Improving the Accuracy of Outdoor Temperature Prediction by IoT Devices

Nevena Golubovic, Rich Wolski, Chandra Krintz
Computer Science Dept.
Univ. of California, Santa Barbara

Markus Mock
Dept. of Mathematics and Statistics
Univ. of Applied Sciences, Landshut Germany

***Abstract*—In this paper, we investigate new methods for improving the accuracy of outdoor temperature prediction using small, low-cost, single board computers (SBCs) used in Internet-of-Things (IoT) deployments. Predicting temperature without dedicated temperature sensors frees up space on these systems for other sensors and reduces the cost of microclimate sensing (e.g. as used in IoT-based, agricultural applications). Our approach uses multiple linear regression and combines measurements of on-board processor temperature from multiple SBCs with remote weather stations. In addition, it accounts for SBC computational load through the use of smoothing techniques that filter out noise in the measurement time series. We empirically evaluate our approach using multiple IoT deployment scenarios, compare it against prior work, and find that it reduces prediction error significantly for these scenarios.**

***Keywords*-IoT; sensing; multiple linear regression; prediction**

I. Introduction

The Internet of Things (IoT) is quickly expanding to include every "thing" from simple Internet-connected objects, to collections of intelligent devices capable of everything from the acquisition, processing, and analysis of data, to data-driven actuation, automation, and control. Since these devices are located "in the wild", they are typically small, resource-constrained and battery powered. At the same time, low latency requirements of many applications mean that processing and the analysis must be performed near where data is collected. This tension requires new techniques that equip IoT devices with more capabilities.

One way to enable IoT devices to do more is to use integrated sensors to `estimate` the measurements of other sensors, a technique that we call *sensor synthesis*. Since the number of sensors per device is generally bounded by design constraints (e.g. space or power limitations), sensor synthesis makes it possible to free up resources in IoT devices for other sensors, particularly those that are less amenable to synthesis, and to reduce the monetary cost of sensing.

Since sensor synthesis is based on computed estimates rather than actual measurement, it also introduces the possibility of additional error beyond measurement error. In this paper, we show how the overall error (measurement error convolved with error propagation due to composition) can be reduced compared to prior work. The authors of [1] present a technique for estimating outdoor temperature from CPU temperature for IoT applications in agriculture. They use a simple regression technique to "synthesize" (in our parlance) and lower the cost of microclimate temperature monitoring on farms. They estimate outdoor the temperature from the processor temperature sensor built into board computers (SBCs), e.g Raspberry Pi devices [2]. This past work relies on data cleaning (both pre and post regression) that employs computationally expensive methodologies that must be performed on full-featured resources (e.g. in a cloud).

In this paper, we examine how a larger ensemble of measurements improves the accuracy of "synthetic" temperature measurement (beyond the $1-2^{\circ}$ Fahrenheit errors reported in [1]) while, at the same time, not requiring the use of powerful computational resources. Specifically, we propose new methods for estimating outdoor temperature from SBC processor temperature.

Reducing the prediction error is not only academically interesting, rather, precision has a direct impact on the cost and efficiency of what has become known as precision agriculture or precision farming. In precision agriculture, farmers use technology to increase the efficiency of farming techniques increasing crop yields and reducing costs. Having more precise temperate data reduces the cost of frost prevention (by avoiding the unnecessary use of frost mitigation systems (e.g. fans)) and prevents excessive resource use without negatively impacting crop production. Consequently, we believe that our approach can contribute to improved farming outcomes, enable water and energy savings, and help reduce carbon emissions, by providing high-quality data to data-driven, IoT-based agricultural applications.

Key to our approach is the combined use of processor temperatures from multiple devices with outdoor temperature from high-quality, remote weather stations used to train a multiple linear regression model. We use this model to predict the future outdoor temperature at a particular device location that is not part of the model. We also investigate the efficacy of computationally simple smoothing techniques (based on sliding window reductions) to reduce noise.

Moreover, we investigate how well our approach performs when the processors on the devices experience load. Load may affect processor temperature and thus negatively impact the accuracy of our outdoor temperature estimates. To address this, we develop techniques that successfully deal with the perturbations caused by load variability, which is an

978-1-7281-2714-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ICIOT.2019.00030

important requirement to make our sensor synthesis practical in the field (and which went uninvestigated in prior work).

Finally, to evaluate the practical effectiveness of our approach, we deploy multiple Raspberry Pi Zero devices in an agricultural setting where citrus trees are grown. To compare the values of our synthesized sensors with measured temperature values, we equip the devices with temperature sensors, which we use to establish ground truth. We evaluate different combinations of explanatory variables[1] with and without smoothing, and with and without a computational load on the processor, as part of our multiple linear regression models. Our results show that our approach reduces mean absolute prediction error (MAE) over past work and is robust to processor load. We next detail our approach and its empirical evaluation.

II. Predicting Outdoor Temperature from Processor Temperature

The goal of our work is to reduce the prediction error associated with sensor synthesis of outdoor temperature from the processor temperature by single board computers (SBCs), for IoT-based agricultural applications. Because temperature is used to guide water use, greenhouse control, and frost mitigation strategies, it is critical that we be able to estimate temperature with very high accuracy. If we are able to do so, we can reduce the number of sensors required and lower the cost of sensing in agricultural settings, while using temperature estimates to automate, actuate, and control farm operations.

