar power generated into the electric grid. This increasing grid-tied solar installations is imposing operational challenges on utilities in balancing electricity's real-time supply and demand. Even when aggregated across many deployments over a larger region, solar generation is still more stochastic and less predictable than aggregate demand, since changes in dynamic cloud cover (the primary weather metric that affects aggregate solar output) are inherently more localized and stochastic than changes in temperature (the primary weather metric that affects aggregate net demand).

While the advancements in solar models (primarily forecasting) are enabling utilities to better monitor, predict and react to variations in intermittent solar power in grid. Unfortunately, pure solar generation data required by solar forecast models for training is often not available, as over 60% of the solar capacity growth is from the small-scale rooftop deployments (<10 kW) that are "behind the meter (BTM)", such that the smart meter data exposed to utilities represents only the net of a building's solar generation and its energy consumption. In order to address these issues, both academia and utilities have a strong interest in solar data analytics that are useful for solar energy modeling and predicting.

1.2 Black-box Solar Analyzing

In this dissertation, we propose a "black-box" approach to solar data analytics to address the above issues. Most recent solar analytics works focus on either Machine Learning (ML) techniques or physical modeling approaches.

"Black-box" ML-based approaches. The ML approaches are often "off the shelf" and do not leverage well-known physical models of solar generation based on fundamental physical properties. Furthermore, most of the recent ML-based analyzing techniques are not real "black-box" approaches, as they require a significant amount (months-to-years) of training data from the solar site under test to build a reasonable accurate model for solar data analytics.

"White-box" physical modeling approaches. In contrast, prior physical modeling works are "white-box" approaches that require detailed information (e.g. tilt, orientation, size, efficiency, nominal operating cell temperature) from a deployment, and some model inputs (e.g. dust build up, air velocity) that rare difficult to accurately measure.

Instead, we present a new "black-box" solar analytics approach that does not require any detailed PVs deployment knowledge, or any training data from a solar-powered building itself. In essence, to archive the best from both of ML-based techniques and "White box" approaches, our "black-box" approach allow users to use physical models for selected parameters (where physical models are available), and uses ML for the other parameters (where physical models are unavailable).

1.3 Thesis Contributions

In this thesis, by evaluating our “black-box” approach, we develop multiple different solar analytics and evaluate their accuracy.

1.3.1 (Mostly) ML Black-box Solar Performance Modeling

Solar performance modeling is prerequisite for a variety of solar data analytics, including solar monitoring, BTM solar disaggregation, anonymous localization, and fault detection. Significant recent works focus on ML-based performance modeling and physical modeling. ML-based modeling requires a significant amount of pure solar generation data and weather condition data for training to build a reasonable accurate model. While, physical performance modeling requires much less data to calibrate (aka to training) than ML modeling, as physical models already embedded detailed knowledge (e.g., orientation, size, tilt, efficiency) about a deployment. However, physical models do not always exist for all the factors that affect solar generation. Therefore, we need a new model that can combine benefits from both worlds, and improve the modeling accuracy.

To address this problem, we first investigate on these existing solar performance modeling works-ML-based modeling and physical modeling, compare the accuracy and the amount of data required for calibration or training for them. We then present a configurable hybrid “black-box” approach that combines the benefits of both world. Our hypothesis is this hybrid approach can achieve the best of both, as it can combine well-known relationships from physical models with unknown relationships learned via ML to improve accuracy.

1.3.2 (Mostly) Physical Black-box Solar Performance Modeling

However, the black-box physical model used in the model above is highly inaccurate and performs significantly worse than black-box ML models. This inaccuracy must derive from either the physical models above being inaccurate, or from the effect of unmodeled physical parameters, such as the other weather metrics, shading from surrounding buildings, or soiling from dust and pollen.

To address these problems, we conduct a large-scale data analysis to determine the primary source of the inaccuracy by isolating the effects of 10 different weather metrics on solar output from nearly 343 million hourly weather and solar readings, or 78,435 aggregate years, gathered from 11,205 solar sites. We show that our empirical physical model accurately describes weather's effect on solar output at all sites, obviating the need for training custom ML models using weather metrics. Instead, we augment our physical model by applying ML to learn only the relationships that are unique to each site, primarily non-weather-based shading. We evaluate our approach on solar and weather data from 100 sites, and show it yields higher accuracy than current state-of-the-art ML approaches.

1.3.3 Solar Disaggregation

Solar performance modelings are enabling utilities to monitor and predict the solar variance in the grid. However, these solar models (especially for forecasting) requires pure solar generation data for training. Unfortunately, these solar generation data are often not available, as the vast majority of the grid-tied solar deployments are "behind the meter (BTM),", such that the utilities can only access to net meter data that represents the sum of each building's solar generation and its energy consumption.

To address this problem, we design SunDance, a "black box" system for accurately disaggregate solar generation from net meter data without access to a building's pure solar generation data for training. It only requires a building's location and as few as two datapoints of historical net meter data. It leverages clear sky maximum solar generation modeling, and identifies an important insight: the Universal Weather-Solar Effect effect, that is, the exact same weather conditions should have the same effect on the maximum solar irradiance potential, regardless of its location and time. To the best of our knowledge, it has not been articulated in the past and is broadly applicable to other solar analytics.

1.3.4 Solar-based Localization

The energy produced by solar-powered homes is monitored by utilities and third-parties using networked energy meters, which record and transmit energy at fine-grained intervals. Such energy data is considered "anonymous" if it is not associated with identifying information, e.g. account number, address. More importantly, according to the U. S. Department of Energy's recently released Voluntary Code of Conduct (VCC), these data can be shared online or even made publicly available without user's consent.

Our key insight is solar energy data is not anonymous and can be localized, since every location on Earth has a unique solar signature (including sunrise, solar noon, and sunset times), and it embeds detailed location information. To localize the solar-powered homes, we first examine the factors that affect the accuracy of solar signature for a given location on Earth, we then designe a localizing technique-SunSpot leveraging two binary searchings to find latitude and longitude for a solar site. We find that SunSpot is able to localize a solar-powered home to small region of interest that is near the smallest possible area give the energy data resolution, e.g., within ∼500 meters and ∼28 kilometers radius for per-second and per-minute resolution. Then, SunSpot identifies solar-powered homes with in this region using crowd-sourced image processing of satellite image data before applying additional filters to identify a specific home. We argue that solar generation data is not anonymous, and SunSpot exposes a new serious privacy threat from energy data, which has not been discussed before.

1.3.5 Weather-based Localization

While, SunSpot work only examines solar generation data, we next look at other energy data, e.g. wind generation, energy consumption. We first compare the relationship between energy data (e.g. wind, solar, energy consumption) and weather condition data to examine whether they actually correlates with each other. Our key insight is: every location on Earth also has a distinct weather signature that uniquely identifies it, as energy consumption, wind, and solar largely correlates with weather metrics, e.g. temperature, wind speed, and cloud cover, respectively.

To localize the source of energy meter data, we design Weatherman, which leverages a suite of big data analytics techniques. We show that Weatherman localizes coarse energy consumption, wind, and solar data to within 16.68 kilometers, 9.84 kilometers, and 5.12 kilometers, respectively. Interestingly, we find that localizing coarse (one-hour resolution) energy data using weather signature is more accurate than localizing solar data (one minute/second resolution) using solar signature. Thus, Weatherman presents not only a serious privacy threat, but also a potential useful tool for researchers working with smart meter data. The location information is highly useful and high sensitive, as it can provide important contextual information to improve big data analytics or interpret their results, but it can also enable third-parties to link private behavior derived from energy data with a particular name and address.

1.4 Dissertation Overview

We organize the rest of the dissertation as follows. Chapter 2 provides the necessary background on solar data analytics. In Chapter 3, we present and evaluate our "black-box" solar performance modeling approaches. Chapter 4 describes the design, implementation, and evaluation of SunDance: "black-box" behind the meter solar disaggregation. Chapter 5 details the design and implementation of SunSpot that localizes the source of solar-powered home using solar signature. Chapter 6 presents and evaluates another solar localization technique – Weatherman, which localizes the source of energy data using weather signature. Finally, Chapter 7 concludes the completed work and future work.

CHAPTER 2
BACKGROUND

In this chapter, we provide background about current solar data analytics approaches required for various aspects of this dissertation.

2.1 Solar Is "Behind" the Meter

The aggregate solar capacity in the U.S. is rising rapidly due to continuing drops in the price of solar modules that have fallen 10% per-year on average over the past three decades. As a result, the return on investment for "going solar" in many locations is now less than five years [57]. In addition, a variety of financing options are now available that lower the barrier to installing solar systems by enabling users to avoid incurring large upfront capital expenses, e.g., by leasing their roof space or entering into a long-term power purchase agreement. Importantly, nearly all solar deployments are "grid-tied," such that they feed any solar power generated into the electric grid. Grid-tied deployments impose new operational challenges on utilities in balancing electricity's real-time supply and demand. In particular, utilities plan "dispatch" schedules for generators in advance based on predictions of future load. Unfortunately, the increasing penetration of grid-tied solar is decreasing the accuracy of net load predictions. Solar power, even when aggregated, is more stochastic and less predictable than aggregate consumption largely because it depends on multiple factors that are specific to each site and highly localized.

In order to monitor and control these intermittent solar energy in the grid, utilities are rapidly installing smart energy meters, which continuously measure and transmit electricity usage at fine-grained intervals using wireless communication techniques, e.g., every minute or less. Unlike traditional analog meters, smart meters dramatically reduce the need for sending meter readers to physically visit customer sites (a.k.a Truck Rolls), and enables two-way power and data flows between customers and utilities to improve the eclectic grid

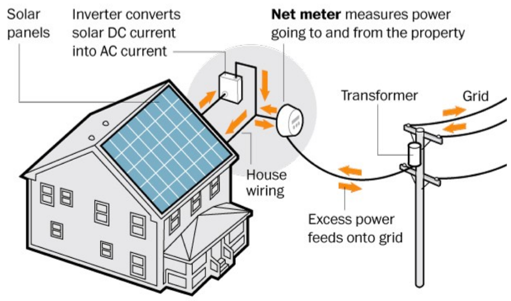

Solar panels
Inverter converts solar DC current into AC current
Net meter measures power going to and from the property
Transformer
Grid
House wiring
Excess power feeds onto grid

Figure 2.1: Solar is “behind the meter” in a grid-tied small scale solar deployment.

management. So far, in the U.S. and Europe, utilities have deployed more than 70 million and 155 million smart meters, respectively, covering over 50% of all households in each region. Thus, ample training data from smart meters is typically available for large residential solar deployments (>10kw) and solar farms, as these deployments are often required to be monitored independently. As a result, these deployments’ meter data represents pure solar data.

However, as shown in Figure 2.1, nearly all the small-scale residential solar deployments (<10kW) that contributes to ∼60% grid-tied solar deployments are “behind the meter” (BTM), such that the smart meter data exposed to utilities represents only the net of a building’s solar generation and its energy consumption, and the pure solar generation data is not revealed. Therefore, the BTM prevents a wide-range of solar data analytics, e.g. monitoring, performance modeling, forecasting, and fault detection, as all these need to access to the historical pure solar generation data. In chapter 4, we present SunDance, a “black-box” technique that accurately disaggregates solar data from smart meter data to to remove this barrier.

2.2 Current Solar Analytics Are Impractical

Prior work on solar data analytics generally takes a “white box” approach that assumes detailed knowledge of a deployment and its location, such as the number of modules and their size, tilt, orientation, efficiency, nominal operating cell temperature, wiring, inverter type, etc. White-box physical models translate this information into the parameters the models require. The PV Performance Modeling Collaborative documents a variety of white-box modeling methods [79]. This approach typically decouples the different effects on solar generation and models them separately. For example, different models exist for estimating ground-level irradiance versus estimating a deployment's efficiency at converting this irradiance to power. The former applies physical models to local or remote sensing data, e.g., ground-level pyranometers or satellites, to estimate irradiance, while the latter applies physical models to estimate the efficiency of converting this irradiance to power. Many tools exist, such as PVWatts [2] and SAM [5], that estimate solar potential using "white-box" models. Unfortunately, while these "white-box" approaches have accuracy, gathering the detailed deployment information at large scales for millions of small-scale deployments is infeasible for utilities.

Significant recent work focuses on learning "black box" models, primarily in the context of forecasting [23, 87], using machine learning (ML) techniques. ML is defined as a set of methods that can automatically detect data patterns, and then use these uncovered patterns to predict future data. "Black-box" approaches are attractive because they use only historical energy and weather data for training. Thus, utilities and third-parties that remotely monitor tens of thousands of solar deployments, e.g., via smart meters and other sensors, can directly apply "black-box" techniques at large scales to vast archives of data. Unfortunately, ML techniques require a significant amount of historical data to train an accurate model. Prior work requires anywhere from months to years [46, 75], while a recent survey states that at least 30 days of data is necessary to train a reasonably accurate model [23]. However, historical data is generally not available for either new deployments or deployments that do not continuously monitor and store the data.

Instead, in this dissertation, we present a hybrid "black box" approach that can achieve the best of both to solar data analytics. Our hypothesis is that the hybrid "black-box" approach can enable a wide range of accurate solar analytics, including modeling, disaggregation, and localization, with limited training data and without knowledge of key system parameters by integrating "black-box" machine learning approaches with "white-box" physical models.

2.3 Anonymized Energy Data Is Not "Anonymous"

Energy data is generally considered "anonymous" if it is not associated with identifying account information, e.g., a name and address, as suggested by the U.S. Department of Energy's recently released Voluntary Code of Conduct (VCC) for managing user energy data [86]. Importantly, the VCC *does not* require user consent to release anonymized energy data with names and addresses stripped. Consent is likely not required because the energy analytics above do not reveal location, which prevents third-parties from associating private behavior above with a specific home. Thus, energy data from these "anonymous" solar-powered homes is often not treated as sensitive: instead, it is routinely transmitted over the Internet in plaintext, stored unencrypted in the cloud, shared with third-party energy analytics companies, and even made publicly available.

A plethora of startups have now arisen to analyze these vast archives of utility energy data, ostensibly to make energy-efficiency recommendations [28, 13, 70]. Prior research has demonstrated the ability to learn a variety of insights into private user behavior by analyzing their energy data [66]. For example, energy data indirectly leaks occupancy [31, 53], which may reveal whether a home's occupants: i) include a stay-at-home spouse, ii) keep regular working hours and daily routines, iii) frequently go on vacation, or iv) regularly eat out for dinner. Energy data can also reveal load power signatures—changes in power unique to a device—for specific appliance brands and models. These behavioral insights and appliance details are valuable to companies in profiling homes and directing advertising campaigns, and may also be exploited by tech-savvy criminals. Thus, some contend that energy data will eventually be worth more than the energy consumed to generate it [68].

Our key insight is that solar energy data is not anonymous: since every location on Earth has a unique solar signature (e.g., a unique sunrise, sunset, and solar noon time) and a unique weather signature (e.g., temperature, wind speed, cloud cover), and they already embedded detailed location information. The localization threat exposes a new privacy threat from energy data, which has never been presented before. In this dissertation, we design two "black-box' localization techniques in Chapter 5 (SunSpot) and Chapter 6 (Weatherman) to explore severity and extent of this privacy threat. This privacy threat explosion is critically important in informing evolving policies by Department of Energy and others for managing “anonymous” energy data, and in emphasizing to users and utilities the need to securely handle energy datasets. Instead of simply removing the account information (e.g., name, address) from energy data, we need to design advanced privacy preserving techniques to protect the energy data before sharing to third-parties or releasing online.

2.4 Mean Absolute Percentage Error

To quantify accuracy of the various solar analytics that we present in this dissertation, we compute the Mean Absolute Percentage Error (MAPE), as follows, between the ground truth solar energy and the solar energy that SunDance infers over all time intervals t. A lower MAPE indicates higher accuracy with a 0% MAPE being perfectly accurate solar disaggregation.

$$MAPE = \frac{100}{n}\sum_{t=0}^{n}|\frac{S_t - P_t}{S_t}| \tag{2.1}$$

Here, S_t and P_t are the actual and inferred average solar power generation, respectively, over time t. We restrict all time periods to between sunrise and sunset, since SunDance is always perfectly accurate at night, as solar generation is always zero. Even so, MAPE is highly sensitive to periods of low absolute solar generation. For example, if sunrise falls near the end of an hour, the absolute generation of a 10kW solar deployment over the hour may only be 50W. If our approach infers a generation of 100W, its MAPE for that period will be 100%. In contrast, the absolute generation during a cloudy mid-day period may be 5kW, such that if our approach infers a generation of 6kW, its MAPE is only 20%. Thus, the absolute error of 50W contributes much more to the average MAPE than the absolute error of 1kW. To put our results in better context, we usually report overall MAPEs, as well as MAPEs for separate time periods.

CHAPTER 3
SOLAR PERFORMANCE MODELING

The increasing penetration of solar power in the grid has motivated a strong interest in developing real-time performance models that estimate solar output based on a deployment's unique location, physical characteristics, and weather conditions. We survey existing work- "white-box" physical modeling and "black-box" ML modeling, and we then present hybrid approaches that combines the benefits of both, and show that they significantly improves solar modeling accuracy.

3.1 Background and Motivation

Solar performance models are useful for a variety of energy analytics, including indirect solar monitoring [40], solar forecasting [23, 87], "behind the meter" solar disaggregation [63, 50, 33], anonymous localization [36], and fault detection [43, 21]. Significant recent work focuses on learning "black box" models, primarily in the context of forecasting [23, 87], using machine learning (ML) techniques. Black-box approaches are attractive because they use only historical energy and weather data for training. Historical and current weather data are freely available form nearly every location in the U.S. form National Weather Service (NWS) and many websites, such as Weather Underground [18]. Thus, utilities and third-parties that remotely monitor tens of thousands of solar deployments, e.g., via smart meters and other sensors, can directly apply "black-box" techniques at large scales to vast archives of data.

Interestingly, these black-box ML approaches are often "off the shelf" and do not leverage well-known physical models of solar generation based on fundamental physical properties. Instead, prior work on physical modeling generally takes a "white-box" approach that assumes detailed knowledge of a deployment, such as the number and type of inverters and solar modules, as well as their rated capacity, efficiency, tilt, orientation, nominal operating cell temperature, and wiring. To develop white-box physical models, experts gather and translate this information into the parameters the models require. The PV Performance Modeling Collaborative distills a series of ten white-box modeling steps [79] implemented as part of the open source PVlib library [22]. Unfortunately, while white-box approaches may yield high accuracy, gathering the necessary information to construct these models at large scales for millions of small-scale deployments is infeasible. Thus, white-box models are typically only developed for utility-scale solar farms.

While recent black-box ML approaches do not require such site-specific information, they also have significant drawbacks. In particular, they require months-to-years of training data to derive accurate models [46, 75, 23], and thus are not immediately applicable to new solar sites coming online, or those that have not archived their historical data. In addition, "off the shelf" ML approaches often do not incorporate well-known physical models of solar generation based on fundamental properties, which reduce their accuracy. To address the problem, we develop an approach to physical black-box modeling that leverages many of the same fundamental properties as existing white-box models. However, rather than derive physical model parameters from a manual site inspection, our approach calibrates them by finding the values that best fit the data. As we show, this calibration requires much less data than training a ML model, as the physical model embeds detailed information about the relationship between the input parameters and solar output.

We survey prior work on solar performance modeling, and then compare black-box approaches using machine learning versus physical modeling [26, 38]. We examine both a canonical "pure" machine learning technique from prior work [63] and a "pure" analytical approach from prior work, which leverages several well-known physical properties of solar generation [33]. We show that a significant drawback of black-box physical modeling compared to ML is that simple physical models i) do not exist for all the variables that potentially affect solar generation, especially the dynamic factors that degrade output, and ii) may require inputs that are difficult to accurately measure. For example, there are no simple physical models that quantify degradation in output due to dust build up, high humidity, or air velocity on solar conversion efficiency [62]. In addition, physical models of cloud cover's impact on solar irradiance requires accurately quantifying cloud cover, which is difficult to measure. In contrast, ML techniques automatically learn these unknown relationships from observed data, and adapt as they change over time. Thus, while black-box physical models have the potential to be more accurate than data-driven ML models, they are generally less accurate in practice.

In this chapter, we compare the accuracy of black-box physical and ML solar performance models, as well as the amount of data required for calibration or training. We then present hybrid solar performance modeling techniques that combine elements of both approaches. Our hypothesis is that a hybrid approach can achieve the best of both worlds by combining well-known relationships from the physical models with unknown relationships learned via ML to improve accuracy, while requiring no more training data from the deployment under test than the pure physical model. However, as we discuss, by normalizing the output of our ML model based on physical solar properties, this training data need not be gathered from the deployment under test. In evaluating our hypothesis, we make the following contributions.

Pure Solar Modeling Approaches. As reference points, we first discuss both a pure ML approach to black-box solar performance modeling from prior work [63] and a pure physical approach, which combines several well-known physical models of solar generation.

Hybrid ML Black-box Solar Modeling. We present a configurable hybrid model that combines ML and physical approaches. In essence, the hybrid approach uses physical models for selected parameters (where physical models are available), and uses ML for the other parameters (where physical models are unavailable). We show that the hybrid approach significantly improves the accuracy of the pure ML and physical approach. In addition, we evaluate multiple variants of our hybrid approach by selectively adding more parameters with physical models. We show that the accuracy of the hybrid model incrementally improves as we model more of the input features using physical models.

Empirical Physical Solar Modeling. Unfortunately, the black-box physical model above is highly inaccurate and performs significantly worse than black-box ML models. This inaccuracy must derive from either the physical models above being inaccurate, or from the effect of unmodeled physical parameters, such as the other weather metrics, shading from surrounding buildings, or soiling from dust and pollen. Therefore, we conduct a large-scale data analysis to determine the primary source of the inaccuracy by isolating the effects of 10 different weather metrics on solar output. Our analysis shows that only 2 weather metrics affect solar generation—temperature and cloud cover—and that their effect is universal and independent of time and location after normalizing for a deployment's physical characteristics. We improve these existing cloud cover models in developing our approach, which estimates a site's solar output at any time based on widely-available temperature and cloud cover readings. Unlike prior ML approaches, which require months-to-years of data to train accurate models, our model requires as few as 2 datapoints to calibrate a specific solar site.

Hybrid Physical Black-box Solar Modeling. Of course, there are unique aspects of each solar site that do affect solar output, which our physical model does not capture. These aspects primarily derive from non-weather-based shading, e.g., from nearby buildings, trees, and mountains, and soiling, e.g., from dust and pollen. Thus, we augment our approach by applying ML to learn only how much these unique site-specific shading and soiling effects decrease the solar output expected by our physical model. As we discuss, these site-specific effects are largely a function of the Sun's azimuth and zenith angles. We show that our ML-enhanced physical black-box model further yields much higher accuracy than current state-of-the-art ML approaches across all 100 sites in our evaluation, especially at sites and during periods with significant shading from obstructions.

3.2 Black-box Solar Modeling

There are significant prior works on ML-based solar modeling and physical solar modeling. In this section, we survey and compare these two different kinds of solar modeling techniques.

3.2.1 Black-box ML-based Modeling

Prior work on ML-based black-box solar modeling has the same broad characteristics. Since solar generation varies based on weather conditions, input features include a variety of weather metrics that are publicly available, e.g., from the National Weather Service (NWS) or Weather Underground, such as temperature, dewpoint, humidity, wind speed, and sky cover. Note that all approaches assume a deployment's location, and thus its weather is well-known. The dependent output variable is often the raw solar output. Given historical weather data and raw solar output, a variety of supervised ML techniques, e.g., regression, neural nets, Support Vector Machines (SVMs), can learn a model that maps the weather metrics to raw solar output. However, since solar generation potential varies significantly each day and over the year, this approach requires learning a separate model for each time period [75]. This significantly increases the training data required to learn an accurate model, as each sub-model requires distinct training data.

To reduce the size of the training data, ML-based modeling can normalize the input and output variables, such that it can use each datapoint to learn a single model [46]. Our pure ML-based approach normalizes these variables without using detailed physical models of the system [63, 65]. In particular, the approach normalizes the output variable by dividing the raw solar power by the solar capacity, defined as the system's maximum generation over some previous interval, which it calls the solar intensity. While the prior work does not specify this interval, in this chapter, we divide by a solar deployment's maximum generation over a year. In addition, the approach also adds the time of each datapoint to the input features along with the time of sunrise and sunset. The time information enables the model to automatically learn the solar generation profile. For example, a time closer to sunrise or sunset will have a lower solar intensity, even in sunny clear sky conditions, compared to a time closer to solar noon. The approach then uses a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel to learn a model from the training data. SVM-RBF is common in solar modeling, since it attempts to fit a Gaussian curve to solar data and solar profiles are similar to Gaussian curves [75, 63, 27]. Figure 1 depicts a typical solar profile and its best fit Gaussian curve. As the figure shows, the Gaussian curve fits well in the middle of the day, but diverges at the beginning and end of each day.

Note that the approach above is completely data-driven and does not incorporate any physical models of solar generation, other than the insight that solar curves vary over time and are similar in shape to Gaussian curves. While the approach requires multiple months of training data to learn an accurate model, the authors claim that the normalization enables them to train the model on different solar deployments than they test on, since all

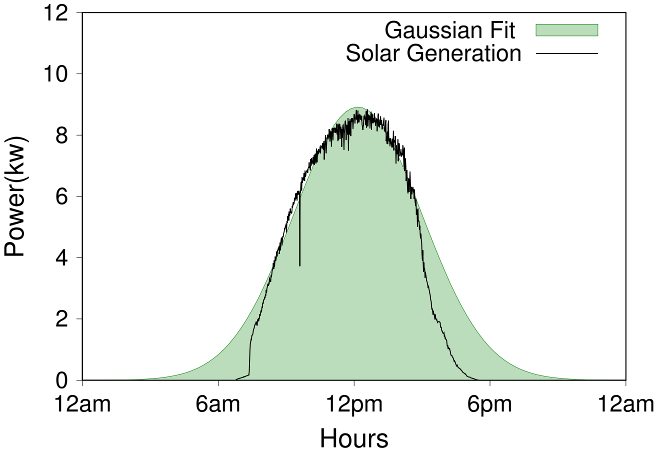


Figure 3.1: Solar data along with a best fit Gaussian curve.

solar profiles exhibit the same Gaussian shape. In fact, this model was developed for solar disaggregation, where solar data from the deployment under test is unknown, thus requiring the model to be trained using data from separate deployments. Of course, as we show,, the model is more accurate when trained data from the deployment under test due to physical differences between deployments that affect solar output.

3.2.2 Black-box Physical Modeling

Our approach to physical modeling leverages several well-known relationships that govern solar generation. Our physical model leverages existing models that estimate the clear sky solar irradiance at any point in time at any location based on the Sun's position in the sky. Many clear sky irradiance models have been developed over the past few decades with varying levels of complexity [60]. There are multiple libraries available that implement these models [3, 1] with the simplest models requiring as input only a location, i.e., a latitude and longitude, and time. The output is then the expected clear sky irradiance (in W/m^2) horizontal to the Earth's surface. This is the maximum solar energy available to a solar module to convert to electricity.

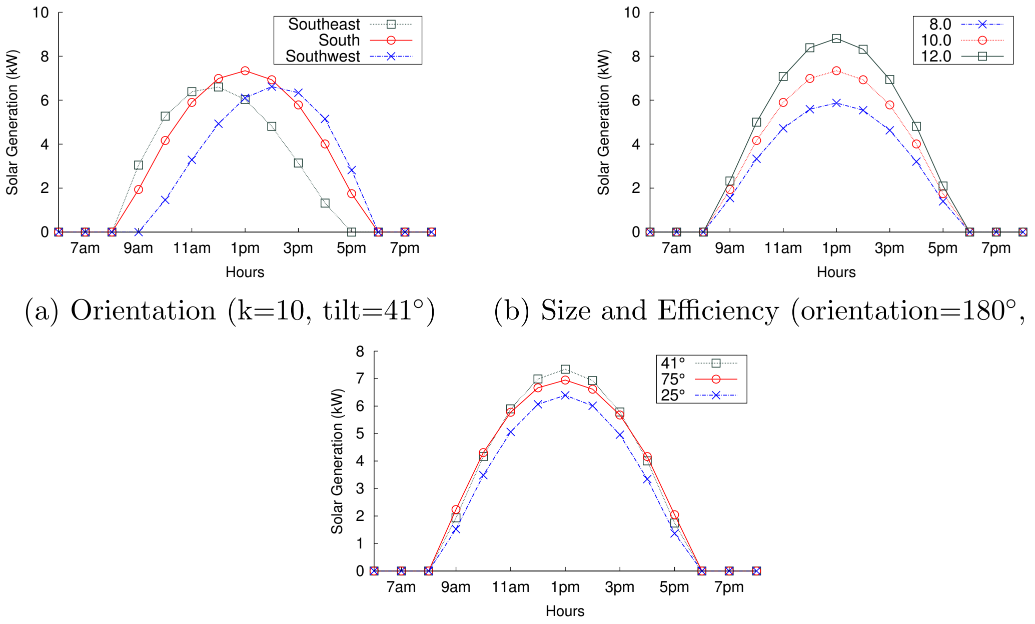
3.2.2.1 Computing Clear Sky Irradiance

Solar irradiance is the power transmitted to the Earth by the Sun, and is measured in units of kilowatts per meter squared (kW/m^2). While the Total Solar Irradiance (TSI) that strikes perpendicular to the Earth's atmosphere is relatively constant and estimated at ∼1.361 kW/m^2, the irradiance that reaches the ground is much less due to atmospheric losses (even under clear skies). The magnitude of these losses is largely a function of the Air Mass (AM) that light must travel through to reach the Earth, such that that the larger the AM the lower the fraction of TSI that reaches the ground. The AM is, in turn, a function of the Sun's position in the sky. For example, the fraction of TSI that reaches the ground is less closer to sunrise or sunset, as the Sun's light must pass through much more of the Earth's atmosphere at those times.

Since the Sun's position in the sky is a well-known function of location and time, it is possible to use the AM along with measurements of other atmospheric parameters to estimate the *clear sky irradiance* under a cloudless sky at any point on Earth at any time. There are many clear sky irradiance models that range from simple geometric formulas involving only the AM, the Sun's position, and experimentally-derived constants to highly complex models that require detailed data on the specific location's atmospheric conditions [60]. Note that evaluating the accuracy of these models is outside the scope of this work, and has been the focus of significant prior work [60]. While our approaches are compatible with any of these models, our implementation in this chapter uses a simple model that requires only a location's latitude and longitude and the Sun's position, which is a function of location and time.

