

No-Sense: Sense with Dormant Sensors

Chayan Sarkar, Vijay S. Rao, R. Venkatesha Prasad
Embedded Software Group
Delft University of Technology, The Netherlands
{C.Sarkar, V.Rao, R.R.VenkateshaPrasad}@tudelft.nl

Abstract—The lifetime of a wireless sensor network mainly depends on battery capacity and energy consumption at each node for operations such as, sensing, processing and communication. A popular approach to save energy has been to restrict the frequency of these operations, rendering lower quality data at the sink. In this article, we propose *Virtual Sensing Framework* (VSF), which reduces the frequency of the above mentioned operations at each node while not compromising on the sensing interval, and hence resulting in higher quality data at the sink. VSF creates virtual sensors at the sink to exploit the spatio-temporal correlations among sensed data. Using an adaptive model at every sensing iteration, the virtual sensors can predict *multiple consecutive* sensor data for dormant physical sensors. We show that even when the sensed data represents different parameters (e.g., light, temperature), our proposed technique works well. Applying our technique on the real-world data sets, we attain substantial reduction in energy consumption per node while maintaining high accuracy of the sensed data. To achieve higher energy reduction, VSF has to be used in conjunction with various layers and protocols of the communication stack. Thus, it has the potential to open up new research insights to make the best use of statistical properties of collected sensor data in a network.

I. INTRODUCTION

Wireless sensor networks (WSNs) have enabled continuous monitoring of an area of interest (body, room, region, etc.) while eliminating expensive wired infrastructure. Typically in such applications, wireless sensor nodes report the sensed values to a sink node, where the information is required for the end-user. WSNs also provide the flexibility to the end-user for choosing several parameters for the monitoring application. For example, placement of sensors, frequency of sensing and transmission of those sensed data. Over the years, the advancement in embedded technology has led to increased processing power and memory capacity of these battery powered devices. However, batteries can only supply limited energy, thus limiting the lifetime of the network. In order to prolong the lifetime of the deployment, various efforts have been made to improve the battery technologies and also reduce the energy consumption of the sensor node at various layers in the networking stack. Of all the operations in the network stack, wireless data transmission and reception have found to consume most of the energy. Hence many proposals found in the literature target reducing them through intelligent schemes like power control, reducing retransmissions, etc. In this article we propose a new framework called *Virtual Sensing Framework* (VSF), which aims to sufficiently satisfy application requirements while conserving energy at the sensor nodes.

In VSF, we define a *Virtual Sensor* (VS) as a software

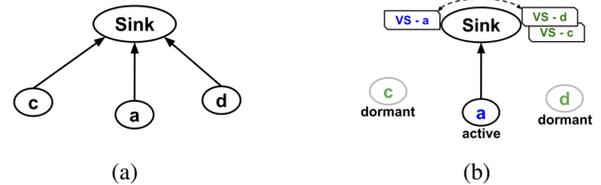


Fig. 1: (a) Data collection scenario in a WSN (star topology); (b) Data collection with virtual sensing framework.

construct in the sink that represents a Physical Sensor (PS)¹ in the WSN. A PS can either be dormant (lowest power mode or sleep mode) or be active (microcontroller, sensors and/or radio are on). When a PS is active, the frequency of its sensor data reporting varies adaptively, which is described later. A VS holds the measured sensor data reported by its corresponding PS, when active, or predicts sensor data on behalf of its PS, when dormant, by exploiting the temporal and spatial correlations among sensors' data in the WSN. It is important to note that the VSF does not consider any *a priori* knowledge about the correlation patterns in the data collected from the sensors. Rather, it utilizes adaptive prediction schemes for each sensor based on inherent correlation among the sensors' data. This is known as non-model approach [15].

Each VS saves energy for its PS by following a collaborative technique to balance between the accuracy of the predicted values and the energy consumption at its PS. Since VSs “reside” in the sink/base station, the collaboration has insignificant overheads; the only overhead is notifying the PSs whether to be in sleep mode, or in active mode for a certain duration. The concept of VSF is depicted in Fig. 1. With this background we list below the major contributions of this work:

- To the best of our knowledge, we are the first to propose a generic energy saving scheme for sensors based on data prediction that can be applied to any type of data collection network. The generality of the method lies in prediction of *multiple consecutive* sensor values for dormant sensors without having prior knowledge (or model) of the sensed data, physical nature of the sensing and deployment information of the nodes.
- Our proposal exploits both the spatial and temporal correlations among sensors. It can track the changes in the correlation and adapts to the situation through the use of blind adaptive filtering technique.

