Operationalizing Solar Energy Predictions for Sustainable, Autonomous IoT Device Management

Frank Alexander Kraemer, *Member, IEEE,* David Palma, *Member, IEEE,* Anders Eivind Braten, Doreid Ammar

***Abstract*—For sustainable Internet of Things (IoT) systems, solar-power prediction is an essential element to optimize performance, allowing devices to schedule energy-intensive tasks in periods with excess energy. In regions with volatile weather, this requires taking the weather forecast into account. The problem is how to provide such solar energy predictions with high accuracy for large-scale IoT systems with various devices in an autonomous way, without manual adaptation effort. We present a detailed study on machine-learning approaches for the prediction of solar power intake for large scale IoT systems. We examine which machine learning models, feature sets and sampling rates gain the best results for a medium-term forecasting horizon. We also explore an operational setting in which devices are deployed without prior data and machine learning models are re-trained for each sensor continuously as a form of online learning. Our results show that prediction errors can be reduced by 20 % compared to the state of the art, despite strong weather volatility.**

***Index Terms*—Device Management, Energy Harvesting, Solar Energy, Machine Learning.**

I. Introduction

ENERGY HARVESTING via solar panels allows wireless devices to replenish their energy buffers and is thus one element towards a sustainable, maintenance-free Internet of Things (IoT) with perpetual operation, as it removes or reduces the need to switch batteries [1]. Use cases where solar power has great potential cover a wide range of domains, like smart cities [2], [3], harbors [4] and agriculture [5]. Rainforest Connection [6], for example, creates acoustic monitoring systems to detect illegal deforestation, using recycled phones powered by solar panels. Operating sustainably, with minimal or no maintenance, is crucial for the feasibility and economic aspects of such use cases. The better the predictions, the more strategically, and hence optimally, IoT devices can act: Apart from adjusting their sensing intervals, they can schedule energy-intensive tasks in periods of energy surplus. Such tasks can include software updates, transmission of aggregated sensing data, or re-training of machine learning models. Improved energy management helps to minimize the required energy buffer and solar panel size of IoT devices, making them simpler, easier to deploy and less obtrusive. This makes systems cheaper, or possible at all, and further facilitates approaches that even integrate solar energy supplies directly onto chips [7].

F.A. Kraemer (kraemer@ntnu.no), D. Palma, and A.E. Braten are with the Department of Information Security and Communication Technology, Norwegian University of Science and Technology, NTNU, Norway.

D. Ammar is with the data science team at emlyon business school, France.

To plan energy budgets effectively also under volatile weather conditions, Sharma et al. [8] and Renner et al. [9] argue that IoT devices require access to solar energy predictions that also take the weather forecast into account. With the availability of new machine learning methods and computational power in general, this leads to the question of how these methods can improve the prediction of solar energy intake. While machine learning has been applied for that purpose in the domain of renewable power, less attention has been paid to solar prediction for IoT nodes, which require medium-term predictions (up to several days ahead) for energy budget planning. There is also the challenge of heterogeneity in large-scale deployments, where devices operate in different settings, for example regarding their position towards the sun or local obstacles such as trees or buildings. To avoid manually modeling these differences, which is prohibitive for large IoT installations, prediction models should hence be individual, that is, per IoT device and work autonomously. We envision that IoT devices are supplied with solar energy predictions as part of the device management, offered by edge or cloud services. For such an approach to work efficiently at scale, it is required that the operation of the prediction models is feasible. Instead of manual adjustment and tuning, predictions must refer directly to individual sensors' expected energy, based on the previous energy intake and on features that are easy to acquire, like a public weather forecast.

In this work, we explore the use of various machine learning techniques combined with public weather forecasts for the prediction of solar energy. This is the first work that discusses the issue of machine learning for medium-term solar energy prediction for IoT devices, paying the necessary attention to operational aspects. We have previously examined how different machine learning methods can be used in a constrained sensor setting [10]. We now go further and (i) present the performance of various machine learning techniques, (ii) conduct an ablation study to identify the most useful features, (iii) introduce scaled forecast metrics that allow us to compare prediction performance independent of seasonal changes, (iv) study the influence of sampling frequency (i.e., how often solar energy should be sampled as training data), and (v) investigate how the accuracy of the predictions develops after a deployment in an operational setting. Our results show that the machine learning models based on weather forecasts outperform other methods by more than 20 %.

We start with an introduction to solar harvesting in IoT in Sect. II that also provides the system context, and a study of related work around energy planning and solar energy prediction in Sect. III. This is followed by the discussion of our method, which highlights the techniques for model and

feature selection and introduces the metrics for our evaluation. We then systematically analyze the type of machine learning models, the suitability of features for the predictions, and the sensitivity to the sampling frequency in Sect. V. In Sect. VI, where we combine all insights, we explore and discuss the performance of the prediction models in an operational setting, and compare them to the state of the art.

II. Solar Energy Harvesting Prediction in IoT

We first provide an overview of the significance of solar energy prediction in IoT and explain then how energy predictions can be integrated into device management that constitutes our system context.

A. The Need for Solar Energy Predictions

The challenge with energy harvesting is its stochasticity, and that energy is not always available when needed by an application [11]. Some of the stochasticity is compensated by energy buffers like batteries or super capacitors. But this only helps to a certain degree, as the required capacity of the buffers must be limited to reduce device cost and physical dimensions [12]. Therefore, energy planning is required [13], [14], [15], aiming at aligning the application energy demand closer with the availability of harvestable energy. This can be achieved for instance by adjusting the duty cycle of the application, allocating tasks to nodes with better energy budgets [13], or allowing a tradeoff between sensing accuracy and energy consumption [16]. Some tasks, such as training of machine learning models, are also tolerant to delays and can hence be scheduled in time slots where more energy is available or when the demand from other tasks is lower.

A basis for effective energy budget planning is the availability of highly accurate predictions for the incoming energy [3], [12], [13], [14], [15], [17]. Available solar energy often follows quasi-cyclic diurnal patterns [11], which motivates approaches that estimate the incoming solar energy based on historic data, which we will review in Sect. III. While such approaches may be suitable for short time horizons (up to 3 hours) or long-term horizons (beyond several days), they are not sufficient for medium-term horizons (3 hours to 3 days), as Sharma et al. [8] and Renner [9] conclude. This is because the arriving energy is not only dependent on the position of the sun relative to the solar panel but also the coverage of the sky with clouds at various levels, which can vary considerably with the local weather conditions. Sharma et al. [8] and Renner [9] therefore highlight the importance of also taking the weather forecast into account, and report significant improvements in accuracy compared with approaches that only rely on historic data.

Another aspect of sustainable and cost-efficient IoT solutions is operational: due to the system scale, devices must operate autonomously. Prediction models must not require manual fitting or oversight for the individual devices. We therefore turn our attention to off-the-shelf machine learning methods and want to explore how they can improve the prediction accuracy of solar prediction when taking the weather forecast into account. We focus in our study on a medium-term time horizon, as this time horizon is significant for energy budget planning in devices with typical energy buffer sizes, and approaches based soley on historic data do not perform well for this horizon. The novelty of our work is the thorough exploration of machine learning options, selection of features and sampling frequencies to achieve better prediction performances than the current approaches as the basis for IoT device energy budget management.

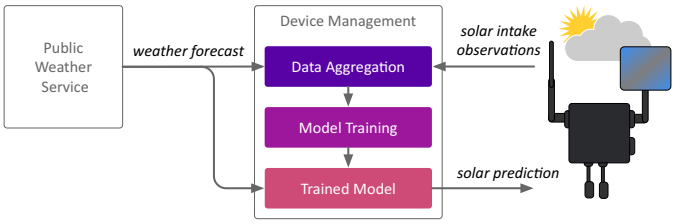

Fig. 1. Operational setting, including IoT device and device management platform.