3.2.2.2 Modeling Physical Characteristics

Solar cells harness the photovoltaic effect to translate the Sun's irradiance into electrical energy. However, the efficiency of solar cells depends on a variety of physical characteristics specific to each solar deployment. For example, the efficiency of commercial solar modules varies widely due to different materials and manufacturing processes, e.g., mono- versus poly-crystalline modules. In addition, a number of other physical characteristics further reduce solar module efficiency. The most important physical characteristics that affect efficiency are a solar module's size, tilt, and orientation. For example, the clear sky irradiance models above assume a 100% efficient solar module lying flat on the ground, such that its directional orientation and vertical tilt are equal to 0°. However, if a solar module is tilted



(a) Orientation (k=10, tilt=41°)

(b) Size and Efficiency (orientation=180°, tilt=41°)

(c) Tilt (orientation=180°, k=10)

Figure 3.2: Maximum clear sky solar generation potential near NYC on 1/1/2016 for different physical deployment characteristics, including different orientations α (a), sizes and efficiencies k (b), and tilts ϕ(c).

upward and facing away from the Sun, not all of the available solar irradiance will reach it. As before, the effect of solar module size, tilt, and orientation are well-known and can be expressed using the closed-form equation below that relate a module's solar power generation P_s to the solar irradiance incident on the module $I_{incident}$ and the physical characteristics above.

$$\begin{multline} P_s = I_{incident} * k * [\cos(90 - \Theta) * \sin(\beta) * \cos(\phi - \alpha) \\ + \sin(90 - \Theta) * \cos(\beta)] \tag{3.1} \end{multline}$$

Here, Θ is the Sun's zenith angle above (such that 90-Θ is the Sun's elevation angle), α is the Sun's azimuth (or orientation) angle, β is the solar module's tilt angle, and ϕ is the solar module's azimuth (or orientation) angle. The Sun's zenith angle ranges from 0° (when the Sun is directly overhead) to 90° (at sunrise or sunset). Similarly, a solar module's tilt angle ranges from 0° when lying flat on the ground to 90° when vertical. The orientation angles for both the Sun and the module range from 0° (directly north) to 180° (directly south). Finally, the k parameter represents a combination of a solar module's size and its efficiency, expressed as a percentage of the incident solar irradiance $I_{incident}$ it converts to electrical energy. For example, a solar module that is 2× larger but half as efficient as another solar module would have the same value of k.

Figure 3.2 illustrates the physical effects on clear sky generation potential at a location just north of New York City, at 41° latitude and -74° longitude, on January 1st, 2016 for different solar module orientations (a), sizes and efficiencies (b), and tilts (c). As the figure shows, orienting the solar modules west or east shifts the peak solar generation later or earlier, respectively. In addition, since the k parameter from Equation 3.1 is a constant scaler it simply scales the curve up and down. The tilt parameter (β) has a similar effect as k, in that it also tends to scale the curve up and down for practical values, but is not a scaler, and thus also affects the orientation shift.

White-box models can directly measure the module angles, size, and efficiency. While black-box models cannot directly measure these values, given the relationships above, it can search for these parameters via curve fitting. In particular, P_s(t) follows the equation above and $I_{incident}$(t) is known from existing clear sky models. To search, we can set the tilt and orientation to their ideal values (a tilt equal to the location's latitude and a south-facing orientation in the northern hemisphere), and then conduct a binary search for the k that both minimizes the Root Mean Squared Error (RMSE) with the observed data and represents a strict upper bound on the data, as we know generation should never exceed the maximum dictated by the clear sky irradiance. After fitting k, we then conduct a similar binary search for orientation and tilt. We iterate on the search until the parameters do not significantly change. In Chapter 4, we show that this searching method results in highly accurate values for k and the orientation and tilt angles.

3.2.2.3 Modeling Weather Effects

The model found above assumes that k is static and never changes. However, module efficiency changes over time based on numerous dynamic conditions, such as temperature, rain, snow, humidity, dust, etc. In particular, the effects of temperature on module efficiency are well-known, and are described by a variety of physical models.

Temperature Effects. While multiple weather metrics may affect solar cell efficiency, the most significant metric is the ambient temperature. The closed-form equation below estimates the cell temperature based on the temperature of the ambient air [73]. The simplest model is the Nominal Operating Cell Temperature (NOCT) model, which specifies the cell temperature based on the ambient air temperature and the cell temperature at 1kW/m^2 in 25C. For every degree increase (or decrease) in T_{cell}, module efficiency drops (or rises) by roughly a constant percentage, which varies between modules, but is ~0.5% per degree Celsius.

$$T_{cell} = T_{air} + S * \frac{NOCT - 20}{800} \tag{3.2}$$

Here, T_{cell} is the cell temperature in Celsius, T_{air} is the ambient air temperature in Celsius, S is the solar irradiance that is striking the panel (in W/m^2), and $NOCT$ is the Nominal Operating Cell Temperature. The NOCT varies between solar modules, but generally ranges from 33°C to 58°C with 48°C as a typical value. Importantly, for every degree increase (or decrease) in T_{cell}, the efficiency drops (or rises) by a constant percentage. While the precise temperature-based efficiency loss varies between modules, it is typically ~0.5% per degree Celsius.

To account for temperature effects, we can re-calibrate our model by adjusting the original value of k above based on the temperature at each datapoint using the equation below, where $T_{baseline}$ is the temperature at the datapoint that is closest to the upper bound solar curve in the model above. Note that the relationship between cell temperature and air temperature is a constant. While efficiency varies strictly based on cell temperature, the cell temperature's relationship to air temperature differs only by an additive constant, which cancels out when subtracting two cell temperatures (leaving only the air temperature below). The baseline temperature should represent the coldest point in the year that has a clear sky. Again, we search for the value of c that minimizes the RMSE with the observed data but remains a strict upper bound on the data.

$$k'(t) = k * (1 + c * (T_{baseline} - T_{air}(t))) \tag{3.3}$$

Cloud Cover Effects. The adjustment above represents a temperature-adjusted clear sky solar generation model. Of course, skies are not always clear, such that the solar irradiance that reaches Earth is much less than the clear sky solar irradiance. The amount of cloud cover is the primary metric that dictates the fraction of the maximum solar irradiance that reaches the ground. As above, there are numerous well-known physical models [67, 89] that translate cloud cover into a clear sky index, which is the solar irradiance that reaches the Earth's surface divided by the clear sky solar irradiance [59]. For example, one well-known cloud cover model is below [4].

$$I_{incident}/I_{clearsky} = (1 - 0.75n^{3.4}) \tag{3.4}$$

Here, $I_{incident}$ represents the solar irradiance that reaches the Earth, $I_{clearsky}$ represents the solar irradiance from the clear sky model, and n represents the fraction of cloud cover (0.0-1.0). This cloud cover (or sky condition) is typically measured in *oktas*, which represents how many eighths of the sky are covered in clouds, ranging from 0 oktas (completely clear sky) through to 8 oktas (completely overcast). The sky conditions reported by the NWS translate directly to oktas [19]. For example "Clear/Sunny" is <1 okta, "Mostly Clear/Mostly Sunny" is 1-3 oktas, "Partly Cloudy/Partly Sunny" is 3-5 oktas, "Mostly Cloudy" is 5-7 oktas, and "Cloudy" is 8 oktas. While the sky condition reported by the NWS (and other sources) is a rough measure of cloud cover, more accurate measures can be extracted from satellite images [42]. However, this is non-trivial and these measures are not reported by weather sites.

Thus, using the equation above we can adjust the output of our physical model by multiplying the solar output in our temperature-adjusted model above by the fraction $I_{incident}/I_{clearsky}$. Note that, while Equation 3.4 is in terms of solar irradiance and not solar power, the ratio of observed solar power to maximum solar generation potential after the temperature adjustment (from Equation 3.3) are equivalent, since the effect of the physical characteristics cancel out. Recent work refers to this value as the clear sky photovoltaic index [40]. We could continue to adjust our model downwards based on physical models for other conditions, such as humidity, air velocity, and dust buildup [62]. Unfortunately, similar types of simple models are not readily available for these parameters.

One benefit of the physical model above is that it requires very little data to calibrate. In the limit, it requires only two datapoints during clear skies with a significant difference in temperature. In recent work, we show that physical models of clear sky generation (without the cloud cover adjustment) built with only two days of data have similar accuracy to those built with a year's worth of data [33]. However, unlike the ML-based models, our physical model is necessarily custom to each deployment based on its unique location, tilt, orientation, efficiency, and sensitivity to temperature. Our physical model also does not account for shade from surrounding structures, e.g., buildings and trees, or multi-module systems with different tilts, orientations, and efficiencies that are wired together, e.g., in series, parallel, or a combination. While accounting for these effects in the physical model is possible, it would significantly increase its complexity. In contrast, the ML-based model is capable of inherently incorporating these effects into its model.

Other Weather Effects. The NWS and other weather sites, such as Weather Underground, report numerous other weather metrics, including dew point, humidity, visibility, pressure, precipitation intensity, precipitation probability, wind speed, and wind bearing. While some work has examined the effect of a few of these metrics on solar output [62, 74, 85], their effects are still not well understood and there are no commonly-used white-box physical models for them. Black-box ML approaches generally include these additional weather metrics as input features in case they do affect solar output.

3.3 (Mostly) ML Black-box Solar Modeling

The black-box ML and physical solar performance models from the previous section have both benefits and drawbacks. The ML model generally requires months of training data to build an accurate model. As we show, while we can train the pure ML model on data from one set of solar deployments, and then use it to model a separate set of solar deployments, this significantly decreases the model's accuracy, since the approach does not take into account different physical system characteristics, e.g., tilt, orientation, size, and efficiency. In contrast, while our physical model requires little data to calibrate, it is generally less accurate than the ML model in practice because it i) depends on coarse measurements of cloud cover that are often inaccurate and ii) does not incorporate the effect of other conditions that degrade output, such as additional weather metrics, complex multi-panel characteristics, dust and snow buildup, and regular shading patterns from nearby structures. Thus, to leverage the benefits of both approaches, we present a configurable hybrid approach that combines both approaches.

Our hybrid approach first builds a physical model of solar output, as in section 3.2.2, based on a deployment's location, tilt, orientation, size, efficiency, and any other relevant parameters where physical models exist. The approach then trains a ML classifier, similar to the one in section 3.2.1, that includes as input features any relevant parameters not included in the physical models. However, a key difference relative to section 3.2.1 is that the dependent output variable is not the raw power normalized by the (static) solar capacity, but is instead the raw power normalized by the generation potential from the physical model above. Thus, *the dependent output variable represents the additional percentage reduction in solar generation beyond that estimated by the physical model due to the parameters in the ML model.* For example, the physical model might estimate a solar output of 1kW based on the current location, time, temperature, and cloud cover. However, based on the other metrics, the ML model may then estimate the actual output to be 80% of this 1kW output. In this case, the labeled data in the training set for the ML model would include any input features that are not physically modeled with an output variable of 0.80.

Thus, our hybrid model estimates solar output by multiplying the estimated output from the physical model by the fraction specified in the ML model. Note that, when the physical model includes only the metrics that affect module efficiency, e.g., tilt, orientation, size, and temperature, this ratio represents the clear sky (photovoltaic) index [40]. Our hybrid approach is configurable because we can either model input features with physical models, or using the ML model. For example, in our evaluation, we examine different hybrid variants that physically models different sets of parameters.

Note that, since the physical model is already a function of time, our ML classifier does not need sunrise, sunset, or current time as input features, unlike the pure ML model from section 3.2.1. In addition, by specifying our output variable as a function of the physical model, its normalization naturally takes into account the physical differences between solar deployments. Thus, as with our pure ML model, our hybrid approach can accurately train its ML model on data from one set of solar deployments, and then apply it to a separate set of deployments with widely different physical characteristics. Of course, for any new deployment, we would still need to calibrate a physical model of the system, as described in section 3.2.1. However, as we discuss, this only requires a minimal amount of data. In some sense, our physical model captures *how efficiently* a solar deployment translates the available solar irradiance into electricity, while our ML model captures *how much* solar irradiance actually reaches the module. As we show in recent work [33], the latter is primarily due to weather effects that are general and not dependent on specific physical deployment characteristics.

In this chapter, we use the same classifier (SVM-RBF) in our hybrid ML model as we do in the pure ML model [63] to provide a direct comparison. More sophisticated ML modeling techniques could potentially learn the physical models above from training data without requiring the manual identification we perform in our hybrid approach. However, for systems, such as solar deployments, where the physical effect from a subset of inputs on a dependent output variable is well-known, and independent of the other inputs, it is more straightforward to simply calibrate the input directly from the data using the model. As we show, this approach significantly increases accuracy using straightforward ML techniques.

3.3.1 Implementation

We implement the ML-based, physical, and hybrid black-box solar performance models using python. We use the *scikit-learn* machine learning library to implement our ML-based models. We implement the pure ML-based model as specified in prior work [63, 65] using the same input features, dependent output variable, and SVM-RBF kernel. In particular, we use one hour resolution weather metrics (from Weather Underground) including the sky cover, dewpoint, humidity, temperature, and windspeed. We translate the sky cover string into a cloud cover percentage using the standard okta translation [19]. Our physical model leverages the PySolar [3] library for computing the clear sky irradiance at any location and time, which it uses to find the tilt, orientation, size, efficiency, and temperature coefficient that best fits the data. Our basic hybrid ML model uses the same weather metrics as with the pure ML-based model [63, 65], and thus does not include temperature and cloud cover as part of the physical model. We implement two other hybrid variants: one that physically models temperature and thus takes it out of the training set of input features, and one that physically models both temperature and cloud cover, which also takes cloud cover out of the ML model's training set.

We evaluate the accuracy of each model on data from 6 rooftop solar deployments at different locations with widely different physical characteristics. Since our weather data has one-hour resolution, we use average power data at one-hour resolution in our evaluation. We examine model accuracy using two different training scenarios, where we train the ML models (both pure and hybrid) using data from either i) the same deployment we test on or ii) different deployments than we test on. In the former scenario, we perform cross-validation across one-year of data to split the dataset into a training and testing set (in a 2:1 ratio). In the latter case, we train the ML model using one year of data from 4 other deployments, and then apply the model to estimate solar output over one year from the 6 deployments. Since, due to Figure 1, the Gaussian fit is most inaccurate during the morning/evening, we evaluate accuracy over both the entire day and over mid-day between 10am and 2pm. Finally, we quantify model accuracy using the Mean Absolute Percentage Error (MAPE), as shown in Equation 2.1.

3.3.2 Experimental Evaluation

Figure 3.3 quantifies model accuracy for the 6 buildings with rooftop deployments in our test set across multiple scenarios. Buildings #1-#6 are located in Pennslyvania, Texas, New York, Arizona, Washington, and Massachusetts, respectively. The deployments have a wide range of sizes: buildings #1-#6 consist of 110, 16, 93, 36, 17, and 30 solar modules, respectively, with a standard size of 165cm×99cm which typically have a rated capacity of ~230-330 based on the module type. The top graph is the scenario where we train a ML model for each deployment using its historical data, while the middle and bottom graphs train a ML model on 4 separate homes (not in the set of six) and then apply that same model to each of these 6 homes. The top two graphs compute MAPE over each day (across

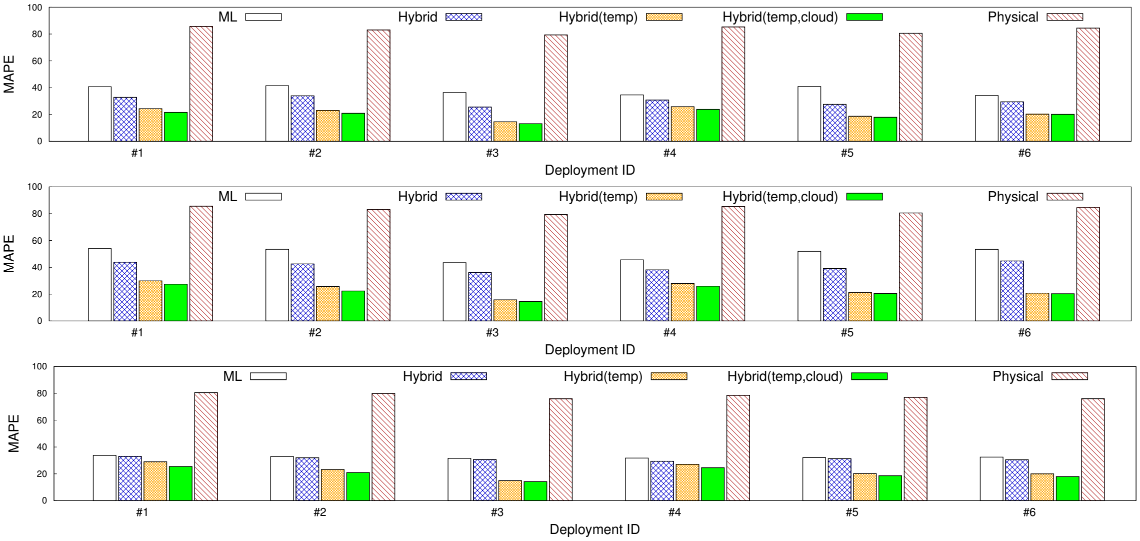


Figure 3.3: Solar model accuracy when training the ML model on the same deployment (top), training on different deployments on different deployments (middle), and the accuracy (when training on different deployments) during the middle of the day.

a year of data), while the bottom graph computes it from 10am-2pm. Note that the physical model requires no training; we include it in all the graphs for comparison.

The experiment shows that the physical model performs significantly worse than all the models that use ML. This is primarily due to i) the coarseness and imprecision of the cloud cover metric, and ii) that it cannot account for conditions that do not permit physical modeling, including the effect of other weather metrics [62]. As part of future work, we are leveraging various satellite images to better quantify real-time cloud cover, such as via the HELIOSTAT method [42], which should improve the results of the analytical model. Unfortunately, an accurate and precise cloud cover metric is not available via common weather services and APIs. In contrast, the pure ML approach can inherently incorporate such effects and achieves a significantly higher accuracy in all cases. Importantly, though, the hybrid model, even without including temperature and cloud cover, significantly improves on the pure ML approach. For example, for deployment's #3 and #5 in the top graph, the improvement is over a 30% reduction in MAPE. Significant, although slightly lesser, improvements are also apparent in the middle graph. The reason for this reduction stems from normalizing the output variable of the hybrid approach's ML model based on a custom physical model of the deployment's output over time, rather than a static capacity value as in the pure ML model.

In addition, as we incorporate more physical parameters into the hybrid model, the more accurate the model becomes. This is most evident when shifting temperature from the ML model to the physical model, which results in another significant decrease in MAPE in all cases. Further, even though cloud cover is a coarse and imprecise metric, by incorporating it into the physical model (along with temperature), we again observe a slight reduction in MAPE in all cases, relative to the hybrid model that only incorporates temperature. These results hold whether we train a ML model for each deployment using its historical data (top) or train a general model using data from other deployments (middle). As expected, the former results in significantly higher accuracy in all cases compared to the latter. However, as the bottom graph indicates, much of this inaccuracy is due to imprecision at the beginning and end of each day. When quantifying only the mid-day accuracy, the pure ML-based approach is only slightly less accurate than our basic hybrid approach, since the Gaussian fit is much more accurate in the middle of the day. However, our hybrid approach significantly improves upon the pure ML model when incorporating the physical models for temperature and cloud cover (even during the mid-day hours in the bottom graph), especially for deployments #3, #5, and #6.

Overall, our results indicate that the hybrid approach achieves much better accuracy than either the pure ML or pure physical approach in all cases. In addition, by training the ML model on separate deployments than we test on, the hybrid model requires only a small amount of training data (as few as two datapoints) from the system under test to calibrate an accurate model.

The model error of our black-box approach is likely higher (∼15-25) MAPE than that of highly-tuned white-box approaches. However, a direct comparison is difficult as prior work uses a wide range of error metrics. In many cases, these metrics are not normalized, and thus vary based on a deployment's capacity. In addition, the variability of weather at a location also affects model accuracy. For example, solar performance models are likely to be more accurate in San Diego, where there is little variation in weather, compared to Massachusetts where weather has more day-to-day and season-to-season changes. As part of future work, we plan to incorporate more accurate estimations of ground-level irradiance from visible satellite imagery, such as offered by SolarAnywhere and SoDa. We expect this to significantly improve accuracy relative to the coarse cloud-cover metrics in standard weather data.

3.4 (Mostly) Physical Black-box Solar Modeling

The black-box model above is entirely based on well-known physical models that incorporate the effect of module size, efficiency, location, time, temperature, cloud cover, tilt, and orientation on solar output. The model is black-box, since it can determine the unique parameter values for each site, given its location, using as few as 2 datapoints (primarily to determine the temperature coefficient). Of course, black-box ML models that are purely data-driven can potentially learn these relationships, given enough training data from a site. However, since these relationships are based on fundamental physical properties common across all solar sites, re-learning them at every site is wasteful and unnecessary. *Unfortunately, as we show, the black-box physical model above is highly inaccurate and performs significantly worse than black-box ML models.* This inaccuracy must derive from either the physical models above being inaccurate, or from the effect of unmodeled physical parameters, such as the other weather metrics, shading from surrounding buildings, or soiling from dust and pollen. In the next section, we conduct a large-scale data analysis to determine the primary source of the inaccuracy by isolating the effects of 10 different weather metrics on solar output

3.4.1 Large-Scale Weather Data Analytics

To isolate the effect of any single weather metric on solar output, we must normalize the effect of all the other variables. We use the models of physical characteristics from section 3.2.2.2 to normalize for the effect of module/array location, size, efficiency, tilt, orientation, and the time of the day and year. This normalization divides the raw solar power observation by the maximum solar generation estimate under clear skies from our model, which, as we discuss in section 3.2.2.3, should be the same as ratio of the observed GHI to the clear sky GHI, modulo any module shading or soiling effects. Figure 3.4 verifies this by

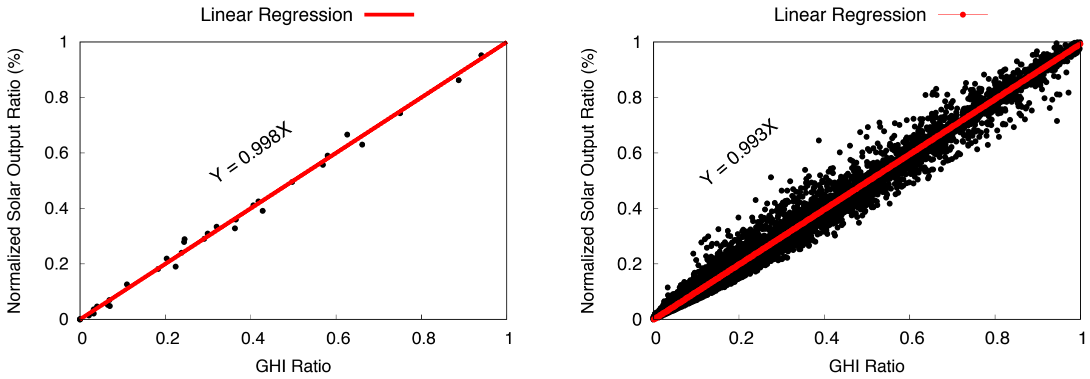


Figure 3.4: Normalized solar output as a function of GHI from a solar radiation sensor (left) and satellite imagery (right).

plotting our normalized solar output ratio (on the y-axis) as a function of the GHI ratio (on the x-axis), which is also called the clear sky index. Figure 3.4(a) uses a solar radiation sensor deployed at a single unshaded solar site to determine the GHI ratio. However, since most sites do not have an on-site solar radiation sensor, (b) computes the GHI ratio using the Heliosat-3 algorithm, which estimates it from visible satellite imagery [42]. Both figures show that our normalized solar power ratio ratio is close to the GHI ratio, with a linear regression line close to y=x.

Importantly, our normalization makes the solar output from many different sites directly comparable. Thus, our data analysis normalizes hourly solar output data from nearly 343 million hourly weather and solar readings, or 78,435 aggregate years, gathered from 11,205 solar sites. We gathered this data from public sources, including Pecan Street's DataPort [13] and PVoutput [16]. We gathered weather data from Weather Underground's API [18], which includes current and historical data from 180k weather stations in the U.S. To isolate the effect of 10 weather metrics on solar output, we examine clusters of datapoints where *all 10 weather metrics have nearly the same value.* One reason we use so much solar data is that finding a statistically significant number of hours where all 10 metrics are the same is not possible at a single site (or even a few sites), as the weather across these 10 metrics varies too much. Even in our massive-scale dataset, identifying sufficiently large clusters is challenging. As a result, our clusters only ensure that each of the 10 metrics is within a small range. We suspect this massive data requirement is one reason a similar analysis has not been conducted in prior work. In contrast, the Kasten-Czeplak cloud cover equation was derived from 10 years of hourly data at only a single site [51].

Universal Weather-Solar Effect. A key insight of our work, which we will leverage in the next section, is that the *exact same weather should have the same proportionate affect on the maximum solar irradiance potential I_{total} that reaches the ground, regardless of its magnitude,* which varies widely over time time at different locations. That is, if two different locations A and B experience the *exact same weather conditions* at two different times then the solar irradiance that reaches the ground $I_{incident}$ will be $c * I_{total}^{A}$ at location A and $c * I_{total}^{B}$ at location B, where c is a constant based on the weather and I_{total} is the maximum clear sky solar irradiance at those locations at those times. We call this the *Universal Weather-Solar Effect.*

Figure 3.5 plots the normalized solar output ratio on the y-axis for 5 such clusters, which each include 7500 hourly datapoints on the x-axis. We use the multi-dimensional k-Means algorithm to identify these clusters, such that we define a threshold for the distance to each cluster's centroid to control the number of datapoints. For each cluster, we select the minimum distance threshold necessary to ensure 7500 datapoints. Table 3.1 shows the weather metric value at the centroid of each cluster. Note that these hourly datapoints are from different times from numerous solar sites with different locations and physical characteristics: the only commonality is the value of the 10 weather metrics within each cluster. We select these illustrative clusters to yield different ratios on the y-axis. Importantly, Figure 3.5 shows that the same weather conditions have the same percentage effect on the normalized solar output ratio (or, equivalently, the GHI ratio) at any site, regardless of the time, a site's location, or its other physical characteristics. That is, the effect of weather is *universal* across all solar sites.

Given our 5 clusters, we can now isolate the effect of any single weather metric by removing it from the cluster, and including all datapoints where the other 9 metrics are within the cluster, but the removed metric can take on any value. Doing so, empirically demonstrates the effect of only that single weather metric on the normalized solar output ratio (modulo any shading or soiling effects) across our massive dataset. Note that we have already applied the temperature adjustment to all these ratios based on each site's tem-

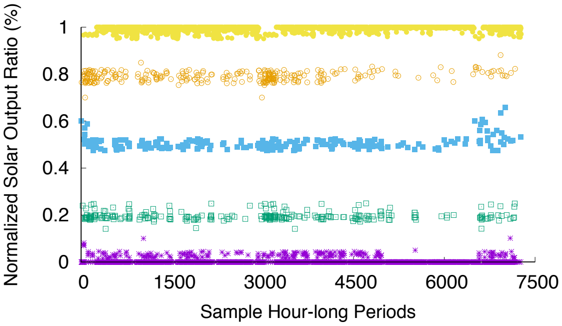
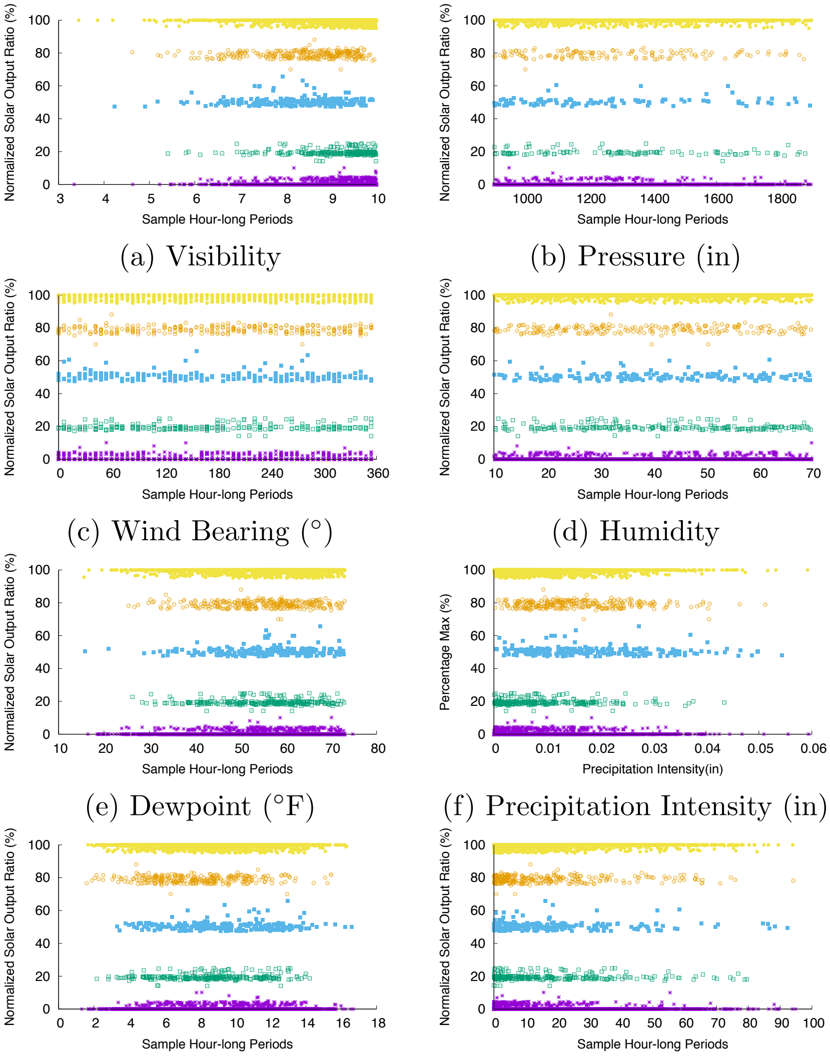


Figure 3.5: Normalized solar output ratio across many solar sites and hour-long time periods for 5 different clusters, indicated by the 5 colors, with the same weather.

Metrics	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Visibility	9.69	11.53	11.77	9.58	11.47
Pressure (in)	1216.51	1211.06	998.10	1103.03	1198.61
Wind Bearing	189.25	203.90	179.07	169.33	180.03
Dewpoint (°F)	44.83	43.31	41.69	55.01	49.84
Precipitation Intensity(in)	0.008	0.019	0.022	0.017	0.011
Wind Speed(mph)	7.81	9.05	8.41	6.11	8.32
Humidity (%)	48.15	48.36	58.39	38.60	35.71
Precipitation Probability(%)	33.13	31.21	35.03	29.18	38.19
Cloud Cover(%)	94.79	77.86	40.97	21.85	1.83

Table 3.1: The centroid for each of the 5 clusters of weather metrics.

perature coefficient using the physical model from section 3.2.2.3. We empirically validated the linear NOCT physical model that describes the temperature effect using our data, but omit the graph due to space constraints. Figure 3.6 shows the results that isolate the effect of the 8 weather metrics without physical models. The graphs show these weather metrics have no significant effect on solar output, as the normalized solar output ratio remains the same regardless of the value of the metric, whether extremely high or low. That is, the lines are horizontal with a value equal to that from Figure 3.5. Thus, these metrics are not useful in estimating solar performance, and need not be included when training black-box ML models.



(a) Visibility

(b) Pressure (in)

(c) Wind Bearing (°)

(d) Humidity

(e) Dewpoint (°F)

(f) Precipitation Intensity (in)

(g) Wind Speed (mph)

(h) Precipitation Probability

Figure 3.6: The isolated effect of 8 unmodeled weather metrics on the normalized solar output ratio for our 5 clusters.

