Black-box Solar Performance Modeling: Comparing Physical, Machine Learning, and Hybrid Approaches

Dong Chen and David Irwin
University of Massachusetts Amherst

ABSTRACT

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. Solar models are useful for a variety of solar energy analytics, including indirect monitoring, forecasting, disaggregation, anonymous localization, and fault detection. Significant recent work focuses on learning "black box" models, primarily for forecasting, using machine learning (ML) techniques, which leverage only historical energy and weather data for training. Interestingly, these ML techniques are often "off the shelf" and do not incorporate well-known physical models of solar generation based on fundamental properties. Instead, prior work on physical modeling generally takes a "white box" approach that assumes detailed knowledge of a deployment. In this paper, we survey existing work on solar modeling, and then compare black-box solar modeling using ML versus physical approaches. We then i) present a configurable hybrid approach that combines the benefits of both by enabling users to select the parameters they physically model versus learn via ML, and ii) show that it significantly improves model accuracy across 6 deployments.

1. INTRODUCTION

The penetration of intermittent solar power in the grid 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 2× from 2009 to 2015 (from ∼$8/W to ∼$4/W) [17]. As a result, the return on investment for "going solar" in many locations is now less than five years [24]. 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. However, this increasing solar penetration is placing pressure on grid operators, which schedule generators to maintain a balanced supply and demand. Even when aggregated across many deployments over a large region, solar generation is more stochastic than aggregate demand, since changes in 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 demand).

The increasing impact of solar on the grid has motivated a strong interest in developing custom performance models that *estimate a deployment's real-time solar output based on its unique location, dynamic and static physical characteristics, and weather conditions*. Solar performance models are useful for a variety of energy analytics, including indirect solar monitoring [16], solar forecasting [9, 33], "behind the meter" solar disaggregation [28, 22, 13], anonymous localization [14], and fault detection [19, 7]. Significant recent work focuses on learning "black box" models, primarily in the context of forecasting [9, 33], using machine learning (ML) techniques. 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.

Interestingly, these ML techniques 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 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 [32], and has implemented them as part of the pvlib library [8]. 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. Note that our work focuses on accurately modeling the real-time output of existing solar deployments under current conditions, and not the potential output of future solar deployments. Many tools exist, such as PVWatts [2] and SAM [5], that estimate solar potential using white-box models.

Prior work also leverages stochastic ML techniques to estimate irradiance, and then uses white-box models for estimating conversion efficiency [23]. An example of such a white-box tool is PlantPredict.[1] Unfortunately, while these white-box approaches have high accuracy, gathering this information at large scales for millions of small-scale deployments is infeasible. As a result, these tools are typically only used for utility-scale solar farms. Of course, while less well-studied, black-box physical modeling using the same fundamental properties is also possible: as we discuss, it simply requires determining the model parameters by finding the values that best fit the data. Such physical modeling generally requires much less data to calibrate (akin to training) than ML modeling, as the physical models embed detailed information about the relation-

Greenmetrics '17, June 5, 2017, Urbana-Champaign, IL USA.

[1] https://plantpredict.com/

ship between its 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 [10, 15]. We examine both a canonical "pure" machine learning technique from prior work [28] and a "pure" analytical approach from prior work, which leverages several well-known physical properties of solar generation [13]. 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 [27]. 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.

Unfortunately, ML techniques require a significant amount of historical data to train an accurate model. Prior work requires anywhere from months to years [20, 31], while a recent survey states that at least 30 days of data is necessary to train a reasonably accurate model [9]. However, historical data is generally not available for either new deployments or deployments that do not continuously monitor and store the data. In contrast, black-box physical models require calibration, and not training, and thus require significantly less historical data, since the physical properties that govern solar generation are well-known. Physical models can often be calibrated using only a few datapoints, mitigating the need for a monitoring infrastructure to gather and store data for training.

