A Testbed and an Experimental Public Dataset for Energy-Harvested IoT Solutions

Melisa Kuzman*, Xavier del Toro García†, Soledad Escolar†, Antonio Caruso‡, Stefano Chessa§ and Juan Carlos López†

*Universidad Nacional del Mar del Plata, Mar del Plata, Argentina. Email: melisakuzman@fi.mdp.edu.ar

†School of Computer Science, University of Castilla-La Mancha, Ciudad Real, Spain

‡Department of Mathematics and Physics Ennio de Giorgi, University of Salento, Collegio Fiorini, Lecce, Italy

§Department of Computer Science, University of Pisa, Pisa, Italy

***Abstract*—The Internet of Things (IoT) paradigm poses a great variety of application domains where million of devices work uninterruptedly to improve some aspect of our lives. To support the continuous execution of the applications working on the devices, energy harvesting systems enable to extract the energy found naturally in the environment (for instance from the sun or from the wind) and convert it into energy able to either sustain the device's operation and recharge its batteries which, in conjunction with an appropriate scheduling strategy, led to the device to an electrically sustainable state (i.e. an energy-neutral state). Most of the works found in literature oriented to achieve energy neutrality are however evaluated by means of simulation which means that, in spite of precisely modeling hardware features and energy productions, lack of the realism that we find in a real deployment. A minor part of the works are based on a real deployment but do not share the collected data that permit to replicate the analysis. With this purpose in mind, in this article we describe a testbed designed for outdoor monitoring purposes in the IoT context, equipped with several sensors for weather conditions monitoring and with a solar panel to provide application lifetimes potentially infinite. The testbed was deployed on the roof of a building and it executed uninterruptedly an application able to generate a dataset with the collected information over a period of more than two months. This dataset has been online published to be used for different researching purposes, as for instance, prediction models of the energy production.**

***Index Terms*—IoT, Deployment, Testbed, Solar Energy Harvesting, Dataset.**

I. Introduction

The impressive growth of Internet of Things (IoT) market — the expectations for 2020 is to have over 50 billion devices connected to the Internet [1] — is paving the way for many new applications and services. In many cases, such applications involve the monitoring/control of outdoor scenarios consisting in large deployments of stationary devices. In such deployments, however, the use of battery-powered devices may result, in many cases, unfeasible due to high costs incurred for maintenance. A viable alternative is to use energy harvesting devices that may operate uninterruptedly for very long time periods, with a potentially unlimited lifetime. The use of photovoltaic (PV) panels and batteries is arguably the most common and mature technology for energy harvesting. The purpose of a rechargeable battery is to provide an energy buffer to use when the energy production of the panel is insufficient or null (e.g. in a cloudy day or at night). Nevertheless, even with the battery and depending on its capacity, if the period of low energy production is too long (for instance, in a sequence of very cloudy days) the device may stop working, until the solar panel starts its production again. This fact has motivated several recent research works aimed at introducing strategies to modulate the energy consumption of the device, by either controlling its duty cycle or its functionalities [2], in order to match it with the actual and expected energy production [3], [4], on the base of an estimation of the next energy production of the panel [5], [6]. In most cases, these works evaluate the performance of the proposed solutions based on simulations and on energy consumption models of the devices. This is motivated by the fact that deploying a testbed and running it for a sufficiently long period is a time-consuming task that incurs in additional efforts and costs. On the other hand, assessing a solution based on real equipment in real environments and conditions would provide a significant added value to the research. It is in this context that the design and deployment of an energy-harvested IoT testbed is of interest to develop and validate new energy-aware algorithms.

In other research fields similar problems have been solved by producing and making publicly available suitable datasets [5], [7], [8]. Some open datasets concerning solar power already exist, but they are not specifically targeting IoT installations, but rather, home or grid-level PV installations [9]. On the other hand, some prototype that may be used to this purpose already exist. Among these, we mention [10] that develops a prototypal testbed to assess the proposed models of energy production (but the dataset is not published). Another testbed [11] addresses the use of an extended version of the Linux kernel in energy harvesting real-time systems, which, however, addresses a rather different class of IoT than the one (low-power devices) we address in our work. The work in [12] describes the implementation of a flexible platform for the monitoring and control of reconfigurable energy harvesting systems, and another interesting platform for wireless sensor networks that also include energy harvesting capabilities is presented in [13], and two experimental platforms for energy harvesting are also introduced in [14] and [15]. However, neither of these proposals address the development of a dataset for the assessment of algorithms/models of energy harvesting devices.

With the aim of capturing the particularities of solar energy harvesting for IoT applications, this work aims to provide a public dataset for such purpose. To this end, the experimental IoT testbed includes data logging functionalities and sensors to provide detailed measurements about the instantaneous energy production obtained by the PV panel, the battery state of charge and IoT node power consumption. Furthermore the wind power resource is also measured by means of an anemometer. At the same time, the testbed can be programmed to run applications that modulate different energy consumption, so to experiment under different conditions.