Past work [1] on this topic uses a combination of single spectrum analysis [3] (to filter noise) and simple linear regression (Ch.3 of [4]) to model the relationship between the response variable (outdoor temperature) and the explanatory variable or predictor (the processor temperature). The authors use the model to predict outdoor temperature using different IoT devices and settings, and report a MAE of $1-2$°F for the best case and $14°F$ for the worst case.

Our approach applies multiple linear regression to reduce this error. In particular, we consider processor temperature measurements from multiple SBCs (deployed in other on-farm microclimates), and outdoor temperature from a remote weather station, as possible predictors. We use the term processor and CPU interchangeably throughout.

A. Deployment and Datasets

We deploy four Raspberry Pi (RPi) Zero [2] devices (named Pi1, Pi2, Pi3, and Pi4) equipped with temperature sensors, at different locations (microclimates) in an agricultural setting (citrus trees). We place a pair of RPis within 3 feet of each other, in two different trees, spaced 10 feet apart. Pi1 and Pi2 monitor tree #1 and Pi3 and Pi4 monitor tree #2. Each device is housed in an inexpensive plastic enclosure and has an on-board processor temperature sensor that is part of its hardware/software interface.

The devices read their processor temperature sensor value every 5 minutes and can process, store, or wirelessly transmit their measurements. We label the measurements CPU-1, CPU-2, CPU-3, and CPU-4, for the CPUs of Pi1 through Pi4, respectively. The RPi devices then transmit the measurements to an on-farm computer for aggregation and analysis.

Each RPi is additionally equipped with an AM2302 DHT22 digital temperature and humidity sensor [5], which we use to measure ground truth. The devices read and transmit these values every 5 minutes (labeled DHT-1, DHT-2, DHT-3, and DHT-4, with temperature value DHT-{i} representing the temperature measured by the DHT22 sensor attached to the Pi{i}) along with their CPU temperature readings to a remote analysis system. We only use this DHT22 data as ground truth (to compute prediction error), i.e., it is not used as part of modeling or prediction.

Finally, we also consider the use of freely available, high-end weather station data from the Internet weather service WeatherUnderground [6]. The closest weather station is 2640 feet (800m) away from our field deployment. We collect the temperature reported by the WeatherUnderground station closest to the deployment site every five minutes (labeled WU-T). We align the measurements (CPU, DHT22, and WU) using the nearest timestamp. If there is data dropout, i.e, if one of the three temperature values is missing, we skip all measurements for that five-minute interval.

B. Linear Regression Models

We model the outdoor temperature that surrounds a single RPi, using one or more predictors. Predictors can include the CPU temperature of RPi itself, the CPU temperature of neighboring RPis, and the outdoor temperature reported by a high-quality, remote weather station. We estimate model parameters $\theta \in \mathbf{R}^n$ by minimizing the residual sum of squares:

$$RSS(\theta) = (y - X\theta)^T (y - X\theta)$$

where $y_i \in \mathbf{R}, i \in \{1, \ldots, N\}$ represents the ground truth outdoor temperature and $X \in \mathbf{R}^{N \times n}$ represents the entire training set, where each row $x_i \in \mathbf{R}^n$ represents the values that predictors take, and n is the number of predictors.

In Section III, we analyze models with testing windows of size one hour to two weeks, which correspond to 12 and 4032 data points respectively. To measure error, we compute the mean absolute error (MAE) (versus R-squared) because of its direct utility in our IoT agriculture applications. In

[1]In linear regression, an explanatory variable is an independent variable that is used to predict a value. In our context, the independent variables are the CPU temperatures and weather station temperature (gathered from a weather station that is in the area of the SBCs but not necessarily co-located), which we use in the model to predict the synthesized sensor. Explanatory variables are also called predictors in the literature. Since we use multiple regression, we use more than one predictor in our synthesis.

particular, we are interested in using the models to make predictions and not in their explanatory power. We compute MAE as the average absolute distance between estimated temperatures and their corresponding ground truth values.

Finally, we evaluate the efficacy of smoothing the training data prior to performing regression. We investigate rolling mean, minimum, and median smoothing methods. In our experiments, rolling mean produces the smallest error for the datasets we investigate. We thus report results using only this smoothing technique, for brevity. To implement rolling mean, we use a window of size w and replace each element with the mean value of the previous w elements including the current element. More formally, we replace X in the *RSS* equation with S where

$$s_{ij} = \begin{cases} \sum_{l=j-w}^{j} \frac{x_{il}}{w} & j >= w \\ \sum_{l=o}^{j} \frac{x_{il}}{j} & j < w \end{cases}$$

For all the experiments presented in Section III we use a window size $w = 6$, which corresponds to 30 minutes.

III. Empirical Evaluation

In our experiments, we use four RPi-based, single board computers (SBCs) deployed outdoors as described in Section II-A. We denote the processor temperature measurements from each as CPU-1, CPU-2, CPU-3, CPU-4. We refer to the outdoor temperature measurements from a nearby WeatherUnderground stations as WU-T.