In contrast, Figure 3.7(left) isolates cloud cover, which demonstrates a clear non-linear relationship with the normalized solar output ratio. At first glance, the relationship appears similar to the Karston-Czeplak model from section 3.2.2.3. However, Figure 3.7(right) plots the normalized temperature-adjusted solar output ratio as a function of cloud cover for a larger random sampling of our entire dataset (and not just from the 5 clusters), as the entire dataset is too large to fit on a graph. The graph shows that, while imprecise, the datapoints follow the same trend as in Figure 3.7(left). The imprecision is not surprising, given the imprecision inherent to measuring oktas. In addition, module shading and soiling, which

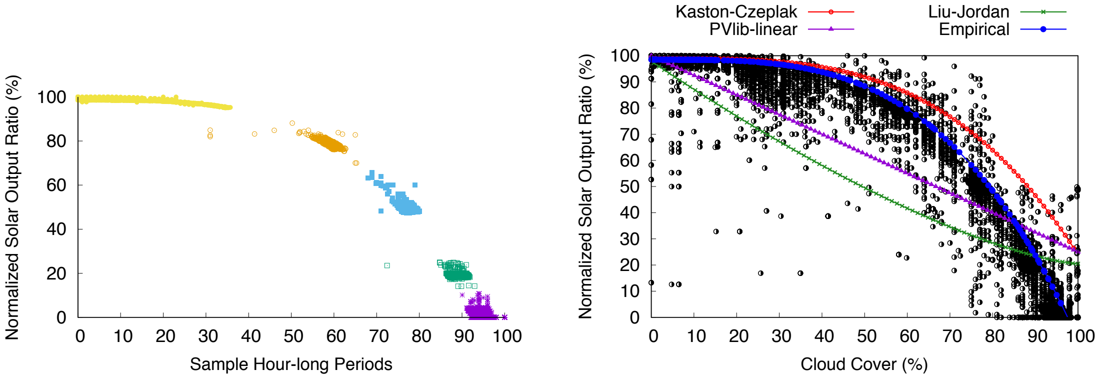


Figure 3.7: Isolated effect of cloud cover on normalized solar output ratio for 5 clusters (left). Normalized solar output versus cloud cover for a larger random sample of clusters, along with existing models and our new empirical one (right).

can cause the ratio to be lower than expected based solely on the weather, also contributes to the imprecision. We also graph the Kasten-Czeplak equation [51], as well as PVlib's models in their default configuration. In this case, for the Liu-Jordan model, we set the zenith angle to 45°, as our data normalizes for the zenith angle.

Similar to the linear model, the Liu-Jordan model is linear for any given zenith angle. As a result, both of PVlib's models are poor fits for the normalized data. The Kasten-Czeplak model is a better fit, but becomes increasingly imprecise as the cloud cover increases, with errors greater than 2× for cloud covers above 90%. Thus, we improve on Kasten-Czeplak by keeping the same model form, but finding parameters that provide a tight upper bound on the bulk of the datapoints. We assume the high outlier values are incorrect okta measurements, and use k-Means clustering to filter them out. As before, we fit the tightest upper bound that minimizes the RMSE with the data, which automatically filters out low values due to shading, soiling, or imprecise okta measurements. Our corrected empirical cloud cover equation is below and in Figure 3.7.

$$I_{incident}/I_{clearsky} = (0.985 - 0.984n^{3.4}) \tag{3.5}$$

3.4.2 Integrating ML-based Modeling

The previous section i) demonstrates that weather's effect on a site's normalized solar output ratio is universal across all solar sites, ii) shows the only weather metrics that affect solar output are temperature and cloud cover, and iii) derives a new physical model to account for cloud cover's effect. The other physical models in section 3.2.2 are also universal across all solar sites, and account for module/array size, location, time, efficiency, tilt, and orientation. Since these models are universal across all sites, there is no need to separately learn them at each site using ML. However, ML is potentially useful for learning the effect of unmodeled parameters that *are* unique to each site. These unmodeled parameters include shade from surrounding buildings and trees, soiling from dust or pollen, or the wiring configuration, e.g., series, parallel, or a combination, of multiple modules with different tilts and orientations. Importantly, all of these unmodeled parameters are a direct function of the position of the Sun in the sky, i.e., its azimuth and zenith angle.

Thus, we enhance our physical model using ML to learn each site's unique shading effects from training data. Our ML model's input features are the Sun's azimuth and zenith angles, while the dependent output variable is the observed solar output divided by the solar output estimated by our physical model. Thus, with no effects from unmodeled parameters, the output variable should be 1, while with significant effects, the output variable should be <1. Note that the Sun's azimuth and zenith angles are a function of time at a given location, and many prior ML models include time as an input feature, enabling them to indirectly learn such shading effects. However, since the solar angles are unique at every point over the year, using time either requires i) multiple years of input data to learn shading effects at each unique time in the year, or ii) rough approximations that assume the same time of day on different days are equivalent over a week or month. In contrast, directly using solar angles at each datapoint as features makes all points comparable and eliminates the need for approximations, which reduces the training data necessary to learn an accurate model.

3.4.3 Implementation

We implement our solar performance model using a mixture of python and C++. To build the model, we require a site's location, i.e., its latitude and longitude, and some time-stamped solar generation data as input. We use the location to fetch historical hourly measurements of temperature and cloud cover at the time of solar generation from Weather Underground's API [18]. Once built, the model estimates solar output at any time t based on the weather at t. Our implementation requires a clear sky GHI model for calibration. We implement a clear sky GHI model from first principles using an open source C++ implementation of the PSA algorithm, which computes the Sun's azimuth and zenith angles to within 0.0083°. Prior work describes in detail how to compute the clear sky GHI given the solar angles, which are a function of location and time [33, 15]. We could also use the clear sky GHI models available in open source libraries, such as PVlib [22], PySolar [3], and NREL's implementation [1]. We use the scikit-learn ML library in python to train our ML model based on solar angles, as well as the ML models we compare against in our evaluation [17]. Our approach in Section 3.4.2 is compatible with any ML modeling technique, such as SVMs or DNNs. Our current implementation uses SVM-RBF, similar to prior work on solar modeling [75, 63, 27]. We also use NumPy [11] and Pandas [12] libraries for weather and energy data processing. We plan to release both the dataset from this paper and the implementation of our model as open source.

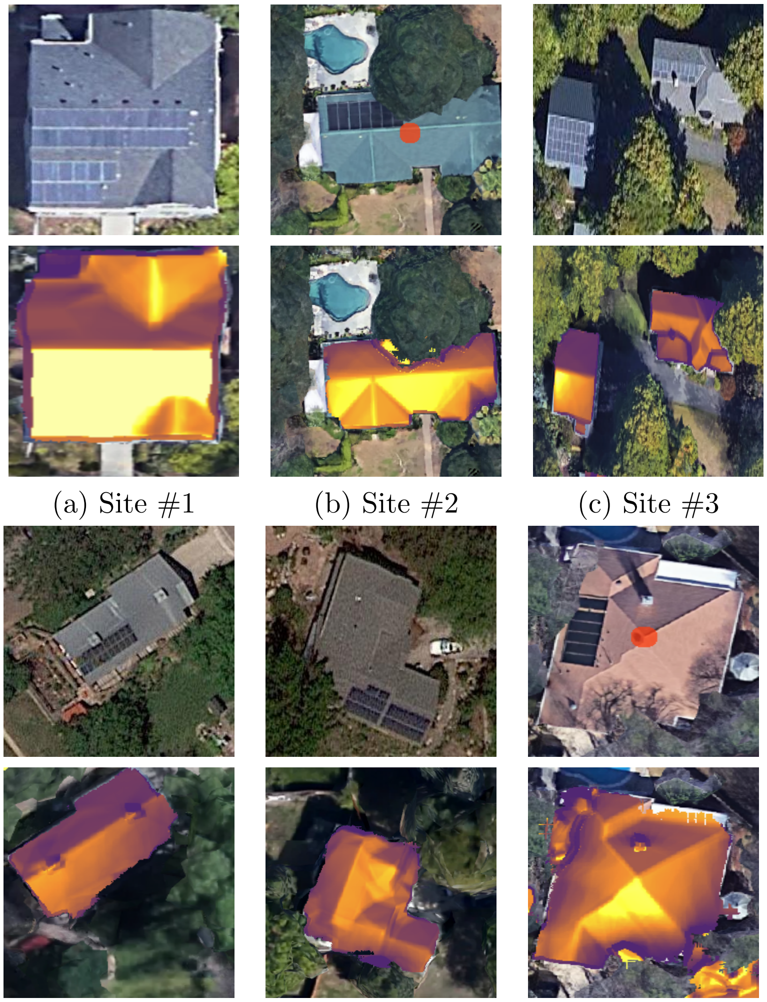
We compare our approach with multiple other approaches to black-box solar performance modeling. We implement a *pure physical* approach using the physical model in section 3.1 including the Kasten-Czeplak cloud cover model. Note that we did not implement either PVlib model, since they would result in significantly worse accuracy than Kasten-Czeplak based on Figure 3.7. We also implement the *pure ML* approach described at the beginning of Section 3.1, which uses all 10 weather metrics and each day's sunset and sunrise times as input features for training, and solar intensity as its output variable [63]. In addition, we implement a *hybrid ML* approach from prior work, which is similar to the pure ML approach, but, instead of solar intensity, uses the same normalized solar output ratio as our physical model in Section 3.1 for its output variable. The hybrid ML model also adjusts its output variable using the NOCT temperature and Kasten-Czeplak cloud cover models, and thus removes them as input features for ML training [32]. Finally, we implement another performance model that uses visible *satellite* imagery to estimate solar output instead of weather data. Geostationary satellites provide visible images of cloud cover every 15 minutes for nearly the entire world. The Heliosat family of algorithms [42] analyze these images to estimate the effect of cloud cover on ground-level GHI. To estimate solar output, our satellite model uses the physical model from Section 3.1 to estimate a site's maximum solar output (including the temperature adjustment), and then multiplies this value by the GHI ratio estimate from satellite imagery in lieu of using the cloud cover equation. Since we derive our physical model empirically, we label it as *empirical.* Note that our model is equivalent to the pure physical model above, but using our improved cloud cover equation. Finally, we also evaluate our physical model after enhancing it with ML, as described in Section 3.4.2, which we label as *empirical ML*.

We train all of the ML-based performance models on 5 years of solar generation data for each site using 20-fold cross-validation with a 70-30% split of training data to test data. For a fair comparison of accuracy, we calibrate the model of maximum solar output for the pure physical, satellite, and our empirical physical model using the same training data. We quantify model accuracy using the Mean Absolute Percentage Error (MAPE) (shown in Equation 2.1) between the ground truth solar power ($S(t)$) and the solar energy estimated by each model ($P_s(t)$) at each time t in our test set, which spans 13,140 hours for each solar site. Lower MAPEs have higher accuracy with 0% being a perfect model. We only evaluate MAPE between sunrise and sunset.

3.4.4 Experimental Evaluation

We evaluate and compare the accuracy of our solar performance model with the other models described in Section 3.4.3 on data from 100 solar sites at different locations with widely different physical and shading characteristics. Of course, the accuracy of the models varies across these sites. To better understand the attributes that affect model accuracy, we first examine in-depth the 6 homes pictured in Figure 3.8. This figure shows both a photograph from a satellite and from Google's Project Sunroof [10], which leverages LIDAR data to estimate a site's solar potential based on the roof tilt, orientation, and surrounding shading. Brighter colors indicate more solar potential. As the figure shows, some sites, such as (a), have little shade and are ideally positioned for solar generation, while other sites, such as (f), have non-ideal orientations and significant shading. We order the solar sites left to right from least shaded to most shaded.

Figure 3.9 then shows the accuracy in MAPE for all the performance models for each of these 6 solar sites. We order the performance models left to right from least accurate



(a) Site #1 (b) Site #2 (c) Site #3

(d) Site #4 (e) Site #5 (f) Site #6

Figure 3.8: Satellite images (top) and Google Sunroof images (bottom) depicting 6 illustrative solar sites and their shading level. The site-specific shading level increases from left (a), with 0% shade, to right (f), with 60% shade.

to most accurate. The top graph shows the MAPE over the entire day, while the middle graph shows the MAPE over just the middle of the day (10am - 3pm), where shading has the least effect, and the bottom graph shows the MAPE over just the beginning and end of each day (Sunrise - 10am and 3pm - Sunset), where shading has the most effect. All the graphs show the impact of the inaccurate Kasten-Czeplak cloud cover model (Equation 3.4) on the pure physical performance model, as it has by far the largest MAPE across all sites.

Surprisingly, the satellite-based performance model is the next least accurate across all sites and time periods. This likely demonstrates the limitations of using visible satellite imagery to estimate ground-level GHI. These limitations were also evident in the inaccuracy of

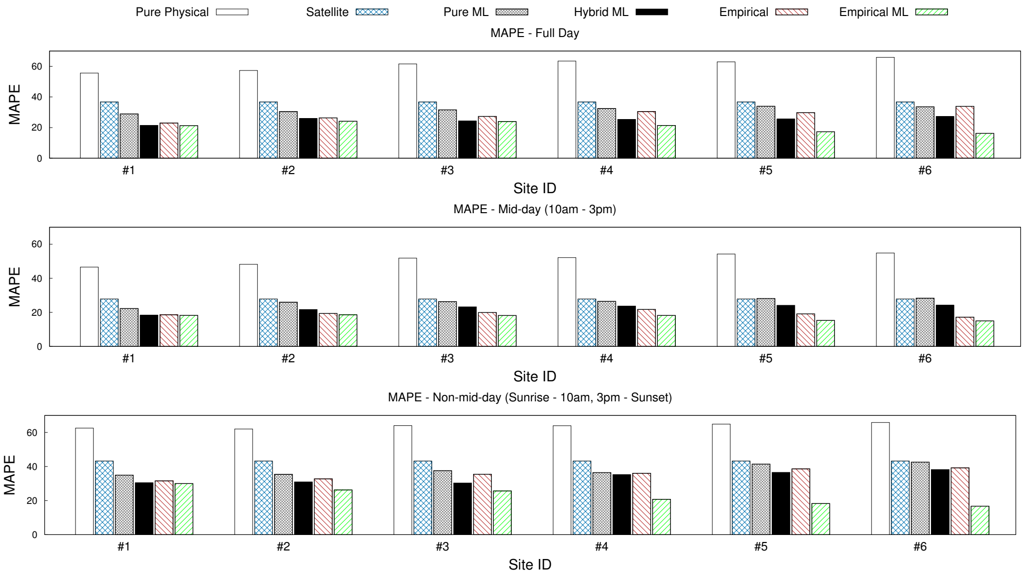


Figure 3.9: MAPE for 6 different black-box solar performance modeling techniques for the 6 solar sites in Figure 3.8.

Figure 3.4(bottom). Ultimately, visible satellite imagery can only detect the reflectivity of the tops of clouds, and cannot asses their thickness and the amount of solar radiation that ultimately reaches the ground. In addition, the Heliosat algorithm requires an accurate estimate of ground reflectivity as a baseline, which varies substantially across locations, seasons, and time periods. For example, the presence of snow dramatically changes the baseline ground reflectivity. As a result, snow detection is critically important for accurate GHI estimates from satellite imagery. In addition, the ground reflectivity changes throughout the day based on the solar angles, and are less accurate at sunrise and sunset as the pictures become darker, making it more difficult to distinguish between the ground and the clouds. Finally, the current generation of satellites only has a resolution of 10×10km, which introduces imprecision by averaging cloud cover effects across a wide area.

Both the pure ML and the hybrid ML performance models are more accurate than the satellite-based model even though they use coarse measurements of cloud cover using oktas. However, while coarse, in contrast to satellite imagery, these measurements are taken at ground level near the solar site. In addition, Weather Underground and other websites process data from multiple weather stations, such that the reported cloud cover often derives not from a single measurement but from multiple independent measurements. For all sites and time periods, the hybrid ML model has a slightly higher accuracy than the

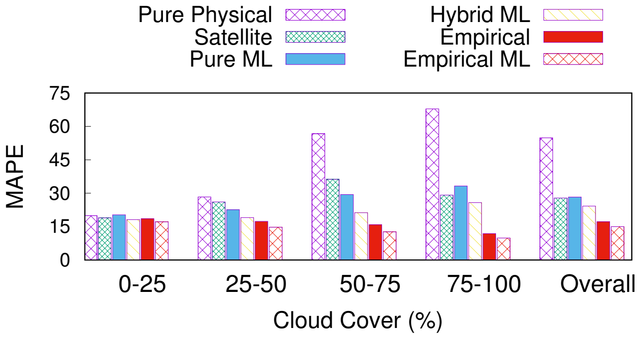


Figure 3.10: MAPE for mid-day solar generation during different weather conditions for solar site #3 from Figure 3.9.

pure ML model, likely because it uses feature engineering based on the physical models in section 3.2.2 before training its ML model. Both the pure ML and hybrid ML models have decreased accuracy at the beginning and end of each day, where shading exhibits a greater effect, compared to the middle of the day. This indicates that both ML models have issues learning shading. The pure physical and satellite models have similar issues, as they also exhibit a lower accuracy at the beginning and end of each day.

Our empirical physical model, which is the same as the pure physical model but with an improved cloud cover equation, substantially increases the accuracy of the pure physical model. In all cases, our empirical model is also more accurate than the satellite model (even though it uses imprecise okta measurements for cloud cover) and the pure ML model (even though it does not incorporate shading effects). In contrast, the hybrid ML model is slightly more accurate than our empirical physical model over a full day, likely because it both incorporates some physical models and indirectly accounts for shading effects during its training. However, over mid-day, where shading effects are minimal, our empirical physical model, which requires much less data for calibration, has equal or better accuracy across all sites. Further, our enhanced empirical ML physical model, which uses ML to account for unique site-specific shading effects, substantially improves the empirical physical model's accuracy. This improvement in accuracy increases as the site-specific shading effects increase from left to right, such that our empirical ML model reduces MAPE by ∼2× for the most shaded site #6. Not surprisingly, the accuracy of our empirical ML model is similar to that of our empirical model over mid-day, where shading effects are minimal, but is significantly better at the beginning and end of each day, especially as site shading increases.

The only difference between the pure physical model and our empirical physical model is the cloud cover model. To illustrate this, Figure 3.10 shows a breakdown of accuracy based on the percentage of cloud cover for site #3. The pure physical model's accuracy becomes steadily worse as the cloud cover increases, due to increasing inaccuracy in the Kasten-Czeplak model, while our empirical model accuracy is consistent. The figure also shows how the empirical ML model improves upon the empirical model. Under minimal cloud cover, all the models exhibit similar accuracy (∼15%). We suspect that some of the inaccuracy derives from okta and satellite measurement error and not model error, as indicated in Figure 3.4(right) and Figure 3.7(right), respectively. Imprecise measurements ultimately bound the accuracy of solar performance modeling.

Finally, Figure 3.9 shows results for all of the models across 100 rooftop solar deployments at different locations with various climates and shading levels. From top to bottom, the graphs show the pure physical, satellite pure ML, hybrid ML, empirical, and empirical ML models described earlier. Note that the y-axis range is [0-80] with a dotted line at 20 as visual reference point for comparison. Since space constraints prevent us from including pictures of all 100 sites, we manually divide the deployments into different general shading levels and group them together. Within each shading level, we order sites based on their average cloud cover, such that less cloudier sites within a group have a lower ID (and are on the left). In general, our results from 100 sites echo our results from 6 sites. Interestingly, in all models except for empirical ML, the average accuracy slightly decreases as the shade increases. In contrast, the accuracy of the empirical ML model are more consistent across all shade levels, and even slightly better for sites with higher levels of shade. Overall, across all 100 sites, the average MAPE of our empirical ML model was 20.7%, or 18% and 49% better than the average MAPE of the state-of-the-art hybrid ML model (25.3%) and the satellite model (40.6%), respectively.

3.5 Applications and Related Work

While there is significant prior work on ML-based solar modeling, most of it is in the context of solar forecasting, as detailed in recent surveys [23, 87, 54] that cite well over one hundred papers on the topic. Unfortunately, this prior work generally conflates modeling

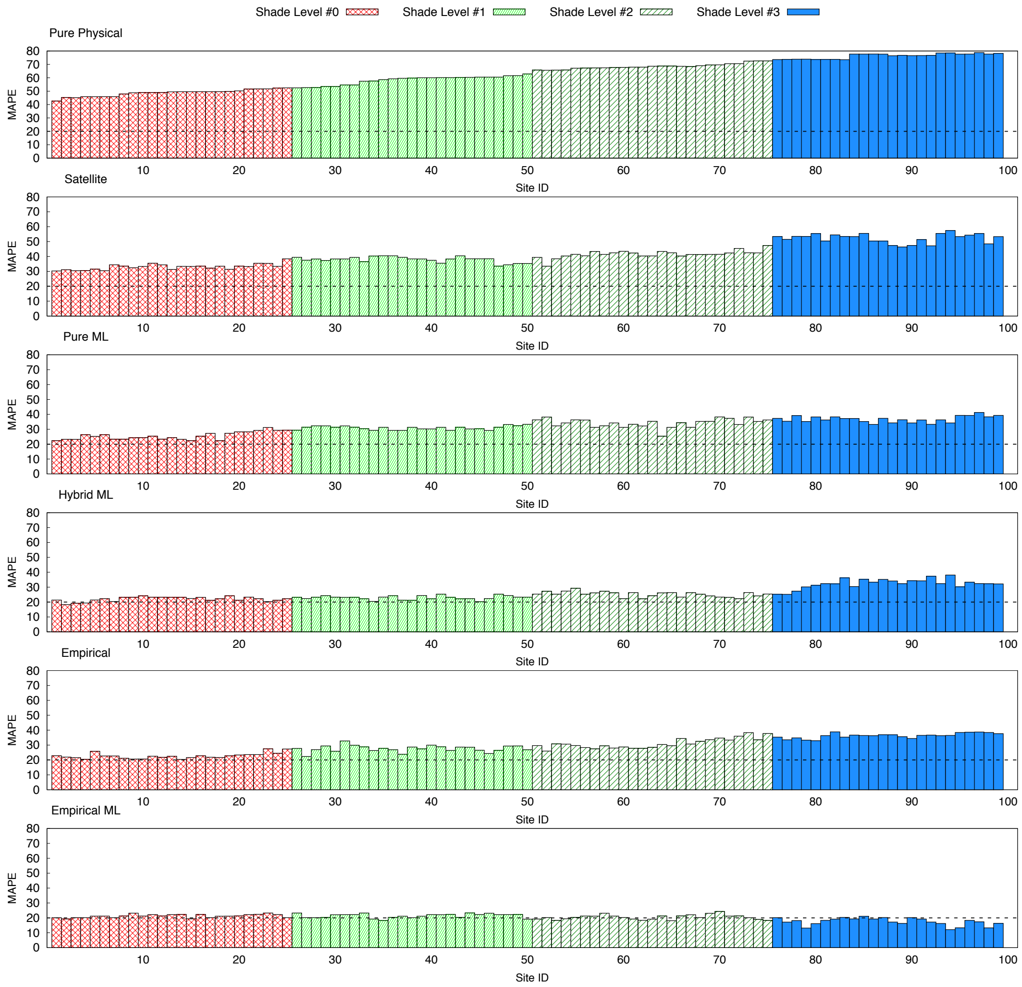


Figure 3.11: MAPE of 6 solar performance models across 100 solar sites over 1.5 years. Sites are grouped by their shade level.

and forecasting, and thus does not evaluate them separately. In addition, these forecasting approaches often implicitly embed assumptions about their specific problem variant, such as its temporal horizon, temporal resolution, spatial horizon, i.e., forecasting one solar deployment versus many deployments, spatial resolution, performance metrics, weather data, and deployment characteristics. These variants are generally not relevant to solar modeling, which simply estimates solar output (at some resolution) given a set of known conditions, e.g., the location, weather, and time. As a result, extracting a solar performance model from prior work on ML-based forecasting is non-trivial. Thus, for our pure ML-based technique we instead adapt a technique originally proposed for solar disaggregation, which focuses on separating solar generation from aggregate energy data that also includes consumption [63]. However, instead of applying the technique to disaggregate such “net meter" data, we use it to model pure solar data. The technique has been patented by Bidgely, Inc. [65] and is in production use [49].

As discussed in section 3.2.2, prior work on physical modeling generally takes a white-box approach [38, 26]. Our approach to black-box physical modeling is similar to these white-box approaches, in that it uses the same well-known physical models, but instead of directly measuring the necessary input parameters for a deployment, we infer them by searching for values that best fit the available data.

Our approach combines the best aspects of white-box modeling, which leverage physical models of solar generation based on fundamental properties but requires manually derive model parameters *a priori*, with black-box modeling, which automatically learns model parameters using ML but requires months-to-years of training data to learn accurate parameters. We compare with numerous other black-box ML approaches in section3.4.4. We plan to compare our approach with PVlib's white-box modeling as part of future work [22, 14]. However, we expect PVlib to be less accurate under cloudy skies given the inaccuracy of its cloud cover models. Solar performance modeling is also a foundation for many solar analytics.

Solar Monitoring. Our solar performance model enables indirect solar monitoring if the sensors directly monitoring power generation fail, by simply replacing the sensor data with the model output. We can also easily adapt our performance model to enable us to infer a site's solar output from other nearby sites if weather data is not available, as in prior work [40]. In this case, we can simply multiply the normalized solar output ratio from a nearby site by our model's estimate of a site's maximum power generation.

Solar Forecasting. Our solar performance model also enables solar forecasting by providing as input a future time, and forecast for temperature and cloud cover at that time. Weather Underground forecasts temperature and cloud cover each hour over the next 240 hours. Recent research focuses on black-box ML approaches to solar forecasting similar to ours [46, 75, 30, 87, 77]. These techniques are generally off-the-shelf, do not incorporate physical models, and require months-to-years of training data to learn accurate models. Unfortunately, prior work often conflates weather forecast error, model error, and measurement error, which makes it difficult to isolate the accuracy of forecast models independently of weather forecast error. As future work, we plan to apply our performance model to solar forecasting, compare its accuracy with existing black-box ML approaches, and quantify the weather forecast, modeling, and measurement error. We expect our model to perform as well or better than existing forecast techniques, since we expect much of its error derives from imprecise okta measurements.

Solar Fault Detection. Our solar performance model enables anomaly and fault detection if its accuracy deviates from its expected accuracy. In addition, a recent approach uses a sophisticated black-box ML algorithm for detecting anomalies by analyzing and comparing with the solar output of nearby sites [45]. Our solar performance model enables a similar function by simply comparing the normalized solar output ratios of nearby sites after adjusting for shading effects. Since these ratios should be equal across sites experiencing the same weather, any divergence signals an anomaly.

Solar Disaggregation. Prior work on solar disaggregation, which separates solar generation from energy consumption in combined net meter data, implicitly leverages the insight that the effect of weather on solar generation is universal [63, 33]. That is, these approaches learn black-box ML models that infer solar output on labeled data from one set of solar sites for training, and then assume they can use these models to accurately infer solar output at other sites, where raw solar data is not available. To disaggregate, these approaches simply subtract the inferred solar output from the net meter data to infer energy consumption. This paper empirically verifies this assumed universal weather effect across data from tens of thousands of solar sites, demonstrates that temperature and cloud cover are the only weather metrics that affect solar output, and derives an improved equation between cloud cover and GHI.

Solar Localization. Our current approach requires a site's location to compute the clear sky GHI and determine the local weather conditions. Prior work assumes the location of solar data is unknown, and instead determines the location from anonymous data based on its unique solar signature [36] or weather [34] signature. We could apply these techniques to automatically determine a site's location, although accurately estimating a site's location may increase our approach's data requirements. However, integrating automated solar localization would enable accurate performance modeling directly from solar data without any input from the user.

3.6 Conclusion

This chapter surveys and compares different approaches to black-box solar performance modeling. We compare a pure ML model from prior work [63, 65], a black-box physical model based on well-known relationships in solar generation [33], and a configurable hybrid approach that combines the benefits of both by achieving the most accurate results with little historical data. Our results motivate using physical models when relationships are well-known, and leveraging ML to quantify the effect of unknown relationships.

Due to inaccuracy of the black-box physical modeling, we then conduct a big data analytics to determine the primary source to the inaccuracy and then develop a physical black-box solar performance model, which requires minimal data for calibration, based on fundamental properties of solar generation. In particular, we leverage a large-scale data analysis across tens of thousands of years of solar and weather data in aggregate to i) demonstrate that weather's effect on a site's normalized solar output ratio is universal across all solar sites, ii) show the only weather metrics that affect solar output are temperature and cloud cover, and iii) derive a new physical model to quantify cloud cover's effect on solar generation. We show that our physical black-box model yields similar accuracy as state-of-the-art black-box ML approaches. We then enhance our physical model with a ML model that learns each site's unique shading effects. We show that our enhanced model has significantly better accuracy than state-of-the-art ML models, as well as a model based on GHI estimates from visible satellite imagery.

CHAPTER 4
SOLAR DISAGGREGATION

As discussed in Chapter 2 and Chapter 3, solar modeling is enabling utilities to better monitor, predict, and response to the variations of solar power in electric grid. Unfortunately, these models require historical solar generation data for training that is often not available. To address this problem, we design SunDance, a “black-box” technique for accurately disaggregating solar generation from net meter data that requires only a building’s location and a minimal amount of historical net meter data, e.g., as few as two datapoints.

4.1 Motivation

Due to its increasing importance in grid operations, numerous prior and ongoing work focuses on accurately forecasting the grid’s solar generation [24, 37, 44, 46, 52, 58, 76, 77]. While many of these forecast models offer coarse grid-level predictions of net load, recent work increasingly focuses on automatically generating customized forecast models via machine learning for each solar deployment based on its unique characteristics [46, 76]. These models can then be combined to generate a more accurate fine-grained grid-level forecast of solar generation and net load. Importantly, these custom solar forecasting techniques leverage supervised machine learning (ML) technique: they use a site’s historical solar generation as training data to automatically learn a model that maps weather metrics to solar output at each time interval. The models then use standard forecasts of these weather metrics as input, e.g., from the National Weather Service (NWS), to predict future solar output.

Thus, the key to constructing sophisticated forecast models is access to historical solar generation data for training. Utilities are rapidly installing advanced or “smart” meters, which record energy flow at fine-grained intervals ranging from five minutes to every hour, that can provide such historical data. Smart meter installations are estimated to hit 70M by the start of 2017 and 90M by 2020 [47]. Thus, ample training data from smart meters is typically available for large residential solar deployments ($\geq$10kW) and solar farms, as these deployments are often required to be independently metered. As a result, these deployments' meter data represents pure solar data. However, nearly all small-scale residential solar deployments (<10kW) are "behind the meter" (BTM), such that the smart meter data exposed to utilities represents only the net of a building's solar generation and its energy consumption. Thus, constructing the forecast models above for BTM solar is not possible, as there is no pure solar data available for model training.

To address the problem, we present a new system, called SunDance, that accurately separates (or "disaggregates") a building's net meter data into its solar generation and energy consumption.[1] Importantly, SunDance employs a "black box" technique that requires *no training data* from the building itself, i.e., historical data separated into solar generation and energy usage, and instead only requires a minimal amount of net meter data and a location, both of which are available to utilities. In lieu of training data, SunDance leverages multiple insights into fundamental relationships between location, weather, physical characteristics, and solar irradiance. In particular, SunDance combines two key insights.

Clear Sky Generation Model. Our first insight is that it is possible to build an accurate customized model of each solar deployment's maximum "clear sky" generation potential based on fundamental relationships between the Sun, the Earth, and a deployment's location and custom physical characteristics, *even when using noisy net meter data that combines solar generation with significant energy consumption.*

Universal Weather-Solar Effect. Our second related insight is that the same weather conditions reduce the maximum clear sky solar irradiance potential by the same percentage regardless of the magnitude of this solar irradiance, which is a well-known weather-independent function of time at each location. This property, which we call the *Universal Weather-Solar Effect*, enables SunDance to i) build a general model using supervised machine learning that maps weather metrics to the expected fraction of the maximum solar

[1] Note that solar disaggregation differs from energy disaggregation [25], as it only separates out solar generation from energy data and not appliance-specific consumption.

irradiance potential for locations where solar training data is available, and then ii) apply that model to accurately infer solar generation at other locations, *where solar training data is not available.*

SunDance combines the insights above to develop an accurate customized model of a solar deployment's maximum solar generation potential using only its noisy net meter data and location, and then determines the fraction of this maximum generation the deployment actually produces by using a general model of weather's fundamental impact on the maximum solar irradiance potential. In developing SunDance, we make the following contributions.

