# Solar Harvest Prediction Supported by Cloud Cover Forecasts

Christian Renner
University of Lübeck
Lübeck, Germany
renner@iti.uni-luebeck.de

## ABSTRACT

Solar harvest prediction is used in energy-harvesting sensor networks to achieve perpetual node operation. Existing approaches only exploit local knowledge and thus fail in unforeseeable, changing weather conditions. We investigate the benefit of incorporating global knowledge in terms of fractional sky cloudiness, so-called cloud cover. We propose and evaluate two methods that combine local information of a node's harvest pattern with global cloud cover forecasts. We evaluate their performance with solar traces collected by three solar-harvesting sensor nodes and compare the results with existing prediction algorithms. We find that (i) harvest predictions using cloud cover forecasts improve overall prediction precision, (ii) prediction errors in changing weather conditions are considerably reduced, and (iii) coarse-grained cloud cover forecasts require low extra network traffic while sacrificing little prediction precision.

## Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; G.3 [**Probability and Statistics**]: Time-Series Analysis

## General Terms

Algorithms, Performance, Design, Measurement

## 1. INTRODUCTION

Energy-harvesting sensor networks enable perpetually operating sensor networks. In particular, solar harvesting has drawn considerable attention [12, 3, 8], since it is applicable in most outdoor deployments. The amount of harvested energy, also called harvest, is yet limited, so that sensor nodes must still employ power-saving mechanisms such as radio duty cycling. Moreover, the harvest is unsteady due to massive inter-day and intra-day fluctuation, seasonal effects, dirt, and hardware aging. In addition, the environment has a notable impact, e.g., shades of buildings and trees reduce the harvest considerably. For a network of tens or hundreds of nodes, planning the dimensions of the solar cells and the duty cycle for each node is noneconomical or even impossible.

For this reason, algorithms have been devised that adjust the duty cycle or task schedule of sensor nodes online to achieve energy-neutral (i.e., depletion-safe) operation [5, 11, 4, 15]. Rather than saving as much energy as possible, these algorithms aim at maximizing the energy-neutral node consumption in order to achieve goals such as low network delay or high sensing rates. They rely on methods to predict the future harvest [5, 13, 1, 2], preferably long-term predictions of at least one day. Existing prediction algorithms only exploit local historical observations of the harvest. Particularly in changing weather conditions, when a cloudy day follows a sunny day and vice versa, these methods fail and produce erroneous predictions. In the worst case, they hazard energy-neutral, perpetual operation with too optimistic predictions, and they waste energy by causing full energy buffers with too pessimistic predictions.

Using global weather forecasts has the potential to elevate the precision of harvest predictions. Here, cloud cover is promising, because it exhibits a correlation with solar irradiance [14, 6, 10]. In addition, cloud cover forecasts have a small data size—they usually have a resolution of up to one hour and are provided as eights of the sky being covered with clouds—so that they can be distributed into the network within a single packet per hour. However, cloud cover information cannot reflect the local harvest pattern, that is created by the node's environment, so that local observations are still required. A combination is hence appealing.

In this paper, we propose methods for harvest prediction that integrate globally available cloud cover forecasts with local observations of a node's individual harvest pattern. We evaluate the methods and compare them with existing algorithms based on local information only. For this purpose, we use solar traces of three energy-harvesting sensor nodes and freely available cloud cover forecasts from an online meteorological service. In addition, we analyze the impact of the time resolution of the forecasts and discuss methods to distribute cloud cover information into the network.

The remainder of the paper is organized as follows. In Sect. 2, we introduce and discuss existing algorithms for harvest prediction in sensor networks. We introduce and explain our methods that integrate cloud cover forecasts in Sect. 3. Sect. 4 evaluates and compares our methods with existing algorithms, and Sect. 5 concludes the paper.

## 2. HARVEST PREDICTION FOR SENSOR NETWORKS

Several researchers have tackled the problem of light-weight algorithms for generating solar harvest predictions for sensor nodes. Here, the main focus is on low computation power and local reck-
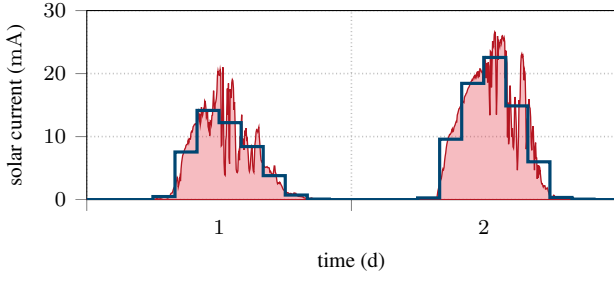
**Figure 1: Excerpt of a solar current trace (red / filled curve) and mean harvest in slots (blue) for $S = 12$**

oning, i.e., preventing data exchange and distribution of weather forecasts. They share in common that they exploit the quasi-cyclic pattern of the harvesting sources—e.g., the diurnal pattern of solar energy—to make predictions. We give an overview of existing algorithms and discuss their suitability for creating the required long-term predictions of at least one day.