The goal of this evaluation is to illustrate the degree to which it is possible to make an accurate prediction of outdoor temperature based on a combination of CPU temperature measurements and temperature measurements from the WeatherUnderground station. In this study, “ground truth” – the true outdoor temperature – comes from DHT22 sensors connected externally to each RPi. We do not use the measurements from the DHT22 sensors in any prediction. However, we use them to determine the mean absolute error (MAE) between a prediction based on CPU and WU-T values and ground truth as established by the DHT value and thereby determine our prediction accuracy. Our RPis are equipped with a 1GHz ARMv7 processor, 512MB memory, 32GB of SSD storage, and Wifi communication. All the temperature readings in the experiments are reported in degrees Fahrenheit.

A. Experimental Results

As a baseline, the upper triangle of the matrix in Table I shows the average difference in temperature, pairwise, between all pairs of temperature measurement traces we include in our study. Thus, for example, the average difference in temperature between CPU-1 and DHT-1 (the DHT connected directly to the RPi hosting CPU-1) is given in row 2, column 6 of the table as $29.23°F$ marked in bold in the table (assuming the header and row labels are row 1 and column 1 respectively). This data spans 72 hours beginning August 27th, 2018 and includes 864 temperature measurements gathered at 5-minute intervals.

Overall, this baseline illustration shows that

- CPU and external DHT measurements differ by approximately $30°F$;
- average differences among DHT22 sensors (ground truth) vary from $1°F$ to $2.6°F$ (despite their proximity); and
- the differences in local temperature from the one reported by the nearby weather station vary from $3.61°F$ to $4.31°F$.

Since the matrix of comparisons is symmetric, we only show values in the upper triangle.

For frost prevention, the application is attempting to determine when a small difference in temperature between warm air aloft and colder air near the ground will result in frost avoidance if the air is mixed. Specifically, large wind machines move the warm air downwards to raise the temperature enough near the ground to prevent frost from forming. The temperature differences are on the order of a few degrees Fahrenheit putting a premium on accurate measurement. The baseline in Table I shows the errors that result when each temperature sensor is used directly to predict another. That is, it is the “worst case” prediction in the sense that it includes no prediction mechanism – only the raw data.

In order to provide a more accurate prediction of local temperature based solely on the devices’ CPU temperatures and the nearby weather station, we combine multiple linear regression with smoothing. We hypothesize that the relationship between outdoor temperature and nearby CPU temperatures measured at the same time is linear. Further, particularly if one or more of the CPUs are loaded, we use one-dimensional smoothing of the CPU temperature series to improve the “signal” from the CPU temperature sensor.

For the regressions, the explanatory variables are a subset of CPU and a weather station temperature (CPU-1, CPU-2, CPU-3, CPU-4, WU-T), as indicated at the top of each results tables. Also, when smoothing is performed, we indicate this in the table header.

In each case, we separate the experimental period under study into a “training” period followed immediately by a “testing” period. The regression coefficients are computed only from data in the training period. We then use the coefficients for the entire duration of the testing period.

Table II shows the MAE between the temperature that our method predicts and the outdoor temperature for two “ground truth” sensors – DHT-1 and DHT-3 – using two separate subsets of explanatory variables for each. On the lefthand side of the table, we show the MAE (both with and without smoothing) when predicting DHT-1 using CPU-1 alone (a univariate regression) and also when using all CPUs and WU-T (a multiple linear regression, denoted as *All*). On the righthand side of the table, we show the same results for

Device	CPU-1	CPU-2	CPU-3	CPU-4	DHT-1	DHT-2	DHT-3	DHT-4	WU-T
CPU-1	0.00	4.78	7.15	3.20	**29.23**	30.12	30.78	29.84	32.26
CPU-2	-	0.00	4.07	3.40	24.51	25.37	26.06	25.12	27.55
CPU-3	-	-	0.00	4.86	23.09	23.99	24.68	23.71	26.16
CPU-4	-	-	-	0.00	27.87	28.76	29.45	28.50	30.95
DHT-1	-	-	-	-	0.00	2.07	2.60	2.15	3.61
DHT-2	-	-	-	-	-	0.00	1.32	1.23	4.31
DHT-3	-	-	-	-	-	-	0.00	1.00	3.75
DHT-4	-	-	-	-	-	-	-	0.00	4.01
WU-T	-	-	-	-	-	-	-	-	0.00

Table I: Average absolute difference in temperature measurements among CPU and DHT22 sensors from four RPi's (Pi1, Pi2, Pi3, and Pi4) measured during the 72 hours period on August 25th, 26th, and 27th, 2018.