The remainder of this paper is organized as follows. In Section II we detail the design of our testbed from a hardware perspective and in Section III we describe the application that was implemented on the testbed to generate the dataset. Section IV presents an analysis of the collected dataset and shows how it can be used for prediction of the energy production. Finally, in Section V we draw the main conclusions of this work and make suggestions for further research.

II. Testbed Description

The general diagram of the testbed developed is shown in Figure 1. In order to enable different hardware configurations and processing capabilities, a modular system has been designed, comprising an IoT node to implement different energy-aware solutions and a data logger (DL) to record the most relevant variables of the testbed. A picture of the testbed PCB design is shown in Figure 2, where the main components are identified. Depending on the application and the equipment location, it might be necessary to have a system with different processing power, energy harvesting dimensioning, variable power consumption and different communication technologies. Hence, for the design of the testbed the following set of preliminary features, based on the identified needs, has been considered:

- To provide flexibility to test a wide range of solutions.
- To have a data logging functionality and an IoT node energetically independent.
- To perform data logging with a reliable and continuous power supply.
- To adopt a commonly-used architecture for IoT and embedded systems, namely the ARM architecture.
- To provide the possibility of outdoor deployment.
- To provide the possibility of only using the DL, IoT node or both.
- Integration of different energy harvesting (EH) and storage technologies, currently only a photovoltaic (PV) panel and a battery.

The following subsections describe the components depicted in the general scheme of the testbed shown in Figure 1.

A. Processing

With the purpose of implementing several technology options, the family of development boards known as Adafruit Feather has been selected to design the modular testbed. These modules are both standalone and stackable. There are several

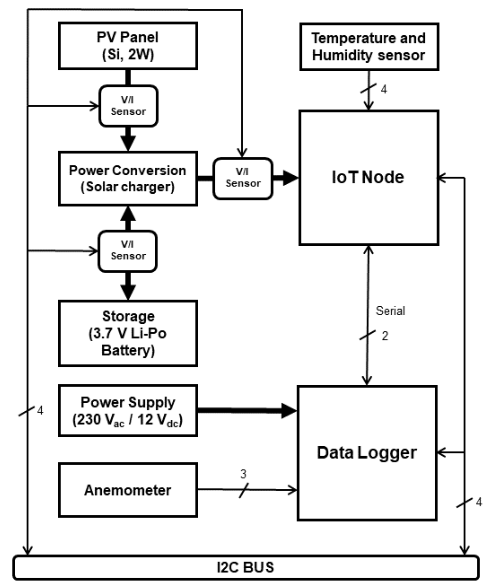

Fig. 1. General testbed diagram.

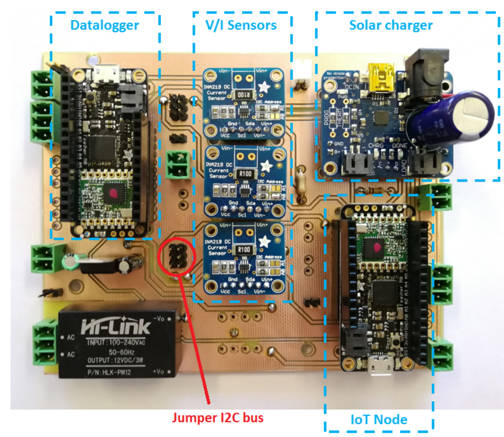

Fig. 2. Picture of the testbed PCB.

Feather boards that combine different microcontrollers (e.g. ATSAMD21, ESP8266, 32u4, ATmega32u4, ESP32) with the possibility of using different communication technologies (e.g. Lora, Wifi, Bluetooth, 2G/3G, RF). This allows to test different processing capabilities or communications technologies with the same board layout (i.e. form factor). Furthermore, a broad family of stackable boards is available, the so-called Feather Wings, to provide additional functionalities to the system.

By means of several jumpers the platform can be configured to work with two microcontrollers, as shown in Figure 2, or a single microcontroller (either the IoT node or the DL). Also, it is possible to choose how the microcontrollers are supplied depending on the system application. The aim of separating power supplies of the IoT node and of the DL is to obtain accurate measurements of the node consumption with no energy overhead for the IoT node. For this reason, each board supplies their associated sensors. While the IoT node is always connected to the energy harvesting system, the DL can be either supplied by the harvesting system, by an independent 12 V battery or by the 230V AC mains, depending on the availability in a particular site.