## 2.1 Harvest Predictions with Time Slots

A day is divided into $S$ time slots, where each slot is assigned a slot value $\mu_{d,s}$ on day $d$. Most algorithms assume static time slots of equal length, e.g., 12 slots with a length of 2 h each. In each day $d$ and time slot $s$, the mean value $h_{d,s}$ of the harvest—which can be, e.g., the solar current or power—is calculated and assigned to the corresponding slot value, i.e., $\mu_{d,s} = h_{d,s}$. Figure 1 shows an example of the actual harvest and the mean values $h_{d,s}$. The fundament of all harvest prediction algorithms is to smooth the values $\mu_{d,s}$ of the same slot $s$ along subsequent days $d$. These smoothed values, denoted $\bar{\mu}_{d,s}$, are then used to predict future slot values $\hat{\mu}_{d,s}$. In the following, we explain different methods for averaging slot values and deriving predictions.

### EWMA: Exponentially Weighted Moving Average
In [5] Kansal et al. obtain smoothed slot values by applying an EWMA filter:

$$\bar{\mu}_{d,s} = \alpha \cdot \bar{\mu}_{d-1,s} + (1 - \alpha) \cdot \mu_{d,s} \qquad (0 < \alpha < 1) . \quad (1)$$

Their intention is to compensate daily fluctuation, seasonal effects, and harvester aging. The averaged value $\bar{\mu}_{d,s}$ of a time slot $s$ in day $d$ is used as harvest prediction for the same time slot $s$ in the next day $d + 1$, i.e.,

$$\hat{\mu}_{d+1,s} = \bar{\mu}_{d,s} . \quad (2)$$

The method is light-weight, because only $S$ values have to be stored and the arithmetic complexity is low. Theoretically, predictions with a horizon of multiple days are possible. However, this comes at the cost of achieving predictions with low precision as discussed in [16, 1]. In particular, there is no mechanism to react to changing weather conditions.

### WCMA: Weather-Conditioned Moving Average
To solve the shortcomings of EWMA predictions w.r.t. changing weather conditions, the authors of [13] proposed the WCMA prediction algorithm. It differs in three aspects:

1. It averages slot values over the latest $D$ past days with

$$\bar{\mu}_{d,s} = \frac{1}{D} \cdot \sum_{i=0}^{D-1} \mu_{d-i,s} \qquad (D \geq 1) . \quad (3)$$

2. A scale factor $\theta$ describes the relation of averaged slot values with those of the current day through

$$\theta = \left( \sum_{k=0}^{K-1} \frac{1}{k+1} \right)^{-1} \cdot \sum_{k=0}^{K-1} \frac{1}{k+1} \cdot \frac{\mu_{d,s-k}}{\bar{\mu}_{d,s-k}} \quad (4)$$

to compare the trend of the current day with the average.

3. The actual prediction of the time slot directly following slot $s$ is a weighted sum of the mean harvest of slot $s$ and the scaled average value of the following slot:

$$\hat{\mu}_{d,s+1} = (1 - \beta) \cdot \mu_{d,s} + \beta \cdot \theta \cdot \bar{\mu}_{d-1,s+1} . \quad (5)$$

Storage space is $S \cdot D$ slot values and computation cost is higher than that of EWMA. WCMA is designed to predict the directly following slot but can also be used to predict slot values other than the directly following ones. For this case our previous work in [16] suggests $\beta = 1$ due to a lack of correlation between the mean values of non-adjacent slots.

### WCMA-PDR: WCMA w. Phase Displacement Regulator
Bergonzini et al. have improved WCMA in [2]. They noted a pattern of prediction errors that they call phase displacement. Their algorithm tracks the prediction error in each slot with an EWMA filter. A phase displacement value is calculated by comparing the prediction errors of adjacent time slots and is added to the WCMA prediction value. The evaluation in [2] shows an improvement of prediction precision for predictions of the following slot only. However, WCMA-PDR increases the memory and computation demand compared to WCMA and introduces additional parameters. Since the algorithm is particularly tailored for the case $\beta < 1$, we do not consider WCMA-PDR for long-term predictions.