B. System Context and IoT Device Management

We suggest to include the weather forecasts into prediction models as part of the device management [18]. This allows the training of machine learning models for solar energy prediction and the actual prediction to be executed off-device, in cloud or edge hardware. Fig. 1 provides an overview of the system, explained in the following. To allow the device management to be specific to the individual IoT devices' settings and micro-environments, the device management distinguishes different device instances:

- IoT devices record their individual solar intake observations and send them to the device management module. In Sect. V-C, we will discuss the significance of the reporting frequency.
- The data aggregation step combines the solar intake observation with the weather forecast data of a region covering a device, further explained in Sect. V-A.
- This data is the input for training prediction models, which is the main focus of our work. Which input features to use is discussed in Sect. V-B, and the amount of training data to store is discussed in Sect. VI-A. In principle, models can be retrained with the arrival of every new observation, but for most use cases a daily training is sufficient, as shown in our final evaluation Sect. VI-B. We note that the computational effort for retraining a model is manageable compared with the typical tasks of device management, further discussed in Sect. VI-C.
- The trained model is then used to provide solar energy predictions to the device, taking weather predictions as input. The computational effort for this prediction is negligible. Depending on the planning mode of the device, predictions can be provided every hour, or for instance at midnight for the entire next day.

In this paper, we focus on the performance of the prediction models and which features and frequency of data they require as input.

III. Related Work

There is a wide range of approaches for solar power prediction, which vary in terms of input data, forecast horizon and temporal resolution. Wireless sensor nodes are typically constrained in computation, which motivates approaches that use the exponentially weighted moving average (EWMA), like Kansal et al. [12]. They divide a day d into N time slots (for instance $N = 48$) and observe the energy intake x in each time slot n. For each time slot n they iteratively compute the EWMA $\bar{x}$ using

$$\bar{x}_n^{(d)} = \alpha \bar{x}_n^{(d-1)} + (1-\alpha) x_n^{(d)},$$

where $\bar{x}_n^{(d-1)}$ is the averaged value of time slot n from the previous day. The prediction of a slot in the future is the EWMA of the observations for that slot on previous days, making use of the diurnal pattern of outdoor solar energy. This type of model is attractive for embedded systems as it only requires previous observations that can be accumulated locally, and the EWMA only requires to store one value for each of the N time slots. However, the performance of such approaches depends on the stability of weather conditions. As an improvement, Piorno et al. [19] propose a weather-conditioned moving average (WCMA), which corrects EWMA-based average values with a factor that indirectly depends on the weather. This factor is calculated based on the solar intake of the day so far, compared to that of the previous days, and hence limits the prediction to a short-term horizon, only a few time slots ahead. UD-WCMA [20] poses another improvement by choosing weighting parameters autonomously. However, the short forecasting horizon remains. Saidi et al. [21] use a Kalman fiter with an autoregressive model to predict solar energy intake, but also this approach only considers a short forecasting horizon, until the next time slot.

Persistence models are another type of forecasting model only considering past observations, which are used as baselines in solar forecasting for the power grid [22], [23]. Instead of averaging over past observations, they take the value from the previous day $d-1$ as forecast for day d. The *smart* persistence model corrects the historic observations with the diurnal variance of the solar irradiance [24]. The global horizontal irradiance (GHI) represents the potential amount of energy that can be harvested by a solar panel. It depends on the angle between the sun and the plane of the solar panel and the travel length through the atmosphere. The GHI can be calculated by using a model that estimates the clear sky global irradiance directly, such as the simplified Solis model described by Ineichen et al. [25], which is coherent for the solar elevation angles at most latitudes and calculated as

$$GHI_S = I'_o \cdot e^{\left(-\frac{\tau}{\sin^g(h)}\right)} \cdot \sin(h), \tag{1}$$

where I'_o is the extraterrestial irradiance modified by the atmospheric radiation component, h is the solar elevation angle, τ is the global total optical depth, and g is the corresponding fitting parameter for the GHI. Therefore, future energy intake can be predicted using

$$\hat{E}_{in}(t+24h) = \begin{cases} E_{in}(t)\frac{GHI_S(t+24h)}{GHI_S(t)}, & GHI_S(t) > 0.1 \\ E_{in}(t), & GHI_S(t) \leq 0.1 \end{cases} \tag{2}$$

where E_{in} is the observed energy intake at a given time t, and GHI_S is the irradiance given by the simplified Solis clear sky model. The used threshold of 0.1 can be adjusted to avoid unrealistic high levels of irradiance at sunrise and sunset.

The above-mentioned techniques, which only rely on past observations, do not perform well in locations with volatile weather and for medium-term prediction horizons, as Sharma et al. [8] point out. They instead propose the inclusion of weather forecasts in the form of cloud coverage C for the prediction of solar power P, and formulate a model $P_{sun} = P_{max} \cdot (1-C)$, where P_{max} is approximated by a quadratic model with coefficients for each month, that are derived by manually selecting sunny days. While this work indicates the benefits of including weather forecasts, it has the drawback of manually fitting models, which is not realistic in a large-scale IoT setting with heterogeneous devices and environments.

Renner [26] combines cloud-cover information of the weather forecast with an EWMA-based model. This combination is similar to that of WCMA, but uses actual weather forecasts. IoT devices are provided with access to the cloud coverage forecast (CCF) for each time slot, and use this value to determine what corresponds to the clear-sky value for each time slot. The EWMA of these values is then used as basis that is again combined with the CCFs to compute the actual predictions. Together with the previously described methods EWMA and smart persistence (SP), we use this approach – CCF – as another baseline in Sect. VI.

Another domain for which solar energy prediction is relevant is the power grid and renewable energy. In this domain, machine learning techniques are much more common, and used at different forecasting horizons, from short-term prediction in terms of minutes to react to fluxes of solar power, to long-term predictions to reason about the feasibility of solar installations. Voyant et al. [27] provide a comprehensive review of machine learning methods for solar radiation forecasting. For instance, Bacher et al. [28] use autoregressive models and find that for medium-term horizons, numerical weather predictions increase the accuracy considerably. Many approaches employ various machine learning techniques with the use of distinct prediction variables as input. Yadav et al. [29] provide an overview of neural networks as prediction models, while Sharma et al. [30] study machine learning based on weather forecasts, including the sky coverage, using support vector machines. Alternatively, Dahl and Bonilla [31] use Gaussian Processes as a forecasting model, which can also quantify the confidence level in the prediction estimate. Benali et al. [32] use separate models for the different components of radiation. Similarly, the use of blended learning with a mixture of models has been addressed by several authors [33], [34], [35].

Tang et al. [36] use an approach based on the least absolute shrinkage and selection operator (LASSO) for the short-term solar prediction. In Wang et al. [37] this approach is extended by long short-term memory units (LSTM) of neural networks into a mixture model based on different weather types. They use weather observations instead of the public weather forecasts, and identify temperature and humidity as valuable features. This is probably due to their short forecasting horizon,

and the absence of cloudiness in the weather observations, as these turn out to be the most important prediction features for medium-term horizons in our analysis.

Altogether, while there is considerable attention on solar forecasting, there is a lack of discussion on the operational aspects of the prediction models relevant for IoT, which are crucial for making them work in a scalable and autonomous way for constrained devices.

IV. Methodology

The main goal of our work is to identify machine learning models and corresponding features to increase the performance of solar energy prediction models. To that end, we start with a set of standard machine learning models which we test on an exhaustive set of feature combinations, resulting in a selection of models and features based on their performance. In the following, we discuss the significant aspects of our research method. This includes data collection, the chosen metrics, and the approaches for feature selection, prevent data leakage and ensure results under realistic conditions.

A. Solar Energy Data Collection

We use real data that we collected over more than two years, so that we were able to cover all seasons throughout a year more than once. The source of our training data is a solar panel with horizontal orientation on the top of a university building in Trondheim, Norway. We measured the voltage on a resistor and logged the data every minute, which resulted in a data set covering two years, starting in October 2017.[1] We denote each measurement with $a_i^{(d)}$, where d is the day and $i \in I_d$ the index of the value within the day. Fig. 2 shows the solar energy intake over three days, which also shows their extreme volatility from day to day.