1) IoT node: An Adafruit Feather M0 [16] with an RFM95 LoRa radio chip is used. Among the most relevant characteristics, this module has 20 General Purpose Input Output (GPIO), enabling up to 10 analog inputs, 8 Pulse-Width Modulation (PWM) outputs and UART, SPI and I2C communication interfaces. As Figure 1 shows, the most relevant connections of the IoT node are:

- Serial communication (UART) with the DL.
- A temperature and humidity sensor connected to 2 digital GPIO (two-wire communication interface).
- Jumpers to connect the microcontroller I2C interface to the general I2C bus.
- Power supply coming from the energy harvesting system. The supply voltage is sensed with an analog input.

2) Data Logger: Another Adafruit Feather M0 with RFM95 LoRa Radio was chosen as the DL. This module receives data from the IoT node by using a serial communication. It also measures both the system currents and voltages with the I2C interface to obtain the IoT node consumption, the PV Panel energy production and the battery state of charge (SOC). Additionally, it obtains the wind speed by means of an anemometer connected to an analog input. The obtained information is stored into a micro SD card with a timestamp that is obtained from a real-time clock (RTC). Three Feather Wing boards are stacked on top of the Feather M0 to add functionalities to the DL:

- A latched Mini Relay measures the PV Panel short-circuit current to estimate the solar irradiation.
- An OLED monochrome display shows to the operator the actual system state (i.e. warnings, errors, data stored).
- An Adalogger module provides a battery-backed RTC and a micro SD card storage. The RTC uses the I2C bus, and the micro SD socket is connected to the SPI port pins, requiring an additional pin to enable the Chip Select.

B. Energy Harvesting

According to the power supply requirements and the location of the testbed, the energy harvesting system is designed with a PV panel as energy harvester, a solar charger and a Li-Po battery as energy storage, as shown in Figure 1.

1) Energy Harvester: Due to the environmental conditions and the system energy requirements, a 2 Watts PV panel [17] was selected as the energy harvester. This panel provides a nominal current of 340 mA with an output voltage of 6.5 V (i.e. maximum power point in standard conditions). The PV panel construction characteristics (waterproof, scratch resistant, and UV resistant) make it an appropriate option for long term outdoor use.

2) Power conversion: In order to efficiently extract the energy harvested by the PV panel and control both the battery charging process and load supply, a solar charger circuit is used (https://www.adafruit.com/product/390). This circuit has one input, the PV panel, and two different outputs: battery and load. When the energy supplied by the PV panel is sufficient, the current directly goes to the load output, where the IoT node is connected. Furthermore, if the battery is not completely charged the surplus current from the PV panel is used to charge it. If the current required is higher than the PV panel production, the current is supplemented from the battery.

3) Energy Storage: In order to store the surplus energy harvested by the solar panel, a Li-Po battery is used. The battery supplies the IoT node when the solar energy is not enough due to the weather conditions, or at night time. The selected 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).

C. Sensors

1) Temperature and Humidity: Because the wide range of operation, resolution and the robust and integrable design, the SHT10 Mesh-Protected and Weather-Proof sensor [18] is selected to measure the temperature and the humidity. The sensor output is a fully-calibrated digital signal and the information is transmitted with a 2-wire protocol. The chip has been designed by Sensiron and it operates between -40 to 123.8°C, and 0 to 100% relative humidity. The measurement resolution can be configured changing the status register by command. The settings used in this paper allow to measure the temperature and humidity with a resolution of 0.01°C and 0.05%, respectively.

2) Anemometer: We use the anemometer [19] to measure the wind speed. This sensor has an analog output with the range 0.4 2V, sensing the wind speed from 0.5 to 50 m/s.

3) Current sensor: Three INA219 [20] sensors are used in the testbed to measure the battery, IoT node, and PV currents and voltages. This module support the I2C interface, operating only as a slave device and it features up to 16 programmable addresses. Using this interface allows changing the quantity of sensors without wiring new tracks for each device. The default register configuration is used, enabling to measure a range of ± 3.2 A, with a resolution of ± 0.8 mA.

III. Dataset Description

The testbed described in Section II has been designed to support the prolonged execution of outdoor monitoring applications, by means of an energy harvester, a solar panel, which let the battery level grow depending on the application consumption and the weather conditions. Since the final purpose of our testbed is to evaluate different IoT scenarios with energy-neutrality requirements, i.e. to evaluate the capacity of the system for running indefinitely, applications with different energy consumptions that execute under different weather conditions should be implemented. To build these applications, we have developed a function library on the microprocessor Adafruit Feather M0 by using the development environment Arduino 1.8.7. The proposal of this library is to ease the development of applications on top of the proposed testbed by abstracting away the hardware complexity and by hiding the programmer the division of tasks carried out by the DL and by the IoT node. Due to the usage of this library, from the programmer perspective the platform is just one single device that integrates the capabilities of sensing (through several current sensors, and a temperature/ humidity sensor), storing into an SD card, communication by using different technologies (i.e. WiFi, LoRa, Bluetooth), energy harvesting from a solar panel and energy storing by means of a 2000 mAh Li-Po battery. Thus, our library is composed of a set of functions that provide basic services to request operations to the overall platform: sample each sensor, read and write from/to the SD card, send/receive from a radio, and other utility functions.