### Pro-Energy
The algorithm in [19] identifies characteristic harvest patterns (slot values) of typical days online—the authors name sunny, cloudy, and rainy days as examples. Harvest predictions are generated by picking the pattern that has the closest match within a given comparison window. The algorithm supports short-term and mid-term predictions (few minutes up to several hours) and increases the prediction precision compared to EWMA and WCMA. Long-term predictions (of a whole day) have not been evaluated by the authors. It is also not clear how well the prediction algorithms works during the night in absence of harvest, when patterns are equal.

## 2.2 Model-Based Harvest Prediction
Sharma et al. present a method for using global solar irradiance and cloud cover forecasts to predict the future harvest in [17]. They use a polynomial model for the relationship between time of a day and solar power. To cope with seasonal effects, one parameter set is generated for every month of the year. The authors suggest a linear down-scaling of the anticipated maximum solar power by means of the per cent cloud cover. The method does not handle influences of, e.g., hardware aging and shades caused by buildings or trees.

Machine learning for predicting solar power was discussed in [18]. SunCast predicts sunlight conditions by finding the most similar

historical sunlight trace [9]. Unfortunately, these algorithms over-shoot the resource demands offered by sensor nodes.

# 3. HARVEST PREDICTION WITH CLOUD COVER

Existing harvest prediction algorithms for sensor nodes succeed in identifying the average harvest pattern of a node. This is an important feature due to the individual harvesting patterns as explained in Sect. 1. In changing weather conditions, however, they fail due to their restricted, local knowledge. Using global knowledge, such as cloud cover forecasts, mitigates this weakness, but it does not incorporate a node's individual harvest pattern. We hence argue that combing both methods satisfies the need for incorporating both local harvest patterns and global weather conditions into predictions.

In the following, we present methods that combine the advantages of EWMA predictions with cloud cover forecasts. First, we explain the relation between solar harvest (current or power), solar irradiance, and cloud cover. Second, we present our prediction methods. Third, we briefly discuss how to distribute cloud cover forecasts in the network.

## 3.1 Linking Cloud Cover and Solar Harvest

The photovoltaic effect causes a solar cell to produce a current that exhibits an almost linear dependency on solar irradiance $R$, if the voltage of the solar cell stays reasonably below its maximum voltage. This relationship also holds for solar power and solar irradiance, if the terminal voltage of the solar cell is constant.

Solar irradiance and cloud cover are also related [14, 6, 10]. In specific, analytical models have been proposed that describe the relation $\varrho$ between irradiance $R_0$ at a clear sky and irradiance $R$ at a relative cloud cover $C$. Here, $C$ is the fraction (usually in eighth) of the sky that is covered with clouds, and $\varrho$ is a function of $C$ with

$$R = R_0 \cdot \varrho(C) \tag{6}$$

Two well-studied models are those of Kimball and Laevastu [14, 6, 10]. Kimball proposes the linear model

$$\varrho(C) = 1 - 0.71 \cdot C, \tag{7}$$

and Laevastu suggests using

$$\varrho(C) = 1 - 0.6 \cdot C^3. \tag{8}$$

Other models are either variations of the presented ones or use additional information, such as the solar altitude at noon or the latitude of the deployment. Comparative studies as in, e.g., [6] have shown that these bring limited or no improvement over the methods of Kimball and Laevastu.

## 3.2 Harvest Prediction

We aim at providing a light-weight algorithm—in terms of computation power, memory demand, and the number of parameters—for harvest prediction using cloud cover forecasts. The general idea is to calculate for each slot the possible harvest under a clear sky with Eq. (6) and use it as slot value. To predict the future harvest, the slot value is scaled using the cloud cover forecast.

We suggest two methods that we elaborate in the following. Both methods are based on the EWMA algorithm and use the mean harvest $h_{d,s}$, the mean relation $\varrho_{d,s}$ in slot $s$ of day $d$, and the cloud-cover forecast $\hat{\varrho}$. While $h_{d,s}$ is locally computable by a sensor node, the values of $\varrho_{d,s}$ and $\hat{\varrho}$ have to be distributed in the network (cf. Sect. 3.3). Note that $\hat{\varrho}$ embraces multiple values, one for each slot in the prediction horizon.