Since measurement values are taken in regular intervals Δt (one minute in our raw data), the total energy collected during a day can be approximated by summing over the individual values $a_i^{(d)}$,

$$E(d) = \gamma \cdot \Delta t \sum_{i \in I_d} a_i^{(d)},$$

where γ is a factor including the solar panel's size, its efficiency, and the efficiency of the power converter. The specific value of this factor is not relevant here, as it depends on each individual IoT device and is an internal variable in the prediction process. Fig. 3 shows $E(d)$ over two years and reveals large seasonal difference between winter and summer. We also calculate the exponentially weighted moving average (EWMA) of the solar intake $\bar{E}(d)$ for each day

$$\bar{E}(d) = \begin{cases} E(d), & d = 1 \\ \alpha \cdot E(d) + (1-\alpha) \cdot \bar{E}(d-1), & d > 1. \end{cases} \quad (3)$$

Fig. 3 shows $\bar{E}(d)$ with $\alpha = 0.095$, which also illustrates the differences within the same month from year to year. The average energy harvested in June 2018, for example, deviates significantly from the same period in the year 2019.

[1]The data set of this study is available at https://github.com/falkr/iot-solar-energy-prediction.

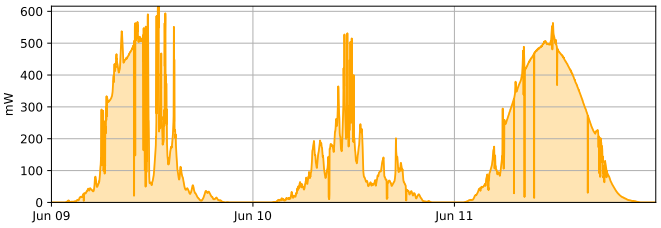

Fig. 2. Three days in June 2018 illustrating the volatility of solar energy intake from day to day.

B. Prediction Metrics

Although various metrics to evaluate solar energy forecasting exist [38], their scaling is significant to ensure comparability, as we will show next. Two standard metrics for the prediction performance on a day d are the mean absolute error (MAE) of the individual prediction values, as well as the total absolute error for the entire day (TAE):

$$MAE(d) = \frac{1}{|I_d|} \sum_{i \in I} |a_i^{(d)} - p_i^{(d)}| \quad (4)$$

$$TAE(d) = \left| \sum_{i \in I_d} a_i^{(d)} - \sum_{i \in I_d} p_i^{(d)} \right|. \quad (5)$$

As the MAE also considers intra-day accuracy, a good score with the MAE also implies a good TAE. Yet, considering the TAE could reveal good predictors for the overall day that are just imprecise with their timing. For IoT energy management, we will in the end address how we can combine these two aspects, but we first need to address the problem of seasonality.

Fig. 4 shows the MAE for the smart persistence prediction model from in Sect. III. (The TAE shows similar behavior.) Both metrics are scale-dependent (see [39] for a discussion), and vary with the seasons. This is problematic for our purposes, as we do not know whether to attribute changes in the score of a prediction to changes of a predictor's quality or just seasonality. To eliminate this scale-dependency and seasonality, we consult the corresponding relative *percentage* errors MAPE and TAPE, which scale the error to the actual value $a_i^{(d)}$:

$$MAPE(d) = \frac{100}{|I_d^*|} \sum_{i \in I_d^*} \frac{\left| a_i^{(d)} - p_i^{(d)} \right|}{a_i^{(d)}} \quad (6)$$

$$TAPE(d) = \frac{100}{\sum_{i \in I_d^*} a_i^{(d)}} \left| \sum_{i \in I_d^*} a_i^{(d)} - \sum_{i \in I_d^*} p_i^{(d)} \right| \quad (7)$$

The modified set I_d^* includes only the indices of observations that are non-null, to prevent division by zero in (6). These percentage errors prevent seasonal variations, but have the drawback that errors on days with very little energy get very large, as the high variation in Fig. 4 for the MAPE shows. In line with the normed errors RMSE and ME in [26] we introduce the *scaled* mean absolute percentage error (SMAPE)

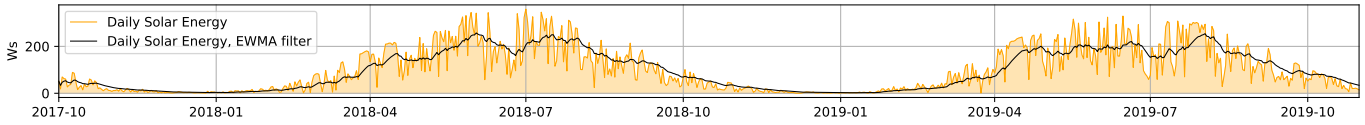

Fig. 3. Measured daily solar power intake over two years. The solid black line shows the exponentially weighted moving average (EWMA) of the daily intake with $\alpha = 0.95$, corresponding to a span of ca. 20 days. Daily values and average show the high daily and seasonal variations.

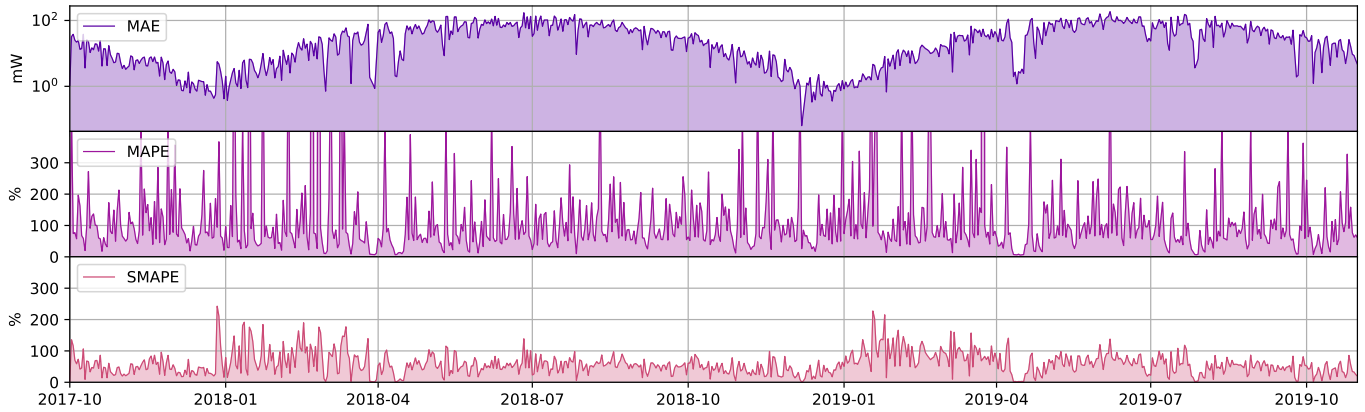

Fig. 4. Mean absolute error (MAE), mean absolute percentage error (MAPE), scaled mean absolute percentage error (SMAPE) of the smart persistence predictor. The corresponding TAE, TAPE and STAPE metrics are not shown here for brevity, but they exhibit the same characteristics as their counterparts.

and the scaled total absolute error (STAPE), which are scaled to the moving average introduced in (3)

$$SMAPE(d) = \frac{100}{\bar{E}(d)} \sum_{i \in I_d} \left| a_i^{(d)} - p_i^{(d)} \right| \tag{8}$$

$$STAPE(d) = \frac{100}{\bar{E}(d)} \left| \sum_{i \in I_d} a_i^{(d)} - \sum_{i \in I_d} p_i^{(d)} \right|. \tag{9}$$

Neither of them exhibit the challenges of the previous metrics. STAPE is hence a measure of how many percent, relative to the average intake during that time, the total energy for a day is off, while SMAPE is also taking into account how accurate the prediction is within a day. Taking percentages instead of absolute values, as in [26], has the benefit that scores can be also compared across different IoT devices. As a single-number metric we use the arithmetic mean between SMAPE and STAPE, which we for simplicity call the scaled absolute percentage error, SCAPE:

$$SCAPE(d) = \frac{1}{2}(SMAPE(d) + STAPE(d)). \tag{10}$$

This metric balances between the total prediction and the intra-day accuracy.