¹A physical sensor is nothing but an wireless sensor node in the WSN.

- The energy saving technique proposed in this article ensures that every node spends energy almost evenly. We show that the overall energy consumption of the whole deployment is reduced, while maintaining sufficiently high accuracy of the predicted sensed values.
- We propose to predict consecutive sensor data to keep the sensor nodes dormant for a longer duration whereby reducing the overhead of switching between low power and active modes.

II. RELATED WORK

While many energy saving techniques/protocols have been proposed for wireless sensor networks, we only review the works that are relevant to our proposal. One common approach to reduce energy consumption is to select a subset of nodes among all the sensors deployed in the network. As the sensors usually have spatial correlation, missing sensor data could be reconstructed from the subset of sensor data. Most of the proposals found in the literature do not find the correlation among sensors explicitly. Rather a good correlation structure among the sensors is assumed to be known *a priori* [4], [5], [6], [8], [10], [14]. However, we do not assume any predefined correlation structure and our adaptive scheme tracks the correlation amongst the sensors in real-time. Further, it should be noted that, to support our prediction, we do not draw any inference based on the physical nature of sensing and deployment of sensors.

A LMS-based adaptive prediction technique has been exploited by Santini *et al.* [15], which does not consider prior knowledge about the sensor data. While our strategy is similar, we exploit spatial correlation to estimate the sensing values blindly. That is, our scheme can predict multiple consecutive sensing values while the sensor node remains dormant. A real-time, blind prediction scheme has also been proposed by Li *et al.* [11]. However, they assume predefined spatial correlations among the sensors and it is required to have a very high spatial correlation between the sensors.

Guestrin *et al.* [7] provided a technique where sensor nodes save energy by restricting their data transmission. The nodes only transmit the parameters of estimation model – instead of the data itself – only if significant changes are noticed in the sensor data. In contrast with this method, our approach can predict consecutive sensor values accurately while the node is in sleep mode. There are some works where estimation techniques have been used to predict sensor data in order to fill the missing data points and complete the sensor data set [13]. Thus, they are not suitable for online sensor data prediction for consecutive sensing instances.

III. VIRTUAL SENSING FRAMEWORK (VSF)

VSF aims to reduce energy consumption of a sensor network by reducing its activity, i.e., reducing frequency of sensing, processing and data transmissions. The data collection is complemented by predicting sensor data at the sink. To predict the sensor data, a virtual sensor (VS) is created for each physical sensor (PS) in a deployment at the sink as shown in Fig. 1. A VS instructs its associated PS on its state and activity for the forthcoming sensing intervals. VSs also collaborate among themselves in order to save more energy and maintain

high data accuracy. In the subsequent sections, we discuss how sensors collaborate among themselves and then discuss how the prediction is done.

A. Energy saving technique

Usually physical parameter values (e.g., ambient temperature) do not change abruptly in a short time span. Therefore, these values have correlation with their immediate past values (temporal correlation). Thus, a sensor value can be predicted by exploiting its temporally correlated data. In order to increase the energy savings, VS should predict successive values while its PS remains dormant. As a result, changes in the physical parameter, during the longer dormant periods, might not be captured by the temporal correlation based method. Prediction in this case can be improved with spatial correlations: if two sensors have had very high correlation in the recent past, it is safe to say that both the sensors will behave in a similar fashion for some time in the future too. Hence in VSF, we choose to exploit temporal as well as spatial correlations in the data collected from the sensors to fine-tune the prediction. With increased accuracy of predictions and correlated sensors, one sensor can remain dormant for the duration of prediction with the help of the other sensor. The latter sensor *aka* the helper is referred to as *companion*. The companion, therefore, has to be an active PS. Please note that the companion of a dormant sensor is not predefined and fixated i.e., it can change over time based on changing correlation between the sensors.

