Experimenting Forecasting Models for Solar Energy Harvesting Devices for Large Smart Cities Deployments

Antonio Caruso[1], Stefano Chessa[2], Soledad Escolar[3], Xavier del Toro[3], Melisa Kuzman[4], Juan C. López[3]
[1]*Dept. of Mathematics and Physics "Ennio de Giorgi", University of Salento, Lecce, Italy*
[2]*Computer Science Department, University of Pisa, Pisa, Italy*
[3]*School of Computing Science, University of Castilla-La Mancha, Ciudad Real, Spain*
[4]*University of Mar del Plata, Mar del Plata, Argentina*

***Abstract*—To make sustainable large IoT deployments in smart cities, a promising approach is to develop a new generation of solar energy harvesting IoT devices based on the concept of energy neutrality. Key to this concept are the models for the forecast of energy production, which provide input to the energy-neutral schedulers governing the activities of the IoT devices. The development of such models however need to be validated against real-world conditions. To this purpose we propose a testbed aimed at the collection of real-world dataset about the energy parameters of energy harvesting IoT devices, and, on the base of such a dataset, we perform a comparative assessment of state of the art and novel energy production forecast models.**

***Index Terms*—IoT, testbed, dataset, solar energy harvesting, forecasting algorithms.**

I. Introduction

The fast development of Internet of Things (IoT) technologies (recent estimates expect over 50 billion of devices connected by 2020) is rapidly investing almost all human activities (from homes to industries), and cities are not an exception. IoT, in combination with cloud technologies can play a strategic role to build smart cities, due to their ability to obtain data from the most diverse sources in a diffuse and pervasive way and to provide fresh, high quality reports to the rulers based on analytics on such data.

Although part of the IoT devices are currently operating under the direct control of human beings (for example wearable devices and smartphones), the largest fraction of such devices is given by environmental sensors that operate autonomously in unattended mode, and such a fraction is expected to grow more and more in the future, simply because there is a limit to the number of devices that humans can directly control.

Especially in a smart city, where any potential IoT deployment may cover a very large space and involve a very large number of devices, it is important to make them even more autonomous and long-lasting, to keep the maintenance costs affordable. Under this respect, energy harvesting solutions are expected to play a key role, and, among these, photovoltaic panels are the most promising. IoT devices harvesting solar energy (hereafter we will just refer them as devices) exploit a photovoltaic panel to produce energy and a rechargeable battery to keep an energy buffer large enough to guarantee a continuous and potentially unlimited operation.

An essential question in the design of such devices is how to size their battery and panel to let them never run out of energy, which leads to the problem of making them energy neutral [1]. The concept of energy neutrality opens a new perspective in the optimized design of the devices, because it allows to modulate the load, i.e. the power consumption of the device, to match with the expected energy production. Lots of effort have been devoted by the researchers to identify different approaches to energy modulation, either by a suitable scheduling of communications [2] and/or by scheduling the sensing and processing tasks of the device, leveraging on a forecast of the energy production [1], [3], [4]. Recent approaches are particularly relevant for IoT devices since they exploit the public weather forecast obtained from the Internet to feed the solar energy production models [5].

An aspect that remains unanswered is however how these approaches would work in practice in a real environment. To the purpose, several research groups began developing prototypes to be used for experimentation in real conditions. Among these we cite [6]–[11], which, however, did not lead so far to the production of freely available datasets that researchers can use for the analysis and comparison of different solutions for energy harvesting management in IoT devices.

For this reason, we have developed a new prototype, which combines a solar-panel energy harvesting subsystem, a programmable IoT node equipped with sensors for the detection of weather conditions, and an independent microsystem acting as logger and capable of measuring all the energy-related parameters of the IoT node and of the harvesting subsystem. The IoT node can be programmed to implement different strategies concerning the execution of the sensing task and the management of the energy, so to experiment and compare different solutions. We have used this device to collect a dataset about energy production and weather conditions.