C. Prevention of Data Leakage

A proper split between training and test data is important to prevent data leakage and ensure applicability and generalization of the results. Since the observed solar energy intake tends to be similar from one minute to the next, a standard randomized training/test split would effectively result in data leakage. For example, a measurement from 12:00 could serve as training data and an almost identical entry from 12:01 could end up as test data. We therefore only assign only entire days to the test set. For that, days of 2018 (which we use for the first parts of the experiment) are numbered consecutively, and every fourth day is taken into the holdout test set. Data from these days will not be used to train models or tune parameters, they are only used to validate the results after models have been developed. The days for validation are evenly assigned into two of ten cross-folds.

D. Feature Selection Method

Feature selection is one of the core concepts in machine learning and involves selecting the most relevant features that yield the best model performance. In an IoT setting, omitting irrelevant features is especially interesting since it may reduce training and inference time and requires to store and transmit less data.

Many different feature selection methods exist in literature and they are being widely used [40]. An optimal feature selection method is the exhaustive feature search [41]. The main strength of the exhaustive feature selection algorithm is that it is guaranteed to find the best set of features. However, the main drawback of this algorithm is the complexity cost.

In our context, the size of the dataset, the number of required features, and the computing resources allow the exhaustive feature selection to be computationally feasible. Therefore, to identify the best set of features, we employ an exhaustive feature selection algorithm to evaluate all possible feature combinations. More specifically, we train several thousand prediction models and evaluate them independently. Following that, we conduct a study of different features, referred to as ablation study, to evaluate the performance impact of removing a given feature from the machine learning model.

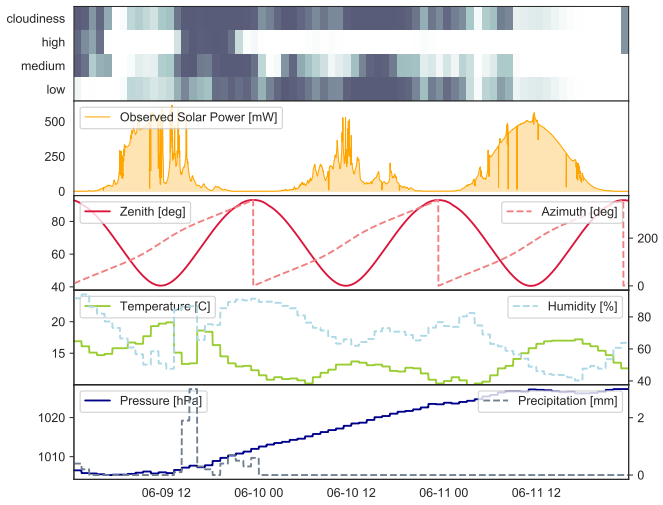

Fig. 5. Example weather forecast values for three days, together with the observed solar energy. The heatmap at the top shows the cloudiness overall and at the three levels high, medium and low. Darker shades imply more clouds.

E. Evaluation in Operational Settings

To ensure the relevance of our approach, we carried out the main evaluation of our work as a case study under quasi-realistic conditions, in an operational setting further detailed in Sect. VI-B. This means that we train the selected machine learning models with the same data they would receive if deployed in reality, and evaluate them with the metrics from Sect. IV-B.

For a comparison of our work with current state-of-the-art solutions, we selected the baselines introduced in Sect. III. Smart persistence (SP) is the standard reference for solar energy forecasting [22], [23], and EWMA is a fundamental prediction technique in wireless sensor networks [12]. The cloud coverage forecast (CFF) itself outperforms other techniques as shown in [26]. Together, these different forecasting techniques constitute a relevant baseline for our approach.

V. Weather-Based Machine Learning Models

In the following, we will discuss the preparation of training data in more detail and then proceed with the identification of suitable machine learning models and selection of the most valuable features, and close with the condsideration of the significance of the sampling intervals for the solar intake observations.

A. Weather Forecast Data

We use the weather forecast provided by the public application programming interface (API) of the Norwegian Meteorological Institute [42]. Weather forecasts are usually issued three times a day, around every 5 to 8 hours. Each issue contains a forecast for the upcoming 60 hours with an hourly resolution. We extract, for each hour, the publication and forecast timestamps, temperature, humidity, pressure, precipitation and the amount of clouds covering the sky. Fig. 5 illustrates the weather data for three sample days. Cloudiness is provided at several levels. Internally, the weather model calculates the amount of clouds at 65 vertical levels in the atmosphere. In the forecast, we use the cloudiness at four different aggregation levels: low clouds (below 2.5 km), medium clouds (2.5–5 km), high clouds (above 5 km) as well as a total cloudiness percentage, calculated from the entire stack of cloud levels.

We merge the solar data collected every minute with the hourly slots of the weather forecast. We clean the data by dropping a negligible number of days where the weather forecast could not be collected or the solar panel was out of order. Based on the timestamp, we add the solar angles zenith and azimuth for our location. This results in a data set with the features $f_1 \dots f_{10} \in F$ that represent zenith, azimuth, temperature, precipitation, pressure, humidity, cloudiness, lowclouds, mediumclouds and highclouds. The truth value for the data set is the observed solar energy intake.

B. Feature Selection

For the exhaustive search, we use machine learning models from Scikit-Learn [43], and specifically a random forest regressor (RFR) with 30 estimators, an artificial neural network (ANN) with a single hidden layer of 100 perceptrons, and a deep neural network (DNN) with three hidden layers of 30 perceptrons each. For both neural networks, we used ReLU activation functions.

We check the performance of all the three base models on different feature sets, i.e., combinations of the features f_1 to f_{10} from F. As the zenith (feature f_1) is the dominating variable describing the position of the sun, we include it in all feature sets. The set FS of all feature sets that include f_1 is then described by

$$FS = \{x \,|\, x \in \mathcal{P}(F) \wedge f_1 \in x\}, \tag{11}$$

where $\mathcal{P}(F)$ is the power set of F. This results in a total of $2^9 = 512$ feature sets.

For each model and feature combination, we calculate the performance with at least 3 crossfolds and take the average of them. Fig. 6 shows the results for all models and all feature sets. Each point shows the performance in terms of SMAPE and STAPE for a specific combination of machine learning model and feature set. The different colors distinguish the different models (RFR, ANN, DNN). We observe that there is in general a strong correlation between the SMAPE and STAPE metric. The magnification to the right shows that RFR produces the best results. This is also confirmed by the histograms in Fig. 7, which shows the distribution of errors of the different models trained with different feature sets. RFR manages to achieve the best mean and median results over all feature sets.

We have tried to improve the scores of the neural networks by tuning their hyper-parameters and applying different activation functions, optimizers, and architectures regarding the hidden layers. However, we have not been able to achieve the same robust and consistent performance as with the relatively simple RFR model, which is why we continue with the RFR models in the following.

For the ablation study, we use the RFR model due to its general good performance as shown above. Based on each of

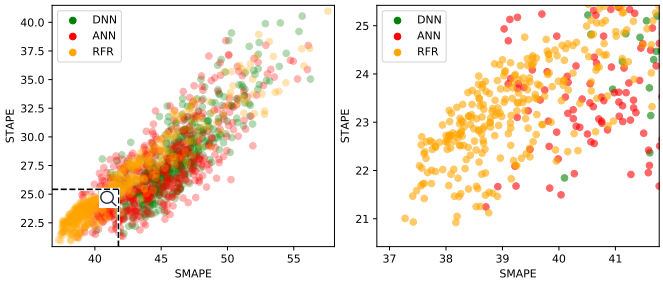

Fig. 6. SMAPE and STAPE of the various basic machine learning models for different feature sets. The right graph is a magnification as indicated by the dashed lines to the left.