As an example, consider the *Dataset* application, which is intended to periodically gather data from each sensor and write these data into the SD memory card. The variables of interest collected to produce the dataset 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), Temp (°C), Hum (%), and Volt (V). Algorithm 1 lists the pseudocode to build this dataset. A complete description of the variables used is provided in Table I.

Algorithm 1 A *Dataset* application with 1 minute period.

```
Require: Period = 60 seconds
Ensure: Date, Time, Load, Battery, Panel, Wind,
  Temp, Hum, Volt
  loop
    t0 = getTimeRTC(); // t0 in ms
    Load = readINA_0();
    Battery = readINA_1();
    Panel = readINA_2();
    Wind = readAnenometer();
    Temp = readSHT10Temp();
    Hum = readSHT10Hum();
    Volt = readBatteryVolt()
    writeSD(getDateRTC(), getTimeRTC(), Load,
    Battery,Panel, Wind, Temp, Hum, Volt)
    t1 = getTimeRTC(); // t1 in ms
    delay(Period*1000-(t1-t0))
  end loop
```

Note that the duty cycle (DC) of this application can be easily computed as $DC = \frac{(t1-t0)}{Period}$. After running the application, we obtained in average values of $(t1-t0)$ equal to 2150 ms, which results into a DC of approximately 3.6%.

IV. Experimental Evaluation

We have deployed the testbed on the roof of one of the buildings of the University of Castilla-La Mancha in Ciudad Real, Spain (38°59'00"N, 03°56'00"W), as shown in Figure 3, over the period between 13:07:15 of 31 July 2018 to 12:43:11 of 4 October 2018. This area of central Spain is characterized by a Mediterranean climate and a high solar resource. During this period the application described in Section III executed uninterruptedly, since the solar production during these months was enough to feed the circuitry and to recharge the battery, guaranteeing thus the energy neutrality of the system. As a

TABLE I
Variables used in the *Dataset* application

Variable	Units	Description
Date	dd/mm/yy	Date provided by the RTC
Time	hh:mm:ss	Time provided by the RTC
Load	mA	IoT node consumption
Battery	mA	Current supplied (> 0) or absorbed (< 0)
Panel	mA	Short-circuit current of the PV panel
Wind	m/s	Wind speed provided by the anemometer
Temp	°C	Ambient temperature
Hum	%	Relative ambient humidity
Voltage	V	Battery terminal voltage

result of this execution, the application generated a dataset that comprises 93438 data records, where the records are regularly spaced every minute. The dataset is available online at https://github.com/arco-group/energy-harvesting-dataset.git. Next, we analyze the dataset and show how it can be used to develop simple prediction models of the expected solar panel production, like in [6].

Fig. 3. Testbed installation at University of Castilla-La Mancha in Ciudad Real, Spain (38°59'00"N, 03°56'00"W).

A. Dataset Analysis

This first real experiment was performed for testing purposes and circuit validation in terms of reliability. All the variables in the dataset were measured with a periodicity of 1 minute. All the variables were measured directly by the DL, except the temperature and humidity, which were measured by the IoT node and then sent to the DL by serial connection. Note that, however, from the perspective of the programmer, the library described in previous section hid such a division of tasks. The IoT node was always in active mode and Lora Radio chip was inactive, yielding to a constant consumption of $13 - 14$ mA. We show in Figures 4 and 5 the plots of all the variables in the dataset from the entire period of deployment.

In the first graph of Figure 4, with a blue line, the curve represents the short-circuit current of the panel which is proportional to the energy production. Thus, it increases with the solar irradiance and then, as observed, it draws a parabola very similar to the daily energy production in favorable solar conditions: it increases up to approximately the noon and decreases up to the sunset. A better view of the solar production is presented in Figure 6, where we represent the data from days 1 to 13 of September, aggregated (averaged) over intervals of 30 minutes. We see from this figure abrupt oscillations on the third day, and in other days due to the presence of clouds. The temperature follows a similar trend, while the wind and humidity levels seems less correlated to the solar production. It is interesting to observe very high temperatures that range between 20 and 40 degrees, while humidity oscillated inversely to the temperature.