This paper presents briefly the prototype and the collected dataset, and then focuses on the experimentation with this dataset by analyzing the state of the art of algorithms that exploit public weather forecasts to estimate the forthcoming energy production of the solar panel. In particular, the contributions of this work are: (i) the prototype used for experimentation (which embeds a double subsystem, an IoT node and a data logger); (ii) a library that abstracts the double nature of the device to allow the programming of both the IoT node and of the data logger with a single program; and (iii) an experimental comparison of different production forecasting models, including models taken from the literature and some novel variations of such models. The remainder of this paper

is organized as follows. After reviewing the related works in Section II, we detail in Section III the design of our testbed from a hardware perspective. In Section IV we present an API intended to facilitate the applications writing on top of our testbed and describe an application so built to generate a dataset with the periodic readings of the sensors. In Section V we provide the results of evaluating different forecasting models of the solar energy production. Finally, in Section VI we draw the main conclusions and make suggestions for further research.

II. Related Works

This section reviews firstly some of the strategies addressed to achieve energy neutrality, which are generally evaluated by simulation, and secondly, energy-harvesting devices used in real deployments that may, therefore, provide accurate data to feed algorithms with different purposes.

A. Energy-Neutral Schedulers and Algorithms

Many works in literature have contributed to achieve energy neutrality, i.e. the ability for keeping indefinitely the device operation by means of some energy harvester (typically solar panels) and some optimization strategy that aligns the workload allocation with the energy availability. The first approach from Kansal and Srivastava [12] was the allocation of distributed tasks among the harvesting devices of a network according to their energy level. The same authors proposed years later in [5] to maximize the performance of the devices by dynamically scaling their duty cycles. This proposal was validated by means of a real experiment where an Heliomote platform (based on a Mica2 mote) was deployed on a residential area in Los Angeles and that ran uninterruptedly for 67 days during the summer of 2005. The results demonstrated that the application used an adaptive duty cycle close to the optimum. LSA [13] is an optimal real-time scheduling algorithm that finds the feasible optimal schedule where the task deadlines can be met. By simulation the authors demonstrate the feasibility of their approach providing how many tasks and in which order they will be executed, and quantify the overhead imposed by the scheduler on a BTnode mote.

We have addressed the energy neutrality problem in a progressive manner. In our first work [14] we consider an only energy harvester device and a basic task model where each task has a quality related to its sampling frequency. A scheduler found an assignment of tasks to time-slots over the reference period such that the overall quality was maximized in that period. The decision about *which task is assigned to the slot* $i+1$ is made at the end of slot i depending on the battery level, which is computed as a function of the real production in past and current slots, the consumption of the applications already executed, and an estimation of the energy production in the forthcoming slots. This work was later extended in [15] with a more refined task model, where each task has an execution time, a weight, a period, and where several scheduling plans are built on the basis of a set of these tasks. The quality of an scheduling plan is optimized under the constraint of energy neutrality. Here, our scheduler assigned initially the most efficient scheduling plan to all the slots and then followed a greedy strategy to upgrade (increase) or downgrade (decrease) the quality of the assignment depending on the energy produced and the battery level in that slot. In [16], [17] we extend the scenario to networked devices, first a sensor and a sink and then a star-connected network, where the scheduler keeps energy neutrality and maximizes the quality of all devices. To do that, the devices share their battery levels, energy productions and assignments, and a centralized scheduler recomputes their assignments accordingly. In [3] we consider an only harvester device and propose an scheduler that finds the optimal assignment that maximizes its overall utility while guarantees energy neutrality, where *utility* is a measure of the degree in which the application satisfies the user requirements, based on the idea that the same function can be implemented with different utility degrees, and a larger utility requires a larger energy consumption. We prove that finding such assignment is an NP-problem and show, by simulation, the optimal assignment that guarantees energy neutrality by assuming the hardware features of three popular IoT platforms: Raspberry PI, Arduino and TMote.