Solar Background. We discuss in detail the fundamental physical relationships that govern solar generation over time based on location, position of the Sun, physical characteristics, weather, temperature, etc., and provide empirical evidence for SunDance's key insights above. These relationships dictate each location's unique solar signature, and the impact of weather on solar generation.

SunDance Design. We present SunDance's solar disaggregation technique summarized above. To construct a customized model of a solar deployment's maximum clear sky generation, SunDance searches for a valid solar signature that represents the tightest strict upper bound on the noisy net meter data. SunDance then learns a general model that captures the Universal Weather-Solar Effect at all location(s) where solar training is available. SunDance then combines these models to disaggregate a location's net meter data.

Implementation and Evaluation. We implement SunDance and evaluate it on net meter data from 100 buildings. We show that SunDance's accuracy, in terms of its Mean Absolute Percentage Error (MAPE), when inferring solar generation *without access to any solar training data from the buildings under test* is comparable to the accuracy of a customized machine learning model built with complete access to a building's historical solar data for training.

4.2 Solar Background

SunDance assumes access to average power data $P_{net}(t)$ from a building smart meter, which represents the sum of solar power generation $P_s(t)$ and energy consumption $P_c(t)$, as shown below, where $P_s(t) \geq 0$ and $P_c(t) \leq 0$.

$$P_{net}(t) = P_s(t) + P_c(t), \forall t > 0 \tag{4.1}$$

Given only $P_{net}(t)$ and the meter's location, SunDance's task is to infer $P_s(t)$ and $P_c(t)$ at each time t.[2] Below, we provide a brief background on the fundamental relationships that determine i) the maximum amount of solar irradiance that reaches the Earth's surface at any time at any location, ii) the physical characteristics of solar cells that dictate how much of this irradiance is converted to electrical power under ideal weather conditions, and iii) the impact of non-ideal weather conditions. We identify insights based on these fundamental relationships in designing SunDance.

Computing Clear Sky Irradiance and Effect of Physical Characteristics. SunDance uses the models in section 3.2.2.1 to compute the clear sky irradiance for a specific solar deployment. While SunDance is compatible with any of these models, our implementation in this chapter uses a simple model that requires only a location's latitude and longitude and the Sun's position, which is a function of location and time. Thus, given k, β, and ϕ, we can compute a solar module's maximum power generation potential P_{smax} in clear skies at any location at any time by setting $I_{incident} = I_{total}$ from 3.2.2.1, as the other parameters are a function of location and time. In 6.3, we show how SunDance infers k, β, and ϕ from net meter data.

While module size, efficiency, tilt, and orientation have the largest impact on solar module output, other physical effects also exist that are not precisely modeled by the closed-form equation above. For example, a module's operating voltage affects its efficiency based on a solar module's IV curve. In this chapter, we assume solar modules always operate at their maximum power point using standard tracking algorithms. In addition, while both i) multiple solar modules with different placements that are wired together (either in series

[2] Note that this time t also includes the day and month of the year.

or parallel) and ii) modules that track the Sun by changing their tilt and orientation also permit similar closed-form models, they are more complex. We focus use simple models, which apply to the vast majority of solar deployments, and leave extending them to more complex deployments as future work.

Efficiency Effects. The relationships in section 3.2.2.2 model the energy a solar module generates at any location at any time in ideal weather, e.g., under clear skies at an optimal temperature. Of course, non-ideal weather conditions can reduce both solar module efficiency and the amount of solar irradiance that reaches the ground. As we discuss in section 3.2.2.3, the ambient temperature has a significant effect on solar module efficiency, while other weather conditions, such as clouds and humidity, affect the solar irradiance that reaches the ground. We do not model the effect of other weather metrics on efficiency, as these effects are typically not significant [62].

Irradiance Effects. A key insight of our work in section 3.4.1 , which we will leverage in the next section, is that the *exact same weather should have the same proportionate affect on the maximum solar irradiance potential I_{total} that reaches the ground, regardless of its magnitude,* which varies widely over time time at different locations. That is, if two different locations A and B experience the *exact same weather conditions* at two different times then the solar irradiance that reaches the ground $I_{incident}$ will be $c * I_{total}^{A}$ at location A and $c * I_{total}^{B}$ at location B, where c is a constant based on the weather and I_{total} is the maximum clear sky solar irradiance at those locations at those times. We call this the *Universal Weather-Solar Effect*, and, as we show, it is key to SunDance's approach.

4.3 SunDance Design

Given only a solar-powered building's net energy meter data and its location, SunDance disaggregates the data into the two separate components in Equation 4.1: the building's solar generation $P_s(t)$ and its energy consumption $P_c(t)$. SunDance's design includes three key steps, which we summarize below, before detailing each.

1. Build a Custom Model of Maximum Solar Generation.

SunDance uses historical net energy meter data to build a *custom model* of a solar deployment's maximum clear sky solar generation potential at any given time based on its location. This model incorporates each deployment's unique physical characteristics from the previous section, and its temperature effects, but focuses narrowly on modeling maximum generation potential and thus *does not* model any other weather-related effects, e.g., due to clouds, humidity, precipitation, etc. SunDance builds this model by finding the valid solar curve dictated by the fundamental relationships in the previous section that best "fits" the data. Since this model focuses narrowly on maximum generation potential, it is possible to build an accurate model using only noisy net meter data.

2. Build a General Model of Weather's Effect on Irradiance.

Separately, SunDance builds a general model that maps multiple weather metrics to the expected percentage reduction in clear sky solar irradiance potential. Due to the Universal Weather-Solar Effect, this model is general and can be built using solar training data from any (or multiple) locations, but then applied to accurately quantify the effect of weather on the clear sky solar irradiance potential at other locations, where such training data is not available.

3. Apply the Two Models Above to Disaggregate Solar Power.

Given the two models above, disaggregating net energy meter data is trivial. SunDance first uses weather data for the location as input to its general weather model to infer the percentage reduction in maximum clear sky solar irradiance potential. SunDance then applies this percentage reduction to the deployment's maximum solar generation, which is computed using the custom model in step one, to infer the absolute amount of solar generation $P_s(t)$ at each time t. Finally, to complete the disaggregation, SunDance subtracts this solar generation $P_s(t)$ from the net meter data $P_{net}(t)$ to yield the energy consumption $P_c(t)$ at the same time t.

4.3.1 Building a Maximum Generation Model

Inferring Physical Characteristics. The relationships in Section 3.2.2.1 and Section 3.2.2.2 enable us to define a range of valid solar curves at any location, which dictate the shape of maximum clear sky solar generation potential over each day of the year based on a deployment's physical characteristics, e.g., ϕ, k, and β from Equation 3.1. Examples of these solar curves are depicted in Figure 3.2.

SunDance builds a maximum clear sky generation model for a solar deployment by finding the ϕ, k, and β that defines the valid solar curve that best "fits" the location's energy data. We first discuss building this model for pure solar generation data and then describe how to translate it to net meter data that combines solar generation and energy consumption. Even pure solar generation data is stochastic, exhibiting many rapid variations in power due to changing weather conditions that diverge from its maximum power. For example, Figure 4.2(a) depicts solar generation on a partially cloudy day for a 10kW residential solar deployment, where output dips in the morning. Since generation deviates from its maximum in the morning, finding the valid solar curve that simply minimizes the Root Mean Squared Error (RMSE) with the data is not appropriate: the non-ideal weather will *always* result in fitting a solar curve that is lower than the maximum clear sky solar generation.

As a result, SunDance instead finds the best fit valid solar curve that represents the tightest upper bound on the data, since we know that the observed solar generation should never exceed the maximum clear sky generation. That is, among the valid solar curves that are equal to or greater than all datapoints, we find the one that minimizes the RMSE with the data. As a result, the curve SunDance finds will be dictated entirely by *the single datapoint that experiences the highest percentage of its maximum generation potential.* Thus, even if a day is cloudy, if there is even one datapoint that is near the maximum generation, this datapoint will dictate the best fit for the entire day (since the best fit must be a strict upper bound on the data). For example, even on the cloudy day in Figure 4.2(a), the best fit curve closely matches the ground truth maximum solar generation (which we approximate using the next day's solar generation under a clear sky), since it is dictated by the points in the day that are sunny. SunDance can apply this approach to multi-day time periods where the likelihood of a deployment experiencing its maximum generation at one or more datapoints is high.

SunDance must search for the ϕ, k, and β from Equation 3.1 to find the best fit. This search is challenging since the parameters are dependent, e.g., modifying the tilt changes the effect of orientation, and conducting a brute force search across the entire parameter space is too computationally expensive. However, searching the entire parameter space appears necessary, as the tilt β and size and efficiency k have a similar effect on generation, which can lead to finding local maxima in isolated areas of the parameter space. For example, in one part of the parameter space, we may find a best fit curve that has a high tilt and low k, whereas the actual deployment has a high k and low tilt.

To address this problem, we observe that, while the physical characteristics of solar deployments are not always ideal, installers generally attempt to make them as ideal as possible. As a result, SunDance is able to accurately estimate a starting condition for its search based on the ideal physical characteristics to ensure it starts in the "right" region of the parameter space. In particular, the ideal orientation angle is south-facing in the northern hemisphere (and north-facing in the southern hemisphere) with a tilt angle equal to the latitude. Given these starting conditions, SunDance conducts an iterative search that first finds the value of k that best fits the data using a binary search, while keeping the other values constant. Given the new value of k, the search process then proceeds iteratively by next searching for the orientation that best fits the data using a binary search. After finding this orientation, we adjust the tilt in the same way. The search continues iteratively by adjusting each parameter in turn until they do not change significantly.

In practice, this approach efficiently finds tilt and orientation angles that are close to the ground truth tilt and orientation angles, since most solar deployments have physical characteristics near the ideal. For example, the ground truth tilt and orientation in Figure 4.2 are 35° and 190°, respectively, while the tilt and orientation angles SunDance finds in (a) when using pure solar data are 36° and 189°.

Modeling Temperature Effects. While the approach above finds a model with tilt and orientation angles that are close to the ground truth tilt and orientation angles, it is not accurate when applied over the entire year due to the effect of temperature on solar cell efficiency, which is captured by the k parameter. Since the best fit curve must be an upper bound on the data, this curve is dictated by the point that achieves the highest percentage of its maximum clear sky generation potential at the lowest temperature, which is the most efficient operating point for the solar cell. Thus, to adjust for these temperature effects, SunDance finds the datapoint that is closest to the initial upper bound solar curve found above and then finds the location's ambient temperature at that time to use as a

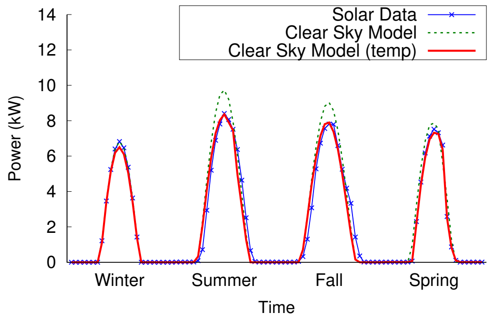


Figure 4.1: SunDance's maximum clear sky generation model both before and after adjusting for temperature effects.

baseline $T_{baseline}$. This datapoint represents the coldest time period that maximizes solar cell efficiency under clear skies. SunDance then applies a temperature adjustment to k in the model, as discussed in Section 3.2.2.3.

The temperature adjustment function reflects the constant factor c increase (or decrease) in efficiency when the ambient temperature is below or above the baseline temperature. Here, T_{air} is the location's ambient temperature at time t. While a typical value of c is 0.5% for solar modules [78], SunDance searches for the precise value of c for each deployment that represents the tightest upper bound on the data. While efficiency is a linear function of cell temperature, and not ambient temperature, since the temperature adjustment function subtracts the current temperature from the baseline it cancels out the constant values in Equation 3.2.

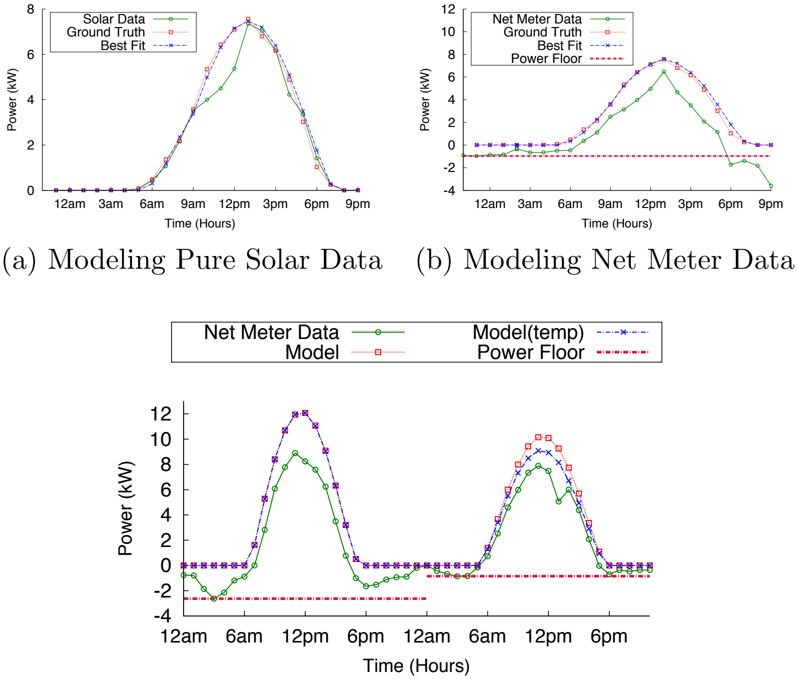
Figure 4.1 shows a maximum generation model for a sunny day in each season both before and after our temperature adjustment. Before the temperature adjustment, the model is highly accurate in January, since these cold weather days represent the most efficient operating points that dictate the upper bound, but is highly inaccurate in July when the temperature is 40C greater than on the coldest days. This is expected, since with a typical solar module, a 40C increase in temperature decreases the efficiency (and maximum generation potential) by $40 * 0.5\%$=20%. After the temperature adjustment, the maximum generation model closely matches the generation on these sunny days. In this case, the factor c we found was 0.57%, which is near the typical value of 0.5%.

Modeling using Net Meter Data. The discussion above builds a model of maximum clear sky solar generation using pure solar generation data. Modeling the maximum solar generation using net meter data differs in two key respects, which require simple extensions to our methodology above.

First, adding energy consumption introduces additional "consumption noise" to pure solar data that causes it to deviate from its maximum generation. Of course, this consumption-induced noise has the same effect as variable non-ideal weather conditions in decreasing the recorded generation. However, as discussed above, our modeling approach above is robust to non-ideal weather conditions, since the best fit must be a strict upper bound on the data, which is dictated the datapoint(s) that are closest to the maximum generation potential. This logic also applies to the non-ideal "weather" created by adding energy consumption: as long as datapoints exist where solar generation is near its maximum potential and energy consumption is low, e.g., when a home is unoccupied on a sunny day, our best fit upper bound model will be dictated by these few datapoints. This insight enables us to model a deployment's maximum solar generation even on noisy net meter data, where we cannot directly model the actual (disaggregated) solar generation.

Second, energy consumption in modern buildings generally never drops to zero. Thus, SunDance must estimate a building's minimum power consumption floor for the datapoint(s) above that dictate the model. To do so, SunDance simply uses the minimum power consumption at night, when solar generation is guaranteed to be zero. SunDance subtracts this power consumption floor from the data before constructing its model, where the minimum nightly consumption in the adjusted net meter data is zero. Figure 4.2(b) shows our model built using net meter data from the same deployment and time as in Figure 4.2(a). Note that, with (negative) consumption included, the net meter data is strictly less than the solar model; the figure also highlights the power floor SunDance uses to adjust the net meter data. Recall that the model in (a) finds a tilt of 36°, an orientation of 189°, a k of 10.6, and a c of 0.57%. Our model in (b) using the net meter data is similar, finding a tilt of 34°, an orientation of 186°, a k of 10.9, and a c of 0.72%.

Historical Data Requirements. SunDance requires remarkably little data to construct an accurate custom model of solar generation. In the limit, our approach needs only two



(a) Modeling Pure Solar Data (b) Modeling Net Meter Data

(c) Modeling using Minimal Data

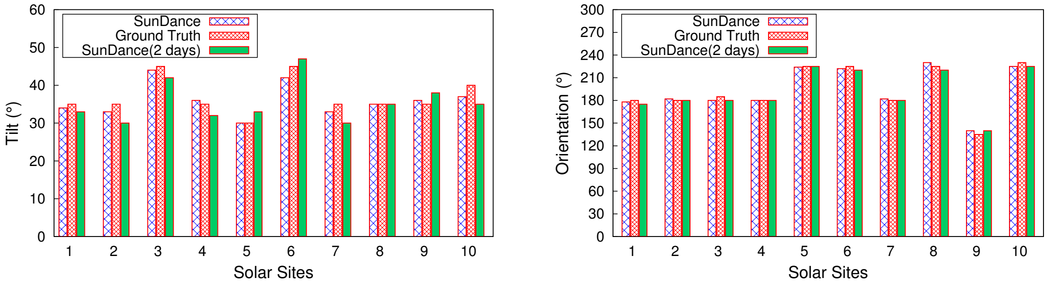
Figure 4.2: SunDance's maximum clear sky generation model when built on pure solar data (a), net meter data (b), and on net meter data using historical data from only two days (c). In contrast, both (a) and (b) represent the best fit over one year of data.

datapoints during clear skies with low energy consumption, such that there is a significant temperature difference between the two points. Since the model finds the tightest upper bound on the available data, it is entirely dictated by the single point of maximum net generation (or, equivalently, the minimum net consumption). An additional point is needed at a different temperature to model the effect of temperature on efficiency. To illustrate, the models in Figures 4.2(a) and (b) were built by finding the tightest upper bound across an entire year of training data. In contrast, Figure 4.2(c) shows our model (with and without a temperature adjustment) using two sunny days in January on net meter data. Using only these two days, instead of a year, SunDance finds similar model parameters, with a tilt of 36°, an orientation of 185°, $k = 11.6$, and $c = 0.69\%$.

Table 4.1 summarizes the model parameters found on the different datasets and model variants in Figure 4.2. In all cases, the tilt, orientation, and c SunDance finds are close to the ground truth. In addition, the k value, which is the product of a module's size and

	Days	*Tilt*	*Orientation*	*k*	*Area* (m^2)	*c*
Ground Truth	NA	35°	190°	12.3	48.88	NA
Pure Solar	365	36°	189°	10.6	48.18	0.57
Net Meter	365	36°	188°	10.7	48.63	0.58
Net Meter (Temp)	365	34°	186°	10.9	49.55	0.72
Net Meter (Temp)	2	36°	185°	11.6	52.73	0.69

Table 4.1: The model parameters SunDance finds in the Figure 4.2 variants are all similar to the ground truth parameters.



(a) Tilt Angle

(b) Orientation Angle

Figure 4.3: SunDance is able to find tilt and orientation angles near the ground truth from as little as two days of noisy net meter data.

efficiency at the baseline temperature, is also accurate. In this case, the module is 48.88m^2. While we cannot separate a module's size and efficiency, if we assume a typical commercial module with ∼22% efficiency at 25C, SunDance can estimate the module size. As the table shows, SunDance finds sizes close to the ground truth for this module. While we evaluated SunDance across 100 buildings, we were only able to verify ground truth tilt and orientation angles of buildings that were clearly visible from Google street view data. Figure 4.3 shows that using noisy net meter data from 10 buildings where we could manually verify the ground truth tilt and orientation angles, SunDance consistently finds angles that are near the ground truth even when using only two days of data.

4.3.2 Building a General Weather Model

The models described above are highly customized to each deployment, incorporating its unique orientation, tilt, size, efficiency, and temperature effects. In contrast, our weather model, which leverages the Universal Weather-Solar Effect from Section 3.4.1, is general and thus applies to *any solar deployment*. As a result, our weather modeling is a one-time exercise that can use any solar irradiance (or solar power data) from any location.

We construct our weather models similar to prior work on solar power forecasting models that use supervised machine learning [46, 76]. These approaches train a model based on labeled data that associates standard weather metrics, such as sky condition, temperature, humidity, dew point, precipitation, etc., with a deployment's solar generation.

Thus, the output of these existing models is a deployment's absolute solar generation, which is not general, but instead custom to each deployment's unique physical characteristics, particular its size. In contrast, SunDance generalizes these models by changing the output to be the fraction of maximum solar irradiance potential that reaches the ground. Based on the Universal Weather-Solar Effect, this approach can include solar irradiance data from *many* locations (and many times) to use for training a single general model. In addition, since pyranometer deployments, which record solar irradiance, are rare, SunDance can equivalently use any pure solar generation data (adjusted for temperature effects) that is available to build these models. Based on Equation 3.1, when dividing a deployment's solar output by its maximum solar generation potential, the factors based on the physical deployment characteristics cancel out, such that the resulting ratio is equivalent to the ratio of observed solar irradiance to maximum solar irradiance potential.

Importantly, our insight above means that *we can build a general weather model using pure solar power data from one (or many) deployments where it is available, and then use that model to accurately infer the reduction in solar power from its maximum potential at other solar deployments, where pure solar power data is not available.* This is a significant insight not only for our work on solar disaggregation, but also for work on solar forecasting based on pure solar data. For example, recent work highlights the importance of reducing the amount of training data necessary to build custom solar forecast models, especially for new solar deployments coming online [46]. However, based on the insight above, our general weather model *requires zero training data* from a new solar deployment under test. In addition, as discussed above, we can build an accurate maximum generation model using as few as two datapoints. In contrast, prior work requires from months [46] to years [76] of historical data to construct an accurate model.

Prior work has evaluated a wide range of supervised machine learning techniques for modeling the effect of weather on solar output, including least squares regression, Support

Vector Machines (SVMs) using different kernel functions, and deep neural nets. While we evaluate different modeling techniques in Section 4.5, SunDance is orthogonal to the specific machine learning technique. SunDance's contribution instead lies in identifying the Universal Weather-Solar Effect and designing input and output features to leverage it to build a general weather model. We use the weather metrics from Weather Underground as input features to our model, including temperature, humidity, dew point, barometric pressure, precipitation, and sky condition. We map the qualitative descriptions for sky condition, e.g., scattered clouds, sunny, etc. to a numerical percentage of cloud cover using the mapping suggested by the NWS. More precise numerical percentages, which would improve model accuracy, can be derived from infrared satellite imagery.

4.3.3 Disaggregating Net Meter Data

Given the two models above, solar disaggregation is trivial. For each datapoint in the net meter data, we use the weather metrics at that time as input to our general weather model above to infer the fraction of its maximum generation a solar deployment will output. To infer a building's actual solar generation $P_s(t)$, we then multiply this fraction by the maximum solar generation we infer based on our customized model in Section 4.3.1 at that time. We then simply subtract our inferred solar generation from the building's net meter data $P_{net}(t)$ to compute the corresponding energy consumption $P_c(t)$.

One limitation of SunDance's current modeling approach is that it does not adjust for dynamic local conditions that reduce solar output and are not reflected in the set of weather metrics, such as shade from nearby buildings and trees or dust buildup on the solar modules. Thus, a decrease in solar generation due to these factors will be incorrectly associated with increased energy consumption. However, since these factors typically have a minimal impact on generation [62], they also generally have a minimal impact on SunDance's accuracy. In addition, it is possible to adjust for such dynamic conditions by adjusting our maximum model above based on solar output during clear sky weather periods. If solar output is significantly less than the maximum generation model (after the temperature adjustments) during these periods where weather has no effect, then we can infer that it is due to a dynamic factor. Of course, correcting for such factors may require much more historical data, as we can only use solar data during clear skies. We leave extending our model to account for dynamic factors to future work.

4.4 Implementation

We implement SunDance using a mixture of python and C++. We use simple well-known geometric formulas to compute a location's clear sky solar irradiance based on its latitude, longitude, elevation, time, and the Sun's position in the sky. To derive the Sun's position in the sky, we use the PSA algorithm, which takes as input the UTC time (to the second) and a location's latitude and longitude and outputs the Sun's precise azimuth and zenith angles [29]. High performance implementations of the PSA algorithm are publicly available that are accurate to within 0.0083° of the Sun's true position. We then compute the AM relative to an AM of 1 when the Sun is 90° overhead based on the well-known formula below.

$$AM = \frac{1}{\cos(\Theta) + 0.50572(96.07995 - \Theta)^{-1.6364}} \tag{4.2}$$

Given the AM above, we estimate the direct solar irradiance I_{direct} that reaches the ground using the Laue Model [55] as follows, where h is the location's elevation above sea level and 1.361 kW/m^2 represents the solar constant.

$$I_{direct} = 1.361 * [(1 - 0.14 * h)0.7^{AM^{0.678}} + 0.14 * h] \tag{4.3}$$

In addition, while variable, the amount of diffuse irradiance that is scattered by the atmosphere is generally estimated at ~10% of the direct irradiance on a clear day. Thus, we compute the total solar irradiance I_{total} from 3.2.2.1 at any location as follows.

$$I_{total} = 1.1 * I_{direct} \tag{4.4}$$

Note that there are also packages available that implement other clear sky solar irradiance models, including PySolar [3] and NREL's library that implements the Bird model [1]. We leave evaluating SunDance's accuracy across these different models as future work.

Given a location's latitude and longitude, our implementation fetches historical weather data at one-hour granularity using Weather Underground's API. Since Weather Underground only has one-hour weather data, we can only disaggregate net meter data at the granularity of an hour. However, SunDance's approach is general and can be applied to weather and energy data at any granularity. We use the *scikit-learn* machine learning library in python to build our general weather model. The library supports multiple techniques including Support Vector Machines with different kernel functions and multiple linear regression models. We also use NumPy and Pandas for weather and energy data processing.

4.5 Experimental Evaluation

We first evaluate SunDance's accuracy across 100 solar-powered buildings using one year of hour-level interval energy data. We then focus on a representative "net zero" building to understand the effect of energy consumption patterns, weather, and time on SunDance's accuracy. To quantify accuracy, we compute the Mean Absolute Percentage Error (MAPE) described in Equation 2.1.

We also compute the MAPE between the actual and inferred energy consumption using the same approach. We restrict all time periods to between sunrise and sunset, since SunDance is always perfectly accurate at night, as solar generation is always zero. Even so, MAPE is highly sensitive to periods of low absolute solar generation. For example, if sunrise falls near the end of an hour, the absolute generation of a 10kW solar deployment over the hour may only be 50W. If SunDance infers a generation of 100W, its MAPE for that period will be 100%. In contrast, the absolute generation during a cloudy mid-day period may be 5kW, such that if SunDance infers a generation of 6kW, its MAPE is only 20%. Thus, the absolute error of 50W contributes much more to the average MAPE than the absolute error of 1kW. To put our results in better context, we report overall MAPEs, as well as MAPEs for separate time periods and under different weather conditions. In particular, as in solar forecasting, we prioritize accuracy during cloudy periods in the middle parts of the day, where a significant amount of solar generation may fluctuate.

4.5.1 Comparing with a Supervised Approach

We compare SunDance's black-box approach to a fully supervised machine learning approach that has access to an entire year of historical solar generation and energy consumption data that has already been separated. In this case, the supervised approach works

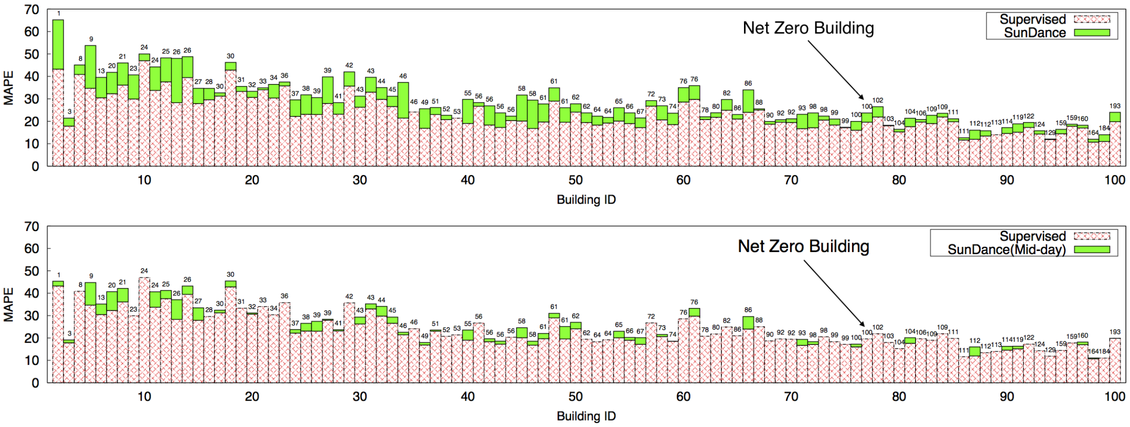


Figure 4.4: Daytime (top) and mid-day (bottom) MAPE for solar disaggregation using a supervised approach and SunDance for 100 buildings over a year. Buildings sorted by their ratio of solar generation to energy consumption (listed atop each bar).

exactly like SunDance, except that, instead of our general weather model, we build a supervised model using the custom solar training data from each specific solar site. Thus, unlike other machine learning approaches [76, 63, 27, 46], our supervised approach incorporates the same physical solar models as SunDance. In recent work, we have show that this supervised approach is significantly more accurate than existing supervised approaches that do not incorporate physical solar models [32]. Our supervised approach represents a lower bound on the MAPE (and an upper bound on the accuracy) that SunDance can expect. As in prior work, we use a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel for our supervised approach [76, 63, 27]. SVM-RBF is common in solar modeling, since it attempts to fit a Gaussian curve to solar data and solar profiles are similar to Gaussian curves.

Figure 4.4 compares SunDance's accuracy with that of the supervised approach for each of the 100 buildings. In the graph, each stacked bar represents a building, such that the lower bar is the MAPE of a supervised approach, and the upper bar represents the increase in MAPE when using SunDance. The graph shows that across all buildings, the increase for SunDance in MAPE is generally small relative to the supervised approach. This result suggests the accuracy of Universal Weather-Solar Effect, as the only difference between the two approaches is data used for training.

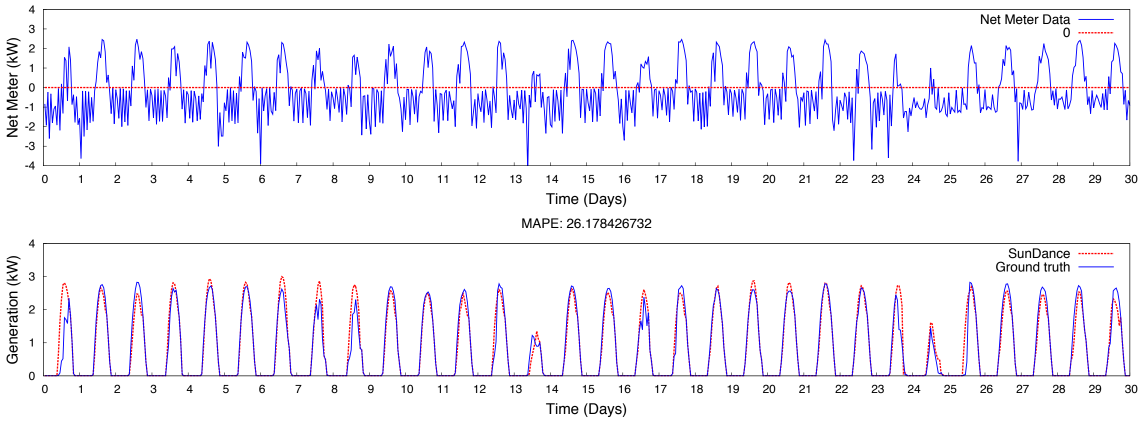


Figure 4.5: One month of net meter data (top) and ground truth and inferred solar generation (bottom) from a net zero building.