In this paper, 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 a hybrid solar performance modeling technique that combines 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. Importantly, our hybrid approach is configurable: it can either apply a physical model to quantify the effect of an input parameter on solar output or it can learn the effect via ML from training data. 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 [28] and a pure physical approach, which combines several well-known physical models of solar generation.

Hybrid Solar Performance 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).

Implementation and Evaluation. We implement the ML, physical, and hybrid modeling approaches above and evaluate their accuracy across 6 solar deployments with widely different characteristics. 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.

2. BACKGROUND

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 [9, 33, 23] that cite well over one hundred papers on the topic. Unfortunately, this prior work generally conflates modeling 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 [28]. 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. [29] and is in production use [21].

As discussed in Section 1, prior work on physical modeling generally takes a white-box approach [15, 10]. 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.

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 [31]. 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 [20]. Our pure ML-based approach normalizes these variables without using detailed physical models of the system [28, 29]. 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 paper, 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 en-

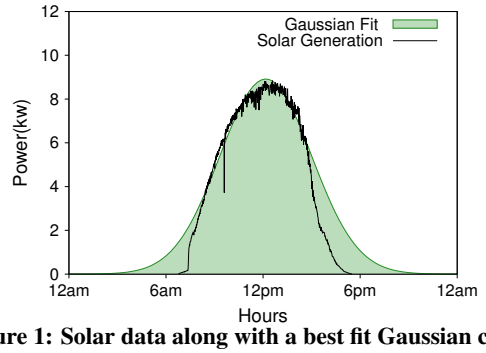

Figure 1: Solar data along with a best fit Gaussian curve.

ables 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 [31, 28, 11]. 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 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.

2.2 Black-box Physical Modeling

Our approach to physical modeling leverages several well-known relationships that govern solar generation. Note that our approach is adapted from an approach we proposed in recent work [13]. However, our prior work, as above, focuses on solar disaggregation of net meter data and not solar performance modeling of pure solar data. 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 [26]. 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.

Of course, solar modules cannot convert all the available solar energy into electricity. Their efficiency is based on the type of module, e.g., poly- versus mono-crystalline, as well as their orientation and tilt. The simple well-known equation below describes the amount of power a solar module generates based on its tilt (β) and orientation (ϕ) relative to the Earth's surface, and the Sun's zenith (Θ) and azimuth (α) angles (which are a direct function of the location and time [12]). Assuming clear skies, $I_{incident}$ is the clear sky solar irradiance, and k is a module-specific parameter that is combines conversion efficiency (as a percentage) and module size (in m^2). Similar expressions exist for deployments that track the sun, or consist of multiple modules with different tilts and orientations.

$$P_s(t) = I_{incident}(t) * k * [\cos(90 - \Theta) * \sin(\beta) * \cos(\phi - \alpha) + \sin(90 - \Theta) * \cos(\beta)] \quad (1)$$

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 prior work, we have shown that this searching method results in highly accurate values for k and the orientation and tilt angles [13].

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. 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 $\sim$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.

$$k'(t) = k * (1 + c * (T_{baseline} - T_{air}(t))) \quad (2)$$

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 [30, 34] 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 [25]. For example, one well-known cloud cover model is below [4].

$$I_{incident}/I_{clearsky} = (1 - 0.75n^{3.4}) \quad (3)$$

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 [6]. 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 [18]. 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 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 1) are equivalent, since the effect of the physical characteristics cancel out. Recent work refers to this value as the clear sky photovoltaic index [16]. We could continue to adjust our model downwards based on physical models for other conditions, such as humidity, air velocity, and dust buildup [27]. 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 [13]. 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.

3. A BLACK-BOX HYBRID MODEL

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 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 2.1, that includes as input features any relevant parameters not included in the physical models. However, a key difference relative to Section 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 [16]. 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 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 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 [13], the latter is primarily due to weather effects that are general and not dependent on specific physical deployment characteristics.