B. Energy Harvesting Devices in Real Deployments

Most of the works pursuing energy neutrality are evaluated by means of simulations, which model on one hand, the energy states of the hardware components of the device for estimating its consumption and, on the other hand, the energy production that will permit increase (or decrease) its energy availability. The solar energy, generally employed as energy source, is uncontrolled but predictable, which means that it is not possible to know in which moments and how much energy will be generated, but however its behavior can be modeled for prediction with some error margin. Thus, the energy production used is generally an estimation of the production based on the ideal daily production, which may be reduced by the presence of clouds, shadows, and other meteorological events that are not generally included in the simulations. Even when these events are introduced in the simulation, the evaluation lacks of the realism that we find in a real deployment. This fact motivates the need of using real testbeds for collecting data of energy production that can serve as basis for algorithms with different purposes, as finding the optimal scheduling and energy predictions. For example, in [18] is described the testbed ViSE equipped with an x86-processor, sensors, and a weather station that includes a wind turbine and a solar panel. The data collected from the weather station together with observational and forecast data from the National Weather Service enable to predict the power output from the solar panel/wind turbine. This work demonstrates that the forecasts may predict the future better than the immediate past does. The results show small differences between the prediction and the real power generated and an outperformance of 25% better with regard to the prediction done by PPF (past to predict the future). In [19] the authors use TelosB motes equipped with photovoltaic cells and wind micro turbines

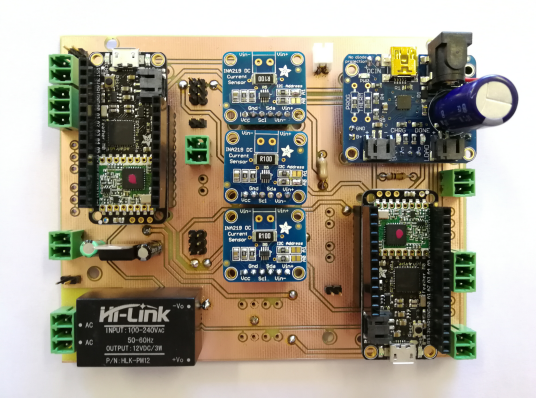

Fig. 1. The testbed.

for defining multi-source energy prediction models that are based on past energy observations to forecast future energy availability. The authors of [5] employ Waspmotes equipped with several sensors, a LoRa communication module, a battery and a solar panel to implement several machine learning methods to predict the solar power based on available weather public data. The three works have online published the dataset collected as result of their deployments. Differently to these works, we propose here a device specifically designed for evaluating different energy harvesting algorithms and we use the data collected during a real deployment for analyzing algorithms that use public weather forecasts to estimate the solar panel energy production.

III. Testbed Description

The testbed has been designed to provide a flexible and modular tool for the experimentation of different energy harvesting solutions. To this purpose it combines two independent microsystems, namely a full IoT device programmable to implement different energy management strategies and an additional data logger that independently records energy-related parameters (about the battery and the harvesting subsystem) while the IoT node is working (Figure 1 highlights the main components of the testbed). The testbed also comprises an energy harvesting subsystem that is currently based on a solar panel but that can be extended to include other harvesting sources (for example wind turbines), and a library that can be used to speed up the development of different energy management solutions. Note that the two subsystems have separate power supply, so to avoid interferences in the measurement of the energy-related parameters by the logger.

From the point of view of processing, the current version of the IoT node is powered by an Adafruit Feather M0 [20] with a RFM95 LoRa radio chip. This module offers 20 general purpose I/O pins, with up to 10 analog input pins. For our experimentation we have configured the IoT node with a serial line connecting to the data logger, a temperature and a humidity sensor, and power supply from the rechargeable battery which, in turn, is refilled by means of an energy harvesting subsystem based on a solar panel.

The data logger is also implemented with an Adafruit Feather M0 with a RFM95 LoRa radio chip. The data logger measures the currents and voltages of the IoT node (to obtain its power consumption), of the rechargeable battery and of the solar panel. Each measured data is associated by the logger to a time stamp. The logger also estimates the solar irradiance by a latched mini relay that measures the short-circuit current of the panel. Finally, the data logger also has an OLED monochrome display to show the current system state.