In addition, the buildings are sorted by their ratio of solar energy generation to energy consumption, which is listed at the top of each bar. As the graph shows, the MAPE is partially a function of this ratio, such that a higher ratio generally yields more accurate results. This is intuitive, as increased energy consumption represents additional "noise" that SunDance must filter out. Note that a ratio of 100% represents a "net zero" building that has equal solar generation and energy consumption. Here, we see average MAPEs of ∼22% for SunDance on net zero buildings. Much of the imprecision derives from the absolute error in estimating each building's energy floor, which essentially requires an informed guess.

We also plot the same graph but only for the middle hours of the day (11am-3pm) to reduce the effect of small absolute errors that yield large percentage errors at the start and end of each day. This graph shows that MAPEs reduce by ∼22% during these important periods to an average of ∼17% for a net zero building. In addition, in many cases for the mid-day results, SunDance performs as well as the supervised approach, in large part, because any small absolute error in the energy consumption floor has less effect on the MAPE as the absolute solar generation increases.

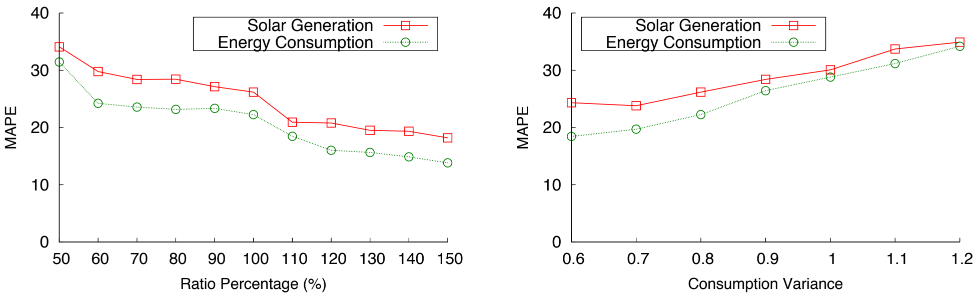
Result: *SunDance's black-box approach achieves similar accuracy without access to any solar training data from a site as a fully supervised approach with complete access to such training data.*

4.5.2 Quantifying SunDance's Accuracy

We next evaluate the different conditions that affect SunDance's accuracy on a representative net zero building (labeled in Figure 4.4). To provide a qualitative sense of SunDance's accuracy, Figure 4.5 shows the raw net meter data (top), as well as the ground truth and disaggregated solar generation (bottom). The figure shows that SunDance's inferred solar generation closely matches the ground truth solar generation, despite the stochasticity in the net meter data. In this case, the MAPE for solar generation is ∼26%, while the MAPE for energy consumption is ∼22%. The inferred energy consumption MAPE is typically lower because it is less affected by low absolute values, e.g., in the morning and evening.

We also examine the effect of changing both the ratio of solar generation to energy consumption and altering the variance of the energy consumption. In this case, to change the ratio, we alter the building's energy consumption at each time by a constant factor to increase and decrease the ratio. Similarly, we alter the variance by scaling the difference in energy consumption between two time periods by a constant factor, such that a value of 0 results in a completely flat consumption that never changes from the initial value. In both cases, the alterations produce a new set of net meter data, which we feed to SunDance for disaggregation. Figure 4.6 shows the results. As expected, as the ratio increases (a), and there is more solar generation to consumption, we see a linear decrease in MAPE (and corresponding increase in accuracy). Similarly, a low variance in consumption enables SunDance to more accurately model the solar generation and energy consumption floor (even if the ratio is large). Thus, in (b) we see a linear decrease in accuracy (and increase in MAPE) as the energy consumption variance increases.

We also break down our results based on weather conditions and time. Figure 4.7 breaks down accuracy based on weather conditions. In this case, we capture weather based on the percentage of the maximum generation a solar deployment is producing at any given time. Thus, if a solar deployment is only generating between 0% and 25% of its maximum clear sky potential, we assume that the weather is not good. Figure 4.7 shows that, as expected, our MAPE improves as the weather conditions improve. Importantly, for weather conditions that result in a ratio greater than 25%, SunDance yields near the same accuracy, indicating it performs well even under highly adverse weather conditions. While the MAPE is quite



(a) Generation:Consumption Ratio (b) Energy Consumption Variance

Figure 4.6: Higher ratios of generation to consumption result in higher disaggregation accuracy (a). Lower variances in consumption result in higher disaggregation accuracy (b).

high during the worst weather conditions, this is largely due to small absolute errors from low generation that result in large percentage errors. To quantify this effect, we also plot the MAPE of energy consumption. Since this is a net zero building, we see that the small absolute errors in inferred solar generation during the worst weather conditions have little on effect on the energy consumption MAPE, which has similar accuracy across all weather. **Result:** *SunDance accuracy is a linear function of the ratio of solar generation to energy usage and the variance in the energy usage. SunDance has the highest accuracy during the most critical period: adverse weather where solar generation is difficult to infer.*

4.6 Related Work

The most similar work to SunDance is a recent approach for solar disaggregation (and product) from Bidgely, Inc. [63, 65]. Similar to SunDance, Bidgely trains a machine learning model (also using an SVM-RBF kernel) that maps weather metrics to normalized solar output on data from a set instrumented deployments, and then applies that model to estimate solar generation on a separate set of deployments. However, while SunDance normalizes solar output by constructing a maximum generation model based on the underlying physical characteristics of the deployment, Bidgely normalizes using a static value representing the maximum capacity of each deployment. As a result, Bidgely's model is significantly less accurate than SunDance's model, as we show in recent work [32]. Prior work also performs solar disaggregation by using data from microsynchrophasors at the feeder-level [50]. This approach differs in that it requires data from grid-level sensors.

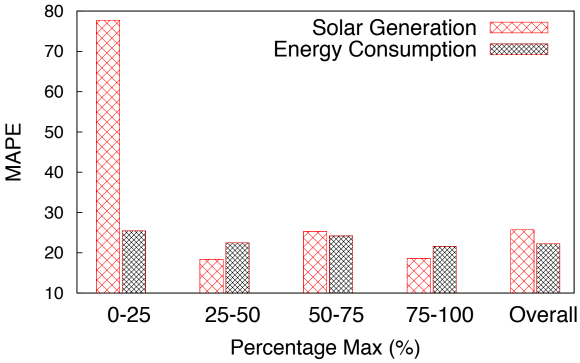


Figure 4.7: Solar generation and energy consumption MAPE during different weather conditions.

SunDance has many commonalities with existing solar forecast models based on machine learning [30, 46, 76]. However, these techniques use pure solar data to train their models, while SunDance focuses on disaggregating solar data from net meter data. SunDance's weather model uses a similar machine learning approach as prior forecasting techniques, except that its output is a fraction of the maximum clear sky generation, which varies over time at each location. This model is general due to the Universal Weather-Solar Effect. As a result, unlike prior forecasting approaches, SunDance requires *no training data* from the location under test to accurately model weather's effect on solar output.

Prior work on solar forecasting in SolarCast also performs feature engineering to reduce the amount training data necessary to build an accurate model that maps weather to solar output [46]. Similar to SunDance, SolarCast leverages the relationship between clear sky solar irradiance and solar output to build a single model of weather-to-solar output by normalizing its training data across time, e.g., by multiplying weather metrics by the clear sky irradiance. However, unlike SunDance, SolarCast's models are custom for each site, and not general, as their output is expressed in terms of raw solar power.

Finally, recent work shows how to extract the location where "anonymous" solar energy data was generated [36]. SunDance suggests the same approach can extract the location of anonymous net meter data that includes solar generation by first disaggregating the solar energy data. The potential to extract location from net meter data has serious privacy implications.

4.7 Conclusion

In this chapter, we design SunDance, a new black-box technique for disaggregating BTM solar generation from net meter data. Importantly, SunDance requires only a deployment's location and a minimal amount of historical net meter data, e.g., as few as two datapoints. SunDance then leverages multiple insights into well-known fundamental relationships between location, weather, physical characteristics, and solar generation to build an accurate model of a deployment's solar generation. We implement SunDance and evaluate it on 100 buildings. Our evaluation shows that SunDance's black-box approach achieves similar accuracy without access to any solar training data from a deployment, as a fully supervised approach that has complete access to historical solar training data. SunDance also enables a wide range of NILM related works that disaggregate total energy consumption into the energy readings of individual loads over a period of time for a home, as SunDance separates energy consumption from net meter as well.

CHAPTER 5
SOLAR-BASED LOCALIZATION

The energy data produced by solar-powered homes is considered “anonymous” if it is not associated with identifying account information, e.g., a name and address. Thus, these energy data is often not handled securely, and even made publicly available over the Internet. Our key insight is that solar energy data is not anonymous: since every location on Earth has a unique solar signature, it embeds detailed location information. To explore the severity and extent of this privacy threat, in this chapter, we design SunSpot to localize “anonymous” solar-powered homes using their solar energy data.

5.1 Background and Motivation

Utilities and third-parties monitor the energy produced by solar-powered homes using networked energy meters, which record and transmit energy data at fine-grained intervals. Such energy data is generally considered anonymous if it is not associated with identifying account information, e.g., a name and address. Thus, energy data from these “anonymous” solar-powered homes is often not treated as sensitive: instead, it is routinely transmitted over the Internet in plaintext, stored unencrypted in the cloud, shared with third-party energy analytics companies, and even made publicly available.

For example, Figure 5.1 shows a screenshot of 1Hz energy data an anonymous solar-powered home has made publicly available on the Internet via a networked energy meter, such as the TED, eGauge, BrulTech, or Enphase Envoy. These meters connect to the Internet and upload energy data to the cloud in real time, where it is then stored to enable queries on archival data. Solar installers typically add networked meters to enable homeowners to monitor energy generation and consumption via web dashboards or smartphone applications. For simplicity, in many cases as in Figure 5.1, accessing the data does not require a password, as there is an assumption the data is anonymous and cannot be associated

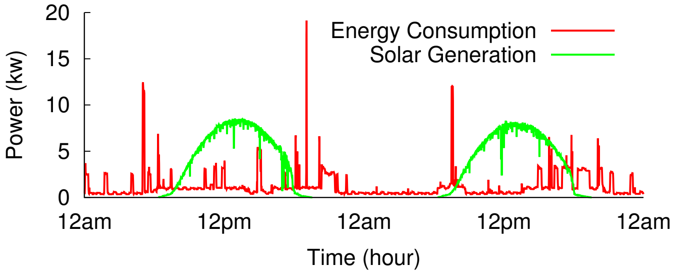


Figure 5.1: Example data from solar-powered home that is making its 1Hz solar generation and energy usage publicly available on the the Internet under the assumption of anonymity.

with a particular home. The example in Figure 5.1 is from one of the 28,000 anonymous homes we have found uploading solar generation and energy consumption data to the public Internet. While the example makes the data publicly available for simplicity, similar data is also being intentionally gathered and released by various research institutions to support energy analytics research. As above, these datasets often include detailed solar and energy usage data from thousands of volunteer anonymous homes.

While users may choose to not install (or securely configure) the meters above, they are *forced* to allow utilities to monitor their energy usage. In addition, to receive reimbursements for solar generation, some states also require users to upload their utility energy data to an external database managed by a third party [71]. This energy data is becoming increasingly detailed, as utilities employ "smart" meters that record energy usage at fine-grained intervals. Current smart meters monitor energy usage on the order of minutes [47] with next-generation meters expected to monitor on the order of seconds [83]. In the U.S., utilities have deployed >70 million smart meters [48], and are rapidly accumulating smart meter data, which they may permanently archive for later analysis.

A plethora of startups have now arisen to analyze these vast archives of utility energy data, ostensibly to make energy-efficiency recommendations [28, 13, 70]. Prior research has demonstrated the ability to learn a variety of insights into private user behavior by analyzing their energy data [66]. For example, energy data indirectly leaks occupancy [31, 53], which may reveal whether a home's occupants: i) include a stay-at-home spouse, ii) keep regular working hours and daily routines, iii) frequently go on vacation, or iv) regularly eat out for dinner. Energy data can also reveal load power signatures—changes in power unique to a device—for specific appliance brands and models. These behavioral insights and appliance

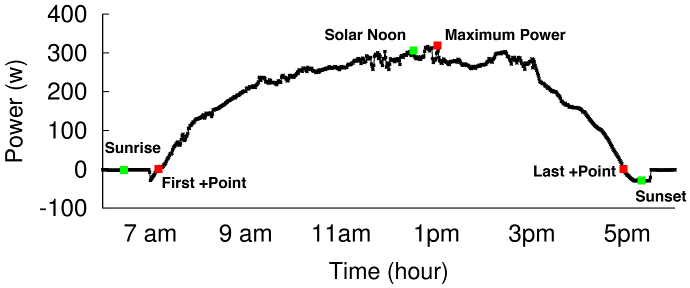


Figure 5.2: The start, stop, and peak of solar generation (red) approximates the time of sunrise, sunset, and solar noon (green).

details are valuable to companies in profiling homes and directing advertising campaigns, and may also be exploited by tech-savvy criminals. Thus, some contend that energy data will eventually be worth more than the energy consumed to generate it [68].

Users and utilities commonly provide energy data to the energy analytics companies above under the assumption the data is anonymous. In many cases, users do not realize their energy data leaks side-channel information. Utilities typically anonymize any energy data they share with third-parties by removing account names and addresses, as suggested by the U.S. Department of Energy's recently released Voluntary Code of Conduct (VCC) for managing user energy data [86]. Importantly, the VCC *does not* require user consent to release anonymized energy data with names and addresses stripped. Consent is likely not required because the energy analytics above do not reveal location, which prevents third-parties from associating private behavior above with a specific home.

Our key insight is that solar energy data is not anonymous: since every location on Earth has a unique solar signature, e.g., a unique sunrise, sunset, and solar noon time, it embeds detailed location information. While there is substantial prior work on estimating solar energy output based on a home's location, we know of no work that does the reverse—estimating the location based on solar output. The localization threat means home energy data that includes solar generation is never anonymous.

As one example of this threat, an attacker could determine when to burglarize the anonymous home in Figure 5.1 by first determining its occupancy pattern from its consumption data (in red) using existing techniques [31, 53], and then analyzing its solar signature (in green) to determine the home's location. As a result, users and utilities should treat such data as highly sensitive by, in particular, not making it publicly available on the Internet or releasing it to third-parties without user consent. To explore the severity and extent of this privacy threat, we design SunSpot, a system for localizing an anonymous solar-powered home by analyzing its solar energy data. Exposing and evaluating this threat is critically important in informing evolving policies by DOE and others for managing anonymous energy data, and in emphasizing to users and utilities the need to securely handle energy datasets that include solar generation. In doing so, we makes the following contributions.

Localization Challenges. We highlight numerous challenges to localization from solar energy data, as a solar module is a highly imprecise sensor for tracking the sun. Solar energy data is affected by numerous unknown variables, including a home's local climate, e.g., frequency of cloud cover and temperature variations, physical characteristics, e.g., tilt/orientation, topography, shading from nearby structures, etc., and properties of the electrical system, e.g., variations in grid voltage, choice of wiring and inverter(s), etc.

SunSpot Design. We design SunSpot, which localizes a solar-powered home to a small region by exploiting multiple insights dervied from the regularity in the Earth's orbit. We leverage crowd-sourced image processing on publicly-available satellite data to identify potential homes in the area with visible solar modules. SunSpot significantly reduces the search area by filtering out areas without man-made structures, and may then apply additional filters, e.g., by matching solar output to module size or local weather patterns, to hone in on a specific solar-powered home.

Implementation and Evaluation. We implement and evaluate SunSpot on publicly-available energy data at both per-second and per-minute resolution from 14 solar-powered homes. We find that SunSpot localizes a solar-powered home to near the smallest possible region given the energy data resolution, e.g., within a ∼500m and ∼28km radius for per-second and per-minute resolution, respectively. SunSpot then leverages Amazon's Mechanical Turk at a cost of \$13.60/km^2 to identify a specific home, after reducing the search space by filtering out regions without man-made structures, which eliminates on average >97% of the search area in the U.S.

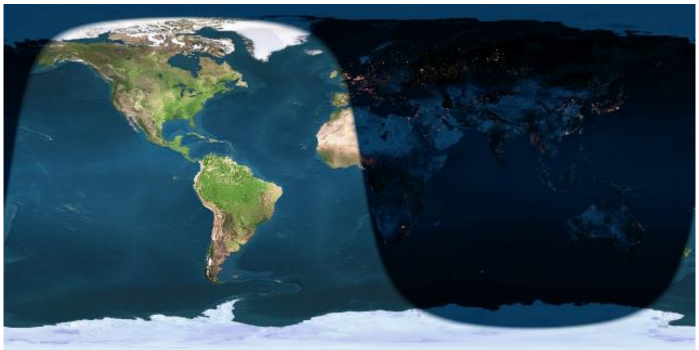


Figure 5.3: Sunlight map of the Earth.

5.2 Localization Challenges

SunSpot assumes an anonymous solar-powered home at an unknown location equipped with a networked energy meter that monitors energy generation over time. Given this solar energy data, SunSpot's objective is to infer a location—a latitude and longitude—where the data originates. Note that we focus exclusively on localizing the source of solar energy data, and not "net meter" data, which is the sum of a home's solar generation and energy consumption. Energy analytics companies have already developed solar disaggregation techniques, which analyze net meter data to separate solar data from consumption data [64], and are actively applying them to utility smart meter data [63]. We also designed a more accurate solar disaggregation system-SunDance in Chapter 5. Combing our solar disaggregation techniques, our localization techniques may be used to localize based on net meter data, and we leave it as future work. In addition, as discussed in Section 5.1, there are already thousands of solar-powered homes, including the home in Figure 5.1, that are separately exposing their solar generation and their energy consumption data.

The basic principle for localizing a solar-powered home from its solar energy data is straightforward. On a clear sunny data, solar generation data reveals a location's unique *solar signature*, which derives from the sun's position in the sky at a particular location and time and determines the amount of solar radiation that strikes the Earth. In particular, a location's unique solar signature dictates a unique time of sunrise, sunset, and solar noon (see Figure 5.2), which correspond to the times when a solar system's generation starts, stops, and peaks each day, respectively. SunSpot leverages this information to infer the location where solar energy data originates.

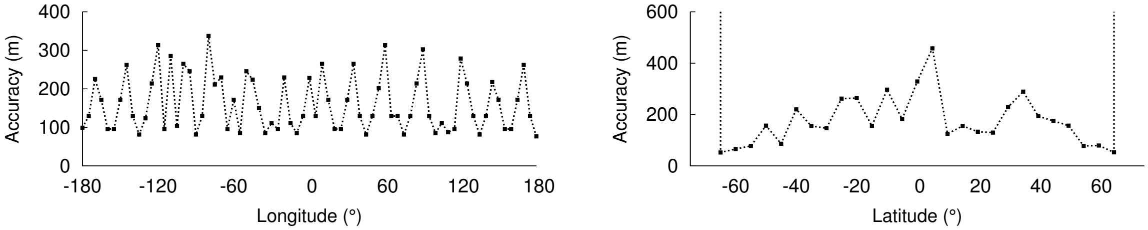
5.2.1 Deriving Location from the Sun

Given the sun's importance to life on Earth, astronomers can derive its movements with incredible precision. For example, the PSA algorithm [29] provides the sun's position, i.e., its azimuth and elevation angles, in the sky to within 0.0083° at any location, given its latitude and longitude, at any time of the year. Open-source code and online APIs are available that implement the sunrise/sunset algorithm, which provides precise sunrise and sunset times (to the second) given a location's latitude and longitude [80, 81].

Interestingly, while technically feasible, there are no commonly available open-source libraries or online APIs that perform the reverse operation, by computing a location from the sunrise and sunset times. Unfortunately, the PSA and the sunrise/sunset algorithm above are not reversible, since they both use trigonometric functions at multiple stages that are not one-to-one, i.e., their inverse yields multiple solutions. Instead, the algorithms for deriving location from sunrise/sunset events are much more obscure, as they are typically only used for celestial navigation of ships without electronic navigation [84]. Unlike the open-source code and online APIs above, these localization algorithms widely published in textbooks do not compensate for the slight irregularities in the Earth's shape and orbit that are required for high precision.

However, as a prerequisite to localizing solar energy data, SunSpot requires a precise algorithm for determining a location based on its sunrise and sunset times. Due to the issues above, rather than implement and refine published algorithms, we develop an approach that uses available APIs, which only work in the opposite direction by computing sunrise/sunset time given a latitude and longitude, as tools to conduct a binary search for a location. Note that in the chapter we use UTC time to eliminate time zone issues.

Deriving Latitude. To determine a location's latitude, we observe that all locations at the same latitude have the same daylength, i.e., the duration between sunrise and sunset, on each day. We also observe that, the daylength gets shorter the further north the latitude in the fall/winter, and gets longer the further north the latitude in the spring/summer. To illustrate, Figure 5.3 shows a sunlight map of the earth in the northern hemisphere's winter, where daylength becomes shorter moving from south to north. We leverage this insight to conduct a binary search to find a latitude that yields our desired daylength, given a sunrise



(a) Longitude Accuracy

(b) Latitude Accuracy

Figure 5.4: We accurately derive location from sunrise and sunset times using existing online APIs that perform the reverse operation.

and sunset time. That is, we pick any longitude value and then compute the daylength using the online APIs for −90°, 0°, and 90° latitude. We then select the region, either [−90°, 0°] or [0°, 90°], that includes the desired daylength. We then compute the daylength for the mid-point of that interval, and repeat the process. We terminate the search when the latitude computed at the next step does not significantly change.

Deriving Longitude. We perform a similar procedure to compute a location's longitude. Longitude is uniquely determined by the time of solar noon, when the sun is at its highest point in the sky, which is always the mid-point between the sunrise and sunset times. In this case, we pick any latitude and then compute solar noon using the online APIs for both 0° and ±180° longitude. We then select the region, either [0°,-180°] or [0°,180°], that includes our desired solar noon. As above, we compute solar noon for the mid-point of the selected region, either 90° or −90°, and repeat the process until the longitude computed at the next step does not change.

Note that, by searching based on daylength and solar noon, the two procedures above are independent of each other. That is, computing the longitude does not depend on knowing the latitude or vice versa. We evaluate our approach across the full range of latitudes and longitudes above using existing online APIs [81], as shown in Figure 5.4. Figure 5.4(a) shows that our derived longitude is always within 400m of the actual location's longitude. Longitude accuracy is a function of the latitude, such that higher latitudes enable higher accuracy at the same data resolution. The Earth's rotation speed decreases by the cosine of the latitude, such that the speed at X° latitude is 465 × cos X°m/s. As a result, the maximum precision possible at the equator with second-level data is 465m, and the accuracy

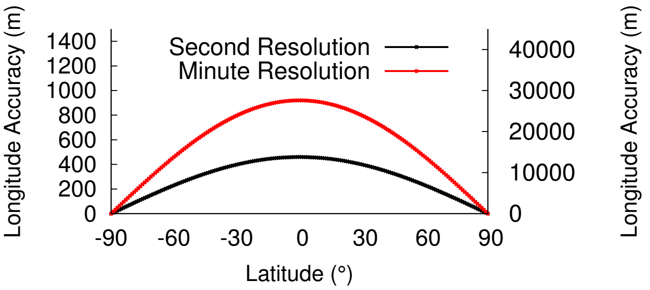


Figure 5.5: The precision possible for computing longitude from solar noon is a function of the latitude and the data resolution.

possible with minute-level data is 27.9km. Figure 5.5 plots the maximum longitude precision possible for second- and minute-level resolution data across all latitudes. To make both lines visible, we plot second-level data based on the left y-axis and minute-level data based on the right y-axis.

Similarly, Figure 5.4(b) shows that our computed latitude is always less than 500m from the actual location. The abrupt increases at ±66.56°[1] indicate regions near the poles where the sun does not rise or set. We ran this experiment on data from June 21st, 2015 (the summer solstice) where the half of the Earth lit by the sun is maximally misaligned with the poles. Note that the solstices represent the days that yield the most accurate results for latitude. Latitude accuracy changes over the course of the year, as shown in Figure 5.6. Since, on the equinoxes (September 22nd and March 20th), every location experiences 12 hours of daylight, it is impossible to distinguish a location's latitude from daylength.

5.2.2 Challenges to using Solar Energy Data

Our approach above derives a location from a known sunrise and sunset time—or equivalently the daylength and solar noon—on a particular day. A naïve approach to compute location from solar energy data is to simply use the times for the first and last positive solar generation of the day as the sunrise and the sunset times, respectively. SunSpot can use these times to directly estimate daylength and solar noon, and then provide these estimates as input into the algorithm above. However, this approach is inaccurate—on the order of hundreds to thousands of kilometers—because a solar system is a highly imprecise sensor for

[1] This latitude is equal to 90° minus the Earth's tilt of 23.44°.

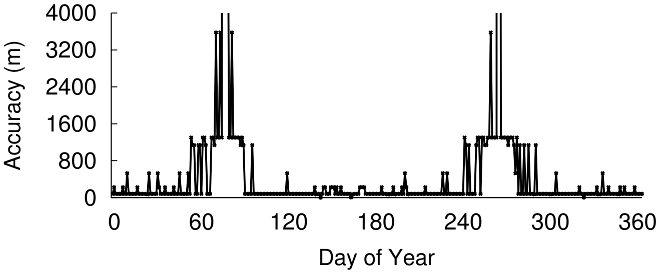


Figure 5.6: The accuracy of deriving latitude from daylength varies over the year and is least accurate near the equinoxes.

numerous reasons. For example, even a few minutes of inaccuracy in solar noon can yield massive errors, as each minute of error translates to roughly 27.9km error in longitude, as mentioned above. Figure 5.2 shows that even on a seemingly ideal day, sunrise, sunset, and solar noon often do not precisely align with the start, stop, and peak of solar generation, respectively. Below we describe the reasons for this error.

Atmospheric Conditions. Solar output depends on changing environmental conditions, namely solar irradiance. For a flat stationary solar deployment, the maximum solar irradiance (in W/m^2) is proportional to the sun's position in the sky. However, atmospheric effects alter the maximum solar irradiance. These effects include not only the presence of visible clouds, but also other conditions, including humidity, rain, dust, snow, pollution levels, etc. As a result, on a cloudy day, the start and stop of solar generation may be tens of minutes after and before sunrise and sunset, respectively.

Generation Inefficiency. Solar modules are not 100% efficient at converting solar radiation to power, but instead range in efficiency from 15-25%. Due to this inefficiency, even under ideal conditions with no clouds, the start and stop of solar generation each day will not precisely align with sunrise and sunset, as shown in Figure 5.2. Solar module efficiency also decreases as the temperature increases. Thus, the lag in detecting the first positive generation after sunrise (and the last positive generation before sunset) varies with temperature. Temperatures may vary significantly over the day (from morning to evening) and year (from winter to summer).

Shading from Nearby Objects. Sunrise and sunset times are derived assuming no topographical effects, i.e., the location and its surroundings are at sea level. This is only true in the middle of the ocean. In reality, the surrounding landscape dictates the horizon. For example, in a valley, the sun will rise from behind the mountains later and set behind them earlier than the official sunrise and sunset times. The opposite will occur at the top of a mountain. Solar deployments, especially on rooftops, are also often obstructed by nearby buildings and trees. The impact of these effects is not consistent, but will vary over time, e.g., when trees lose their leaves.

Physical Properties. The physical properties of a module, namely its tilt and orientation, also affect energy generation. The power output of a stationary deployment oriented toward the equator, e.g., south in the northern hemisphere and north in the southern hemisphere, is proportional to solar irradiance, which is a function of the sun's position in the sky. However, many deployments are not perfectly oriented toward the equator, and may also be tilted to varying degrees. The equation below computes solar output as a function of the sun's position in the sky, and modules' tilt and orientation.

$$S_p = S_i[\cos(\alpha)\sin(\beta)\cos(\psi - \Theta) + \sin(\alpha)\cos(\beta)] \tag{5.1}$$

Here, S_i is the intensity of solar radiation that strikes a flat module, while S_p is the amount of solar radiation that strikes an actual module, given the module's azimuth and tilt angles (ψ and β, respectively), as well as the sun's azimuth and elevation angles (Θ and α, respectively). Figure 5.7 graphically depicts how the orientation affects the output of a module (in the northern hemisphere). An ideal module oriented south ($\psi = 180°$) will experience its maximum production at solar noon (when the sun is at its highest point in the sky, maximizing solar radiation). In contrast, an ideal module with more of an eastward orientation will shift the generation curve earlier, such that its maximum production is earlier than solar noon. The more eastwardly the orientation, the earlier the maximum production point and the earlier the day's first and last generation times. A westward orientation has the opposite effect.

Unlike changes in the orientation, changes in the tilt do not affect either the time of maximum generation or the time of first or last positive generation. However, they do reduce the magnitude of the maximum generation and, thus, result in a more gradual rise and fall of the generation curve. Note that the relationships above dictate the output for a solar system where all modules are tilted and oriented in the same way. If there are multiple modules with different tilts and orientations their output is the sum of each module's individual production based on its own tilt and orientation (assuming each module has its own microinverter, as discussed below).

Electrical Characteristics. A solar deployment's electrical system also affects its output. For example, the output of modules wired in series is dictated by the individual module generating the least current. Each individual module's output is also dictated by its IV curve, which defines the amount of current (and power) a module generates at different voltages. The IV curve of a multi-module deployment connected to a single inverter is a complex function of the IV curves of all the modules and how they are wired, e.g., in series or parallel. As with module efficiency, the shape of the IV curve also varies with temperature and solar irradiance. Inverters actively vary their operating voltage to search for the maximum power point on this complex aggregate IV curve as conditions change, which may lead to periods of operation below the maximum power point. Rather than connect multiple modules to a single inverter, deployments may also attach a microinverter to each module. In this case, each microinverter independently optimizes its module's maximum power point. Thus, the same system using microinverters will generate a different energy profile than when using a single inverter.

Meter Accuracy. Ultimately, solar energy data derives from meters that sense its generation. These meters have varying levels of accuracy, typically ranging from 0.5% to 2%, depending on whether they are certified as utility- or consumer-grade. Energy meters are typically placed in front of the inverter and measure AC power. As a result, the power they record is a function of, not only the current generated by the modules, but also the grid's voltage. While RMS grid voltage in the U.S. is 120V, it may vary by ±5% based on current standards. Thus, recorded power generation will also vary in proportion to these voltage fluctuations.

5.2.3 Summary

The inaccuracy in solar energy data caused by the effects above varies across locations. For example, the power generated by a deployment in Southern California (which has

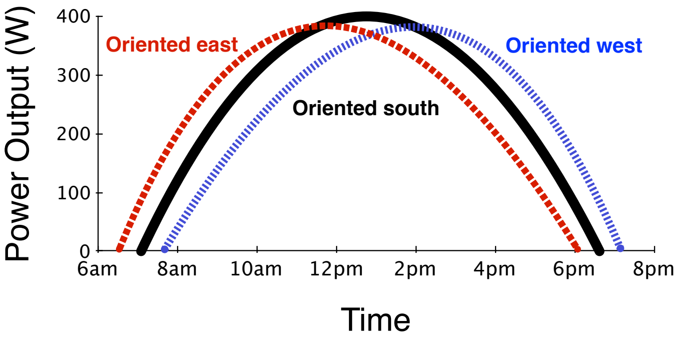


Figure 5.7: Depiction of how solar generation changes based on solar module orientation (in the northern hemisphere).

few temperature variations and cloudy days) that has few obstructions and all modules oriented towards the equator (with the same tilt) will more closely reflect the sun's path than a deployment in a location with a highly variable climate, many obstructions, multiple modules with different orientations and tilts, unstable grid voltage, etc. In essence, the more efficient a solar deployment is at generating power, the closer it tracks the sun's position in the sky, and the more susceptible it is to localization. This general principle—that the more energy-efficient a system, the more vulnerable it is to leaking information via energy data—has been observed in other contexts [31, 53].