VSF also conserves energy in the companion PS, whenever possible, by incorporating a temporal correlation based predictor within it, and VS in the sink. Here the node continues to sense the physical parameter, and also predicts the value. If the prediction error lies within a sufficiently tolerable error bounds, then the sensor does not transmit the sensed data. By withholding data transmissions, significant energy is saved even in the active sensors. The dormant and active nodes are associated with two different types of VSs based on their functionality. We use the terms Type-I VS when the corresponding PS is in dormant state, and Type-II VS when the corresponding PS is active. Note that Type-II VS may or may not send the sensed data depending on the prediction error sought.

Every Type-I VS requires at least one companion, while one companion can help multiple Type-I VS. Multiple spatially correlated sensors can act as companion nodes for a Type-I VS to improve the prediction accuracy, however after a certain threshold number of companions, more such companions will not necessarily improve the prediction significantly. This trade-off between prediction accuracy and energy savings (more active node implies more energy consumption) needs to be balanced. We follow a greedy approach to maximize the energy saving: VSF maximizes the number of Type-I VS nodes, and thus reducing the number of companions.

It is clear that a dormant sensor can conserve more energy than an active sensor. In order to keep the sensor network alive for a longer period, energy consumption of the sensor nodes in the network need to be balanced over time. Thus, state of the nodes switch between dormant and active after certain number of timeslots.

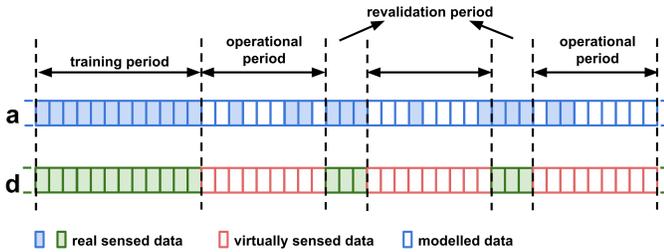


Fig. 2: Data collection phases in virtual sensing technique.

As we do not assume any *a priori* knowledge about the sensor data statistics, VSF needs to capture the correlation among sensors. It should also monitor the change in the correlation and adapt dynamically. To accomplish this, the whole data collection period is classified into three phases – training period, operational period and revalidation period as shown in Fig 2. During the training period, all the PSs collect data and transmit their data to the sink. Using the training data sets, states of the nodes (active or dormant), suitable companion for each dormant node, type of their VSs, and prediction models for each VS are decided. During operational period, the VSs become functional as the dormant and active nodes restrict their activity to conserve energy. The revalidation period resumes all PSs to active mode for a shorter period of time as compared to the training period. The prediction models are validated using the sensor data collected in this period, and the operational period resumes. If the revalidation (and calibration) of the prediction model fails, i.e., the correlations among the sensor data changes drastically, another training period is initiated. The following sections describe each of the phases in detail.

B. Training period

During the training period, T_p training data samples ($T_p > 0$) are collected from all the sensors. Using these samples, a temporal prediction mechanism is created for each type of VS at the end of the training period. To create a prediction mechanism based on temporal correlation, a transversal or tapped-delay line filter is created. First, the training data is stored in a $(T_p - p) \times p$ matrix and $(T_p - p) \times 1$ column vector as inputs (U) and outputs (\underline{d}) of the filter as shown below,

$$U = \begin{bmatrix} d(p) & d(p-1) & \dots & d(1) \\ d(p+1) & d(p) & \dots & d(2) \\ \vdots & \vdots & \ddots & \vdots \\ d(T_p-1) & d(T_p-2) & \dots & d(T_p-p) \end{bmatrix} \quad (1)$$

$$\underline{d} = [d(p+1), d(p+2), \dots, d(T_p)]^T, \quad (2)$$

where p is the order of the filter. Then, the filter coefficients,

$$\underline{\alpha} = (U^T U)^{-1} U^T \underline{d}, \quad (3)$$

are found by minimizing the mean-square error [9]. These p filter coefficients, i.e., $\underline{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_p]^T$ are used to predict the sensor data during the operational period. If the correlation is known a priori, a *Wiener filter* can be developed, which is said to be the *optimum in the mean-square error sense*. As the autocorrelation function is unknown, the filter coefficients can become outdated and may result in erroneous prediction. Using

an adaptive filtering technique, the coefficients are updated at later stages.