The energy harvesting subsystem is currently designed with a 2 Watts photovoltaic solar panel [21] (the energy harvester), a solar charger (to operate the power conversion) and an energy storage based on a lithium rechargeable battery. The solar panel, which is designed for outdoor applications, provides a nominal current of 340 mA with an output voltage of 6.5 V (i.e. maximum power point in standard conditions).

The solar charger can provide the output power to the battery or directly to the load. In particular, it powers directly the load (i.e. the IoT node) if the harvested energy is sufficient (and the surplus of energy is directed to recharge the battery, if necessary). Otherwise, if the requirement of the load is higher than the harvested energy the necessary power is supplied by the battery. The rechargeable battery has a capacity of 2000 mAh with a nominal voltage of 3.7 V, and the output ranges from 4.2 V (when fully charged) to 3.3 V (when fully discharged).

IV. API and Dataset Description

As previously mentioned, the ultimate goal of the testbed described in Section III is to support and evaluate the uninterrupted execution of a wide variety of outdoor monitoring applications, with different energy requirements and under different meteorological conditions, by using as energy harvester a solar panel. To the purpose of building these applications, we have developed a function library on the microprocessor Adafruit Feather M0 by using the development environment Arduino 1.8.7. The software library has been designed to support software development on top of our testbed in a simple way, by hiding the programmer the hardware details, in particular, the existence of two microsystems embedded into the platform, i.e. the IoT node and the data logger as well as the specialization of tasks carried out by each one of them. Thus, the library offers the programmer an abstract view of the prototype as a single device, and thus with a single program it is possible to control both the IoT device and the data logger. At compilation time the functions are then partitioned over the two microsystems.

The library comprises a set of 26 functions that cover both the sampling activities from the IoT node and the sampling activities from the data logger, through the different sensors that they integrate (current, voltage, temperature, wind of speed, and humidity), storing into an SD card, communication by using LoRa technology (note that at this moment of development only LoRa communication has been supported), energy harvesting from a solar panel and energy storing by means of a 2000 mAh Li-Po battery. The library is open source

TABLE I
PROTOTYPE OF THE LIBRARY FUNCTIONS.

Prototype	Description
int initializeRTC(void)	Initialize the Real Time Clock (RTC)
void adjustRTC(void)	Adjust the time for RTC
void getTime(void)	Returns the time from RTC
int initializeLoRa(void)	Initialize the LoRa communication module
int sendLoRa(String)	Send data through LoRa module
String recvLoRa(void)	Receive data from Lora module
float getTemperature(void)	Read the temperature value ($^{\circ}C$)
float getHumidity(void)	Read the humidity value (%)
float getSpeedOfWind(void)	Read the wind speed from anenometer (m/s)
float getBatteryVoltage(void)	Read the battery voltage (V)
void initINA0(void)	Initialize the current sensor INA0
void initINA1(void)	Initialize the current sensor INA1
void initINA2(void)	Initialize the current sensor INA2
float getLoadPower(void)	Read the load power from INA1 (mW)
float getBatteryPower(void)	Read the battery power from INA2 (mW)
float getPanelPower(void)	Read the panel power from INA0 (mW)
float getLoadCurrent(void)	Read the load current from INA1 (mA)
float getBatteryCurrent(void)	Read the battery current from INA2 (mA)
float getPanelCurrent(void)	Read the panel current from INA0 (mA)
int initializeSD(void)	Initialize the micro SD
int open (String, int)	Open a file to read/write on micro SD
void close(void)	Close a file
void writeline(String)	Store data on micro SD
String readline(void)	Read a line from the micro SD
void initializeDisplay(void)	Activate the OLED monochrome display
void displayOLED(String,String)	Display a title and a text on the OLED display

and it is available online at: https://github.com/arco-group/energy-harvesting-dataset.git. Table I presents the prototype of the functions included in our library.