5.3 SunSpot Design

The effects from the previous section are often significant—even for the most efficient deployment—and impossible to accurately model without knowing details of a solar installation, e.g., its location, tilt/orientation, wiring, etc. Thus, accurate localization from single day's solar data (or even a few days or weeks) is challenging, and impossible if the time period is near the equinox (since all locations have a similar daylength near the equinox). However, since utilities, third parties, and current networked energy meters have archives of solar energy data, as discussed in Section 5.1, SunSpot leverages data over multiple days to mitigate inaccuracy from any single day's data. Note that Sunspot *does not require many months of data*, and can operate on even a few weeks of data, as long it includes some clear sunny days. However, as we discuss, for inferring latitude, SunSpot does require data from a separate set of days in the fall/winter and the spring/summer. Of course, in general, data over a longer period increases both accuracy and confidence.

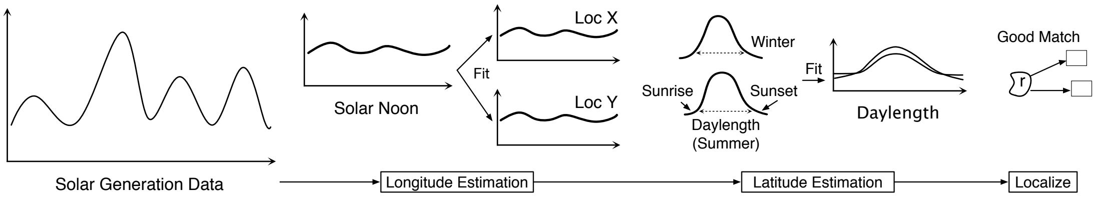


Figure 5.8: Pipeline of operations SunSpot uses to infer the location of a solar array from its energy data.

Similar to the approach in Section 2, SunSpot works by first inferring a location's longitude and latitude separately. SunSpot uses this inferred location to identify a region of interest, since solar energy data alone is not accurate enough to precisely identify a home's location. After identifying a region of interest, SunSpot then uses image processing on publicly-available satellite data within the region to identify candidate homes with visible solar systems. Finally, SunSpot applies filters to further prune this set of homes. Figure 5.8 depicts SunSpot's pipeline of operations.

5.3.1 Identifying a Region of Interest

Inferring Longitude. To infer longitude, SunSpot leverages the time of solar noon. Solar noon is the wall clock time on any given day where the sun is at its highest point in the sky at a specific location. The clock time of noon, e.g., 12pm, often differs from solar noon, as clock times are based on the local time zone, where a large region within the same time zone has the same wall clock time. In contrast, each location within that time zone has a different solar noon time, depending on when the sun rises to its highest point, i.e., nearest zenith, at that specific location. Further, as the Earth orbits the sun, the time of solar noon for each location changes gradually over the course of the year. *SunSpot leverages the fact that the day-to-day changes in solar noon across a year are consistent at every location on Earth.* In particular, the ∼31 minutes of movement in solar noon are the same at every location and dictated by the Equation of Time (EoT), which is imprinted on sundials to reconcile the difference between apparent solar time (which tracks the actual movement of the sun each day) and mean solar time (which tracks an "average" sun where noon is always 24 hours apart).

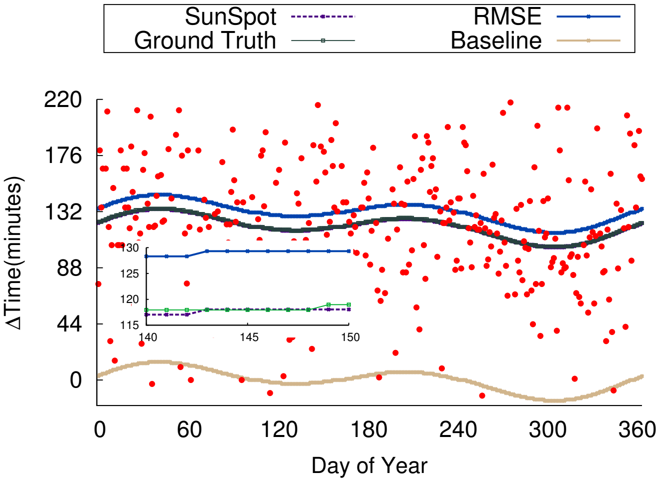


Figure 5.9: SunSpot finds the time where the EoT curve overlaps the most maximum solar generation points over the year. The zoomed-in inset shows SunSpot curve nearly preciesely overlaps the ground truth EoT curve. The y-axis is the change in solar noon time from Janurary 1st.

Figure 5.9 shows the EoT (the bottom line) at 0° latitude over the course of a year, where the y-axis is the change in solar noon time assuming we set solar noon time on January 1st to zero. For an efficient solar system (oriented towards the equator) on a sunny day, *solar noon should correspond to the time of maximum generation.* Thus, the change in the time of maximum generation should precisely track the EoT, regardless of a deployment's location. Note that using solar noon should mitigate the effect of shading from obstructions, as only the most inefficient deployments would be shaded at solar noon. Of course, due to the other effects in the previous section, the maximum time of generation does deviate significantly from that predicted by the EoT over the year. Figure 5.9 includes a scatterplot of the time of maximum energy generation (with energy data at one minute resolution) for a representative home. The scatterplot shows that there are numerous and significant deviations in the time of maximum generation across the year.

Since the EoT is the same for every location on Earth, SunSpot knows the shape of the EoT curve it must "fit" to the data: it need only shift up and down the y-axis to determine where to best place it. In the figure, we use 0° latitude as the baseline EoT. To "fit" the EoT to the data, SunSpot assumes that on ideal (sunny) days the time of maximum generation should often track the EoT, while on non-ideal (cloudy) days the time of maximum generation will be random, e.g., it might be before or after solar noon

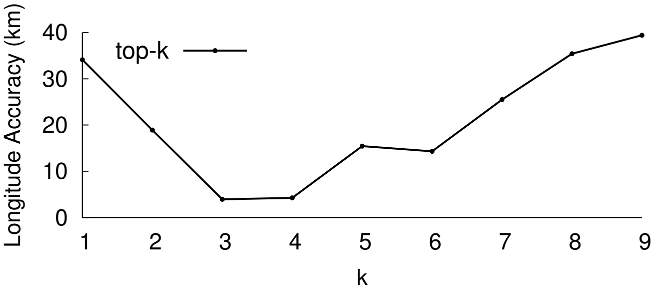


Figure 5.10: Longitude accuracy depends on the top k points of generation we include in the scatterplot when fitting the EoT.

depending on the weather. Given this assumption, we place the EoT curve at the spot on the graph where it overlaps the most data points within some tolerance, e.g., ±1 minute. In Figure 5.9, the placed curve (in dark violet) nearly precisely overlaps the actual ground truth EoT curve for the location—the bottom of the zoomed-in inset shows the two overlapping curves.We tried various other methods for placing the EoT curve, such as placing it to minimize the root mean squared error, or RMSE (in blue), but found that this and similar approaches were not as accurate. This likely occurs because using RMSE assumes the magnitude of the deviations above and below solar noon are the same. However, the local climate may cause the magnitude of these deviations to be biased towards the morning or afternoon. For example, frequent fog in the mornings might increase the probability of the time of maximum generation often occurring much later than solar noon (as in the figure).

After placing the EoT on the graph, SunSpot then infers longitude by taking solar noon time for any day on the EoT and applying our algorithm from Section 6.1 to compute the longitude. Note that solar noon time for any day on the same EoT curve will yield exactly the same longitude. In experimenting with the basic approach above, we found that the time of maximum generation often deviates from solar noon on many sunny days that appear ideal. This likely occurs due to small variations in the various factors listed in Section 2.2, such as slight variations in grid voltage. As a result, we extend this approach by using the top k times of maximum generation. That is, we sort the data points (for any resolution) by their energy generation and plot the top k data points on the graph. Figure 5.10 shows how longitude accuracy changes based on the value of k for 1Hz data.

We typically use $k = 3$, as we have empirically found that it performs best on a large set of solar deployments.

Note that our basic approach above assumes a deployment with a single set of modules oriented towards the equator. Since most deployments strive for efficiency, they are typically oriented towards the equator, which mitigates the impact of our orientation assumption in practice. However, we can extend the basic approach to account for tilts and orientations, as described below.

Based on the PSA algorithm [29] and Equation 1, we can compute a modified EoT that tracks the movement in solar radiation incident on a module of different orientations. The PSA algorithm gives the sun's position in the sky at any time for any location, and Equation 1 computes the expected solar generation given the sun's position and a deployment's tilt and orientation. The PSA algorithm requires a latitude and a longitude: we use the latitude we infer below (which is derived independently of the longitude) and we choose any longitude, since the EoT and our modified EoT will have the same shape at any location. We then compute multiple modified EoT curves for many different orientations, e.g., every 5° from 0° to 180°, and place them based on the procedure above. We choose the curve (and orientation) with the most overlapping points as above, and use Equation 1 to compute the difference between the point of maximum generation and the real solar noon for a module with that orientation. After inferring solar noon (on any day), as above, we infer longitude by taking this solar noon and using our algorithm from Section 2 to compute a longitude. While it may be possible to adjust for other factors that contribute to inaccuracy, e.g., multiple modules with different tilts and orientations, etc., we leave these optimizations as future work.

Inferring Latitude. To infer latitude, we observe that the length of a day—the time from sunrise to sunset—varies with latitude. For example, in the summer, the daylength gets longer as we go from the equator to the north pole and shorter as we go from the equator to the south pole. The situation is reversed in the winter. Thus, latitude is a function of the daylength—the length of the day at a location and how the daylength changes over the course of the year depends on its latitude. To compute the daylength, we must estimate the sunrise and sunset time. SunSpot estimates sunrise and sunset by simply taking the first and last positive points of generation in the day, respectively. However, as discussed earlier, this approach will *always* result in a significantly shorter daylength than the actual daylength. We have found the difference to be on the order of tens of minutes for sunrise and sunset, resulting in latitude errors that approach 1000 kilometers.

SunSpot mitigates this error by leveraging the insight above: *namely, in the fall/winter, daylength becomes shorter moving north, and in the spring/summer, daylength becomes shorter moving south.* As a result, using the approach above, in the fall/winter, SunSpot will always infer a location north of the actual location, and in the spring/summer, SunSpot will always infer a location south of the actual location. SunSpot splits the difference between these two errors by computing latitude separately for each half of the year, and then averaging them. This approach is surprisingly accurate, reducing latitude errors from near 1000km to less than 20km for 1Hz energy data. The accuracy improvement derives, in part, from the technique's ability to mitigate the impact of orientation, shading from structures, etc., as these characteristics affect the inaccuracy in the fall/winter and spring/summer in a similar way. Thus, when averaging, the effects largely cancel each other out.

The approach above requires some energy data from the two different halves of the year. We generally use a few months worth of data to mitigate the inaccuracy of data from any single day. Since daylength is a function of latitude, we can derive a daylength curve that dictates the daylength over a year. Of course, unlike the EoT, the shape of the curve is dependent on the latitude, requiring SunSpot to find the latitude curve that best "fits" the data. In addition, just as above, the curve that minimizes the root mean squared error is inaccurate, as it is "pulled down" by many short daylengths due to cloudy days. Instead, SunSpot defines the best fit as the daylength curve that represents the tighest upper bound on the data. While the data point that defines this tightest bound represents the most ideal day of the year (in that it is the longest day we record relative to the daylength curve), it will still always be shorter than the actual daylength, since a solar deployment cannot generate power until strictly after the sun has risen (and will stop generating strictly before the sun has set). Thus, the tightest bound will never define a longer daylength than the actual one. We find this upper bound separately for data in the spring/summer and fall/winter.

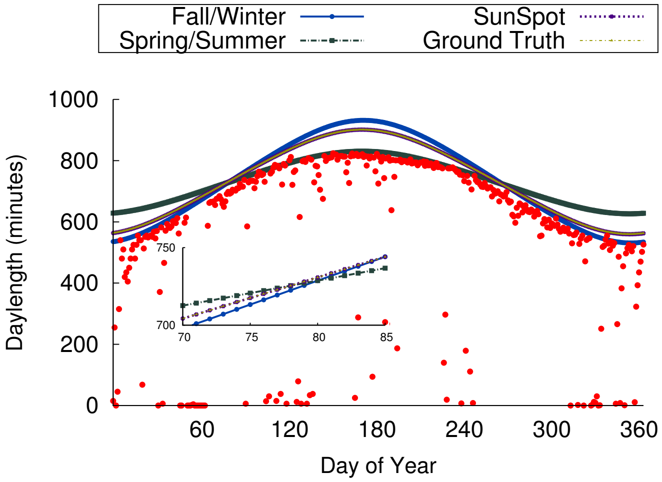


Figure 5.11: SunSpot finds the daylength curve that provides the tightest upper bound on the data in each half of the year. The zoomed-in inset shows SunSpot curve nearly preciesely overlaps the ground truth daylength curve.

Figure 5.11 shows three daylength curves for an example home, representing the upper-bound on the spring/summer data and the fall/winter data, as well as the daylength curve associated with the average of the two derived latitudes. In computing the tightest bound, we adjust for outliers due to sensing errors by removing the data point that defines the tightest bound of the daylength curve, and then find the next tightest bound. We then compare the distance between these two latitudes, and if it is less than a threshold distance n we stop, but if not we iterate again. We continue until the latitude does not change significantly. This approach ensures that at least two points over the year define near the same daylength curve. The figure shows how far apart the tightest bounds of the daylength curve are in each half of the year. The difference typically translates to near 1000km. However, when averaging the two, SunSpot achieves a location near the ground truth.

5.3.2 Localizing a Home

The latitude and longitude define only a region of interest, and are not accurate enough, even with 1Hz resolution energy data, to identify a specific address. SunSpot uses another method to localize a home within the region, as described below. We define a region of interest as being a radius r around the inferred location.

Identifying Candidate Homes. To identify candidate homes, we observe that solar modules are clearly visible from publicly available satellite imagery. Figure 5.12 shows a

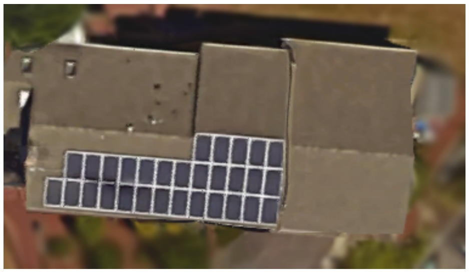


Figure 5.12: Rooftop solar is identifiable from satellite imagery.

representative photo of a rooftop solar deployment. While it is likely possible to identify candidate homes using image recognition, given the consistent and distinctive appearance of solar modules, SunSpot takes advantage of crowd-sourced image recognition on Mechanical Turk, which provides a programmatic interface to hiring people to perform routine tasks, such as image processing. In this case, SunSpot submits tasks to identify whether a solar module appears within the image.

To reduce costs, we also leverage Google Maps' landscape API that colors areas with and without man-made structures differently, allowing us to filter out forests, deserts, bodies of water, etc. Since >97% of land area in the U. S. does not have man-made structures, this optimization significantly reduces the search area, although it becomes less effective the more urban the region. Figure 5.13 shows an example of Manhattan, where man-made structures are colored black and other areas are colored green (land) or blue (water). The figure shows the API is precise at distinguishing areas without man-made structures, as streets, central park, and shoreline are not black. We provide images to the OpenCV image processing library to filter out images with very little black color.

Filtering Sites. There may be many candidate solar homes identified within the region of interest. There are numerous ways to filter this list of candidate homes. Some examples include: computing the area covered by a solar system to estimate its maximum output, and then filtering out homes that deviate from the anonymous solar data; observing panel properties, such as the orientation or the presence of obstructions, and checking if those properties manifest themselves in the data; or comparing how well drops in energy generation align with clouds moving over each candidate home.

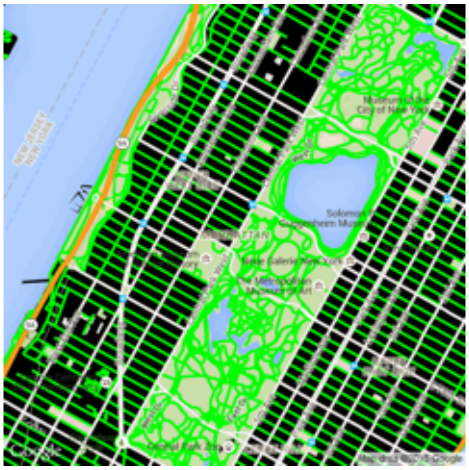


Figure 5.13: Map of Manhattan using Google Maps API that colors areas that include man-made structures black.

The filtering above may not be able to prune the list of candidate homes to only a single one. Solar energy data inherently provides k-anonymity, where k represents the number of nearby homes with similar solar deployments [82]. For example, an Hawaiian neighborhood where nearly every home has solar is less vulnerable to precise localization, despite the sunny weather, compared to a home in the Southeast where few homes have solar.

5.3.3 Preserving Privacy

There are many possible ways to preserve the privacy of solar energy data. We discuss a few below, but, due to space constraints, only focus on localization in this chapter and leave a full treatment of privacy preservation to future work. Simple data transformations that shift all datapoints forward or backwards in time would reduce the accuracy of longitude estimates. Utilities could apply these data transformations (or even remove time labels altogether) before releasing data to third parties. However, there may be legitimate reasons for third parties to know the absolute time of generation. Consumers could apply such transformations themselves by shifting their consumption using batteries. Such shifting would require significantly less battery capacity than is required to prevent Non-Intrusive Load Monitoring [61, 88], since consumers can significantly reduce longitudinal accuracy by shifting perceived generation by only a few minutes. Consumers might also be able to employ background load scheduling to introduce noise at the start, stop, and peak of solar generation. While our approach relies on these three key generation points, more sophisticated approaches may be possible that leverage the entire generation profile for localization. Thus, provably masking latitude poses a greater challenge, since it potentially requires modifying the entire profile.

5.4 Implementation

We implemented SunSpot in `python` using widely available open-source code that computes a location from its sunrise and sunset time[2]; SunSpot could also leverage any of a number of online APIs [81]. Our current implementation determines the region of interest as described in Section 5.3, but does not implement the adjustments to account for different orientations (from Section 5.3.1). After defining the region of interest, our implementation then processes satellite data to filter areas without man-made structures, divides it into many small images, and submits them to Amazon's Mechanical Turk to detect candidate homes. Our integration with Mechanical Turk downloads satellite imagery from Google Maps, which has a maximum zoom of 20 that corresponds to a width of ∼70m in the northern U.S. (but increases to ∼100m near the equator). In some highly rural areas, Google Maps either has much lower resolution or is not available. For these areas, higher resolution satellite imagery, which is available for purchase, may be necessary. The total number of images at a zoom-level of 20 (640x640 pixel) within a 1km^2 radius is 276. We use the Google Maps API to generate the equivalent images with areas with man-made objects black and other areas a different color, and then use OpenCV to automatically remove any images that have more than 5% of their area covered in black pixels. We currently do not apply the additional filters from Section 5.3.2. Thus, our results are conservative, as applying these techniques would only improve SunSpot's accuracy.

5.5 Evaluation

We evaluate our results on publicly-available energy data from 14 solar-powered homes at known locations with visible solar modules in the northern hemisphere. We have per-

[2] See `https://github.com/mikereedell/sunrisesunsetlib-java` and `https://github.com/rconradharris/pysunset/`

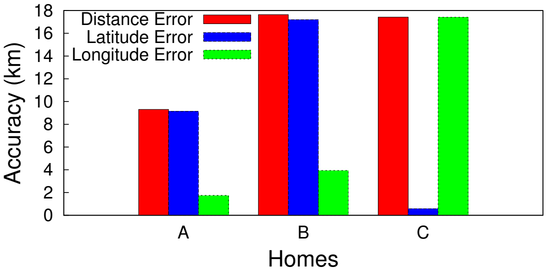
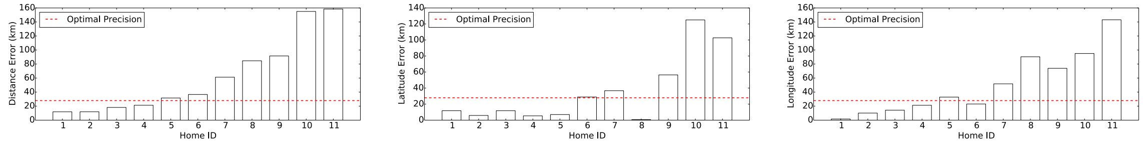


Figure 5.14: Sunspot accuracy when identifying efficient solar deployments using per-second resolution solar energy data.

second solar energy data for three of the homes and per-minute resolution data for the remaining 11 homes. For each home, we have between 6 months and a year's worth of data. Since our initial prototype does not account for irregular module tilts or orientations (as discussed in Section 5.3.1), we focus on homes with mostly south-facing orientations that maximize solar generation. Our evaluation quantifies the localization accuracy for solar deployments per-second and per-minute data resolution. We then evaluate the cost and accuracy of crowd-sourced image processing on Mechanical Turk.

5.5.1 Localization Accuracy

Figure 5.14 shows distance error when localizing the region of interest for the three homes with per-second resolution solar energy data. The latitude error is the north-south accuracy, while the longitude error is the east-west accuracy. We then compute the combined distance error as the hypotenuse of the right triangle formed by the latitude and longitude error. The combined area represents the minimum radius required to include the home in the search area. The figure shows that with per-second energy data the inaccuracy ranges from 10km to 20km. Interestingly, for Homes A and B the error in latitude dominates the total error, while for Home C the error in longitude dominates the total error. We believe Home C's higher longitude error is largely due to its orientation, which deviates the most from south-facing (and our current implementation does not take into account when determining the longitude). The underlying reason for the difference in latitude error is more difficult to determine, as averaging the spring/summer and fall/winter cancel out some, but not all, of the effects of a solar system's irregularities. This may be due to different conditions in



(a) Distance Error (b) Latitude Error (c) Longitude Error

Figure 5.15: Sunspot accuracy when identifying efficient solar deployments using minute resolution solar energy data. The red line depicts the baseline precision possible (27.9km) with minute-level data.

each half of the year, such as the presence of nearby trees, which might provide shade in the summer but not in the winter.

Similarly, Figure 5.15 shows results for 11 homes with per-minute resolution data. We sort the homes based on their error in total distance from the ground truth location in Figure 5.15(a), and again report both the latitude and longitude error in (b) and (c). The red line at 27.9km indicates the baseline precision (at the equator) with minute-level data.[3] Overall, the minimum error is 10km with six homes having an error near or below the baseline precision. The average error is 62km (or near 2× the baseline precision), the largest error is 160km (or near 5× the baseline precision). Again, the larger errors are due to less efficient deployments with orientations that deviate more from south-facing. Since we define the region of interest based on a radius r from the home, the region of interest on average is within $2^2 = 4\times$ the smallest possible region given the minute-level resolution of the data. Interestingly, in this case, the average latitude error is less than the average longitude error, despite the fact that our daylength estimates are much more inaccurate that our solar noon estimates (due to the start and stop of generation not aligning with the sunrise and sunset times).

Our results above demonstrate that 1Hz resolution significantly improves accuracy. To illustrate this, Figure 5.16 shows this scatterplot for five months of data for Home A, along with the inferred EoTs when using minute-level and second-level data over this period. As the graph shows, the time difference between the best fit with minute-level data and the best-fit with second-level data is 22 seconds, which corresponds to an error of ∼41km. In

[3] Note that the actual precision varies by the cosine of the latitude, and these homes are located across a range of latitudes.

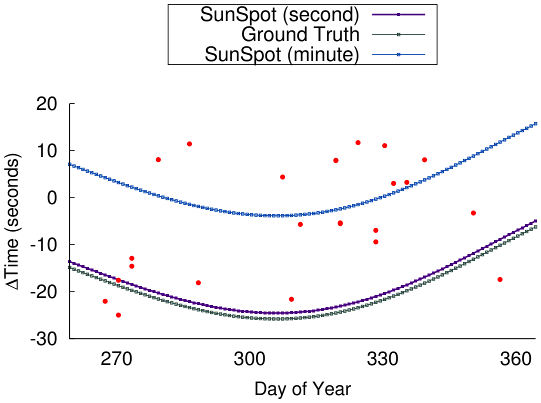


Figure 5.16: The difference in error in placing the EoT when using minute-level and second-level data.

contrast, the error between the ground truth longitude and the longitude inferred by the second-level data is only 1.75km (or 4× the baseline precision for second-level data).

SunSpot's longitude accuracy also depends on tuning the k parameter and the tolerance parameter from Section 5.3.1. Figure 5.17 plots the accuracy of inferring longitude as a function of the tolerance parameter for k=3 using Homes A, B, and C with 1Hz resolution data. Recall that the tolerance parameter is the amount of time above and below the EoT curve we are fitting, such that we count datapoints within this range as overlapping the EoT curve. The graph shows that as we increase the threshold to near 50 seconds the localization for all three homes becomes more accurate. Interestingly, the best tolerance for second-level data is near one minute, which is similar to the one minute tolerance we also used for the minute-level data. Since our dataset includes a wide range of homes from different locations, these results suggest the magnitude of these parameters do not vary significantly across deployments.

Finally, while our current implementation does not adjust for inefficient orientations that deviate significantly from south-facing, we have performed an initial evaluation of this effect. Home C is unique in that it has two separate solar arrays—one south-facing and one east-facing—with two separate inverters and meters. While the data in Figure 5.14 is from the south-facing solar array and has a longitude error of 18km, the longitude error for the east-facing array is 50km. By quantifying the effect of orientation on solar generation under the same weather conditions, this initial result indicates the potential of the optimizations that adjust for orientation. We plan to explore this as part of future work.

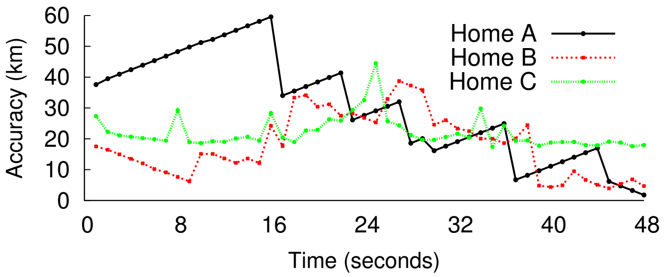


Figure 5.17: Homes A, B, and C show similar trends when tuning SunSpot's tolerance threshold when computing longitude.

5.5.2 Image Processing

To preserve privacy, we did not conduct any real searches on Amazon's Mechanical Turk, but rather ran microbenchmarks to evaluate the accuracy and cost of identifying anonymous solar-powered homes. Here, we took a random urban area with 2km radius (or 12.6km^2). We chose this small area both to limit costs and to enable manual verification of all the solar-powered homes in the area by observing each image. We divided this area into 3481 satellite images from Google Earth, which has a maximum zoom of ~65-70m at 640x640 resolution. Since we chose an urban location, a much higher percentage (82%) of these images contained man-made structures compared to the U.S. average, yielding a total of 2847 images. We manually checked these images for solar-powered homes and found 28 total.

We then submitted the 2487 images as "categorization" tasks on Mechanical Turk. We selected master-level workers for 5% extra cost to ensure high accuracy.[4] Amazon allows a maximum redundancy of two workers per task, so we issued a total of $2487 \times 2 = 5694$ images for categorization into two categories: i) yes, solar modules do exist or ii) no, solar modules do not exist. Each task had a reward of $0.02 and Amazon charges an additional $0.02 per task. Thus, the overall cost of the experiment was $170.82, or $13.60/km^2. Of the 5594 images, 99% were categorized in <30 minutes with the average time per task equal to 42 seconds. The workers agreed on 26 images, and were thus correct in identifying all but two deployments, yielding a 93% accuracy.

[4] Master-level workers have an accuracy >90%.

Since we chose a relatively small area (to minimize experiment costs and perform manual verification) in an urban setting, we were only able to filter out 18% of the search area. However, generally the larger the search radius, the higher the percentage of land area that can be filtered out. For example, in this experiment, if we had chosen a 10km radius near the search radius for Home A, only 60% of the images contained man-made structures (38085 out of 62845 total images). For non-urban areas and larger areas, we expect to be able to filter out an even higher percentage of the images. Thus, even for the relatively large search areas, it is possible to filter out a high percentage of the actual land area when searching. These costs may be reduced using computational image processing, rather than people, recognizing solar modules. Given the uniformity in solar module appearance, automated recognition is likely possible.

5.6 Related Work

There is significant prior work on estimating solar production for a specific location, which solar installers routinely use to give users an estimate of their potential benefits from solar. There is also significant prior work on analyzing building energy consumption data to infer individual appliance energy usage [90], i.e., Non-Intrusive Load Monitoring (NILM) and user behavioral patterns, such as occupancy [31, 53]. NILM researchers have also looked at inferring the generation profile of solar power by treating it as another load (with negative consumption) and disaggregating it [64]. Such solar disaggregation is now included in commercial offerings from third party analytics startups, which actively use it on data from a large number of utilities [28, 64]. As part of future work, we plan to extend SunSpot to localize a home based on net meter data by first disaggregating the solar data.

Security researchers have recognized that the energy analytics above represent significant privacy threats [61, 88, 66, 35]. However, this prior work focuses on using chemical or thermal energy storage, e.g., batteries and water heaters, to mask the changes in energy usage that analytic techniques use to infer behavior. The threat is that utilities can associate behavior learned from energy data with account information, e.g., names and addresses. The threat SunSpot exposes is different, as it reveals that data most people believe is anonymous is actually not anonymous.

SunSpot is also related to prior work on modeling a solar deployment's generation based its various characteristics, e.g., the weather, tilt/orientation, etc. These models are largely used for predicting solar generation in the near-term future based on weather forecasts. Large solar farms may develop detailed models that incorporate specific characteristics of the deployment, e.g., type of panels, tilt/orientation, wiring, etc., and data from co-located irradiance sensors [20]. SunSpot differs from this work in that it does not predict solar output, but instead estimates the location of a solar-powered home based on its output. Prior work also applies machine learning techniques on empirical solar data to develop "black box" models that do not require such deployment-specific details [46]. SunSpot is similar to this work in that it also operates on anonymous solar energy data, although for localization and not prediction. However, SunSpot could potentially improve its accuracy by incorporating information about a solar deployment's characteristics learned via such models, such as tilt and orientation.

5.7 Conclusion

We design SunSpot to localize anonymous solar energy data and expose its threat to privacy. SunSpot extracts the location information inherently embedded in solar data to localize a solar deployment to a small region of interest. The system then uses crowd-sourced image processing to identify a small set of potential solar deployments within the region. We evaluate SunSpot on publicly-available energy data from 14 homes with rooftop solar, and show that its accuracy ranges from 10km to 20km for 1Hz data. SunSpot is then able to narrow this region to a set of candidate sites with visible solar panels using crowd-sourced image processing of publicly-available satellite data on Amazon's Mechanical Turk at a cost of \$13.60/km^2. SunSpot's motivates a reconsideration of what energy data is classified as "anonymous," as current regulations, such as the DOE's Voluntary Code of Conduct for handling energy data, only consider energy data without associated account information to be anonymous. In contrast, our work shows that energy data itself can reveal location.

CHAPTER 6

WEATHER-BASED LOCALIZATION

Solar-based localization in Chapter 5 has a fundamental limit due to Earth's rotation. To further localize towards a specific home, in this chapter, we identify another key insight: every location on Earth also has a distinct weather signature that uniquely identifies it. We then present Weatherman based on this key insight, which leverages a suite of "big data" analytics techniques to expose the source of "anonymous" energy meter data.