	DHT-1				**DHT-3**			
	Original		**Smoothed**		**Original**		**Smoothed**	
TE	**CPU-1**	***All***	**CPU-1**	***All***	**CPU-3**	***All***	**CPU-3**	***All***
1	0.55	*0.39*	**0.38**	*0.40*	**0.32**	***0.32***	**0.37**	*0.21*
3	**0.45**	*0.34*	0.38	*0.33*	0.50	*0.32*	0.47	***0.20***
6	0.46	***0.32***	0.41	***0.28***	0.78	*0.41*	0.83	*0.28*
12	0.48	*0.46*	0.44	*0.43*	0.70	*0.48*	0.74	*0.37*
24	0.55	*0.43*	0.55	*0.44*	0.95	*0.57*	0.99	*0.46*
48	0.62	*0.47*	0.62	*0.46*	1.04	*0.63*	1.04	*0.51*
72	0.70	*0.49*	0.70	*0.49*	1.28	*0.69*	1.21	*0.55*
96	0.75	*0.52*	0.78	*0.53*	1.36	*0.72*	1.31	*0.62*
168	**0.85**	*0.72*	**0.92**	***0.69***	**1.68**	*0.83*	1.64	*0.80*
336	0.79	***0.81***	0.77	*0.66*	1.54	***1.26***	**1.56**	***1.24***

Table II: MAE for different sets of smoothed and non-smoothed explanatory variables and lengths of Test Window (TE) when predicting DHT-1 and DHT-3 temperature based on a 72h train window and a test start day on Aug. 25th.

DHT-3 using CPU-3 in the univariate case. The experiment (testing period) start date is Aug. 25th. For all experiments, we use a training window of 72 hours (864 readings). As mentioned in section II-B, we use MAE as our measure of accuracy since it captures the "distance" between the predicted temperature and the DHT-measured temperature. It is this distance that concerns farmers who are deciding on whether to trust their crops to the methodology.

Note that columns CPU-1 and CPU-3 under the Original column show values corresponding to results based on the method proposed in prior work [1]. Note also that we highlight the minimum and maximum MAE in each column using boldface type.

When predicting DHT-1, we see that errors from univariate regression using only the CPU temperature from Pi1 (CPU-1) are in the range from $0.45°F$ to $0.85°F$. MAE for multiple linear regression with CPU temperatures from all four devices and a nearby weather station data range from $0.32°F$ to $0.81°F$. When predicting DHT-3 from its Pi3's CPU sensor deployed in a similar manner we see MAE values between $0.32°F$ to $1.68°F$ (listed in the left DHT-3 sub-table as CPU-3 column). MAE decreases to a range from $0.32°F$ to $1.26°F$ when we introduce multiple linear regression (*All* column). Note that even though the setup is similar (the same set of devices and outdoor conditions), the readings are influenced by other environmental factors (tree coverage, sun exposure, etc.).

We find that multiple linear regression which includes CPU and nearby weather station temperatures as its predictors reduces prediction error. For DHT-1, the minimum error decreases from $0.45°F$ (minimum error in *CPU-1* column) to $0.32°F$ (minimum error in *All* column) while the maximum error decreases from $0.85°F$ (maximum error in *CPU-1* column) to $0.81°F$ (maximum error in *All* column). For DHT-3 the minimum error is $0.32°F$ for both columns (*CPU-3* and *All*) while the maximum error decreases from $1.68°F$ for *CPU-3* to $1.26°F$ for *All*. If we compare errors per test window length, we note that for DHT-1 all errors but for the 2 weeks test window were reduced (where $0.79 < 0.81$) and for DHT-3 all errors but for 1h test window were reduced (1h row had the same error of $0.32°F$ in both columns).

These results indicate that it is possible to make predictions with an average absolute error of under $1°F$ that require infrequent model refitting (e.g. once per several days) using a combination of CPU and weather station data. Indeed, the accuracy of DHT22 sensors is approximately $0.5°F$. Thus this methodology is approaching the limit of accuracy that is possible using DHT22 sensors as ground truth. Under $1°F$ is acceptable for frost prevention where current manual methods use measurements in the $3°F$ range.

For the smoothing results in Table II, each value (except the first 6) in the training period is replaced by the average of the 6 preceding it in the period (i.e. we use a *sliding window average* to smooth the data in the training period). When comparing the *All* column from *Original* and *Smoothed* columns, we see that the smoothing decreases the mean absolute error (MAE) from the range of $0.32°F$ to $0.81°F$ (original) to the range of $0.28°F$ to $0.69°F$ (smoothed). Similarly, for DHT-3 prediction, the MAE goes from the range $0.32°F$ to $1.26°F$ (original) to the range $0.20°F$ to $1.24°F$ (smoothed).

B. Computational Load: the Effect of Smoothing and Multiple Linear Regression

CPU temperatures are correlated with the CPU load [7], [8] and while the CPUs are idle for much of the time in our setting temporary computational load at the time of temperature recording might influence the prediction error (e.g. if the CPU were performing encryption as part of transmitting the data over the network). We next analyze

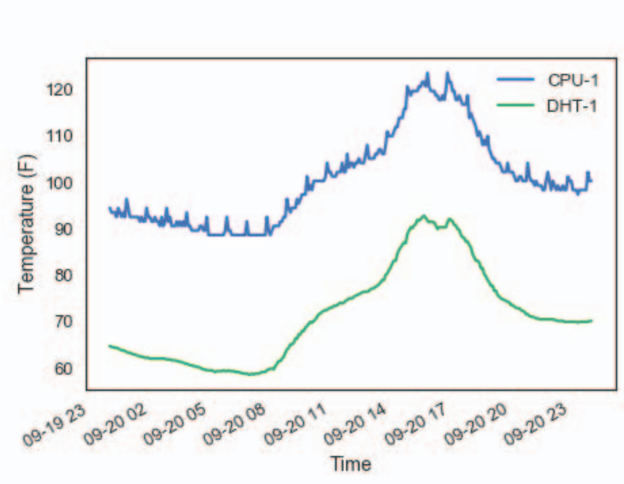


Figure 1: CPU-1 temperature under load and DHT-1 temperature in $^{\circ}F$.