By using this library we have developed several IoT applications on top of our testbed; in particular, we highlight an application that generated a dataset by sampling periodically each sensor and writing these data into the SD memory card. The data collected include information in terms of time stamp, energy resource, battery state, node consumption and meteorological conditions: Date (dd/mm/yyyy), Time (hh:mm:ss), Load (mA), Battery (mA), Panel (mA), Wind (m/s), Temperature (°C), Humidity (%), and Voltage (V). The values of Load, Battery, and Panel correspond to the currents of the IoT node consumption, battery supplied (if > 0) or absorbed (if < 0) and the short-circuit current of the photovoltaic panel, which is proportional to the energy production. The testbed ran this application for a preliminary data collection campaign conducted in Ciudad Real (Spain), from July 31st to October 4th, 2018. The testbed operated in this period without interruptions, collecting all energy-related data and environmental data with a sampling period of 1 minute. The resulting dataset comprises over 93438 data records, it is public and it can be downloaded at https://github.com/arco-group/energy-harvesting-dataset.git.

The first four graphs in Figure 2, represent respectively the production in mA of the solar panel, the temperature, the wind speed and the humidity levels across the entire dataset. The plots below represent the curve of the current flowing into (negative) and out (positive) from the battery, the battery residual level (as percentage of its capacity) and the load on the IoT node. Since the application running in the node simply collect, regularly in time, some environmental data, the load in the node is mainly constant.

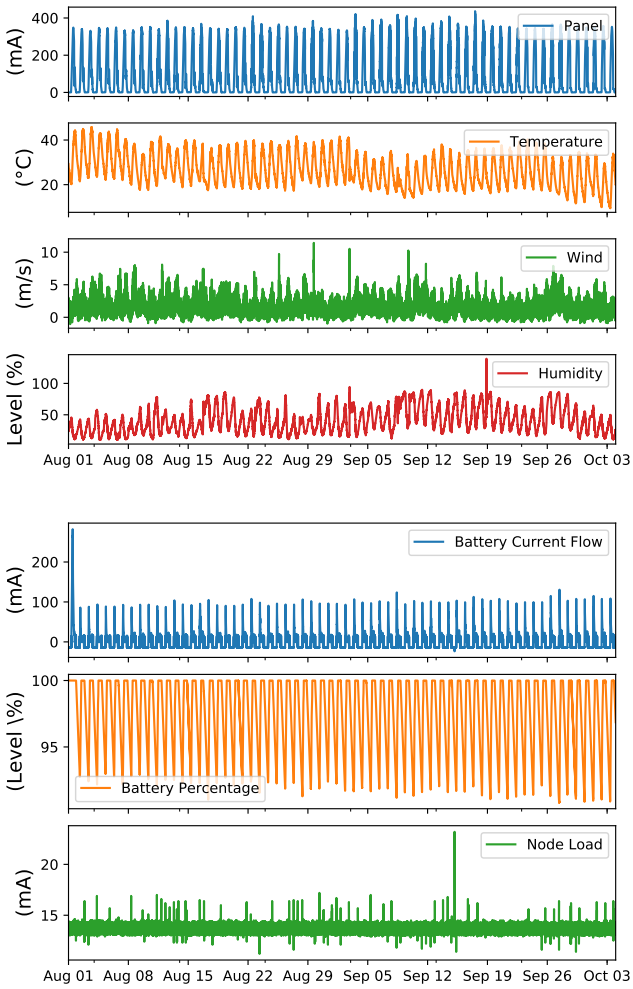

Fig. 2. Plot of all values of the dataset. The solar panel production (mA), the temperature (°C), wind speed (m/s) humidity level (%), and Battery Flow (mA), Battery Level (%), and Node load (mA).

V. EXPERIMENTAL RESULTS

A key ingredient in the design of an energy neutral IoT node is a good scheduling algorithm, capable of dynamically adapt the node load taking into account not only the actual amount of energy harvested by the solar panel, but also the expected energy in the future. We already reviewed in Section II some of them. A scheduling algorithm run periodically in the node, and define a set of possible parameters that affect its load: i.e. the frequency of duty-cycling, the sampling-rate, the radio transmission power and the amount of data processing.

The optimization of these parameters act at different levels: the lowest levels require high frequencies decisions, for example tuning the radio at the level of a single packet transmission; while the higher levels act on more long term parameters, that can remain constant over longer time windows, like the duty cycle of the application or the sampling frequency of

sensors. The variation in the energy harvested is limited over short intervals of time, most algorithms run periodically, using a limited time horizon (typically a day) divided in shorter periods like 30 minutes.