6.1 Background and Motivation

Given the scale of the deployments as we discussed in Chapter 1 and Chapter 2, developing techniques that analyze big energy data to improve energy-efficiency has become an active research area in both industry and academia. Numerous startups, including Bidgely [28], Onzo [69], PlottWatt [70], and Sense [8], are now focused on monetizing insights drawn from big energy data. These insights have the potential to significantly improve energy-efficiency at massive scales, e.g., by providing real-time energy-efficiency recommendations to users, automatically identifying faults in individual buildings or the electric grid, detecting energy usage outliers to select candidates for energy audits, or improving consumption and generation forecasting to inform generator dispatch scheduling. To gain these insights, utilities routinely contract with the third-party big energy data analytics companies above and directly provide them energy meter data, while end-users often link their meters to public APIs that allow analytics companies to directly access customer energy data. These companies often provide analytics services to customers "for free," since the energy data itself provides value to them, e.g., either as training data to improve their techniques or in profiling users' energy usage and behavior.

The big energy data made available to the third-party companies and academic researchers above is often *anonymous* and not associated with a specific location. Anonymous energy data includes only a series of tuples, which each specify a timestamp and energy consumption (or generation). The primary reason for anonymizing energy data is to prevent third-parties from linking sensitive private behavior derived from energy data with a particular name and address [66]. Such behavioral information is potentially valuable. As one example, analyzing energy data can reveal irregular sleeping patterns, e.g., based on sporadic energy usage at night, which pharmaceutical companies could use to inform direct marketing campaigns of insomnia drugs. A publicly-posted job description from one big data analytics startup illustrates a real example of such profiling: it advertises that prospective employees will "[u]se energy data to predict whether the user has a GE or Maytag refrigerator. Very Cool! Imagine the value of that information for Whirlpool to target this house for selling their appliance." To guard against these privacy threats, the Voluntary Code of Conduct (VCC) for managing customer energy data recently released by the DOE recommends utilities remove the name and location of any energy data they share with third-parties [86].

Unfortunately, while entirely stripping energy data of its location prevents the privacy leaks above, it also removes perhaps the most important input for many well-intentioned analytics. For example, knowing location can improve the accuracy of consumption and generation forecasts by enabling analytics to correlate them with local weather forecasts. Similarly, location information can enable comparisons of energy usage across buildings within the same region to profile general energy consumption patterns or identify outliers. Thus, *recovering* the location of energy data can be a useful tool that advances the development and evaluation of such well-intentioned analytics. This is especially true for academic researchers that have limited data access, and must rely on an assortment of public datasets, which often lack metadata. In many cases, energy data is collected in an ad hoc fashion and not handled rigorously, causing the metadata that specifies location to be lost.

In this chapter, we present Weatherman, a suite of big data analytics techniques that extract location from anonymous energy consumption, wind, and solar data. Our key insight is that energy data largely correlates with the local weather, e.g., temperature, wind speed, and cloud cover, and that every location on Earth has a distinct *weather signature* that uniquely identifies it. Weatherman leverages this insight to localize the source of anonymous energy data. To do so, Weatherman combines physical system models with statistical techniques to extract a weather signature from energy data at each location when searching a massive weather database that includes records from 35k locations.

Our goal is to explore the severity of this privacy threat by quantifying the localization accuracy for energy consumption, wind, and solar data. Based on the DOE's VCC, users often do not consider the privacy implications of releasing anonymous energy data to third-parties, assuming the data is anonymous if it is not associated with location information, e.g., an address. Understanding the localization threat is important in i) educating users about the sensitivity of energy data, ii) informing evolving policies on managing energy data, and iii) developing techniques that preserve privacy, while also enabling well-intentioned analytics. Existing techniques for preserving privacy in energy data do not consider localization threats, and thus cannot prevent them [61, 88].

Broadly, Weatherman shows how public access to large "big data" archives of sensor data can introduce serious privacy threats. Our hypothesis is that weather-based localization of energy consumption, wind, and solar data is accurate to a small region. Since wind and solar sites are identifiable via public satellite imagery within the region [56, 36], such localization represents a serious privacy threat, as it is possible to associate data with a specific home. In evaluating our hypothesis, we make the following contributions.

Weather-based Energy Modeling. We present physical models that characterize the energy consumption of buildings and the energy generation of wind and solar sites based on the weather. These physical models show how energy consumption, wind, and solar energy data correlate with specific weather metrics—temperature, wind speed, and cloud cover—in different ways, which dictates the fundamental localization accuracy in each case.

Weather-based Energy Localization. We combine the physical models above with statistical techniques to extract a unique weather signature at each possible location from energy data based on its type. Weather-based localization then involves searching a massive weather database to find a location with weather that best matches the weather signature. Given the scale of the database, a key challenge is making this search both efficient and accurate.

Implementation and Evaluation. Finally, we evaluate Weatherman's localization accuracy on 117 smart meters and show that it localizes coarse (hour-level) energy consumption, wind, and solar data to within 16.68km, 9.84km, and 5.12km regions, respectively, on average. This represents significantly higher accuracy than the work on solar localization in Chapter 5, which i) only localizes solar energy data based on its solar signature, and not its weather signature, and ii) requires fine-grained second- or minute-level data and is not accurate using coarse hourly or daily data.

6.2 Correlation Modeling

Weatherman is given anonymous energy meter data that includes only a time-series of energy readings at a coarse resolution, e.g., every hour, with no other metadata. Weatherman's goal is to then analyze this anonymous energy data to infer the location—a latitude and longitude—of the smart meter that collected it. Weatherman does this by searching a database of historical weather data to find a location where the weather data best matches a *weather signature* extracted from the energy data. While our techniques are general and not specific to any data resolution, Weatherman's accuracy is dependent on the spatial resolution, temporal resolution, and coverage of the weather database. Our database includes the U.S., but could be expanded to include other areas.

Weather archives are available that cover hundreds of thousands of weather stations going back as far as 100 years. For example, in the U.S., the NOAA maintains the Integrated Surface Database (ISD) [7], which consists of hourly observations from over 35k weather stations in a common data format. In addition, Weather Underground collects real-time weather data from a network of 180k weather stations in the U.S. [18], or roughly 1 weather station every 42km^2. However, the density of weather stations is much higher in populated areas, which encompass only 3.5% of total U.S. land area [9], or 1 weather station every 1.5km^2. This is near the density that Weather Underground generates customized forecasts, which are unique for every 4km^2 grid.

Temporal resolution also affects localization accuracy. While modern weather stations typically update their observations in near real-time (every few seconds), most publicly-available weather archives only store average data at a one-hour resolution. As a result, Weatherman currently operates on energy and weather data with a coarse one-hour resolution. As we show, even with such coarse data, Weatherman achieves *higher accuracy* than prior work that localizes *only* solar data using solar signatures [36] and requires either one-minute or one-second resolution data. We expect weather data to be archived and made available at higher temporal and spatial resolutions in the future, as the cost of storage decreases, which will increase Weatherman's accuracy.

Weatherman identifies and localizes three different *types* of anonymous big energy data—energy consumption, wind, and solar data—using a type-specific technique. For each type of energy data, Weatherman leverages a different physical model based on how that energy data relates to the location's weather to extract a weather signature. Below, we describe the simple physical models Weatherman uses to generate a weather signature for each type of data.

6.2.1 Energy Consumption-Temperature Model

The dominant fraction of energy consumption in residential homes is due to space heating and cooling, which accounts for over 48% of energy usage [39]. The energy consumed for heating and cooling generally correlates with the outdoor temperature. This relationship is captured by the *degree-day* metric (in units of degree-time), which is the integral of the degrees above or below a specified base temperature over time for cooling and heating, respectively [6]. The base temperature represents the "balance" point at which no cooling or heating is required, and is typically estimated as 18C (or 65F) for buildings. The energy required to heat or cool a building is modeled as being directly proportional to the number of heating or cooling degree-days, respectively. To illustrate this relationship, Figure 6.1 plots a home's daily energy usage on the y-axis, and the daily degree-days on the x-axis, over summer. We use a base temperature of 18C, so a degree-day less than 0 is a day where the temperature was always less than 18C.

While the degree-days metric linearly correlates with energy consumption, the parameters and base temperature(s) vary significantly across buildings. For example, the slope in Figure 6.1 is related to the tightness of the building envelope, where a larger slope indicates a greater increase in energy usage for every degree rise in temperature. Similarly, the base

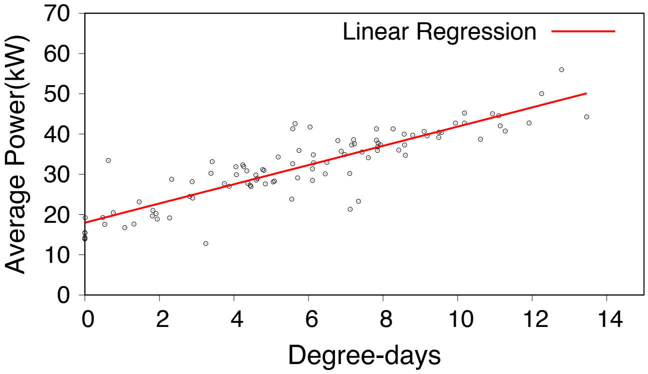


Figure 6.1: Daily average power and degree-days over the summer for a home with central air conditioning.

temperature is also partially a function of user behavior, as it depends on the thermostat setpoint. For example, if it is 30C outside but the thermostat is set to 32C cool, then even though it is quite hot, there will not be an increase in energy usage. In addition, this behavior may change over time, as users may program a thermostat to set different setpoints at different times, e.g., when they are home versus away.

Thus, the accuracy of localizing energy consumption based on outdoor temperature is, in part, a function of a building's insulation and user behavior, since less efficient buildings and users cause energy consumption to more closely track outdoor temperature. Localization accuracy is also a function of the location's temperature variance. For example, a location where temperature varies frequently has more opportunities to distinguish itself from locations where the temperature rarely changes. Since the speed of cold fronts, which have a temperature difference on the order of 10C-20C, ranges from 25 to 45 kilometers (km) per hour, two locations 25-45km apart can experience a wide difference in temperature at the same time, which, as we show, can manifest itself as a difference in energy consumption at even a coarse hourly or daily resolution.

6.2.2 Wind Energy-Speed Model

The relationship between wind speed and wind energy generation is much simpler, since 100% of wind energy is a function of wind speed. Wind power generation is based on the cubic function below, where A derives from the turbine's rotor area, ρ is the air density, and v is the wind speed.

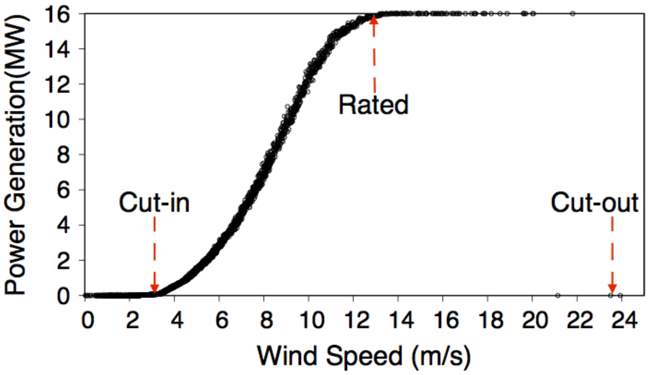


Figure 6.2: Hourly measurements of wind power and speed.

$$P = \frac{1}{2} A \rho v^3 \tag{6.1}$$

Wind turbine designs also dictate cut-in, rated, and cut-out thresholds that represent the wind speed at which power generation starts to increase, stops increasing, and terminates, respectively. At low wind speeds under the cut-in speed, there is not enough power to overcome the friction of the rotor. After the cut-in wind speed, power then increases cubically up to the turbine's rated wind speed, where its generator limits power to a constant output. The turbine generates this constant power up to a cut-out wind speed that can damage it, at which point the turbine engages brakes and power output drops to zero. While these thresholds vary based on a wind turbine's size and design, typical cut-in wind speeds are 3-4 meters per second (m/s), rated speeds are 12-17 m/s, and cut-out speeds are ∼25 m/s. Figure 6.2 is a scatterplot of hour-level wind generation and speed measurements with annotations of the turbine's cut-in speed, cubic function, rated speed, and cut-out speed.

6.2.3 Solar Energy-Cloud Cover Model

Solar energy embeds perhaps the most detailed location information, since the Sun's irradiance in clear skies at every location is a well-known function of time. Thus, prior work shows how to localize the source of anonymous solar energy data using only its *solar signature*, which it defines as the shape of the curve of solar output over time in clear skies [36]. This approach requires second- and minute-level solar data to localize within ∼20km and ∼60km, respectively, which is near the highest accuracy possible given the speed of the Earth's rotation. However, prior work does not consider weather when localizing

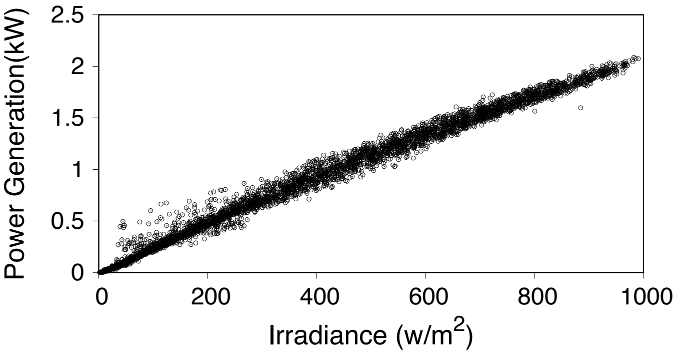


Figure 6.3: Solar energy is a linear function of solar irradiance.

solar data, even though there is a strong relationship between solar output and weather, particularly temperature and cloud cover.

Figure 6.3 shows the solar output of a small rooftop solar site as a function of the measured Global Horizontal Irradiance (GHI) in W/m^2 using a pyranometer at the same location. As expected, the relationship is almost perfectly linear, since solar modules translate irradiance directly into power with an efficiency loss. The small imprecision in the relationship is due to minor temperature effects, which cause efficiency to decrease as the temperature rises. Temperature coefficients for crystalline solar modules range from 0.38-0.50, such that for every degree above 25C the output decreases by 0.38-0.50% and for every degree below 25C the output increases by 0.38-0.50%. However, the slope of the line depends on the site's efficiency, which is a function of lower-level physical and electrical characteristics, such as the module's material and choice of inverter(s).

Unfortunately, unlike temperature and wind speed, most weather stations do not include pyranometers, and thus do not report ground-level irradiance. Thus, to localize solar data, we cannot simply correlate ground-level irradiance measurements with solar output. As a result, we use the coarse sky condition information reported by weather stations, and is typically measured in *oktas*, which represents how many eighths of the sky are covered in clouds and ranges from 0 oktas (completely clear sky) to 8 oktas (completely overcast). The sky conditions reported by the NWS translate directly to oktas [19]. For example, "Clear/Sunny" is <1 okta, "Mostly Clear/Mostly Sunny" is 1-3 oktas, "Partly Cloudy/Partly Sunny" is 3-5 oktas, "Mostly Cloudy" is 5-7 oktas, and "Cloudy" is 8 oktas. While more accurate sky condition estimates can be extracted from visible satellite

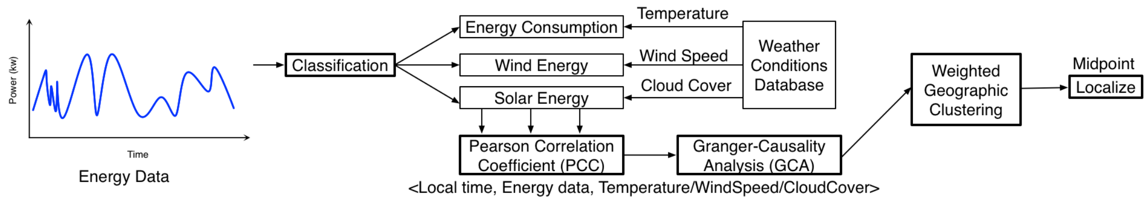

Power (kw)
Time
Energy Data
Classification
Energy Consumption
Wind Energy
Solar Energy
Temperature
Wind Speed
Cloud Cover
Weather Conditions Database
Pearson Correlation Coefficient (PCC)
Granger-Causality Analysis (GCA)
<Local time, Energy data, Temperature/WindSpeed/CloudCover>
Weighted Geographic Clustering
Midpoint
Localize

Figure 6.4: Pipeline of operations Weatherman uses to infer the location from its energy data.

images [42], this process is non-trivial and these measures are generally not reported by weather stations.

6.3 Weatherman Design

Weatherman uses the physical relationships above to search a massive weather database to determine the location with weather that best matches a weather signature extracted from the energy data at each possible location. Figure 6.4 shows the pipeline of Weathman operations. Since Weatherman assumes energy data is anonymous, it makes minimal assumptions about the associated metadata. For example, Weatherman does not assume the type of energy data is given, which requires it to first classify the data. This classification is straightforward using simple rules: if the energy data is rarely zero, we classify it as energy consumption, since nearly all buildings have a non-zero baseload; if the energy data is consistently zero for a multi-hour period every 24 hours, we identify this period as nighttime and classify it as solar; if we do not classify the energy data as consumption or solar, and it exhibits a similar variance over time, we classify it as wind (under the assumption that wind intensity is more similar across day and night than solar). Weatherman also does not require the associated units of energy data, e.g., watt, kilowatt, megawatt, etc., as it uses only the correlation of energy with weather at a location, which does not depend on the magnitude. In addition, we also ignore a positive/negative sign, if any, associated with the energy data, since there is no standard for how it maps to generation or consumption.

Finally, Weatherman supports different assumptions about the metadata information encoded in the timestamp. In effect, the specificity in the timestamp simply increases or decreases the search space. By default, Weatherman assumes the timestamp includes the date and hour, but the local timezone is not known. In this case, Weatherman assumes the timezone is local to each location for which it is extracting the weather signature. However, in some cases, the timestamp may not include the date. In this case, Weatherman must correlate weather signatures for every possible daily time offset against each location in the weather database, which increases the search-space by 365× over a year. Similarly, if the timestamp does not include the hour, it increases the search space by 8760× over a year.

6.3.1 Weather-based Localization Challenges

A naïve approach to weather-based localization is to directly correlate energy data with weather measurements at each location in our weather database. There are many possible functions that quantify how well two time-series correlate with each other, enabling a ranking of locations based on how well energy data matches the weather data. For example, the Pearson Correlation Coefficient (PCC) is a measure of the linear correlation between two variables, computed as the covariance between the variables divided by the product of their standard deviation. It is shown in the equation below,

$$PCC = \frac{\sum_{t=1}^{n}(x_t - \bar{x})(y_t - \bar{y})}{\sqrt{\sum_{t=1}^{n}(x_t - \bar{x})^2}\sqrt{\sum_{t=1}^{n}(y_t - \bar{y})^2} \tag{6.2}$$

where n is the number of samples, x_t and y_t are the single samples-weather data (temperature, wind speed, or sky cover) and energy data (energy consumption, wind generation, or solar generation) indexed with time t, $\bar{x} = \frac{1}{n}\sum_{t=1}^{n} x_t$ (the sample mean); and analogously for $\bar{y}$.

A naïve approach selects the top-ranked location using such a correlation function. Unfortunately, this approach has multiple problems.

Imprecise. The approach is imprecise, as energy data, itself, does not highly correlates with weather. For example, as discussed in 6.2, energy consumption only correlates with weather above (or below) a specified base temperature, while wind turbines define multiple points where the correlation between wind speed and power output abruptly changes. In addition, while changing weather instantly affects wind and solar energy, there is often a lag in the effect on energy consumption as a building heats up or cools down, which simple "instantaneous" correlation coefficients, such as the PCC, do not capture. Thus, as we describe below, *Weatherman extracts a custom signature for each location that accounts for such data type-specific weather effects.*

Inefficient. Searching all locations in a large weather database can be highly inefficient, since Weatherman must extract and compare a weather signature from energy data based on each location's weather metrics over a long period of time, e.g., multiple months to years. While any individual extract and compare operation is not expensive, performing them over tens of thousands of locations across many months in a massive weather database is not efficient, especially for correlations that must account for a variable lag. Thus, Weatherman first extracts weather signatures on coarse day-level data, which it natively stores in its database, to filter the possible locations, as each type of energy also correlates with weather each day, albeit with less precision. *To improve efficiency, Weatherman only then does finer-grained signature extraction and matching on this filtered set of locations.*

Noisy. While the physical models in 6.2 apply to all energy consumption, wind, and solar data, the specific parameters of each model vary widely and only represent a coarse approximation of each system, since, in each case, a large number of hidden variables also affect energy consumption and generation. For example, the actual energy consumption may be less than predicted by a linear model when occupants are away on vacation if they set their thermostat to a high temperature. Similarly, if occupants return and proceed to execute a series of energy-intensive laundry loads (to clean the clothes from their vacation), then their energy consumption may exceed the linear model. This noise makes it challenging to distinguish between locations that are close together, e.g., under 20km, especially with access to only coarse hour-level data. As we discuss, *Weatherman selects different correlation functions at different granularities for different data types to mitigate the impact of noise.*

6.3.2 Basic Weather Localization Approach

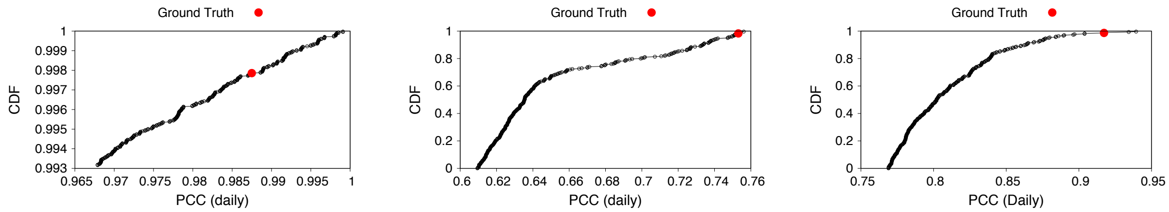
Based on the discussion above, Weatherman uses the same general approach to localize each type of weather data. Weatherman first uses data type-specific methods, discussed below, for extracting a custom weather signature from energy data at each location in its weather database. To improve efficiency, it first extracts and correlates this weather signature at each location with coarse day-level weather using a data type-specific correlation function. Weather databases, including ours, typically store both day- and hour-level data. Using day-level data both reduces the size of the input by 24×, and thus increases efficiency, and, in the case of energy consumption, also mitigates the impact of a variable lag in the energy response to temperature changes (since this lag is only evident at hour-level). Weatherman uses the day-level analysis to filter possible locations by generating a Cumulative Distribution Function (CDF) and only considering locations in the "tail" of the CDF, where the correlations are highest, e.g., within the top 1%. This day-level filtering is especially important if timestamps lack metadata, since this increases the search space.

Weatherman then finds the weighted geographic midpoint of these top candidate locations (based on the magnitude of the correlation with each location) to estimate a final location. Below, we present the weather signature extraction and correlation functions for each energy type.

6.3.3 Energy Consumption Weather Signatures

Based on the degree-days model from 6.1, when correlating with each location, Weatherman removes energy consumption datapoints whenever the corresponding temperature is below the typical 18C base temperature. Since energy consumption is linear with degree-days above a base temperature, we simply compute the correlation using the PCC between daily energy and temperature data. Note that the daily correlation is robust to changes in user behavior, which are most prevalent within a day, e.g., from setting a programmable thermostat schedule that differs over the day, rather than across days. Figure 6.5(a) shows the CDF of the PCC across all locations, with the ground truth location indicated as a red dot. This graph is for the same home as in Figure 6.1. In this case, we filter the list of locations from 30k to 300 (the top 1%). Here, the home's actual location is ranked 94th, and the weighted geographic midpoint of the top five homes is 38.84km from the actual location.

Unfortunately, for higher resolution hourly energy data, there is typically a variable lag between the increase in temperature and the corresponding increase in energy consumption, as it takes time for a building to heat up and for its thermostat to detect this and activate the air conditioner. This lag is variable, as it depends on the thermostat setting, which may



(a) Energy consumption (b) Wind energy (c) Solar energy

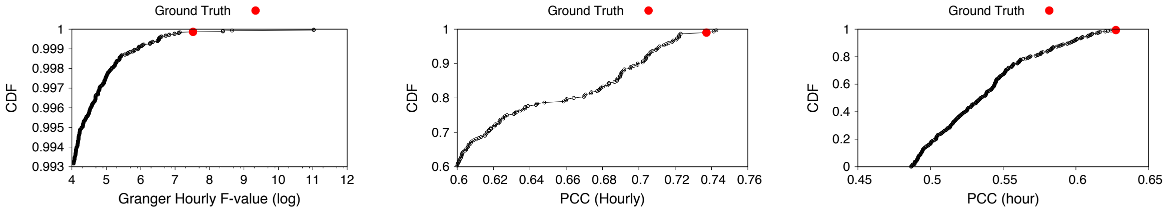
Figure 6.5: CDF of correlation analysis across all locations for daily energy consumption, wind energy, and solar energy data.

vary, and the tightness of the building's envelope. As a result, the impact of a temperature increase is often not observed in energy data for an hour or more. Thus, the PCC does not work well with hour-level data, since it only considers the correlation between each two points in time.

In this case, Weatherman applies Granger causality analysis [41], which captures the extent to which changes in one variable predict (or lag) another over time using an F-test. Note that, unlike the PCC, Granger causality analysis does not require that changes be linearly correlated, only that they lag and have the same direction. Computing Granger causality is more computationally-intensive than computing the PCC, since it searches over multiple possible lag values. As a result, performing Granger causality at hour-level over 35k locations is time-consuming. For example, a full search, assuming the date and hour are well-known, takes 8.5 hours using 80 high-end data center servers. If the date is not included in the timestamp, the search would take $\sim 8.5 * 120 = 1020$ hours (42.5 days) on the same set of servers, since we only conduct this search over the summer months. Thus, we only perform Granger causality analysis over the filtered list of the top 300 (1%) sites using the daily data analysis above, which takes ∼5 minutes.

Figure 6.6(a) shows the CDF of the Granger causality of the hourly energy data (using an F-test with a p-value<0.001). In this case, the final weighted geographic midpoint of these 300 locations results in an estimated location 6.14km from the actual location (and within the same town). Note that the home is 4.1km from the nearest weather station, which has the fifth highest correlation in this case.

As Figure 6.2 shows, the relationship between wind power and speed is defined by a piecewise function based on the cut-in, rated, and cut-out speeds. A simple approach for



(a) Energy consumption (b) Wind energy (c) Solar energy

Figure 6.6: CDF of correlation analysis across all locations for hourly energy consumption, wind energy, and solar energy data.

extracting a wind weather signature would be to focus on just one part of this function. However, this would remove useful information. Instead, Weatherman projects this piecewise function onto the single line $y = 0$ (where y is the energy generation and x is the wind speed), such that the wind power data as a function of wind speed after this projection should be zero at the correct location. Since PCC and other correlation coefficients are undefined when the variance of one variable is zero, we rank locations based on their average absolute value after the projection, i.e., the average perpendicular distance from $y = 0$.

While cut-in, rated, and cut-out speeds vary slightly from turbine to turbine, they are in the same general range. In general, the cut-in speed is ∼3m/s, the rated speed is ∼13-14m/s, and the cut-out speed is greater than ∼22m/s. To account for slight differences in these ranges between turbines, when extracting the weather signature at each location, we remove datapoints around the boundary points. Specifically, we remove datapoints that correspond to wind speeds in the ranges 3-4m/s, 13-14m/s, and 21-22m/s.

6.3.4 Wind Energy Weather Signatures

Weatherman does not alter the energy datapoints that correspond to wind speeds from 0-3m/s and >22m/s, since these should already map to zero. For datapoints in the range 4-13m/s, we first take the cube root of the energy data, perform a linear regression, and then find the distance each datapoint is from this line. We then project these points by replacing their original energy generation value on the y-axis with this distance value on the y-axis. For datapoints in the range 14-21m/s, we find the horizontal line that minimizes the root mean squared error with the datapoints, and then subtract the y-value of this horizontal line from the y-value of each datapoint. After this projection, wind power data that perfectly correlates with wind speed will lie near the line y=0. Figure 6.7 illustrates

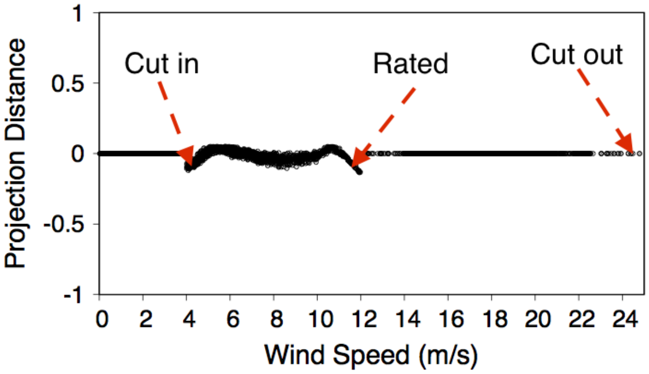


Figure 6.7: Projection of wind data from Figure 6.2 to y=0.

this projection for the same data as in Figure 6.2. As expected, the projected data is near the line y=0, e.g., within 0.1.

We perform the same projection when filtering based on daily and hourly data, as described above. After performing this projection, Weatherman proceeds based on the basic approach above, where the weather metric is wind speed and the correlation function is the average of the absolute value of the projected data. Figure 6.5(b) and Figure 6.6(b) show the CDF of this average across all locations for the daily and hourly data for the data in Figure 6.2, with the ground truth location indicated by the red dot. In this case, we filter from 30k to 300 (1% of locations) using the daily energy data. Here, the nearest weather station to the actual location ranks fifth and the geographic midpoint of the selected locations is 24.37km from the target site. We then perform the same analysis on the 300 sites using the hourly data. In this case, the nearest weather station ranks third, and the geographic mid-point of the 300 locations is 3.87km away from the location. Interestingly, the nearest weather station is 10.31km away, so in this case, the localization is more accurate than any single weather station.

6.3.5 Solar Energy Weather Signatures

Solar power has a near linear correlation with solar irradiance, which is largely determined by cloud cover that is measured by weather stations in oktas, as discussed in 6.1. Unfortunately, raw solar generation does not directly correlate with oktas, as solar output varies over both the time-of-day and the day-of-year. Since these variations are a function of location, Weatherman does not know them precisely. However, we can roughly estimate

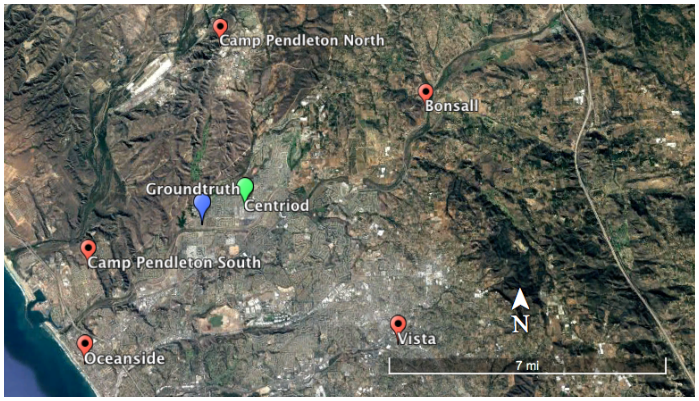


Figure 6.8: Weather-based localization of solar energy.

the maximum generation potential P_s of solar by observing that the average clear sky irradiance, which is a well-known function of time at each location based on a site's efficiency, tilt, orientation, etc. should be an upper-bound on solar output, as described by the equation in section 3.2.2.2.