### 3.2.1 Combined Method

This method applies Eq. (6) for each slot by calculating the quotient of mean harvest and cloud cover and by using it as mean slot value

$$\mu_{d,s} = \frac{h_{d,s}}{\varrho_{d,s}}. \tag{9}$$

The smoothed value $\bar{\mu}_{d,s}$ is calculated with an EWMA filter (cf. Sect. 2.1). The future harvest is predicted using the smoothed value and the cloud cover forecast. For slot $s$ of day $d + k$ ($k > 0$), the predicted harvest is

$$\hat{\mu}_{d+k,s} = \bar{\mu}_{d,s} \cdot \hat{\varrho}_{d+k,s}. \tag{10}$$

The memory footprint of this approach scales linearly with the number of slots $S$. The effort of distributing cloud cover forecasts in the network depends on its forecast horizon and time resolution, i.e., the size of the forecast in bytes. Computation complexity is that of EWMA filtering plus one additional division and multiplication each.

### 3.2.2 Separate Method

The second method tracks the smoothed mean harvest and cloud cover values separately. For each slot $s$, two smoothed slot values $\bar{\mu}_{d,s}^h$ (for the harvest) and $\bar{\mu}_{d,s}^\varrho$ (for cloud cover) are stored. They are fed with the mean harvest $\mu_{d,s}^h = h_{d,s}$ and the mean cloud cover $\mu_{d,s}^\varrho = \varrho_{d,s}$ in slot $s$ using EWMA filtering (with same filter coefficient $\alpha$).

The future harvest is predicted using these two smoothed values and the cloud cover forecast. In particular, the prediction for slot $s$ of day $d + k$ ($k > 0$) is calculated through

$$\hat{\mu}_{d+k,s}^h = \frac{\bar{\mu}_{d,s}^h}{\bar{\mu}_{d,s}^\varrho} \cdot \hat{\varrho}_{d+k,s}. \tag{11}$$

The memory footprint and computation complexity are twice that of the combined method. The effort of distributing cloud cover forecasts in the network is the same.

## 3.3 Distribution of Cloud Cover Data

Until now, we have assumed that each node has access to cloud cover information—either in form of $C$ or $\varrho$—for the current time slot and a cloud cover forecast. Apparently, this data is global information, that has to be distributed in the network. Naturally, this distribution should lead to as low communication overhead as possible. One way to solve this problem is to use a distribution method such as Trickle [7]. Another option is to piggy-back the corresponding data in (software) acknowledgments. This is particularly appealing, if there is regular data traffic in the network and since cloud cover information and forecasts are usually only available in an hourly resolution. We plan to investigate this point in detail in future work.

# 4. EVALUATION

Next, we evaluate and compare harvest predictions with and without cloud cover forecasts. Before presenting the results, we introduce the data sets and evaluation methodology.
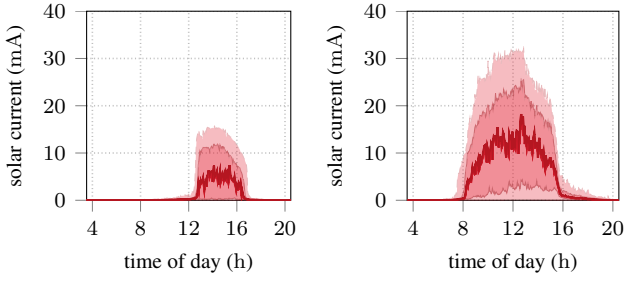
(a) node A: indoor, shaded in the morning
(b) node C: outdoor, partly shaded in late afternoon

**Figure 2: Statistical distribution (minimum, lower quartile, median, upper quartile, maximum) of the harvested solar current (ground truth).**

## 4.1 Data Sets

We deployed three energy-harvesting sensor nodes (A–C) on different window sills of our University building. Node A was placed on the inside, whereas nodes B and C were on the outside. All nodes were equipped with a miniature solar cell and a supercapacitor as energy buffer. The solar cell produces a maximum current of 35 mA, where the current roughly depends on the radiation (weather conditions) only. A shunt in the ground path of the charging circuit serves as solar current sensor. Further details on the hardware are provided in [15].

All nodes sampled the solar current every 3 s and sent 5 min-averages to the base station. We recorded solar current data from April 2013 until July (88 to 95 days of data per node). Figure 2 shows the statistical distributions of solar current for two nodes throughout the experiment time (nodes B and C performed similarly).