We aggregated (averaged) over periods of 30 minutes all values in the dataset. Since the solar panel production exhibits a typical daily seasonality, we plotted in Figure 3 the average of the production over all days at the same half-hour, from 6:30 to 21.30; the vertical red bars represent the standard deviations in the series, i.e. its variability across different days. The plot shows an expected daily pattern: the energy produced, that clearly depends on the position of the Sun (and the relative position and orientation of the solar panel), steadily increases after 6:30 at sunrise, reaches its maximum at midday and starts to decrease until it becomes zero after sunset. Temporary weather conditions like passing clouds affect the variance in the production especially at peak hours, and in the afternoon after 16:00 when the Sun starts to be less perpendicular to the panel, lowering its efficiency until it rapidly drops to zero.

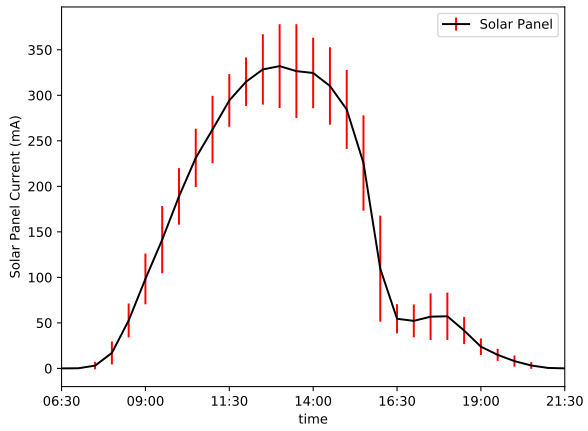

Fig. 3. Plot of the mean solar energy production every 30 minutes, the vertical red bars are the deviations from the mean.

A. Forecasting Models

As a preliminary step in the direction of studying the performance of different scheduling algorithms, we focused this Section on testing the performances of different forecasting models. We review some of them, moving from the simplest to the more complex, several forecasting models are *statistically based*, i.e. they consider the past as a key indication of what could happen in the future. Consider the time series of the energy production until time t, the simplest possible, statistically based, forecast model, that we call *previous half-hour* just compute the forecast $y_{t+1} = x_t$, i.e. it forecasts for the next half-hour the same production of the last 30 minutes.

The quality of any forecast model can be evaluated by computing the *root mean square deviation* (RMSD) i.e. $\sqrt{\mathbb{E}[(x_i - y_i)^2]}$, and dividing by the maximum of the production (360 mA) we get a percentage of error, that is simple to reason about. With our dataset, *previous half-hour* has a surprising small error of 9.24% that we use as a baseline. Note that this depends on the nature of our dataset that comprises only a few months in summer. Another simple model is to use the values observed in the previous day, so $y_{t+1} = x_{t-48}$ or an average between the previous day and the last half-hour.

TABLE II
FORECASTING MODELS AND RMSD FOR EACH OF THEM

Previous Half-Hour	9.24%
Previous Day	8.42%
$(0.448, 0.551)$ Averages of Above	6.63%
EWMA (Kansal)	6.34%
Scaled EWMA	5.97%

In Kansal [12] the exponential weighted average of all previous days is used together with the value observed at the end of the previous 30 minutes as the forecast: i.e. they use $y_{t+1} = \alpha x_t + (1-\alpha) y_{t-48}$, i.e y stores the weighted averages of the solar panel production in the same slot for all the days in the dataset. The optimal parameter $0 < \alpha < 1$ that affects the weight of the current vs the past observations on the forecast is numerically optimized. The relative errors of all this statistically based methods are reported in Table II, and as shown, the baseline is improved from 9.24% to 6.34%. Other statistically based methods like EWMA [1], EEHF [12], or Enhanced-EEHF [22] try to rapidly adapt the forecast to the abrupt changes caused by sporadic clouds.