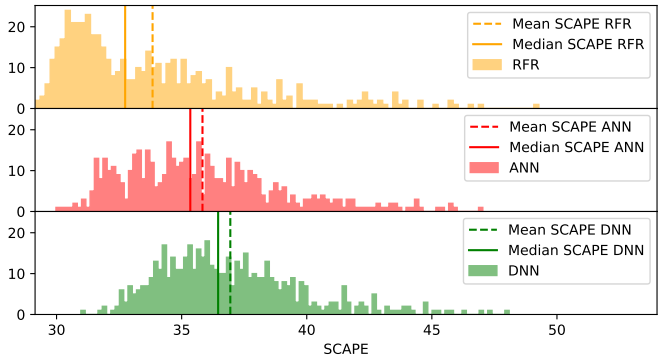

Fig. 7. Distribution of errors (SCAPE) of the three types of base models (RFR, ANN, DNN) for all 512 combinations of features. Mean SCAPE (dashed) and median (solid) errors are also shown.

the optional features $f_2 ... f_{10}$ we define pairs of feature sets, $(F^+, F^-) \in PFS_{f_i}$, where the first set F^+ includes feature f_i and the second feature set F^- does not, that means

$$PFS_{f_i} = \{(F^+, F^-) \,|\, F^+, F^- \in FS \wedge f_i \in F^+ \wedge f_i \notin F^- \wedge F^+ \setminus \{f_i\} \equiv F^-\}. \quad (12)$$

With the total of 512 feature sets in FS, there are 256 pairs for each of the optional features f_i. We then compute the mean performance of the prediction models for all pairs, using 10 crossfolds, and consider scatter plots as shown in Fig. 8. Each scatter plot includes 256 pairs. The x-coordinate is given by the mean SCAPE of the models trained using feature sets F^+ *with* f_i included. Correspondingly, the y-coordinate shows the mean SCAPE of the models trained on the feature sets F^-, i.e., *without* f_i. If a feature is useful, its inclusion should reduce the SCAPE value. Hence, points above the identity line indicate pairs where including f_i improves performance, while points below correspond to pairs where removing f_i is detrimental. Points close to the identity line show that the given feature f_i has little influence on performance. The mean SCAPE for each plot is depicted by a black cross in the intersection between the two dashed lines.

1) Humidity: For the humidity feature in Fig. 8, feature pairs are close to the identity line, especially for good models. There are some improvements for models that score worse in

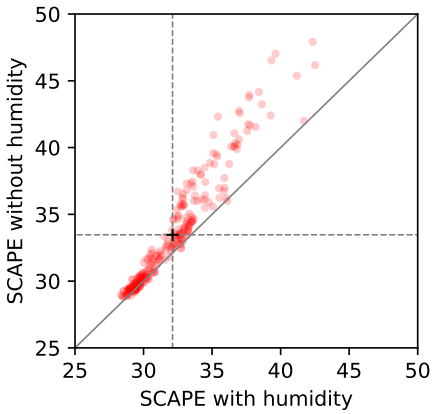

Fig. 8. Ablation study for the humidity feature. Each point denotes a feature set pair, the x-coordinate showing the SCAPE *with* humidity, the y-axis *without*. The dashed lines and cross show the mean values.

general, however they yield significantly poorer results than the best feature sets not including this feature. This reveals that humidity is not a useful feature.

The scatter plots for the other features are shown in Fig. 9.

2) Precipitation: Similarly to humidity, the results for the precipitation feature ablation lie mostly on the identity line, with the exception for very few of the worse performing models. This is also confirmed by the mean SCAPE result which coincides with the identity line, proving that precipitation should not be considered as a feature.

3) Pressure: The pressure feature has most pairs along or even below the identity line, indicating that this feature has little usefulness, being even responsible for decreasing performance in some feature combinations. In fact, the obtained mean SCAPE is located below the identity line, confirming that using pressure as a feature is overall detrimental to the models' quality.

4) Azimuth and Temperature: Azimuth and temperature are both valuable features, showing loss of performance when removed from the feature set. In particular, we found a dependency between these two features. In the azimuth plot, the orange markers are of model pairs that do not include temperature as a feature. Similarly, in the temperature feature plot, the orange markers indicate pairs where no azimuth was present. This means that the azimuth feature is especially valuable if temperature is not a feature, and vice versa. We attribute this to the often observed pattern of both the azimuth and the temperature raising in the morning (see Fig. 5). Since the temperature feature is obtained from an uncertain weather forecast while the azimuth can be calculated precisely, we consider the azimuth to be the better feature of the two and discard temperature.

5) Low-Clouds: When the low-clouds feature is removed, the mean SCAPE of the RFR models increases from approximately 31 to almost 35. In addition, the variability of results increases, which explains the appearance of two vertical columns, with the SCAPE being as high as 45. The coloring of the plot distinguishes the presence of the general cloudiness

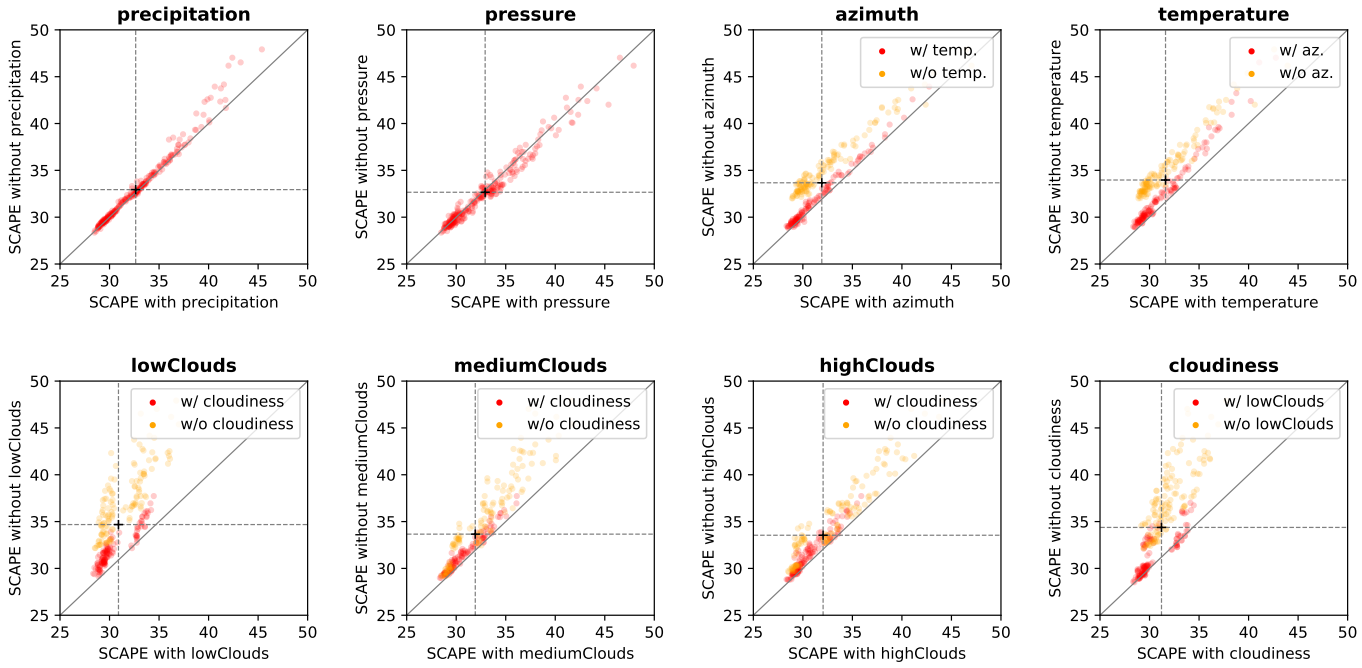

Fig. 9. Ablation study for the remaining features. Different colors are used to distinguish further certain subsets to show feature dependencies.

feature, but interactions between the two features are not obvious here.