For the same time period, we recorded cloud cover forecasts from an online weather forecast service[1]. These forecasts have a resolution of one hour and a forecast horizon of ten days. Updated forecasts were available at every full hour.

## 4.2 Methodology and Metrics

Load adaptation algorithms (e.g., [11]) usually require long-term harvest forecasts to achieve stable node operation. We hence compare the prediction precision of our cloud-cover enabled harvest prediction algorithms from Sect. 3, EWMA and long-term WCMA predictions as described in Sect. 2.1. The prediction horizon is one day, where predictions are updated after every time slot. Initially, all smoothed slot values $\bar{\mu}_{d,s}$ are zero. We performed an analysis for $S = 6, \ldots, 48$ slots and $\alpha$ from 0 to 1 in steps of 0.1. For WCMA we use $D = 3, 5$, $K = 3$, and $\beta = 1$. This setup has shown good performance in [16, 2, 1]. Predictions using cloud cover were generated with the relationship functions $\varrho$ from Kimball and Laevastu and averaged to meet the slot length. We added a suffix to Kimball and Laevastu to differentiate between the combined ("-1", one slot value) and separate method ("-2", two slot values).

To analyze the deviation of the prediction from the real harvest course (ground truth), we calculated the root-mean-square error (RMSE) for every single prediction. This metric shows the deviation within a day and is hence particularly relevant for small energy

[1] www.wetterspiegel.de
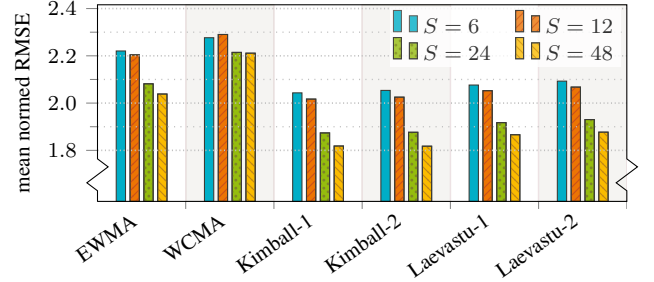


(a) node A
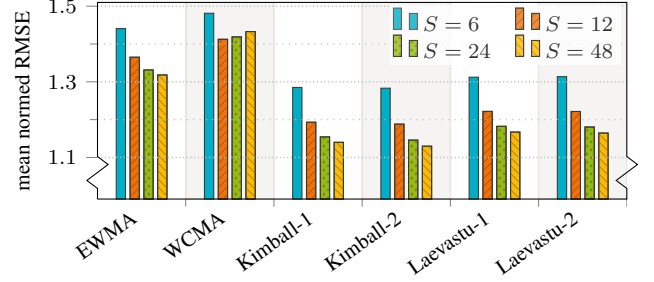


(b) node C

**Figure 3: Comparison of prediction error for two nodes through mean normed RMSE for $\alpha = 0.7$, $D = 3$.**

buffers (e.g., supercapacitors), where intra-day deviation may lead to depletion or a full buffer. To analyze the error at the end of a prediction horizon—i.e., not considering intra-day deviation—we calculated the mean error (ME) for every prediction. This metric is more relevant for large energy buffers that are rarely operated at extreme fill levels (e.g., rechargeable batteries).

Since there is one error value for each individual prediction, there are $S$ values per day of the solar traces. To allow for a setup phase, we only consider predictions after the fifth day (this is the rise time of an EWMA filter with $\alpha \leq 0.8$ to 67% of an asymptotic final value). Both RMSE and ME values are normed by dividing them through the overall mean harvest. Error magnitudes are thus directly compared to the harvest and the mean, energy-neutral consumption of a node. We did not calculate relative errors on a daily basis, because they are extremely large on days with low harvest and a too large prediction and thus distort the results massively.

## 4.3 Results

### 4.3.1 Comparison of Prediction Precision

The analysis of the RMSE shows a benefit of using cloud cover forecasts for harvest prediction. Using $S = 12$ or 24 gives a good trade-off between precision and memory consumption in all cases. Figure 3 portrays the mean normed RMSE for nodes A and C for different numbers $S$ of slots (results for node B are similar to node C). Results for different values of $\alpha$ (and $D$) are similar, where best values are achieved for $0.6 \leq \alpha \leq 0.9$ and $D = 3$.