A Type-I VS also needs to develop a prediction mechanism based on spatial correlation. To calculate a spatial correlation based predictor, VSF uses linear regression analysis. Linear regression is a statistical method that models the relationship between a dependent variable and one/more independent variable(s) [12]. In our approach, we treat the dormant node as dependent variable and a potential companion as an independent variable. The inputs for linear regressor are collected from the companion node and stored in the form,

$$V = \begin{bmatrix} 1 & a(p+1) \\ 1 & a(p+2) \\ \vdots & \vdots \\ 1 & a(T_p) \end{bmatrix}. \quad (4)$$

Then, by minimizing the mean-square error, we calculate the coefficient of the linear regression as,

$$\underline{\beta} = (V^T V)^{-1} V^T \underline{d}. \quad (5)$$

The vector \underline{d} is used as the regressor output. $\underline{\beta} = [\beta_0, \beta_1]^T$ are used for spatial prediction during the operational period, and they are also updated during the revalidation period (discussed in Section III-D).

Finding a suitable companion: Since we do not assume any prior spatial correlation, a companion cannot be chosen beforehand. A common assumption is that two geographically co-located sensors show high spatial correlation. Nevertheless, in reality, they may show poor correlation sometimes, whereas two sensors located relatively far can show high correlation. By finding a suitable companion, we find a spatially correlated sensor that can best predict the sensor data. Again, since we do not consider any prior knowledge about the physical parameters of the sensed data, we can indeed make any sensor a companion for any other sensor during the training period. The linear regression coefficients are created separately for each potential companion using (5). The sensor, which scores the highest in the *goodness of fit* test, is selected as the companion. *Chi-squared statistics* is a well-known method to test goodness of fit [16]. To this end, the error values of the estimated signal need to be known. Using the model parameters and the training data set, first, the sensor values are estimated as $\underline{d}_{spa} = V \underline{\beta}$. Then, the Chi-squared statistics can be obtained by taking normalized sum of the squared-errors. Chi-squared statistic is calculated as,

$$\chi_{spa}^2 = \sum_{i=p+1}^{T_p} \frac{(d(i) - d_{spa}(i))^2}{\sigma^2}, \quad (6)$$

where σ^2 is the variance of the observed signal. To get an inference from the statistics, a reduced Chi-squared statistic can be calculated by dividing it by the number of degrees of freedom. The score of the goodness of fit test, represented as δ , is given by,

$$\delta = 1 - \frac{\chi_{spa}^2}{\nu}, \quad (7)$$

where, ν , the degrees of freedom is equivalent to the number of samples ($T_p - 1$). δ lies between (0, 1), where 0 implies complete failure of capturing the system behavior and 1

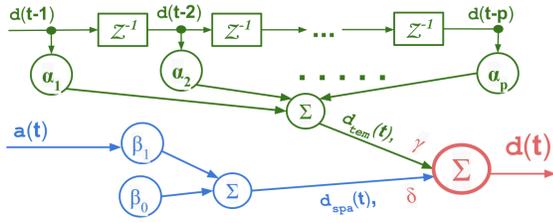


Fig. 3: The filter used for an active virtual sensor is a hybrid of a transversal filter (for temporal prediction) and linear regression (spatial prediction).

implies complete resemblance of the system behavior by the model parameters.

C. Operational period

Once the prediction coefficients are found, the operational period starts (see Fig. 2). In this period, a Type-I VS informs its corresponding PS to remain dormant for O_p timeslots, and it predicts consecutive O_p data points using a hybrid model consisting of temporal and spatial predictions (see Fig. 3). At t^{th} timeslot, first, the temporal prediction is done using

$$d_{tem}(t) = \sum_{i=1}^p \alpha_i \cdot d(t-i), \quad (8)$$

where α_i is found using (3). Then the VS collects the sensor data from its companion. Using the data from the companion and the linear regression coefficients, the spatial prediction is found using

$$d_{spa}(t) = \beta_0 + \beta_1 \cdot a(t). \quad (9)$$

The final predicted value of the missing sensor data is computed by taking a weighted average of these two predicted values as,

$$\hat{d}(t) = \frac{(\gamma \cdot d_{tem}(t) + \delta \cdot d_{spa}(t))}{(\gamma + \delta)}. \quad (10)$$

The weights γ and δ are the *goodness of fit* score of the temporal and spatial predictions respectively. The value of γ is calculated the same way as δ has been calculated while choosing the best companion using (6) and (7).