6) Medium-Clouds: The impact of removing the medium-clouds feature is not as significant as with low-clouds but the mean SCAPE still increases when this feature is not used. The resulting variation also improves slightly, suggesting that the medium-clouds should be considered as a feature.

7) High-Clouds: The obtained performance by removing the high-clouds feature is similar to the medium-clouds feature, except that a smaller variation occurs. By analyzing this third cloud-related feature we can conclude that they complement each other, even though low-clouds have a stronger correlation to the overall performance of the model.

8) Cloudiness: As shown in the ablation plot, having the overall cloudiness feature improves the performance of our model similar to the low-clouds feature. We explain this with the fact that the densest clouds are found in the lower levels. If a large proportion of the sky is covered by clouds in this layer, clouds in the medium or high layers have significant influence on how much light reaches the ground.

The dependencies between the different cloudiness features are not obvious in the pairwise comparison above, which is why we also computed the performance of all combinations of cloudiness-features, shown in Table I. Each row shows the mean performances and variation of the models matched with the combination of cloudiness features given to the left. The table reveals that models score similarly if they have at least three cloud-related features included. When the three cloud-levels (low, medium high) are present, the overall cloudiness feature does not contribute to any improvement. If we only want to select two cloud-related features, *highClouds* and *lowClouds* combined score best. If only one cloud-related feature should be taken into account, it should be *cloudiness*, but it scores ca. 10% worse than the best combinations.

TABLE I
AVERAGE SCORES OF ALL MODELS GIVEN THE AVAILABILITY OF THE VARIOUS CLOUDINESS FEATURES. USING THE RFR MODEL WITH 30 ESTIMATORS, 10-FOLD CROSS-VALIDATION.

cloudiness	lowClouds	mediumClouds	highClouds	feature count	SCAPE	SMAPE	STAPE
	✓	✓	✓	**3**	30.0 (0.3)	37.7 (0.3)	22.3 (0.3)
✓	✓	✓	✓	**4**	30.0 (0.3)	37.7 (0.3)	22.3 (0.3)
✓	✓	✓		3	30.3 (0.3)	38.0 (0.3)	22.7 (0.3)
✓	✓		✓	3	30.4 (0.3)	38.2 (0.3)	22.6 (0.3)
	✓		✓	**2**	30.6 (0.3)	38.5 (0.3)	22.7 (0.2)
✓		✓	✓	3	30.8 (0.3)	38.7 (0.3)	22.9 (0.3)
✓	✓			2	30.9 (0.3)	38.7 (0.3)	23.1 (0.3)
	✓	✓		2	31.1 (0.3)	38.8 (0.3)	23.4 (0.3)
✓			✓	2	31.9 (0.3)	39.9 (0.3)	23.9 (0.3)
✓		✓		2	32.2 (0.3)	40.1 (0.3)	24.3 (0.3)
✓				**1**	33.1 (0.3)	41.3 (0.4)	24.9 (0.3)
	✓			1	33.9 (0.3)	41.8 (0.3)	26.0 (0.2)
		✓	✓	2	35.1 (0.4)	42.9 (0.4)	27.2 (0.4)
		✓		1	35.9 (0.5)	44.0 (0.5)	27.8 (0.4)
			✓	1	37.5 (0.5)	45.7 (0.6)	29.3 (0.5)
				0	40.9 (0.6)	49.4 (0.6)	32.5 (0.6)

C. Training Sample Intervals

We now examine how sensitive the machine learning models are to the intervals in which training samples are collected. Longer intervals are desirable since they reduce (i) the amount

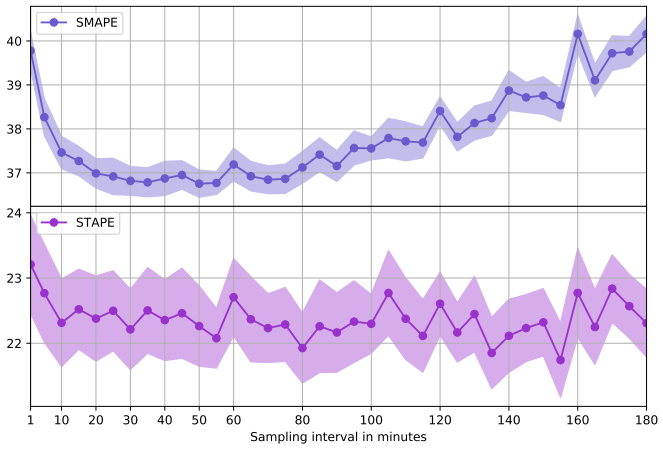

Fig. 10. Scores for models trained with data sampled at different rates. While the STAPE is fairly unaffected once sampling intervals are longer than 10 minutes, the SMAPE is getting better in the range between 30 and 50 minutes.

of data to transfer from the IoT devices to the machine learning models in the edge or cloud, (ii) how much data needs to be stored, and (iii) the runtime of the model fitting, i.e., training time.

For the sensitivity analysis, we prepared data sets with different sampling intervals, ranging from data sampled every minute (the original rate), to data sampled every 180 minutes, by resampling and taking the mean values. For each sampling rate, we trained RFR models for each of the 10 crossfolds, using the feature sets consisting of zenith, azimuth and all cloudiness features. We evaluated the resulting models with the test data from the original 1-minute sampled raw data. Fig. 10 shows the mean SMAPE and STAPE for the 10-folds including their standard error. Interestingly, the lowest sampling interval of 1 minute does not result in the most accurate models measured by either SMAPE or STAPE. Instead, we see an optimum for the SMAPE in the range between 30 to 50 minutes, before the accuracy of the prediction decreases again with growing sampling intervals. We attribute this to the short-term fluctuations in the cloud coverage that are present in the more fine-grained data, which are not represented in the hourly weather forecast. For an operational setting, this is significant: more data does not imply higher accuracy, and by choosing a sampling interval closer to 30 minutes we can both increase the accuracy of the prediction and save costs in transmission, data storage and training time.

VI. Day-to-Day Operation and Evaluation

To ensure the relevance of our results, we now study the performance of the prediction in an operational setting. That means that devices are newly deployed without prior data, and machine learning models are re-trained continuously as training data becomes available. In the previous sections, we analyzed suitable machine learning models, feature sets and sampling frequencies. For the operational settings, we chose trade-offs that allow for quick computation and good performance. We hence chose the random forest regressor (RFR), with *zenith*, *azimuth*, *cloudiness*, *lowClouds*, *mediumClouds*, *highClouds*, as features, and a 30-minute sampling frequency. As training and test data we use the so far unused data of the year 2019.

We assume that an IoT device n is newly deployed on day d^n_{dep} and use d^n_i to describe the i-th day since the deployment of n. At the end of each day d^n_i, the device manager computes a new model M^n_i with the training data from the previous days since deployment, i.e.,

$$train^n_i = \{d \mid d \geq d_{dep} \ \wedge \ d \leq d_i + d_{dep} \ \wedge \ d \geq d_i + d_{dep} - train_{max}\}.$$

The last conjunct constraints the use to only the last $train_{max}$ days for training data. Since a device is deployed without prior knowledge, the model M^n_i only contains training data collected from d^n_{dep} to d^n_i. The model M^n_i created for device n at the end of day d^n_i is then used to create a new energy prediction for day d^n_{i+1}, using the public weather forecast and the solar angles that are calculated from time and location. To ensure a realistic and causal setting, we only use weather forecasts for day d^n_{i+1} that were published on the previous day d^n_i. We then calculate error metrics for all days $d^n_i, i > 0$.

A. Amount of Training Data

As the computational effort of model fitting increases with the number of training samples and hence with increasing $train_{max}$, we examine first how the prediction quality of M^n_i develops with the number of included training days. For that, we calculate the results for $n = 365$ devices, each deployed at a different day in 2019,[2] and study the results of M^{355}_i when predicting day d^{365}_i for increasing $train_{max}$. Fig. 11 shows how the SCAPE reduces with a growing number of training days $train_{max}$. Mean and median error reduce quickly with an increasing amount of training data. Already after around 30 to 50 days with training data the SCAPE is on a level that only marginally reduces with further data.