	DHT-1				DHT-3			
	Original		Smoothed		Original		Smoothed	
TE	**CPU-1**	***All***	**CPU-1**	***All***	**CPU-3**	***All***	**CPU-3**	***All***
1	0.85	*0.54*	**0.19**	*0.46*	0.75	***0.39***	**0.23**	***0.32***
3	**0.71**	***0.49***	0.42	***0.36***	**0.78**	*0.53*	0.30	*0.34*
6	0.73	*0.55*	0.47	*0.37*	0.70	*0.44*	0.27	*0.34*
12	0.73	*0.53*	0.60	*0.42*	0.74	*0.53*	0.42	*0.50*
24	0.85	*0.57*	**0.76**	***0.54***	0.70	*0.52*	0.57	*0.49*
48	0.84	***0.58***	0.69	*0.51*	0.67	*0.50*	0.62	*0.48*
72	0.82	*0.55*	0.66	*0.50*	0.66	*0.50*	0.61	*0.48*
96	0.80	*0.54*	0.66	*0.53*	0.66	*0.52*	0.61	*0.49*
168	0.80	*0.53*	0.62	*0.51*	0.66	***0.53***	**0.62**	*0.50*
336	**0.85**	*0.53*	0.60	*0.51*	**0.65**	*0.51*	0.61	***0.50***

Table III: Prediction error when CPU-1 and CPU-3 experience periodic load. The data shows MAE for different sets of smoothed and non-smoothed explanatory variables and lengths of Test Window (TE) when predicting outdoor temperature for DHT-1 and DHT-3 based on a train window of 72h and with a test start day on Sep 20th.

the effect of the CPU load on the temperature prediction error.

Out of the four devices that we consider, we keep Pi2 and Pi4 unloaded and add hourly jobs to Pi1 and Pi3, which increase the CPU load by encrypting and copying a 1GB file on Pi1 and a 512MB file on Pi3. Figure 1 illustrates CPU temperature measurements from Pi1 with hourly spikes due to the load. The load testing for Pi1 and Pi3 started mid September and we use September 20th as a test start date. Note that Pi2 and Pi4 have no artificial load and are kept idle for comparison. We observe that, compared to the August test, all four Pi's show smaller errors on average, however, we omit these averages for brevity.

Table III shows the MAE for predicting DHT-1 and DHT-3 based on different sets of explanatory variables (listed on the top of the table) for different duration of the test window (TE), while both Pi1 and Pi3 are loaded. For predicting DHT-1 based on CPU-1, we see MAE in the range of $0.71^{\circ}F$ to $0.85^{\circ}F$ and for the DHT-3 of $0.65^{\circ}F$ to $0.78^{\circ}F$. The effect of the CPU load is more pronounced in univariate prediction. Moreover, this effect is mitigated when we include nearby devices' CPU temperature measurements. Including nearby devices in the DHT-1 prediction (*All*) results in MAE in the range of $0.49^{\circ}F$ to $0.58^{\circ}F$ for DHT-1 and in the range of $0.39^{\circ}F$ to $0.53^{\circ}F$ for DHT-3.

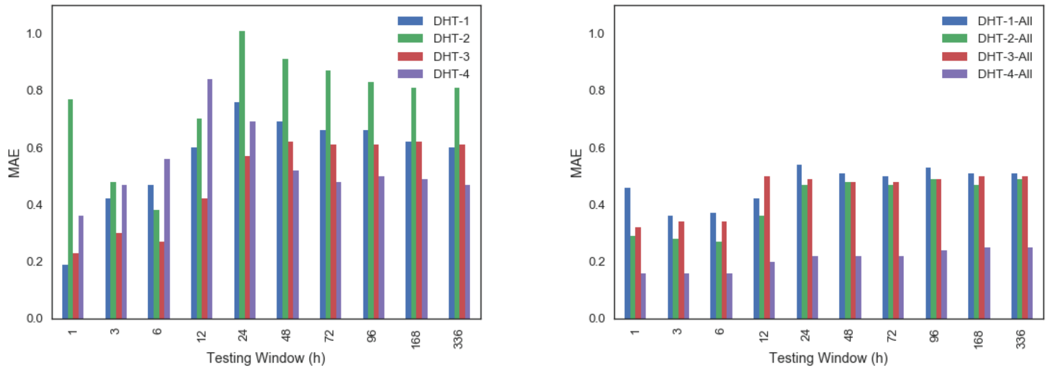
Similar to the results for the unloaded experiments, when the CPUs are loaded we also see improvement in prediction error when we apply smoothing, as shown in Table III. The two columns show MAE for DHT-1 and DHT-3 temperature prediction with the same smoothing technique explained earlier (rolling mean with a window size of 30 minutes or 6 readings). Note that this type of smoothing is computationally simple enough to be performed on each device (rather than as a remote computation requiring a more powerful computational resource (used in past work)).