Figure 5 represents in the last graph the IoT node current (note that the DL current is irrelevant since it is connected to the battery), the current drawn by the battery (blue line), and battery capacity as percentage of its total capacity (orange line). The latter variable is obtained by processing the values in the dataset of the battery voltage. Note that since the load of the node is low compared to the battery capacity and harvested energy, the level of the battery is always over 90% (voltages always ranging between 4.15 and 4.25.) The battery current (flow), in blue, takes a negative value when the current is supplied by the battery (discharge period) and a positive value if the battery is absorbing current from the panel (charge period). As observed, it follows the same daily pattern than the production. The current required to keep the IoT node working is approximately 13 mA, and it is consistent with the specification of the manufacturer. This consumption represents the sum of the individual currents of all active hardware components that are being used and it is constant along the period since all of them kept the same energy state.

B. Simple Prediction Model of the Energy Production

In Figure 6 we show a closer view on the solar panel production, from 1 to 12 September. The data, collected every 1 minute have been averaged over periods of 30 minutes. We note from the plot, that the solar energy production is affected by the weather conditions, it decreases in cloudy days $(3, 8)$, while it is more regular in sunny days. The different production curves in days $(8, 11)$ are highlighted in Figure 7.

In [3] we develop a task scheduling algorithm to dynamically optimize the load of the IoT node as a function of the energy available in the battery and the expected production of the solar panel. Like in [6], to properly optimize the node behavior, it is mandatory to use an energy prediction model that gives an estimation of the future expected energy production. Here, we test the EWMA method [6], suggested in Kansal, and estimate its error. The EWMA method is a weighted exponential moving average of the dataset, calculated over each slot of 30 minutes. The prediction follows this simple equation:

$$p_t = \alpha p_{t-1} + (1 - \alpha) S_{t-1}$$

where p_t is the prediction made at the end of period t, in our case at the end of each day, p_{t-1} is the old prediction (the memory), and S_{t-1} is the energy produced by the solar panel measured at the end of the day; finally, $\alpha \in [0, 1]$ is a smoothing factor that model how importance we give to the history, with respect to the observed production in the previous day. Note that, the above equation, must be used independently for each slot of 30 minutes in order to develop a model for each period of the day.

This prediction model can be evaluated by measuring its *mean square error*, the optimal value of α to be used depends on the single deployment and has been obtained minimizing the error. Figure 8 shows the curve of the error as a function of different choice of α, we see that the optimal value is close to 0.863. Using this value for α we show in Figure 9 the production (green line) vs. prediction (red line) obtained with this method. We see that the prediction is quite accurate when the weather is stable, but it overestimates or underestimates the solar production sometimes.

The error is better shown in the lower plot in the same Figure, that represents the curves of the relative error as a percentage of the real production. Note that we represent in the same plot two curves, the blue line (positive curve) is a forecast that overestimates the real production, while the orange line is the underestimation. We note, as expected, that the error is concentrated in cloudy days, or in periods with a higher variability of weather, or after sunset. Note also that the forecast error could be higher than 50% of the real production, especially in overestimation. So, even if in the curves above, it seems that the forecast and the solar production are quite close, Table II shows a selection of the interval in the dataset with a relative error greater than 50%, the columns are respectively: date and hour, the effective panel production, the forecasted production, the absolute difference between the first two values, and the relative error in percentage when above 50%. We note from this table that the error is naturally higher in days with variable weather, but also in the intervals close to dawn (7:30am) or sunset (20:00pm). This preliminary observations highlight that this model must be improved to support smart IoT applications that respond in a more accurate way to the real level of energy harvested by the solar panel.

V. CONCLUSION

Despite the large number of prototypes and even installation of IoT devices with solar energy harvesting capabilities, many research questions such as the estimation of the forthcoming energy production, the tradeoff between performance and lifetime, the need for energy neutrality are still open. Many works address these questions by proposing suitable algorithms often evaluated on the base of models or on ad hoc datasets