We have also experimented with data augmentation, similar to [44], which takes the existing training data and generates additional data points through various techniques. However, this approach could not further reduce the number of required training days as desired.

B. Operational Setting and Comparison

For the final evaluation and comparison with other baselines, we now examine how the metrics evolve over time, i.e., starting with one day of training data on deployment day d_1, and going forward using a maximum of $train_{max} = 30$ days. Figure 12 shows the results of the operational setting. Again, we simulated N=365, corresponding to the deployment of devices at different days of the year 2019.

Table II compares the results of our proposed approach (RFR) with the approaches described in Sect. III. For the EWMA-based approach based on [12] we used 48 timeslots and $\alpha = 0.7$. Similarly, we used the same setting for the approach using cloud-cover forecasts (CCFs), as they provided

[2]We arranged the days of the year in a circular way, so that periods going beyond the year end draw their days from the beginning of the year.

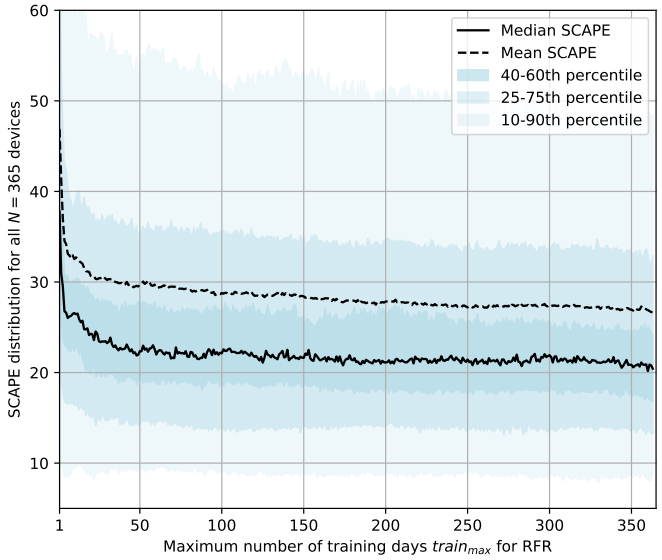

Fig. 11. Distribution of SCAPE with growing number of training days, for N=365 deployments.

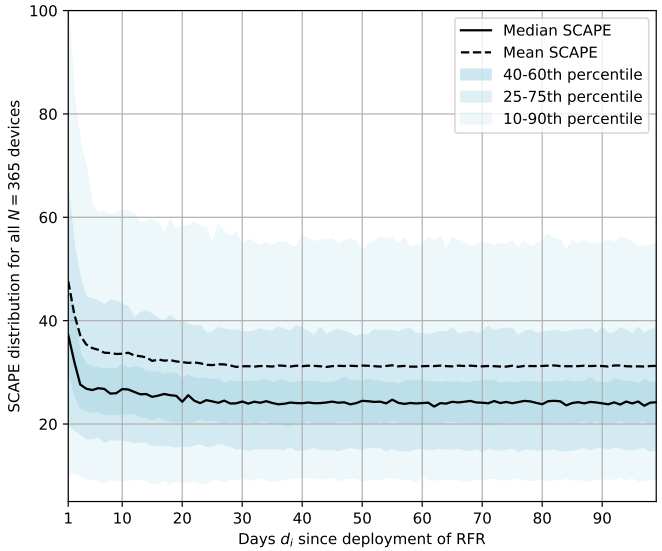

Fig. 12. Distribution of SCAPE for N=365 nodes starting without prior data with growing days d_i since deployment, and a maximum of $train_{max} = 30$ training days.

the best results in [26]. (SP) denotes the smart persistence model. The numbers show the mean and the median of the metrics over all 365 prediction days. As expected, the methods only relying on past data (EWMA and SP) perform worst. The results of CCF show the benefit of taking cloud coverage forecasts into account. The consistently best results are achieved by our proposed RFR model, which scores more than 20 % better than the CCF model for the median SCAPE.

RFR and CCF receive the same weather forecast, but CCF only utilizes the overall cloudiness feature while RFR also takes the other cloudiness features and solar angles into account. The remaining error for RFR is comparatively low,

TABLE II
RESULTS OF THE PREDICTION MODELS IN THE OPERATIONAL SETTING.

	SCAPE		SMAPE		STAPE	
	mean	median	mean	median	mean	median
SP	44.79	38.84	50.56	47.67	39.01	32.96
EWMA	40.30	33.19	45.48	41.06	35.13	27.16
CCF	34.49	30.63	41.53	38.47	27.46	21.11
RFR	**31.20**	**24.10**	**38.00**	**31.50**	**24.40**	**16.10**
Error Reduction from CCF to RFR	-9.54 %	-21.32 %	-8.50 %	-18.12 %	-11.14 %	-23.73 %

and we suspect that most of it is due to imprecission of the weather forecast, i.e., an inherent problem that only reduces with more accurate weather predictions.

C. Computational Effort

One argument for the EWMA-based approaches is their computational simplicity. We argue, however, that the computational effort needed for the RFR is insignificant in a modern IoT system with device management. The effort for storing training data and training the models is low on cloud- or edge-platforms, especially when compared with computation and storage needed for the actual application data in an IoT system. For example, with the setting chosen above ($train_{max} = 30$, sampling rate 30 minutes and all cloud-related features), training time on a PC-grade CPU is on average below 400 ms per day and per device. The prediction time for a single day with 48 time slots is on average below 9 ms. This seems to be acceptable given the potential gains in prediction accuracy that further increase the energy efficiency of the IoT device, which is a much more urgent problem.

VII. CONCLUSION

We presented and evaluated an approach for solar power energy prediction that is suitable for the application in solar-powered IoT systems. The median prediction scores are more than 20 % better than the current state of the art for IoT energy prediction, which we consider significant for the effectiveness of energy budget planning. The input for the prediction models only uses the public weather forecast and solar angles derived from the current time and location, that means, only data that is easily available. The approach scales well in an IoT setting. No manual tuning is necessary for the individual IoT devices despite any differences, for instance in solar panel size, as these differences are learned by the individual machine learning models. The problem of individual adaptation is hence solved by individual but autonomously learning models.

Once part of the device management, energy harvesting models with better performance open up for more strategic energy planning. As indicated in the introduction, IoT devices can then more proactively schedule energy-intensive tasks in predicted periods of energy surplus, avoiding over-dimensioning of systems. This has the potential to reduce the size of solar panels and energy buffers that IoT devices need to operate in a perpetual way, further advancing sustainable operation.

ACKNOWLEDGMENT

We would like to thank Nattachart Tamkittikhun and Pål Sæther for their help with the experimental setup.

REFERENCES

[1] F. K. Shaikh and S. Zeadally, "Energy harvesting in wireless sensor networks: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 55, pp. 1041–1054, Mar. 2016.

[2] D. Ahlers, P. Driscoll, F. A. Kraemer, F. Anthonisen, and J. Krogstie, "A Measurement-Driven Approach to Understand Urban Greenhouse Gas Emissions in Nordic Cities," *NIK Norsk Informatikkonferanse*, pp. 1–12, Nov. 2016.

[3] A. Caruso, S. Chessa, S. Escolar, X. del Toro, M. Kuzman, and J. C. Lopez, "Experimenting Forecasting Models for Solar Energy Harvesting Devices for Large Smart Cities Deployments." *ISCC*, vol. 2019-June, pp. 1177–1182, 2019.

[4] L. Hanschke, J. Heitmann, and C. Renner, "Challenges of WiFi-Enabled and Solar-Powered Sensors for Smart Ports," in *the 4th International Workshop*. New York, New York, USA: ACM Press, 2016, pp. 13–18.

[5] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019.