We see that for any length of test window the error when all the predictors are used (*All* column) is smaller than when any single predictor counterpart is used: CPU-1 for DHT-1, and CPU-3 for DHT-3. With smoothing, the prediction MAE decreases from the range of $0.71^{\circ}F$ to $0.85^{\circ}F$ to the range of $0.36^{\circ}F$ to $0.54^{\circ}F$ for DHT-1, and from the range $0.65^{\circ}F$ to $0.78^{\circ}F$ to a range of $0.32^{\circ}F$ to $0.50^{\circ}F$ for DHT-3. While not strictly lower or higher, these results are similar (in terms of accuracy) to the results for the unloaded case. We conclude that, using a combination of multivariate regression and smoothing, it is possible to obtain high degrees of prediction accuracy (relative to measurement error) regardless of whether the CPU is loaded or not.

To account for the possibility that the specific timeframe may have influenced the results (i.e. outdoor conditions might have been more dynamic in late August than late September), we show comparative results for the September timeframe for loaded and unloaded experiments in Figure 2. The data shown in this figure is taken during the same period as the results shown in Table III. That is, we use the 72-hour period ending on September 20th, 2018 as a training period and the remaining time as a test period (ranging from 1h to 2 weeks). The bars in the figure corresponding to CPU-1 and CPU-3 show the same data as in Table III from the *Smoothed All* columns. For comparison, we show data for two other CPUs – CPU-2 and CPU-4 – taken at the same time, again using smoothing and all explanatory variables in each regression (i.e. *Smoothed All*).

Figure 2a shows the comparison when only the CPU directly attached to the DHT is used as a single explanatory variable (i.e. the "nearest" CPU). In Figure 2b, we show the the results when all explanatory variables are used to predict each DHT.

In Figure 2b, the maximum MAE observed in any experiment does not exceed $0.54^{\circ}F$ across all CPUs, DHTs, and load patterns. These results indicate that the methodology is robust with respect to typical loads that the CPUs might



(a) Single CPU per DHT

(b) All CPUs + WU

Figure 2: MAE when predicting DHT-1, DHT-2, DHT-3, and DHT-4 temperature based on a 72h Train Window for different Test Window sizes (TE) and sets of smoothed explanatory variables (single CPU in the left and all variables of the right). Test start date is Sep. 20th. Pi1 and Pi3 experience additional periodic load, Pi2 and Pi4 do not.

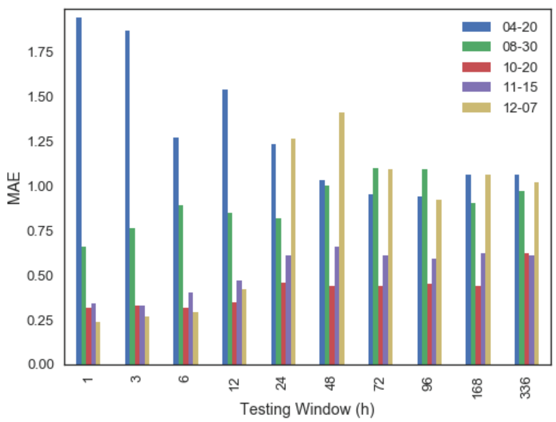


Figure 3: Comparison of MAE when predicting DHT-1 values for five different dates from April 20th to Dec. 7th.

experience in our IoT setting. Comparing Figure 2a to Figure 2b shows that multivariate regression improves accuracy across all DHTs and load patterns.

C. Effects of Seasons and Precipitation

In addition to the two dates in August and September, we observed very similar error rates when testing during different seasons (Summer, Fall, and Winter). This is illustrated in Figure 3 where we predict DHT-1 temperature for different days from April to December. April 20th (04-20) has a higher error because Pi3 and Pi4 were not yet deployed and thus their CPU values were not available as features. December 7th had variable weather conditions with alternating rainy and sunny days, which may have contributed to a somewhat higher MAE. However, even so, the MAE for most of the days it was less than $1.25°F$.

We also tested the accuracy of the model when there were changes in precipitation. From a time series perspective, precipitation could constitute a change-point in each temperature series (due to the sudden onset of evaporative cooling effects). Table IV shows the comparison of errors when training and testing periods had different levels of precipitation. For each column, the training period was 3 days and the test periods listed go from 1h to 3 days. In the first column, both training and testing days were without any precipitation (this data is the same data that is represented graphically in Figure 2b as DHT-1-ALL). In the second column, we show the effects of training using rainy days to predict the temperatures during sunny days. December 4th, 5th, and 6th were rainy days with 2.54, 1.27, and 1.27 inches of rain respectively followed by three days without precipitation that were used for testing the model. In the third column, we show results for training during sunny days followed by prediction during rainy periods. January 2nd, 3rd, and 4th were days without precipitation followed by three days with 1.29, 1.06, and 1.0 inches of precipitation respectively.

The results show that the model trained only on three rainy days had errors slightly higher than when tested on sunny days, while the model trained on sunny days behaved similarly to the models we discussed before, even when tested on rainy days. Part of our future work is to expand test cases to more variable weather conditions (e.g., including changes in wind, solar radiance, etc.). However these results indicate that the prediction errors are robust to what are essentially “shocks” to the temperature time series in the explanatory weather data (WU-T) and the predicted variables (DHT values). Because the CPUs were in sealed containers (and the DHT sensors were exposed to the atmosphere) the effects of precipitation on the CPU series is less pronounced. Still, the errors are largely unaffected by precipitation.