During the operational period, a Type-II VS informs its corresponding PS to continue its sensing measurement for the predefined interval and to avoid any unnecessary data transmissions if possible. At the beginning of the operational period, Type-II VS sends the filter coefficients to the physical sensor. At t^{th} instant in operational period, the sensor predicts the sensor value using the filter coefficients, $\underline{\alpha}$, (see (3)). Moreover, the actual measurement is also available since it is active. Then, it calculates the prediction error as given by (11) using the model parameters. If the absolute error of the prediction is within a permissible (predefined) limit, then sensor node suspends transmission and the corresponding Type-II VS predicts the sensed value using the same filter coefficients. Otherwise, the sensor node transmits the value to the sink and the model coefficients are updated both at the sensor node and at the corresponding VS in the sink. In this way, the model parameters are synchronized at both the places. At the same time, we can ensure that the prediction

error remains within the permissible error limits. To update the model parameters (at both the places), we have used *least-mean-square (LMS) algorithm*. LMS is a widely used adaptive filtering technique in time-series predictions. It requires very less memory and computational capabilities, and performs well [15].

To update the temporal filter coefficients, i.e., the $\underline{\alpha}$ vector, first, the prediction error is calculated given by,

$$e_{temp}(t) = d(t) - d_{temp}(t), \quad (11)$$

where $d(t)$ and $d_{temp}(t)$ are the actual and predicted sensor values respectively. Then the filter coefficients are updated using,

$$\underline{\alpha}(t+1) = \underline{\alpha}(t) + \mu \cdot \underline{d}(t) \cdot e_{temp}(t), \quad (12)$$

where $\underline{d}(t) = [d(t-1), d(t-2), \dots, d(t-p)]^T$ is the input vector of the filter, and μ is the learning rate of the adaptive algorithm [9]. Procedure to set μ can be found in [15].

D. Revalidation period

Since Type-I VS asks the corresponding PS to go to dormant mode, there is a chance that the predicted value might diverge from the ground reality. To tackle this, revalidation of the model parameters are done after O_p sensing intervals (here we skip the discussion on fixing O_p due to paucity of space). During this period, i.e., $R_p (< T_p)$ sensing intervals, all the PSs becomes active. That is, they sense and transmit the data to the sink (see Fig. 2). Then, the temporal prediction coefficients are updated as described in (11) & (12). The spatial prediction coefficient, i.e., $\underline{\beta} = [\beta_0, \beta_1]^T$, is also updated based on the spatial prediction error. As the final prediction of a Type-I VS is dependent on γ and δ (10), an updated *goodness of fit* score, i.e., the *Chi-squared statistics* is calculated using,

$$\chi_{tem}^2 = \chi_{tem}^2 + \frac{e_{tem}^2(t)}{\sigma^2} \quad (13)$$

$$\chi_{spa}^2 = \chi_{spa}^2 + \frac{e_{spa}^2(t)}{\sigma^2}. \quad (14)$$

IV. EVALUATION

In this section, we evaluate our virtual sensing technique on the real data sets obtained from the Lausanne Urban Canopy Experiment [2]. The data were collected in the EPFL campus between July 2006 and May 2007, where 97 sensor nodes monitored various environmental parameters, e.g., ambient temperature, solar radiation, relative Humidity,, etc. These sensor nodes collected data every 30s. We have applied VSF on the ambient temperature data collected from multiple sensor nodes. We provide the most interesting results here. We chose two nodes randomly (which turned out to be node 3 and node 44), and we associated Type-I and Type-II virtual sensors with them respectively. We tested our prediction technique in Matlab. 10^3 data samples from nodes are used. A snapshot of the predicted sensor data by the VSs are shown in Fig. 4. We fixed the operational (O_p) and revalidation (R_p) window sizes to 20 and 5 respectively. The training was done at the beginning of the data streams, which is not shown in the figure. The result shows that the virtual sensors can achieve significant prediction accuracy.

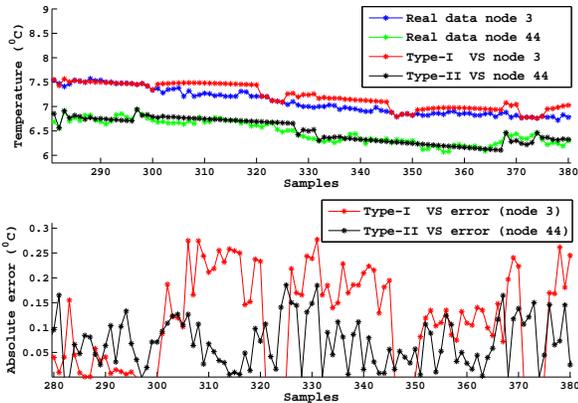


Fig. 4: Absolute error in sensor data prediction using a Type-I and Type-II virtual sensor.