Another class of models that will take on the utmost importance, exploits the connectivity to the Internet, both directly and through fog or edge devices, to download a weather forecast for the next day and use it to directly affect some statistics-based model, or to implement machine learning-based forecasting models. Since that we did not have the opportunity to collect weather forecasting in the days of the experiment, we used the historic forecast provided by the online service AEMET (The Spanish Meteorological Agency). We found that the solar energy production is not directly correlated to these variables, but the second order difference of the forecasted maximum temperature is sufficiently correlated to the solar panel production as shown in Figure 4.

We present here a very simple forecasting model built on an scaled EWMA. We use the change of the maximum temperature to scale the EWMA forecasting appropriately, in particular, we use the second order difference of the temperature. If at time t, (the end of a day) we denote with M_t the *forecast* of the maximum temperature for the next day, and given $M_t^{'} = M_t - M_{t-1}$ and $M_t^{''} = M_{t+1}^{'} - M_t^{'}$, we compute a forecast for the expected panel production simply as $y_{t+1} = EWMA_{t+1} * (1 \pm \frac{|M_t^{''}|}{100})$ with a positive correction if the M'' increases and negative if it decreases. Even this simple model is able to better adapt the EWMA forecast to the changing weather conditions, lowering the error to 5.97%. All values of the relative errors for all forecasting models are reported in Table II.

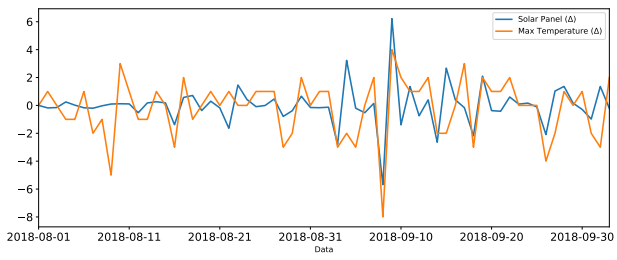

Fig. 4. Plot of the (second order) changes in the Solar Panel Production vs the Forecasted values of the maximum Temperatures.

VI. Conclusion

This paper presents the initial stage of a long-term research activity aimed at the systematic creation of datasets for the experimentation of energy harvesting solutions for IoT devices, and at the consequent systematic assessment of models and algorithmic solutions for the management of such devices.

In the current stage we have developed a device composed of three subsystems: an energy harvester with a rechargeable battery, a programmable IoT device (at this time collecting weather-related data) and a data logger for the real-time collection of several energy-related parameters of the energy harvester and of the IoT device. We have used this device to perform a first campaign of data collection that resulted in a first dataset over which we conducted an experimental comparison and validation of several energy production forecasting models. The next steps will aim at the consolidation of this dataset, by running several data collection campaigns over different periods of the year and in different weather conditions, and in the collection of weather forecast data from public channels to complement the dataset. We expect that the final dataset will result a valuable asset to experiment different solutions for energy harvesting IoT, including energy-neutral schedulers that exploit the weather forecasts to optimize the energy management. In some cases, the scheduler and a more advanced class of weather forecasting models based on machine learning cannot run directly on the nodes. The correct placement of these computations, in the edge devices or in the cloud, represent another important design decision that we would like to explore in the future.

Acknowledgment

This work has been partly funded by the Spanish Ministry of Economy and Competitiveness under projects PLATINO (TEC2017-86722-C4-4-R) and CitiSim Itea3 (TSI-102107-2016-8 ITEA3 Num. 15018) and by the Regional Government of Castilla-La Mancha under project SymbIoT (SBPLY/17/180501/000334).

References

[1] A. Kansal, J. Hsu, S. Zahedi, and M. B., Srivastava, "Power Management in Energy Harvesting Sensor Networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, Sep. 2007.

[2] G. Amato, A. Caruso, and S. Chessa, "Application-driven, energy-efficient communication in wireless sensor networks," *Computer Communications*, vol. 32, no. 5, pp. 896–906, 2009.

[3] A. Caruso, S. Chessa, S. Escolar, X. del Toro, and J. C. López, "A Dynamic Programming Algorithm for High-Level Task Scheduling in Energy Harvesting IoT," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2234–2248, June 2018.