The advantage of cloud-cover-enabled predictions ranges from $10-20\%$. Kimball conversion performs slightly better than Laevastu for our data set; the method of prediction (separate vs. combined) has a minor impact on the RMSE only. WCMA performs worst because its scaling method fails for slots further in the future—e.g., a
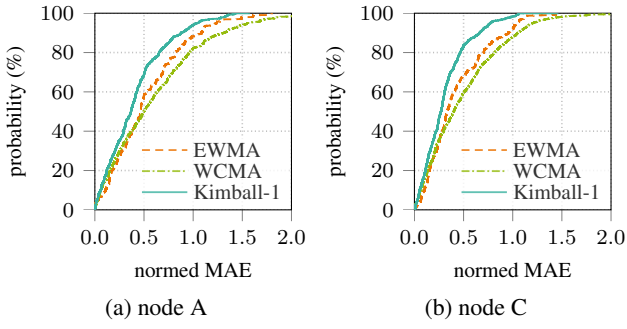
(a) node A        (b) node C

**Figure 4: Distribution of normed mean absolute errors for $S = 24$ and $\alpha = 0.7$ ($D = 3$ for WCMA)**



**Figure 6: Influence of the temporal resolution of cloud cover forecasts on the mean normed RMSE for Kimball-1 with $\alpha = 0.7$ for node C**

better harvest than usual at noon does not (necessarily) imply a better harvest in the morning of the following day. The figure reveals that node C profits earlier from increasing $S$, because its harvest pattern is more wide-spread over the day (cf. Fig. 2). The narrow harvest window of node A requires additional slots to improve the time resolution of prediction.

The RMSE in Fig. 3 may appear unexpectedly high (it ranges from 1.1 to 2.3 times the average harvest). The concept of time slots already produces a large normed RMSE between the actual traces and their slot representation (cf. Fig. 1); e.g., the normed RMSE for node A ranges from 0.81 ($S = 48$) to 1.43 ($S = 6$). For node C, it ranges from 0.50 to 0.96. We omit a comparison of prediction precision without the bias of time slots due to space constraints.

Next, we studied the distribution of the ME and found that all methods produce a relatively unbiased (close to zero) ME. However, EWMA and WCMA produce larger ME values, since they cannot adequately compensate for changing weather conditions. Figure 4 shows the cumulative distribution function (CDF) of absolute ME values. Both plots exhibit a notable improvement of Kimball-1 over EWMA and WCMA; e.g., 60% of all ME values of node A are at most 0.37 for Kimball-1, 0.45 for EWMA, and 0.55 for WCMA. The results for other values of $S$ and $\alpha$ are comparable.

In addition to statistical metric evaluation, we investigated the temporal course of prediction errors. Figure 5 shows harvest and prediction ME for an excerpt of 21 d for node A. It shows that particularly upon changing weather conditions, Kimball-1 has an improved precision (ME closer to zero) over EWMA. This is, e.g., visible on day 61 and the series of days starting with day 71, where the weather conditions and thus harvest increase from day to day. In some cases, e.g., day 69/70, erroneous cloud cover forecasts (not shown in the figure) prevent an improvement. The figure also reveals that WCMA produces small errors particularly on days with low harvest (e.g., days 61 and 62). However, it produces large errors on days 76 and 79.

### 4.3.2  Influence of Cloud Cover Resolution

In the following, we explain the influence of the resolution of cloud cover forecasts. For this purpose, we analyzed the mean normed RMSE produced by Kimball-1 with resolutions ranging from 1 h to 24 h by averaging the Kimball-converted cloud cover values over the corresponding time window (resolution). Note that only the resolution of the forecast is altered. The values used in Eq. (9) still have the resolution of the slot length.
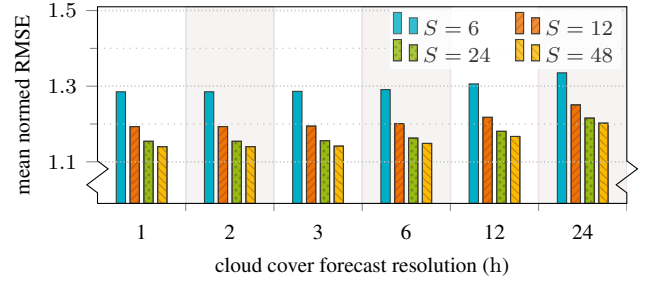
Figure 6 depicts the results for node C and different values of $S$. For a resolution of 1 h to 3 h the results stay virtually unchanged. Only for lower resolutions the RMSE increases slightly; e.g., for $S = 12$, the RMSE rises by less than 1%, if the resolution is changed from 1 h to 6 h and by less than 5% for a change to 24 h. Prediction accuracy with a resolution of 24 h hence stays considerably below those of EWMA and WCMA, cf. Fig. 3. As a result, the energy expenditure due to distributing cloud cover forecasts in the network can be reduced—with very little loss in prediction precision—by reducing forecast resolution.