In this paper, we use the same classifier (SVM-RBF) in our hybrid ML model as we do in the pure ML model [28] 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.

4. 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 [28, 29] 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 [6]. Our physical model leverages the

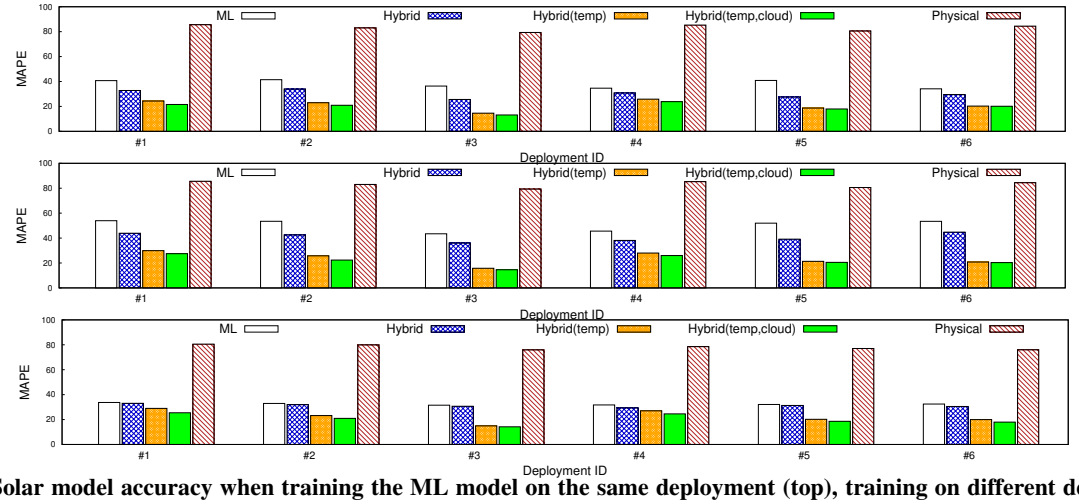

Figure 2: Solar model accuracy when training the ML model on the same deployment (top), training on different deployments on different deployments (middle), and the accuracy (when training on different deployments) during the middle of the day.

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 [28, 29], 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 follows, between the ground truth solar energy (S) and the solar energy estimated by our models (P_s) at all times t. A lower MAPE indicates higher accuracy with a 0% MAPE being a perfect model.

$$MAPE = \frac{100}{n} \sum_{t=0}^{n} |\frac{S - P_s}{S}| \quad (4)$$

5. EXPERIMENTAL EVALUATION

Figure 2 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 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 [27]. As part of future work, we are leveraging various satellite images to better quantify real-time cloud cover, such as via the HELIOSTAT method [18], 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 ap-

proach, 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 ($\sim$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.

6. CONCLUSION

This paper surveys and compares different approaches to black-box solar performance modeling. We compare a pure ML model from prior work [28, 29], a black-box physical model based on well-known relationships in solar generation [13], 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. Our black-box solar modeling has applications to a wide range of solar analytics, which we plan to explore as part of future work. Finally, our methodology is potentially generalizable to other complex physical systems where the physical effect from a subset of inputs on a dependent output variable is well-known, and independent of other inputs, which have an unknown effect.

Acknowledgements. This research is supported by NSF grants IIP-1534080, CNS-1405826, CNS-1253063, CNS-1505422, and the Massachusetts Department of Energy Resources.

7. REFERENCES

[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] Weather.gov Forecast Terms. http://www.weather.gov/bgm/forecast_terms.

[7] 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.

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

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

[10] 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.

[11] 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.

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

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

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

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

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

[17] R. Fares. Scientific American, The Price of Solar Is Declining to Unprecedented Lows, August 27th 2016.

[18] 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.

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

[20] 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.

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

[22] 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.

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

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

[25] 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.

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

[27] 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.

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

[29] 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.

[30] 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.

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

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

[33] 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.

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