[4] S. Escolar, A. Caruso, S. Chessa, X. del Toro, F. J. Villanueva, and J. C. López, "Statistical Energy Neutrality in IoT Hybrid Energy-Harvesting Networks," in *IEEE Symposium on Computers and Communications (ISCC)*. IEEE, June 2018, pp. 444–449.

[5] 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 *Proceedings of the Seventh International Conference on the Internet of Things*. New York, NY, USA: ACM, 2017, pp. 2:1–2:8.

[6] G. Jackson, S. Kartakis, and J. McCann, "Accurate Models of Energy Harvesting for Smart Environments," in *IEEE International Conference on Smart Computing (SMARTCOMP)*, May 2017, pp. 1–7.

[7] L. Borin, M. Castro, and P. D. M. Plentz, "Towards the Use of LITMUS RT as a Testbed for Multiprocessor Scheduling in Energy Harvesting Real-Time Systems," in *VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2017, pp. 109–116.

[8] S. Bader and B. Oelmann, "A concept for remotely reconfigurable solar energy harvesting testbeds," in *2017 IEEE SENSORS*, Oct 2017, pp. 1–3.

[9] A. K. R. Venkatapathy, M. Roidl, A. Riesner, J. Emmerich, and M. ten Hompel, "PhyNetLab: Architecture design of ultra-low power Wireless Sensor Network testbed," in *IEEE Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2015, pp. 1–6.

[10] S. Rao and N. B. Mehta, "Hybrid Energy Harvesting Wireless Systems: Performance Evaluation and Benchmarking," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 4782–4793, Sep. 2014.

[11] M. Masoudinejad, J. Emmerich, D. Kossmann, A. Riesner, M. Roidl, and M. ten Hompel, "Development of a measurement platform for indoor photovoltaic energy harvesting in materials handling applications," in *The 6th Int. Renewable Energy Congress*, March 2015, pp. 1–6.

[12] A. Kansal and M. B. Srivastava, "An environmental energy harvesting framework for sensor networks," in *Proceedings of the International Symposium on Low Power Electronics and Design*, ser. ISLPED '03. New York, NY, USA: ACM, 2003, pp. 481–486.

[13] C. Moser, J. Chen, and L. Thiele, "Dynamic power management in environmentally powered systems," in *15th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Jan 2010, pp. 81–88.

[14] S. Escolar, S. Chessa, and J. Carretero, "Optimization of Quality of Service in Wireless Sensor Networks Powered by Solar Cells," in *IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, July 2012, pp. 269–276.

[15] S. Escolar, S. Chessa, and J. Carretero, "Energy management of networked, solar cells powered, wireless sensors," in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. ACM, 2013, pp. 263–266.

[16] S. Escolar, S. Chessa, and J. Carretero, "Energy-neutral networked wireless sensors," *Simulation Modelling Practice and Theory*, vol. 43, pp. 1–15, 2014.

[17] S. Escolar, S. Chessa, and J. Carretero, "Quality of service optimization in solar cells-based energy harvesting wireless sensor networks," *Energy Efficiency*, pp. 1–27, 2016.

[18] N. Sharma, J. Gummeson, D. Irwin, and P. Shenoy, "Cloudy computing: Leveraging weather forecasts in energy harvesting sensor systems," in *7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, June 2010, pp. 1–9.

[19] A. Cammarano, C. Petrioli, and D. Spenza, "Online energy harvesting prediction in environmentally powered wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 17, pp. 6793–6804, Sep. 2016.

[20] Microchip, "ATSAMD21G18," https://www.microchip.com/wwwproducts/en/ATsamd21g18, 2018.

[21] Voltaic Systems, "2 Watt Solar Panel," https://www.voltaicsystems.com/2-watt-panel, 2019.

[22] K. Kinoshita, T. Okazaki, H. Tode, and K. Murakami, "A data gathering scheme for environmental energy-based wireless sensor networks," in *5th IEEE Consumer Communications and Networking Conference*, Jan 2008, pp. 719–723.