## 5.  CONCLUSION

Existing harvest prediction algorithms for energy-harvesting sensor nodes fail in changing weather conditions. We presented two methods to incorporate global weather information, such as cloud cover forecasts, to resolve this weakness. For each of these methods, we applied two different models to convert cloud cover into a scaling factor for predicting the future harvest from historical, local data. Our methods are applicable to direct and maximum-power-point charging circuits, because they apply to both solar current and power predictions.

Our evaluation shows that the overall prediction error is significantly reduced by exploiting cloud cover information. The actual method of using cloud cover information has a minor impact only. More importantly, a detailed study of the prediction errors vs. real-world solar traces proves that particularly in changing weather conditions, the integration of cloud cover improves over existing approaches. The risk of depleting and wasting harvestable energy is thus decreased. In addition, our analysis indicates that the time resolution of cloud cover forecasts has a relatively low impact on prediction precision. It is hence possible to keep the amount of network traffic needed to distribute forecasts in the network at a low level; e.g., cloud cover forecasts with a resolution of 12 h can be distributed as piggy-backed information with link-level acknowledgments once every 12 h.

We also identified future research directions. First, it would be insightful to analyze further methods for cloud cover conversion and different types of weather forecasts (e.g., sunshine duration). Second, a field test with a real load adaptation algorithm—that, e.g., adapts the radio duty cycle of the nodes—would produce another metric of the actual performance improvement, e.g., in terms of network delay or throughput. Third, it appears promising to analyze if the scaling used by the WCMA algorithm can be combined with our cloud cover methods to compensate imprecise cloud clover forecasts.
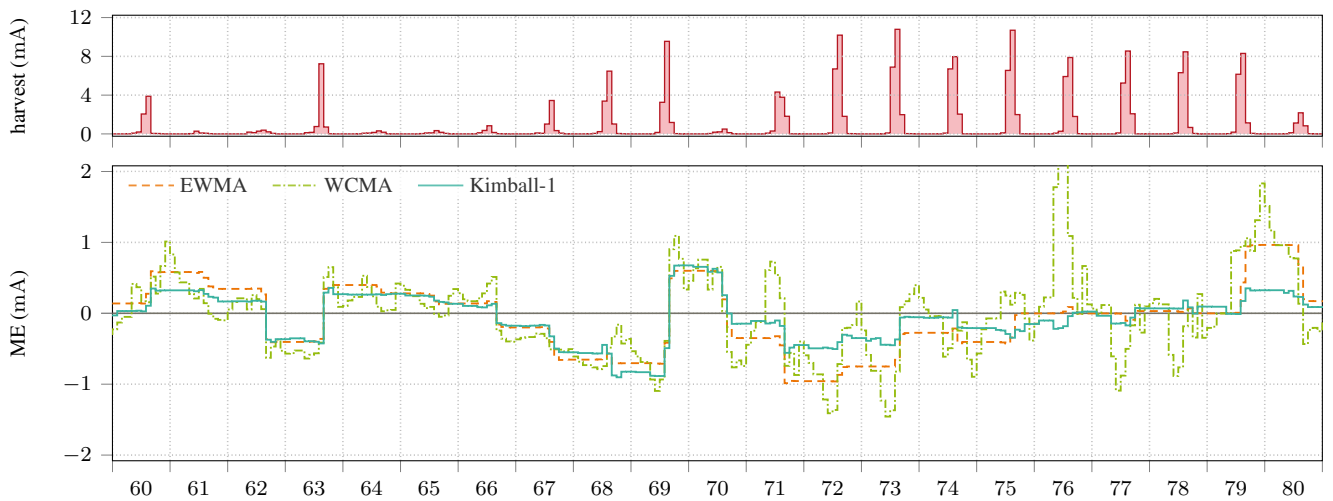
Figure 5: Excerpt of the harvest trace (ground truth, upper plot) of node A and mean prediction errors at the beginning of each slot. Prediction parameters are $\alpha = 0.7$ ($D = 3$ for WCMA) and $S = 12$ with a prediction horizon of one day. Values closer to zero are better.