We search for the parameters above as described in prior work in Chapter 3, 4 and 5. Specifically, we first use prior work on localization using solar signatures to estimate a location by associating the first, last, and maximum hour of generation with the time of sunrise, sunset, and solar noon in Chapter 5. Note that we use this search only to provide a rough estimate of the hourly maximum generation; the latitude and longitude we find are not accurate for localization. Accurate localization based on the solar signature using hourly data is not possible, as it has a maximum accuracy of at most 1656km based on the speed of the Earth's rotation. Given this rough location, we then conduct a binary search for the efficiency, tilt, and orientation parameters that represents the tightest upper-bound on the data, as described in prior work in Chapter 1, since solar generation is bounded by the maximum clear sky irradiance at any time. Weatherman's search only differs in that it does not adjust the solar efficiency of the maximum generation model for temperature at this stage.

The model above does not account for temperature, which increases solar efficiency by some percentage c for every degree Celsius decrease in temperature. Thus, when extracting the weather signature for each location, we use the same model from prior work to adjust for these temperature effects 3.2.2.3. This approach conducts a binary search for the temperature coefficient c that results in the tightest upper bound on the data.

Note that, since temperatures are different at each location, our search must repeat this process at each location. This search provides a temperature-adjusted estimate of the maximum solar generation for each hour and day at each location. Weatherman then normalizes the daily and hourly data in its weather signature at each time period by dividing each data point by this maximum estimated solar generation. This normalized solar output (relative to the maximum possible output in clear skies) should linearly correlate with the sky condition in oktas reported by weather stations. As a result, Weatherman directly uses the PCC to quantify this correlation. While Weatherman could use the latitude and longitude estimated from the solar signature to limit its search space, it does not because the estimates are highly inaccurate, e.g., order of thousands of kilometers.

Figure 6.5(c) and Figure 6.6(c) show the CDF of the PCC across all locations for the daily and hourly solar data. We again filter from 30k locations to 300 locations (the top 1%) using the daily data, and then perform analytics using the hourly data. The nearest weather station ranks fourth in the daily data with the geographic midpoint of the 300 locations 13.53km from the actual location. The nearest weather station ranks second using the hourly data with the geographic midpoint an estimated location 2.05km from the actual location. Again, the nearest weather station is 12.73km away from the actual location, so Weatherman's estimate is closer than any single point. Figure 6.8 shows the top 2% locations that contribute the most to the midpoint.

6.4 Implementation

We implement Weatherman in Python, and plan to make it and our data publicly-available. We use the scikit package, which includes the required correlation functions, e.g., PCC and Granger causality analysis. We implement a standard approach for finding the weighted geographic midpoint.[1] Finally, we use the Pysolar Python package for estimating the clear sky irradiance [3]. Note that we set the thresholds for filtering the daily and

[1] Using the approach at http://www.geomidpoint.com/calculation.html.

Dataset Name	*Granularity*	*Duration*	*Sites*	*Dimension*
Weather	Hour	6/1/2016∼10/1/2016	3,0000	6
Energy Consumption	Hour	6/1/2016∼10/1/2016	100	2
Wind Energy	Hour	6/1/2016∼10/1/2016	7	2
Solar Energy	Hour	6/1/2016∼10/1/2016	10	2

Table 6.1: Datasets used in evaluations.

hourly data based on an empirical analysis. We experimented with different thresholds in this range, and it did not significantly change the results. For the daily data, we select the top 1% of locations with the highest correlation, and for the hourly data we compute the weighted geographic midpoint of the filtered locations.

We build our weather database (with $\sim$ 1 million hourly data) by fetching data from DarkSky's weather data API. [2] Weatherman only uses the temperature, wind, and sky condition data. Our database currently stores hourly and daily weather data from 35k weather stations in the U.S. In general, weather station archives natively include both average hourly and daily measurements. While some of these weather stations include data archives going back multiple decades, our database only includes hourly and daily data over a four-month period in the summer, including June, July, August and September. We focus on data in the summer, since we are localizing electricity consumption, and all cooling is electric.

6.5 Evaluation

Section 6.3 illustrates Weatherman on an example building, wind site, and solar site. In this section, we evaluate Weatherman's accuracy across many sites, and highlight how its accuracy varies across sites with different characteristics.

6.5.1 Datasets

As shown in Table 6.1, our evaluation uses energy consumption data from a sample of 100 homes in the Pecan St. dataset [13], as well as 10 solar sites and 7 wind sites. We compute the accuracy as the difference between the localization estimate and the Pecan St. neighborhood. We know the configuration profiles of AC setup for each PecanStreet

[2] `http://darksky.net`

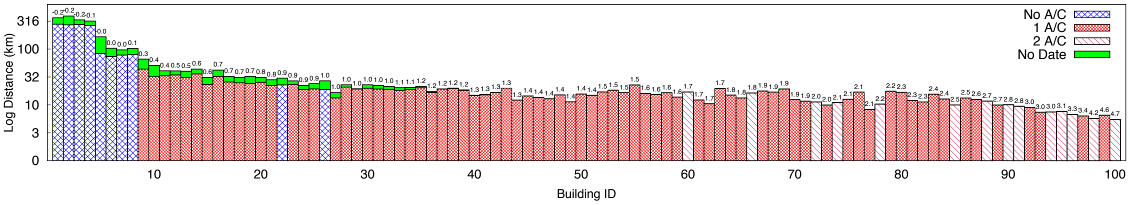


Figure 6.9: Weatherman localization accuracy for 100 different homes in Texas (on a log-scale). Homes are sorted by their degree-day slope, which appears atop each bar. The bar color indicates whether the home has zero, one, or two air conditioners.

house, and the ground truth location for all wind and solar sites. Our weather database has weather condition as: [local time, latitude, longitude, Temperature, Wind Speed, Cloud Cover]. The energy consumption, wind, and solar data have the same format as: [local time, energy in kw].

6.5.2 Energy Consumption

Figure 6.9 shows Weatherman's localization accuracy for 100 homes from the Pecan St. dataset. We sort homes by the slope of their average energy usage versus degree-day line, as depicted in Figure 6.1, which appears as a number above each bar. We also color each bar based on whether the home has zero, one, or two air conditioner circuits sub-metered, as the Pecan St. dataset includes not only each home's aggregate energy usage, but also the energy usage of selected circuits. Multiple air conditioners indicate multiple units for multi-zone cooling systems. We also experiment with two different timestamp assumptions: one where we know each point's date and hour (but not the timezone), and one where we only know the hour (but not the date or timezone). We indicate the decrease in accuracy from removing the date by placing an additional green bar atop each bar. Note that the y-axis has a log scale.

The graph shows that homes without air conditioners have a relatively flat degree-day slope, which indicates that their energy consumption does not change with outdoor temperature. The first four bars of the graph have a localization accuracy of >200km and exhibit a *negative* degree-day slope, such that their energy consumption *decreases* as the temperature increases in the Texas summer. Four other homes with no air conditioners have a non-negative slope in the range 0.0-0.1 and thus have a slightly better localization accuracy of ∼80km. There are two other homes without air conditioners that exhibit much

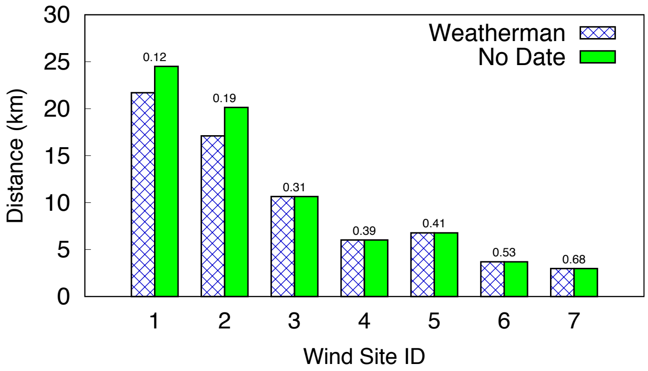


Figure 6.10: Localization accuracy for 7 wind sites, sorted by variance in wind speed, which appears atop each bar.

higher degree-day slopes 0.9-1.0, and thus yield better accuracy of ∼20km. These homes likely operate other temperature-dependent loads.

As expected, homes with a single air conditioner exhibit degree-day slopes ranging from 0.3-4.6, and exhibit much higher localization accuracy, ranging from 5-40km with an average accuracy of 16.98km. Homes with two air conditioners tend to have an even larger degree-day slope and thus a higher average localization accuracy of 11.89km on average. Here, accuracy is a roughly linear function of degree-day slope, indicating that more efficient homes are more difficult to localize. We also observe that removing the timestamp's date does not significantly alter the localization accuracy: for homes with degree-day slopes >1.2 (near Building ID 37) it does not change, and for homes with degree slopes <1.2 it only slightly decreases. This shows that weather signatures are distinct, not only across locations, but also across time.

6.5.3 Wind Energy

Figure 6.10 shows the localization accuracy for 7 wind sites in Washington (#1), Idaho (#2), California (#3), Colorado (#4,6), Wisconsin (#5), and Texas (#7). In this case, we sort the sites by the variance in the wind speed at their location over a year. We use variance as a proxy for the uniqueness of a location's weather signature, since the more the wind speed varies, the more opportunity Weatherman has to distinguish one location from another. As expected, the localization accuracy increases as the variance in wind speed at a location increases. In this case, the highest variance yields the highest localization

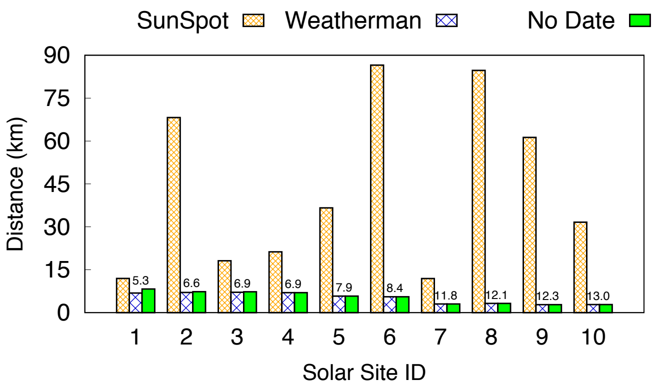


Figure 6.11: Localization accuracy for 10 solar sites, sorted by variance in sky condition, which appears atop each bar.

accuracy of ∼3km, while the lowest variance yields the lowest accuracy ∼21km. Thus, wind localization is slightly more accurate than energy usage localization (for homes with air conditioners). For wind energy, removing the date from the timestamp also has little effect on the accuracy (indicated by the green bars as before).

6.5.4 Solar Energy

Figure 6.11 shows Weatherman's localization accuracy for 10 solar sites, as well as the accuracy for prior work on SunSpot, which localizes using a site's solar signature. The solar sites are in North Carolina (#1), Washington (#2), Colorado (#3-5), Texas (#6), Wisconsin (#7), Massachusetts (#8,10), and Ohio (#9). In this case, for SunSpot, we localize using minute-level data, while for Weatherman we localize using hour-level data. Similar to above, we sort the sites by the variance in their location's sky condition data, which is listed atop each bar. Using the same intuition as for wind, the more variable the sky condition, the more opportunity Weatherman has to distinguish one location from another. As above, we see that Weatherman's accuracy improves as the variance increases, with the most variable site having an accuracy of ∼2km. In addition, we see that solar localization accuracy based on weather is typically higher than either energy consumption or wind with all the sites having an accuracy between 2-7km. In general, the relationship between solar power and cloud cover (and temperature) is more direct than the similar relationships with energy consumption and wind, since solar is a purely electric device, while the other two involve more complex mechanical relationships.

We also see that weather-based solar localization is significantly more accurate using hour-level data than SunSpot using minute-level data. In particular, the *worst site* for Weatherman has an accuracy of 6.86km, while the *best site* for SunSpot has an accuracy of ∼12km. In addition, SunSpot has more variable accuracy, indicating that its solar signature is less robust than Weatherman's weather signature. Finally, we again see that removing the date from the timestamp has a minimal effect on localization accuracy.

6.6 Related Works

The work most similar to Weatherman is recent time-series big data analytics work on SunSpot [36], which localizes pure solar data using a solar signature, specifically by inferring the time of sunrise, sunset, and solar noon. This technique was able to localize solar sites to within ∼20km using second-level solar data and ∼60km using minute-level data. Weatherman shows that weather-based localization is *significantly more accurate* using *much lower resolution data* and requiring *much less data.* In particular, Weatherman's average solar localization accuracy using hour-level data is 5.12km, which is more accurate than SunSpot's accuracy using data that is 60-3600× lower resolution. In addition, SunSpot requires more than six months of data, since it needs data in both the spring/summer and fall/winter to pinpoint an accurate latitude, while our evaluation here only used data from four months in the summer. Finally, Weatherman is more general and also capable of localizing energy consumption and wind energy data to similar (or better) levels of accuracy.

There have been numerous papers focused on preventing big energy data analytics to protect user privacy. These techniques generally focus on obscuring the patterns of high resolution energy data, e.g., second-level or minute-level, using a controllable power source, such as a battery [61, 88], a water heater [35], or a solar inverter [72]. These techniques are likely not effective in preventing weather-based localization, since it requires only coarse day- and hour-level data. In general, the battery, water heater, or solar capacity required to significantly alter day- and hour-level energy usage over a long period is prohibitively expensive. While users could also prevent weather-based solar localization by decreasing their solar output at their inverter, this would decrease solar generation and thus defeat the benefit of solar modules. In addition, we also show that even modifying energy data to eliminating timestamp metadata, e.g., by not including the date or hour, does not significantly affect Weatherman's accuracy. Thus, preventing weather-based localization represents a challenging problem, which we plan to explore as part of future work.

6.7 Conclusion

We present Weatherman, which leverages a suite of big data analytics techniques to localize anonymous energy usage, wind, and solar data. Weatherman shows how access to large archives of publicly-available, and seemingly innocuous, sensor data can introduce serious privacy threats. Our work shows that weather-based localization is highly accurate for multiple types of energy data. In particular, we show that Weatherman localizes coarse (one-hour resolution) energy consumption, wind, and solar data to within a radius distance of 16.68km, 9.84km, and 5.12km. These results are *significantly more accurate* using *much lower resolution* and *much less* energy data than using solar signature in Chapter 5 on energy-based localization, which only localized solar data to within ∼20km using second-level energy data.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

Solar generation capacity is rapidly expanding as solar module prices continue to drop. Accommodating this solar growth, while also balancing electricity's real-time supply and demand, is placing increasing pressure on utilities to monitor, forecast, and respond to changes in solar generation. Thus, there is an increasing interest in accurately solar analytics.

Prior solar analytics are either using "white box" approaches or "black-box" approaches. Unfortunately, none of them is practical for utilities to manage the grid. "White-box" approach requires utilities to manually gather and record detailed deployment information from millions of solar-powered homes. While, "black-box" approach does not incorporate fundamental well-known physical models of solar generation and requires a significant amount of historical pure solar generation data to train an accurate model. And this pure solar data is generally not available for either new deployments coming online or deployments that do not continuously monitor and store the data to train an accurate model.

To address these problems, we present a hybrid "black-box" approach that can achieve the best of both to solar data analytics. We show that the hybrid "black-box" approach can enable a wide range of accurate solar analytics, including modeling, disaggregation, and localization, with limited training data and without knowledge of key system parameters by integrating "black-box" machine learning approaches with "white-box" physical models.

We first investigate on most recent solar performance modeling techniques, including ML-based modeling and physical modeling, and then present a configurable hybrid ML "black-box" approach that combines the benefits of both. Rather than manually determining the values for the physical models, our approach automatically calibrates them by finding values that best fit the data. And the calibration requires much less data as few as 2 datapoints.

Unfortunately, the physical models we used in the above hybrid approach are highly inaccurate and perform significantly worse than pure ML models. This inaccuracy can drive from either the physical models being inaccurate, for from the effect of unmodeled physical parameters, such as other weather metrics, shading from nearby trees or buildings. To address this problem, we conduct a large-scale data analytics to determine the primary source of the inaccuracy. We isolate 10 different weather metrics on solar output using 343 million hourly weather and solar data, and then find that the only weather metrics affecting solar output are temperature and cloud cover. We then derive a new physical model to quantify cloud over's effect on solar generation. Finally, we enhance our physical model with an ML model that leads each site unique shading effects. And we show that this enhanced physical model has significantly better accuracy than state-of-the-art ML models and a model based on GHI estimate s from satellite imagery.

To address the Behind-the-Meter (BTM) problem and get pure solar generation for the training of ML approaches, we present SunDance, a "black box" system for accurately disaggregate solar generation from net meter data without access to a building's pure solar generation data for training. We also identify a new relationship between weather metrics and solar generation–Universal Weather and Solar Generation Effect. This effect has never been discussed before, and is highly useful for other energy data analytics.

Energy data is usually considered "anonymous" if it is not associated with identifying account information, e.g., a name and address. We argue that solar energy data is not anonymous, since every location on Earth has a unique solar signature (including sunrise, solar noon, and sunset times), and it embeds detailed location information. To localize the solar-powered home, we then design "SunSpot" that can localize a solar-powered home within ∼500 meters and ∼28 kilometers radius for per-second and per-minute resolution solar generation data.

To further localize towards a specific home, we find another key insight: besides the solar signature, every location on Earth also has a distinct weather signature that uniquely identifies it, since energy consumption, wind generation, and solar generation largely correlates with weather metrics, e.g., temperature, wind speed, and cloud cover. We then design Weatherman to localize the source of energy data. Interestingly, we find that localizing coarse (one-hour resolution) energy data using weather signature is more accurate than localizing solar data (one minute or one second resolution) using its solar signature. Therefore, Weatherman exposes a severe new privacy threat from energy data, which has not been discussed before.

7.2 Future Work

Smart meter is the most widely deployed sensor in the world. Comparing with other energy data, net metered energy data, which combines energy consumption and renewable data, is much more common. These energy data embeds detailed information about a building's energy-efficiency, as well as the behavior of its occupants, which academia and industry are actively working to extract. However, for the public available net meter datasets, their addresses or locations are typically not associated. We can combine and extend SunDance and Weatherman is: 1) disaggregate the solar generation from anonymized net metered data; 2) localize the pure solar generation data using weather signature to a small region of interest; 3) further localize to a specific solar-powered home using machine learning-based satellite image processing technique. We could apply these techniques to automatically determine a site's location.

BIBLIOGRAPHY

[1] Bird Simple Spectral Model. http://rredc.nrel.gov/solar/models/spectral/.

[2] PVWatts. http://pvwatts.nrel.gov/.

[3] PySolar. http://pysolar.org/.

[4] Solar Radiation Cloud Cover Adjustment Calculator. http://www.shodor.org/os411/courses/_master/tools/calculators/solarrad/.

[5] System Advisor Model. https://sam.nrel.gov/.

[6] BizEE degree Days: Weather Data for Energy Professionals. http://www.degreedays.net/, 2017.

[7] NOAA Integrated Surface Database. https://www.ncdc.noaa.gov/isd, Accessed June 2017.

[8] Sense. https://sense.com/, Accessed June 2017.

[9] United States Census Bureau. https://www.census.gov/newsroom/press-releases/2015/cb15-33.html, Accessed 2017.

[10] Google Project Sunroof. https://www.google.com/get/sunroof, June 2018.

[11] NumPy. http://www.numpy.org/, June 2018.

[12] Pandas: Python Data Analysis Library. https://pandas.pydata.org/, 2018.

[13] Pecan Street. http://www.pecanstreet.org/, June 2018.

[14] PlantPredict. https://plantpredict.com/, June 2018.

[15] PVEducation.org. https://www.pveducation.org/pvcdrom/properties-of-sunlight/air-mass, June 2018.

[16] PVoutput.org. http://pvoutput.org/, June 2018.

[17] scikit-learn: Machine Learning in Python. http://scikit-learn.org/stable/index.html, June 2018.

[18] Weather Underground API. https://www.wunderground.com/weather/api/, June 2018.

[19] Weather.gov Terms. http://www.weather.gov/bgm/forecast_terms, 2018.

[20] S. Achleitner, A. Kamthe, T. Liu, and A. Cerpa. SIPs: Solar Irradiance Prediction System. In *IPSN*, April 2014.

[21] Y. Akiyama, Y. Kasai, M. Iwata, E. Takahashi, F. Sato, and M. Murakawa. Anomaly Detection of Solar Power Generation Systems Based on the Normalization of the Amount of Generated Electricity. In *AINA*, March 2015.

[22] R.W. Andrews, J.S. Stein, C. Hansen, and D. Riley. Introduction to the Open Source pvlib for Python Photovoltaic System Modelling Package. In *IEEE Photovoltaic Specialist Conference*, 2014.

[23] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F.J. Martinez de Pison, and F. Antonanzas-Torres. Review of Photovoltaic Power Forecasting. *Solar Energy*, 136, October 2016.

[24] J. Antonanzas, N. Osorio, R. Escobar, R. Urraca, F.J. Martinez de Pison, and F. Antonanzas-Torres. Review of Photovoltaic Power Forecasting. *Elsevier Solar Energy*, 136, October 2016.

[25] K. Armel, A. Gupta, G. Shrimali, and A. Albert. Is Disaggregation the Holy Grail of Energy Efficiency? the Case of Electricity. *Energy Policy*, 52(1), January 2013.

[26] L. Ayompe, A. Duffy, S. McCormack, and M. Conlon. Validated Real-time Energy Models for Small-scale Grid-connected PV-systems. *Energy*, 35(10):4086–4091, 2015.

[27] M. Benghanem and A. Mellit. Radial Basis Function Network-based Prediction of Global Solar Radiation Data: Application for Sizing of a Stand-alone Photovoltaic System at Al-Madinah. *Energy*, 35(9), 2010.

[28] Bidgely. `http://bidgely.com`, May 2015.

[29] M. Blanco-Muriel, D. Alarcon-Padilla, T. Lopez-Moratalla, and M. Lara-Coira. Computing the Solar Vector. *Solar Energy*, 70(5):431–441, 2001.

[30] P. Chakraborty, M. Marwah, M. Arlitt, and N. Ramakrishnan. Fine-grained Photovoltaic Output Prediction using a Bayesian Ensemble. In *AAAI*, July 2012.

[31] D. Chen, Sean Barker, A. Subbaswamy, D. Irwin, and P. Shenoy. Non-Intrusive Occupancy Monitoring using Smart Meters. In *BuildSys*, November 2013.

[32] D. Chen and D. Irwin. Black-box Solar Performance Modeling: Comparing Physical, Machine Learning, and Hybrid Approaches. In *Greenmetrics*, June 2017.

[33] D. Chen and D. Irwin. SunDance: Black-box Behind-the-Meter Solar Disaggregation. In *e-Energy*, May 2017.

[34] D. Chen and D. Irwin. Weatherman: Exposing Weather-based Privacy Threats in Big Energy Data. In *BigData*, December 2017.

[35] D. Chen, D. Irwin, P. Shenoy, and J. Albrecht. Combined Heat and Privacy: Preventing Occupancy Detection from Smart Meters. In *PerCom*, March 2014.

[36] D. Chen, S. Iyengar, D. Irwin, and P. Shenoy. SunSpot: Exposing the Location of Anonymous Solar-powered Homes. In *BuildSys*, November 2016.

[37] M. Diagne, M. David, P. Lauret, J. Boland, and N. Schmutz. Review of Solar Irradiance Forecasting Methods and a Proposition for Small-scale Insular Grids. *Renewable and Sustainable Energy Reviews*, 27, November 2013.

[38] A. Dolara, S. Leva, and G. Manzolini. Comparison of different physical models for pv power output prediction. *Solar Energy*, 119:83–99, 2015.

[39] Energy.gov Heating and Cooling. `https://energy.gov/public-services/homes/heating-cooling`, Accessed June 2017.

[40] N.A. Engerer and F.P. Mills. Kpv: A Clear-sky Index for Photovoltaics. *Solar Energy*, 105:670–693, July 2014.

[41] C. Granger. Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrics*, 37(3), 1969.

[42] A. Hammer, D. Heinemann, C. Hoyer, R. Kuhlemann, E. Lorenz, R. Muller, and H. Beyer. Solar Energy Assessment using Remote Sensing Technologies. *Remote Sensing of Environment*, 86:423–432, 2003.

[43] B. Hu. Solar Panel Anomaly Detection and Classification. Technical report, University of Waterloo, 2012.

[44] R. Inman, H. Pedro, and C. Coimbra. Solar Forecasting Methods for Renewable Energy Integration. *Elsevier Progress in Energy and Combustion Science*, 39, December 2013.

[45] S. Iyengar, S. Lee, D. Sheldon, and P. Shenoy. SolarClique: Detecting Anomalies in Residential Solar Arrays. In *ACM COMPASS*, June 2018.

[46] S. Iyengar, N. Sharma, D. Irwin, P. Shenoy, and K. Ramamkritham. SolarCast - A Cloud-based Black Box Solar Predictor for Smart Homes. In *BuildSys*, 2014.

[47] J. John. US Smart Meter Deployments to Hit 70M in 2016, 90M in 2020. In *GreenTechMedia*, October 26th 2016.

[48] J. St. John. 50 Million U.S. Smart Meters and Counting. In *GreenTech Grid*, September 16th 2014.

[49] J. St. John. Bidgely Thinks Algorithms Can Replace Hardware to Capture the Impact of Rooftop Solar, July 8th 2014.

[50] E. Kara, M. Tabone, C. Roberts, S. Kiliccote, and E. Stewart. Poster Abstract: Estimating Behind-the-meter Solar Generation with Existing Measurement Infrastructure. In *BuildSys*, November 2016.

[51] F. Kasten and G. Czeplak. Solar and Terrestrial Radiation Dependent on the Amount and Type of Cloud. *Solar Energy*, 24(2), October 1980.

[52] A. Kaur, L. Nonnenmacher, and C. Coimbra. Net Load Forecasting for High Renewable Energy Penetration Grids. *Elsevier Energy*, 114, November 2016.

[53] W. Kleiminger, C. Beckel, T. Staake, and S. Santini. Occupancy Detection from Electricity Consumption Data. In *BuildSys*, November 2013.

[54] J. Kleissl. *Solar Energy Forecasting and Resource Assessment.* Academic Press, Waltham, Massachusetts, 2013.

[55] E. G. Laue. The Measurement of Solar Spectral Irradiance at Different Terrestrial Elevations. *Solar Energy*, 13, 1970.

[56] J. Malof, R. Hou, L. Collins, K. Bradbury, and R. Newell. Automatic Solar Photovoltaic Panel Detection in Satellite Imagery. In *ICRERA*, November 2015.

[57] S. Marcacci. US Solar Energy Capacity Grew An Astounding 418% From 2010-2014, April 24th 2014.

[58] C. Martinez-Anido, B. Botor, A. Florita, C. Draxi, and S. Lu. The Value of Day-ahead Solar Power Forecasting Improvement. *Elsevier Solar Energy*, 129, May 2016.

[59] C. Marty and R. Philipona. The clear-sky index to separate clear-sky from cloudy-sky situations in climate research. *Geophysical Research Letters*, 27(17), September 2000.

[60] Joshua S. Stein Matthew J. Reno, Clifford W. Hansen. Global Horizontal Irradiance Clear Sky Models: Implementation and Analysis. Technical report, Sandia National Laboratories, March 2012.

[61] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting Consumer Privacy from Electric Load Monitoring. In *CCS*, October 2011.

[62] S. Mekhilef, R. Saidur, and M. Kamalisarvestani. Effect of Dust, Humidity and Air Velocity on Efficiency of Photovoltaic Cells. *Renewable and Sustainable Energy Reviews*, 16(5), 2012.

[63] R. Mohan, T. Cheng, A. Gupta, V. Garud, and Y. He. Solar Energy Disaggregation using Whole-House Consumption Signals. In *NILM Workshop*, June 2014.

[64] R. Mohan, T. Cheng, A. Gupta, and Y. He. Solar Energy Disaggregation using Whole-house Consumption Signals. In *NILM Workshop*, June 2014.

[65] R. Mohan, C. Hsien-Teng, A. Gupta, Y. He, and V. Garud. Bidgely, inc., solar Energy Disaggregation Techniques for Whole-house Energy Consumption Data. Technical Report WO2015073996-A3, U.S. Patent Office, October 2015.

[66] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin. Private Memoirs of a Smart Meter. In *BuildSys*, November 2010.

[67] D. Myers. Cloudy Sky Version of Bird's Broadband Hourly Clear Sky Model. In *Annual conference of the American Solar Energy Society (SOLAR)*, July 2006.

[68] D. Perera. Smart Grid Powers Up Privacy Worries. in *Politico*. `http://www.politico.com/story/2015/01/energy-electricity-data-use-113901`, January 1st 2015.

[69] Onzo. `http://www.onzo.com/`, May 2015.

[70] PlottWatt. `https://plotwatt.com/`, May 2015.

[71] Massachusetts CEC, Production Tracking System. `http://www.masscec.com/production-tracking-system`, January 2016.

[72] A. Reinhardt, G. Konstantinou, D. Egarter, and D. Christin. Worried about Privacy? let Your PV Converter Cover Your Electricity Consumption Fingerprints. In *SmartGridComm*, November 2014.

[73] R. Ross. Flat-Plate Photovoltaic Array Design Optimization. In *IEEE Photovoltaic Specialists Conference*, 1980.

[74] C. Schwingshackl, M. Petitta, J.E. Wagner, G. Belluardo, D. Moser, M. Castelli, M. Zebisch, and A. Tetzlaff. Wind Effect on PV Module Temperature: Analysis of Different Techniques for an Accurate Estimation. *Energy Procedia*, 40, 2013.

[75] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting Solar Generation from Weather Forecasts Using Machine Learning. In *SmartGridComm*, October 2011.

[76] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting Solar Generation from Weather Forecasts Using Machine Learning. In *SmartGridComm*, October 2011.

[77] C. Silva, L. Lim, D. Stevens, and D. Nakafuji. Probabilistic Models for One-Day Ahead Solar Irradiance Forecasting in Renewable Energy Applications. In *ICMLA*, December 2015.

[78] E. Skoplaki and J. Palyvos. On the Temperature Dependence of Photovoltaic Module Electrical Performance: A Review of Efficiency/Power Correlations. *Solar Energy*, 83(5), 2009.

[79] J.S. Stein. The Photovoltaic Performance Modeling Collaborative (PVPMC). In *IEEE Photovoltaic Specialist Conference*, 2012.

[80] Sunrise/Sunset Algorithm. `http://williams.best.vwh.net/sunrise_sunset_algorithm.htm`, January 2016.

[81] Sunset and Sunrise Times API. `http://sunrise-sunset.org/`, January 2016.

[82] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, 10(5), 2002.

[83] Smart Meter Implementation Programme. `https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/42737/1480-design-requirement-annex.pdf`.

[84] H. Umland. *A Short Guide to Celestial Navigation.* 1997.

[85] A. Vasel and F. Iakovidis. The Effect of Wind Direction on the Performance of Solar PV Plants. *Energy Conversion and Management*, 153, December 2017.

[86] Voluntary Code of Conduct (VCC). Technical report, U.S. Department of Energy, January 12 2015.

[87] C. Voyant, G. Notton, S. Kalogirou, M. Nivet, C. Paoli, F. Motte, and A. Fouilloy. Machine Learning Methods for Solar Radiation Forecasting: A Review. *Renewable Energy*, 105, May 2017.

[88] W. Yang, N. Li, Y. Qi, W. Qardaji, S. McLaughlin, and P. McDaniel. Minimizing Private Data Disclosures in the Smart Grid. In *CCS*, October 2012.

[89] S. Younes and T. Muneer. Comparison between Solar Radiation Models based on Cloud Information. *International Journal of Sustainable Energy*, 26(3), 2007.

[90] M. Zeifman and K. Roth. Nonintrusive Appliance Load Monitoring: Review and Outlook. *IEEE Transactions on Consumer Electronics*, 57(1), February 2011.