TE	Sep. 20th	Dec. 7th	Jan. 5th
1	0.46	0.24	0.22
3	0.36	0.27	0.40
6	0.37	0.29	0.57
12	0.42	0.42	0.76
24	0.54	1.26	0.56
48	0.51	1.41	0.54
72	0.50	1.09	0.46

Table IV: MAE for models trained and tested during dry periods (Sep. 20), training during a rainy period and testing during a dry period (Dec. 7th) and training during a dry period and tested during a rainy period (Jan. 5th). Models are trained on 3 days to predict DHT-1 temperature based on all five explanatory variables using smoothing.

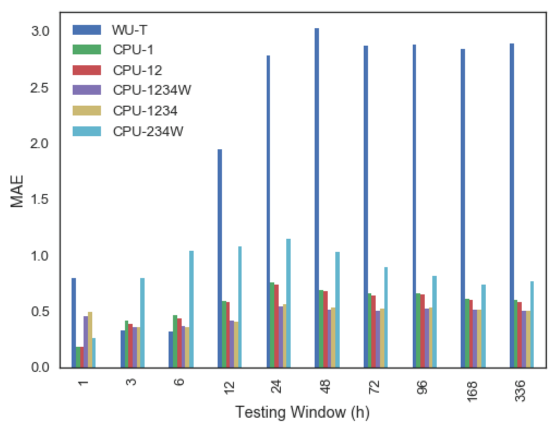


Figure 4: Comparison of MAE when predicting DHT-1 values for different sets of features for Sep. 20th.

Figure 4 illustrates the errors when predicting DHT-1 temperature with different subsets of explanatory variables. We see that if we only rely on the nearby weather station (which is approximately 800m from the nearest DHT) the error (WU-T) is much higher ($2-3°F$) than for a subset that includes at least one of the CPU temperatures ($< 1.15°F$). Farmers, today, often use only a weather station temperature reading when implementing manual frost prevention practices. Often, though, the weather station they choose to use for the outdoor temperature is even farther away from the target growing block than the station we use in this study.

Notice, also, that when the CPU that is directly connected to the DHT is not included (denoted CPU-234W in the figure), the errors are higher than when it is included (all other bars in the figure except for W). Thus, as one might expect, proximity plays a role in determining the error. However using only the attached CPU (CPU-1 in the figure which is necessarily physically closest to DHT-1) generates a higher MAE than all CPUs and the weather station (denoted CPU-1234W in the figure). Indeed, the best performing model is this one that uses all four CPU temperatures and WU-T measurements as explanatory variables, yielding an MAE $< 0.5°F$ across all time frames. Thus using the nearest CPU improves accuracy, but using only the nearest CPU does not yield the most accurate prediction. Finally, while the weather station data does not generate an accurate prediction by itself, including it does improve the accuracy (slightly) over leaving it out.

In summary, our methodology is capable of automatically synthesizing a "virtual" temperature sensor from a set of CPU measurements and externally available weather data. By including all of the available temperature time series, it automatically "tunes" itself to generate the most accurate predictions even when one of the explanatory variables (WU-T in Figure 4) is, by itself, a poor predictor. These predictions are durable (lasting up to 2 weeks without refitting the regression coefficients), with errors often at the threshold of measurement error (for DHT sensors), on average, and relatively insensitive to seasonal and meteorological effects, as well as typical CPU loads in the frost-prevention setting where we have deployed it as part of an IoT system.

IV. Related Work

The proliferation of sensor network technologies enabled more granular sensing of the environment and in turn improved our abilities to model and predict future weather events with greater accuracy. In recent years, we have seen proposals of precision farming end-to-end edge cloud systems that provide sensor integration, data analysis, and actuation [9], [10], [11], [12], [13], [14]. In this work, we focus on temperature prediction since it (together with other weather parameters like humidity, wind speed, solar radiation, cloud cover) influences some of the most energy consuming practices: frost prevention and irrigation [15], [16], [17], [18], [19], [20]. Moreover, our work does so at no additional cost for sensors and frees up SBC ports for use by other sensors.

The authors of the work most related to our own [1] use univariate linear regression to estimate the outdoor temperature based on the CPU temperature of a single co-located SBC. In two experiments tested (non-smoothed and smoothed with the single spectrum analysis [3]), the model does not perform as well when the training window is smaller than 6h ($4.5 - 14.6°F$) or larger than one week ($1.3 - 4.8°F$). We overcame this limitation with multiple linear regression that uses nearby devices' CPU temperature readings as well as the temperature of a close weather station. This yielded more stable models and the error for two weeks tests did not exceed $1.25°F$ under similar training and testing weather conditions. Even in the case of models trained on consecutive rainy days and tested on sunny days, the MAE did not exceed $1.25°F$. Our work also investigates alternative smoothing techniques and the impact of processor load on prediction.