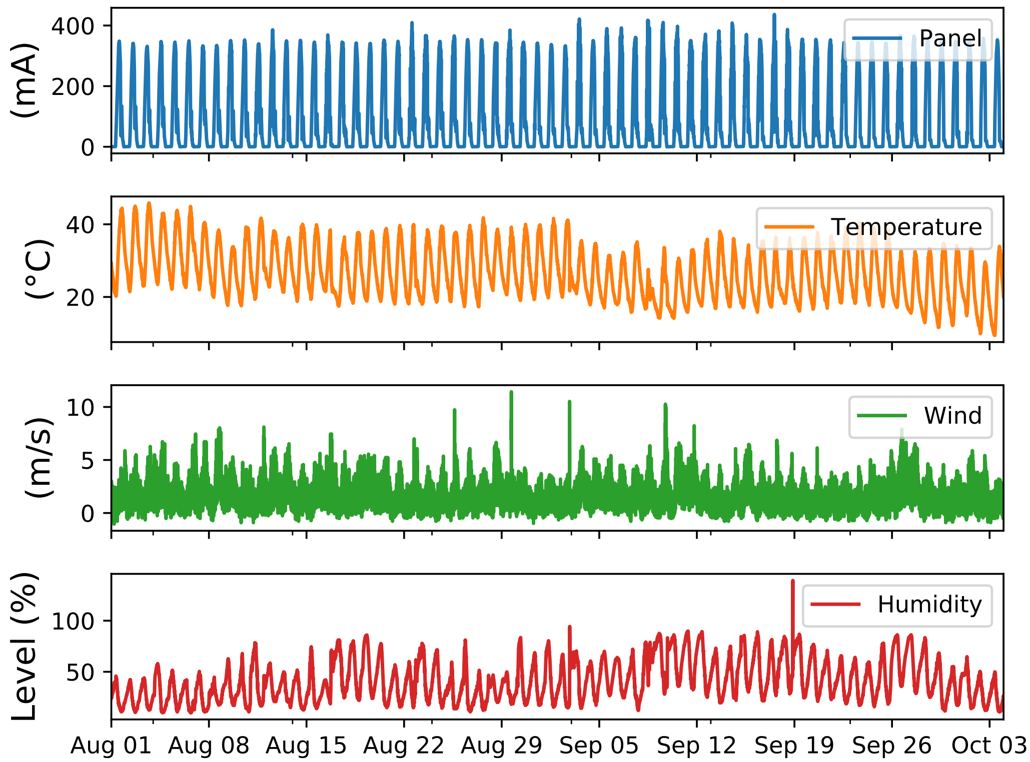

Fig. 4. Energy production from the panel, temperature, wind speed, and humidity collected during the reference period.

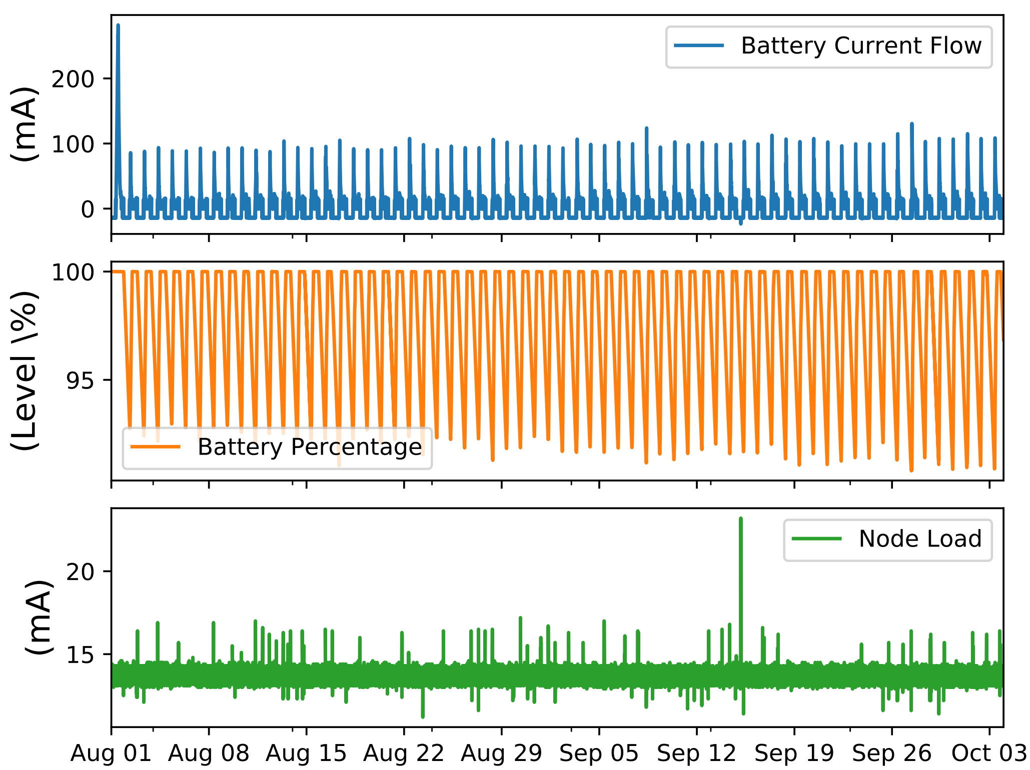

Fig. 5. Current drawn by the battery, battery capacity (%) and IoT node current over the reference period.