## Acknowledgments

## 6. REFERENCES

[1] M. Ali, B. Al-Hashimi, J. Recas Piorno, and D. Atienza. Evaluation and Design Exploration of Solar Harvested-Energy Prediction Algorithm. In *Proc. Conf. on Design, Automation and Test in Europe*, DATE, Mar. 2010.

[2] C. Bergonzini, D. Brunelli, and L. Benini. Comparison of Energy Intake Prediction Algorithms for Systems Powered by Photovoltaic Harvesters. *Microelectronics Journal*, Nov. 2010.

[3] D. Brunelli, C. Moser, L. Thiele, and L. Benini. Design of a Solar-Harvesting Circuit for Batteryless Embedded Systems. *IEEE Transactions on Circuits and Systems I (TCAS-I): Regular Papers*, 56(11), Nov. 2009.

[4] M. Gorlatova, A. Bernstein, and G. Zussman. Performance Evaluation of Resource Allocation Policies for Energy Harvesting Devices. In *Proc. Intl. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, WiOpt, May 2011.

[5] A. Kansal, J. Hsu, S. Zahedi, and M. Srivastava. Power Management in Energy Harvesting Sensor Networks. *Transactions on Embedded Computing Systems (TECS)*, Sept. 2007.

[6] H.-C. Kim and E. Hofmann. Evaluation and Derivation of Cloud-Cover Algorithms for Calculation of Surface Irradiance in Sub-Antarctic and Antarctic Environments. *Antarctic Science*, 17, Mar. 2005.

[7] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks. In *Proc. Symp. on Networked Systems Design and Implementation*, NSDI, Mar. 2004.

[8] C. Lu, V. Raghunathan, and K. Roy. Efficient Design of Micro-Scale Energy Harvesting Systems. *Journal on Emerging and Selected Topics in Circuits and Systems*, 1(3), Sept. 2011.

[9] J. Lu and K. Whitehouse. SunCast: Fine-grained Prediction of Natural Sunlight Levels for Improved Daylight Harvesting. In *Proc. ACM/IEEE Conf. on Information Processing in Sensor Networks*, IPSN, Apr. 2012.

[10] F. Martins, S. Silva, E. Pereira, and S. Abreu. The Influence of Cloud Cover Index on the Accuracy of Solar Irradiance Model Estimates . *Meteorology and Atmospheric Physics*, 99, Apr. 2008.

[11] C. Moser, L. Thiele, D. Brunelli, and L. Benini. Adaptive Power Management for Environmentally Powered Systems. *Transactions on Computers (TC)*, 59(4), Apr. 2010.

[12] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava. Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. In *Proc. ACM/IEEE Intl. Symp. on Information Processing in Sensor Networks*, IPSN, Apr. 2005.

[13] J. Recas Piorno, C. Bergonzini, D. Atienza, and T. Simunic Rosing. Prediction and Management in Energy Harvested Wireless Sensor Nodes. In *Proc. Intl. Conf. on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, VITAE, May 2009.

[14] R. K. Reed. An Evaluation of Cloud Factors for Estimating Insolation Over the Ocean. Technical Report NOAA Technical Memorandum ERL PMEL-8, Pacific Marine Environmental Laboratory, Seattle, WA, USA, Sept. 1976.

[15] C. Renner, F. Meier, and V. Turau. Policies for Predictive Energy Management with Supercapacitors. In *Proc. IEEE Intl. Wksp. on Sensor Networks and Systems for Pervasive Computing*, PerSeNS, Mar. 2012.

[16] C. Renner and V. Turau. Adaptive Energy-Harvest Profiling to Enhance Depletion-Safe Operation and Efficient Task Scheduling. *Sustainable Computing: Informatics and Systems*, 2(1), Mar. 2012.

[17] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy. Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. In *Proc. IEEE Conf. on Sensor, Mesh and Ad Hoc Communications and Networks*, SECON, June 2010.

[18] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy. Predicting Solar Generation from Weather Forecasts Using Machine Learning. In *Proc. IEEE Intl. Conf. on Smart Grid Communications*, SmartGridComm, Oct. 2011.

[19] D. Spenza, C. Petrioli, and A. Cammarano. Pro-Energy: A Novel Energy Prediction Model for Solar and Wind Energy-Harvesting Wireless Sensor Networks. In *Proc. Intl. Conf. on Mobile Ad-Hoc and Sensor Systems*, MASS, Oct. 2012.