More generally, linear regression [4] is used in sensor networks for modeling, summarizing, and data analysis [21]. The work described in [21] was deployed indoors, where there was no need to consider seasonal or sudden weather changes, or where such had a smaller impact on the regression coefficients. Our work differs in that it does not measure the temperature directly but it estimates it from the CPU temperatures of nearby devices, while considering different smoothing techniques and sets of explanatory variables.

V. Conclusion

We have presented a new approach for predicting outdoor temperature from the processor temperature of SBCs in outdoor IoT settings. To enable this, we employ multiple linear regression using nearby SBC processors and weather stations. We use these models to predict microclimate temperatures, which can be used (if sufficiently accurate) in agricultural settings to guide irrigation, frost control, and other IoT applications. We deploy our system in a citrus grove and perform an extensive empirical study using the devices and methodology. In addition, we consider the impact of loaded and unloaded processors as well as alternative smoothing techniques. We train our models for up to three days and evaluate their accuracy for a duration of up to two weeks. We find that our approach enables a prediction error that is less than $1.5 \circ F$, while past work resulted in errors of 1–14 degrees Fahrenheit for similar datasets.

Acknowledgments. This work is supported in part by NSF CNS-1703560, OAC-1541215, CCF-1539586, ACI-1541215 and the California Energy Commission (PON-14-304)

References

[1] C. Krintz, R. Wolski, , N. Golubovic, and F. Bakir, "Estimating outdoor temperature from cpu temperature for iot applications in agriculture," in *International Conference on the Internet of Things*, 2018.

[2] "Raspberry Pi," 2018, [Online; accessed 10-Oct-2018] www.raspberrypi.org.

[3] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for time series*. Springer Science & Business Media, 2013.

[4] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning new york," *NY: Springer*, 2009.

[5] "Adafruit AM2302 Wired DHT22 Temperature and Humidity Sensor," 2018, [Online; accessed 22-Jun-2018] https://www.adafruit.com/product/393.

[6] "WeatherUnderground," 2018, [Online; accessed 22-Jun-2018] http://www.weatherunderground.com/.

[7] J. D. Moore, J. S. Chase, P. Ranganathan, and R. K. Sharma, "Making scheduling" cool": Temperature-aware workload placement in data centers." 2005.

[8] A. M. Haywood, J. Sherbeck, P. Phelan, G. Varsamopoulos, and S. K. Gupta, "The relationship among cpu utilization, temperature, and thermal power for waste heat utilization," *Energy Conversion and Management*, vol. 95, pp. 297–303, 2015.

[9] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. N. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture." in *NSDI*, 2017, pp. 515–529.

[10] C. Krintz, R. Wolski, N. Golubovic, B. Lampel, V. Kulkarni, B. Sethuramasamyraja, B. Roberts, and B. Liu, "SmartFarm: Improving Agriculture Sustainability Using Modern Information Technology," in *KDD Workshop on Data Science for Food, Energy, and Water*, Aug. 2016.

[11] T. Ojha, S. Misra, and N. S. Raghuwanshi, "Sensing-cloud: Leveraging the benefits for agricultural applications," *Computers and electronics in agriculture*, vol. 135, pp. 96–107, 2017.

[12] N. Karimi, A. Arabhosseini, M. Karimi, and M. H. Kianmehr, "Web-based monitoring system using wireless sensor networks for traditional vineyards and grape drying buildings," *Computers and Electronics in Agriculture*, vol. 144, pp. 269–283, 2018.

[13] K. Foughali, K. Fathallah, and A. Frihida, "Using cloud iot for disease prevention in precision agriculture," *Procedia Comput. Sci*, vol. 130, pp. 575–582, 2018.

[14] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, and M. Ismail, "Energy-efficient wireless sensor networks for precision agriculture: A review," *Sensors*, vol. 17, no. 8, p. 1781, 2017.

[15] N. R. Dalezios and G. P. Petropoulos, "6 frost and remote sensing," *Remote Sensing of Hydrometeorological Hazards*, p. 105, 2017.

[16] J. Cai, Y. Liu, T. Lei, and L. S. Pereira, "Estimating reference evapotranspiration with the fao penman–monteith equation using daily weather forecast messages," *Agricultural and Forest Meteorology*, vol. 145, no. 1-2, pp. 22–35, 2007.

[17] Y. Lu, Y. Hu, M. Zhao, and Y. Sun, "Temperature variability during frost event on different terrain of tea fields," in *2017 ASABE Annual International Meeting*. American Society of Agricultural and Biological Engineers, 2017, p. 1.

[18] T. Watteyne, A. L. Diedrichs, K. Brun-Laguna, J. E. Chaar, D. Dujovne, J. C. Taffernaberry, and G. Mercado, "Peach: Predicting frost events in peach orchards using iot technology," *EAI Endorsed Transactions on the Internet of Things*, 2016.

[19] S. A. Nikolidakis, D. Kandris, D. D. Vergados, and C. Douligeris, "Energy efficient automated control of irrigation in agriculture by using wireless sensor networks," *Computers and Electronics in Agriculture*, vol. 113, pp. 154–163, 2015.

[20] D. A. Winkler, M. Á. Carreira-Perpiñán, and A. E. Cerpa, "Plug-and-play irrigation control at scale," in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*. IEEE Press, 2018, pp. 1–12.

[21] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *Intl Symp on Information processing in sensor networks*, 2004.