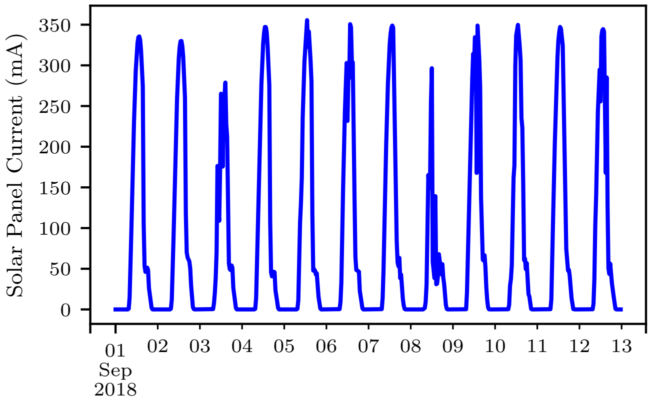

Fig. 6. Panel production (averaged over intervals of 30min) from 1st to 12nd Sep.

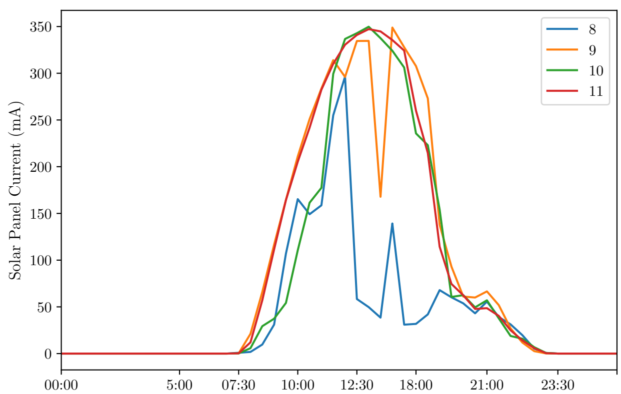

Fig. 7. Production of 4 days in the same plot.

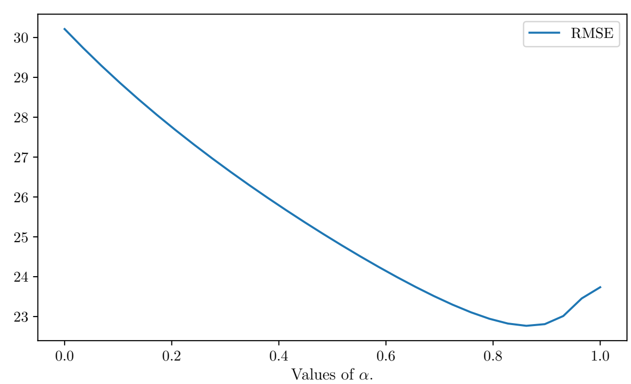

Fig. 8. Exponential Weighted Average, value of error for different α.

collected just for their purposes. We instead propose here an open dataset that can be used to experiment, test and compare different solutions. The development of such a dataset has required the design of a suitable device, able at the same time to run a specific IoT task and to monitor the parameters of energy production and consumption without interfering with such a task. This lead us to the implementation of the testbed described

TABLE II
REAL PV PANEL PRODUCTION VS FORECASTED. ABSOLUTE AND RELATIVE ERROR. INTERVALS WITH ERROR GREATER THAN $> 50\%$.

Day/Hour	Solar Panel	Forecast (EWMA)	|P-F|	|P-F|/P $> 50\%$ Rel. Error (%)
01 20:30	2.08	3.6	1.5	73.5%
02 08:00	12.26	19.2	7.0	57.1%
03 10:30	122.52	233.5	111.0	90.6%
03 11:30	172.96	297.8	124.8	72.2%
03 13:00	175.60	339.8	164.2	93.5%
03 20:30	1.63	2.9	1.3	83.5%
04 20:30	1.78	2.8	1.0	57.2%
05 20:30	1.40	2.6	1.2	89.6%
06 08:00	9.04	15.6	6.6	73.3%
06 08:30	33.06	51.2	18.2	55.0%
06 20:30	1.37	2.4	1.1	81.1%
07 20:00	4.96	8.9	4.0	81.0%
08 10:30	149.12	228.6	79.4	53.3%
08 11:00	158.66	259.8	101.1	63.7%
08 16:00	67.91	102.2	34.3	50.5%
08 20:00	5.31	8.4	3.1	58.6%
09 13:30	167.81	292.8	125.0	74.5%
10 08:30	29.31	46.4	17.1	58.4%
10 10:00	110.53	189.7	79.2	71.6%
12 08:00	7.00	12.7	5.7	82.0%
12 15:00	167.75	254.9	87.2	52.0%
12 20:30	0.70	1.3	0.6	94.5%

with a double, independent subsystems, one IoT node and one for energy-related data logging. Our testbed, in turn, was used to collect a dataset in a data collection campaign, which is now available at this URL: https://github.com/arco-group/energy-harvesting-dataset.git. The paper shows some analysis conducted over the dataset to show its features and an example of how a known algorithm for energy production estimation would perform. As a future work we plan to extend the dataset with further data collection campaign to be conducted over different periods of the year, and to associate the dataset with data obtained from public weather forecasts, which can also be used to experiments algorithms as in [5] that use such data to improve the solar energy prediction. Furthermore, we also plan to continue experimenting and comparing algorithms for energy harvesting, for example those related to the optimization of the quality of service with respect to the energy neutrality constraints, as in [3], [6].