[6] R. Connection. (2019) Prevent illegal deforestation. [Online]. Available: https://rfcx.org/

[7] J. Liu, G. Faulkner, B. Choubey, S. Collins, and D. C. O'Brien, "An Optical Transceiver Powered by On-Chip Solar Cells for IoT Smart Dusts With Optical Wireless Communications," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3248–3256, Apr. 2019.

[8] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, "Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems," in *2010 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*. IEEE, 2010, pp. 1–9.

[9] C. Renner, S. Unterschütz, V. Turau, and K. Römer, "Perpetual Data Collection with Energy-Harvesting Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 11, no. 1, pp. 1–45, Nov. 2014.

[10] F. A. Kraemer, D. Ammar, A. E. Braten, N. Tamkittikhun, and D. Palma, "Solar energy prediction for constrained IoT nodes based on public weather forecasts," in *ACM International Conference Proceeding Series*, Oct. 2017.

[11] C.-W. Yau, T. T.-O. Kwok, C.-U. Lei, and Y.-K. Kwok, "Energy Harvesting in Internet of Things," in *Internet of Things Based on Smart Objects*. Singapore: Springer Singapore, Oct. 2017, pp. 35–79.

[12] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Transactions on Embedded Computing Systems*, vol. 6, no. 4, p. 32, Jan. 2007.

[13] N. Edalat and M. Motani, "Energy Aware Task Allocation for Energy Harvesting Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–14, Jan. 2016.

[14] V. Cionca, A. McGibney, and S. Rea, "MAllEC: Fast and Optimal Scheduling of Energy Consumption for Energy Harvesting Devices," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5132–5140, Jan. 2019.

[15] T. Zou, S. Lin, Q. Feng, and Y. Chen, "Energy-Efficient Control with Harvesting Predictions for Solar-Powered Wireless Sensor Networks," *Sensors*, vol. 16, no. 1, pp. 53–31, Jan. 2016.

[16] F. A. Kraemer, F. Alawad, and I. M. V. Bosch, "Energy-accuracy tradeoff for efficient noise monitoring and prediction inworking environments," in *ACM International Conference Proceeding Series*, Norges Teknisk-Naturvitenskapelige Universitet, Trondheim, Norway. New York, New York, USA: ACM Press, Oct. 2019, pp. 1–8.

[17] N. Ashraf, A. Hasan, H. K. Qureshi, and M. Lestas, "Combined Data Rate and Energy Management in Harvesting Enabled Tactile IoT Sensing Devices," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 3006–3015, Apr. 2019.

[18] A. E. Braten, F. A. Kraemer, and D. Palma, "Adaptive, Correlation-Based Training Data Selection for IoT Device Management," in *The 6th International Conference on Internet of Things: Systems, Management and Security (IOTSMS 2019)*. IEEE, 2019.

[19] J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," in *Proceedings of the 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2009*, Department of Computer Science and Engineering, San Diego, United States. IEEE, Nov. 2009, pp. 6–10.

[20] A. H. Dehwah, S. Elmetennani, and C. Claudel, "UD-WCMA: An energy estimation and forecast scheme for solar powered wireless sensor networks," *Journal of Network and Computer Applications*, vol. 90, pp. 17–25, Jul. 2017.

[21] K. Saidi, W. Ajib, and M. Boukadoum, "Adaptive transmitter load size using receiver harvested energy prediction by Kalman filter," *2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP)*, pp. 1–5, Jul. 2016.

[22] A. Kaur, L. Nonnenmacher, H. T. C. Pedro, and C. F. M. Coimbra, "Benefits of solar forecasting for energy imbalance markets," *Renewable Energy*, vol. 86, pp. 819–830, Jan. 2016.

[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*, vol. 136, no. C, pp. 78–111, Oct. 2016.

[24] H. T. C. Pedro and C. F. M. Coimbra, "Assessment of forecasting techniques for solar power production with no exogenous inputs," *Solar Energy*, vol. 86, no. 7, pp. 2017–2028, Jul. 2012.

[25] P. Ineichen, "A broadband simplified version of the Solis clear sky model," *Solar Energy*, vol. 82, no. 8, pp. 758–762, Aug. 2008.

[26] C. Renner, "Solar harvest prediction supported by cloud cover forecasts," in *the 1st International Workshop*. New York, New York, USA: ACM Press, 2013, pp. 1–6.

[27] C. Voyant, G. Notton, S. Kalogirou, M.-L. Nivet, C. Paoli, F. Motte, and A. Fouilloy, "Machine learning methods for solar radiation forecasting: A review," *Renewable Energy*, vol. 105, pp. 569–582, May 2017.

[28] P. Bacher, H. Madsen, and H. A. Nielsen, "Online short-term solar power forecasting," *Solar Energy*, vol. 83, no. 10, pp. 1772–1783, Oct. 2009.

[29] A. K. Yadav and S. S. Chandel, "Solar radiation prediction using Artificial Neural Network techniques: A review," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 772–781, Jan. 2014.

[30] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *2011 IEEE International Conference on Smart Grid Communications, SmartGridComm 2011*, University of Massachusetts Amherst, Amherst MA, United States. IEEE, Dec. 2011, pp. 528–533.

[31] A. Dahl and E. V. Bonilla, "Grouped Gaussian processes for solar power prediction," *Machine Learning*, vol. 12, no. 5, pp. 1–20, May 2019.

[32] L. Benali, G. Notton, A. Fouilloy, C. Voyant, and R. Dizene, "Solar radiation forecasting using artificial neural network and random forest methods: Application to normal beam, horizontal diffuse and global components," *Renewable Energy*, vol. 132, pp. 871–884, Mar. 2019.

[33] A. E. Braten and F. A. Kraemer, "Towards cognitive IoT: Autonomous prediction model selection for solar-powered nodes," in *Proceedings - 2018 IEEE International Congress on Internet of Things, ICIOT 2018 - Part of the 2018 IEEE World Congress on Services*, Norges Teknisk-Naturvitenskapelige Universitet, Trondheim, Norway. IEEE, Sep. 2018, pp. 118–125.

[34] C. Feng and J. Zhang, "Hourly-Similarity Based Solar Forecasting Using Multi-Model Machine Learning Blending," *2018 IEEE Power & Energy Society General Meeting (PESGM)*, pp. 1–5, Oct. 2018.

[35] A. Fouilloy, C. Voyant, G. Notton, F. Motte, C. Paoli, M.-L. Nivet, E. Guillot, and J.-L. Duchaud, "Solar irradiation prediction with machine learning: Forecasting models selection method depending on weather variability," *Energy*, vol. 165, no. Part A, pp. 620–629, Dec. 2018.

[36] N. Tang, S. Mao, Y. Wang, and R. M. Nelms, "Solar Power Generation Forecasting With a LASSO-Based Approach," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1090–1099, Apr. 2018.

[37] Y. Wang, Y. Shen, S. Mao, X. Chen, and H. Zou, "LASSO and LSTM Integrated Temporal Model for Short-Term Solar Intensity Forecasting," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2933–2944, Apr. 2019.

[38] J. Zhang, A. Florita, B.-M. Hodge, S. Lu, H. F. Hamann, V. Banunarayanan, and A. M. Brockway, "A suite of metrics for assessing the performance of solar power forecasting," *Solar Energy*, vol. 111, no. C, pp. 157–175, Jan. 2015.

[39] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, Oct. 2006.

[40] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16 – 28, 2014, 40th-year commemorative issue. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790613003066

[41] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of the 10th National Conference on Artificial Intelligence, San Jose, CA,*

USA, July 12-16, 1992, W. R. Swartout, Ed. AAAI Press / The MIT Press, 1992, pp. 129–134. [Online]. Available: https://dl.acm.org/doi/10.5555/1867135.1867155

[42] Meteorologisk institutt. (2019) Free meteorological data. [Online]. Available: https://www.met.no/en/free-meteorological-data

[43] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[44] Y. Li, H. Hu, and G. Zhou, "Using Data Augmentation in Continuous Authentication on Smartphones," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 628–640, 2019.