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] D. Evans, "The Internet of Things: How the Next Evolution of the Internet is Changing Everything," http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf, 2011.

[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, cited By 10.

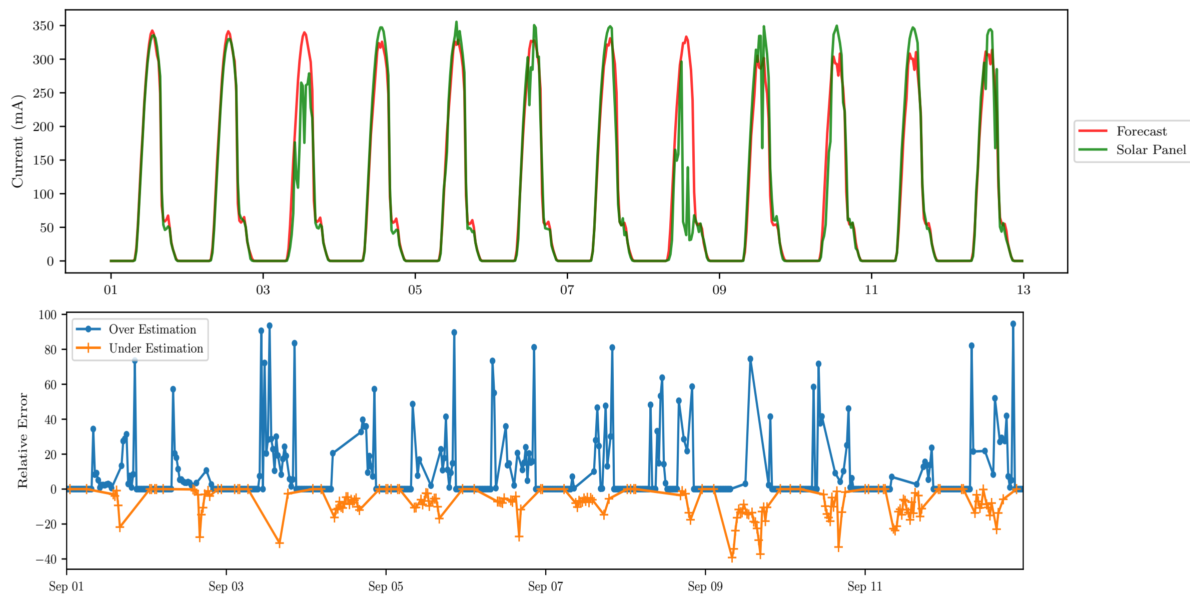

Fig. 9. Solar Energy Forecast using EWMA with optimal parameter, compared with the effective Solar Panel Production and relative error in percentage (negative if underestimate, positive if overestimate).

[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 *2018 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*, ser. IoT ’17. New York, NY, USA: ACM, 2017, pp. 2:1–2:8.

[6] 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.

[7] P. Baronti, P. Barsocchi, S. Chessa, F. Mavilia, and F. Palumbo, “Indoor Bluetooth Low Energy Dataset for Localization, Tracking, Occupancy, and Social Interaction,” *Sensors*, vol. 18, no. 12, 2018.

[8] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, “The Mobile Data Challenge: Big Data for Mobile Computing Research,” L’IDIAP Laboratory, Tech. Rep., 2012. [Online]. Available: http://infoscience.epfl.ch/record/192489

[9] Department of Energy’s SunShot Initiative, the National Renewable Energy Laboratory (NREL), “Solar Datasets,” https://energydemo.github.io/SolarDatasets, 2018.

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

[11] 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 *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, Nov 2017, pp. 109–116.

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

[13] 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 *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2015, pp. 1–6.

[14] 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 *IREC2015 The Sixth International Renewable Energy Congress*, March 2015, pp. 1–6.

[15] 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.

[16] Microchip, “SAM D21 Family, Low-Power, 32-bit Cortex-M0+ MCU with Advanced Analog and PWM. ATSAMD21G18,” https://www.microchip.com/wwwproducts/en/ATsamd21g18, 2018.

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

[18] Sensirion, “SHT1x sensor. Humidity and Temperature Sensor IC,” https://cdn-shop.adafruit.com/datasheets/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf, 2011.

[19] Adafruit, “Anemometer Wind Speed Sensor w/Analog Voltage Output,” https://www.adafruit.com/product/1733, 2019.

[20] T. Instrument, “INA219 Zero-Drift, Bidirectional Current/Power Monitor With I 2C Interface,” http://www.ti.com/lit/ds/symlink/ina219.pdf, 2015.