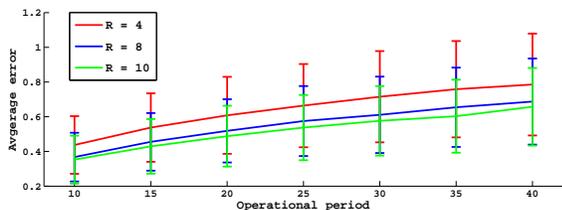


Fig. 5: Average prediction error for various operational period (M) and revalidation period (R).

To further investigate the accuracy of the VSF based predictions, we have conducted simulations by varying O_p and R_p . The average error with the standard deviation of error as a confidence interval for various values of the $\langle O_p, R_p \rangle$ pair is shown in Fig. 5. From this study, as expected, when R increases, the average prediction error as well as the variance of prediction error is reduced. Nonetheless, it is at the cost of energy consumption since more data is transmitted. To tackle this we can also increase O_p . However, with increased O_p , the average prediction error also increases. The choice of these window sizes depends on the applications as accuracy and energy cost go together. From our simulation results, we can conclude that VSF provides the tool to restrict data transmissions from sensor nodes with an insignificant reduction in the quality of data.

Next, we show that the virtual sensing technique can save more energy using blind estimation. Three simulations are conducted for three different lengths of O_p and R_p . For each pair of O_p and R_p , the error threshold is also varied from 0°C to 2°C . Error threshold 0 represents the base case, where all 1000 data samples from the sensors are sensed and transmitted.

The system parameters and their settings are listed in Table I to calculate the energy consumption. The value of each parameter is calculated using methods described in [3], [17]. Using these parameters and the number of times a sensor senses and transmits data, energy consumption per node is calculated. As a Type-I VS (dormant node) is always accom-

TABLE I: System Parameters and Settings

Parameter	Value
Message size	128 B
Transmission power	0 dBm
Energy cost for sending a message	$341 \mu\text{J}$
Energy cost for sensing temperature	$330 \mu\text{J}$
Energy cost in active mode	4.898 mW
Energy cost in low power mode (LPM3)	0.144 mW
Energy cost in switching (LPM to active)	0.016 mW

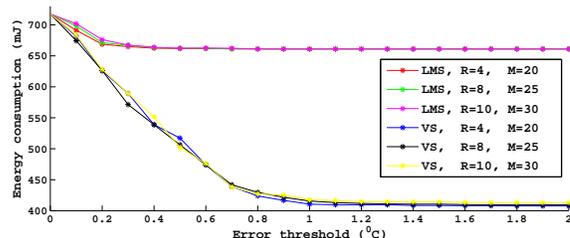


Fig. 6: Combined energy consumption for one Type-I and one Type-II virtual sensors v/s LMS-based scheme for two sensors.

panied by a Type-II VS (active node), a combined energy consumption is calculated and compared with the LMS-based method described in [15]. As the error threshold increases, VSs consume lesser energy at the PSs. Note that energy consumption calculated here is based on only sensing and data transmissions. Using VSF, a Type-I VS can achieve higher energy conservation due to its dormant PS. Furthermore, with lesser number of transmissions, the relay nodes in WSN have to spend lesser energy in forwarding the data packets towards the sink. As a result, further energy consumption is achieved, which is not accounted in this study.

To improve prediction accuracy of a Type-I VS multiple spatially correlated sensors can be used as companions. In Fig. 7, prediction error based on one and two companions are shown. It is evident that the prediction accuracy improves when multiple spatially correlated sensors are used for prediction. Nevertheless, more energy is consumed due to multiple active PSs that act as companions. There is a trade-off between data accuracy and energy spent in the nodes.

As an extension of VSF, we have also tried *heterogeneous virtual sensing*, where the companion sensor is monitoring a different physical parameter. We have tried to predict temperature using a light sensor (see Fig. 8). The light and temperature sensor data are collected from [1]. This shows the effectiveness

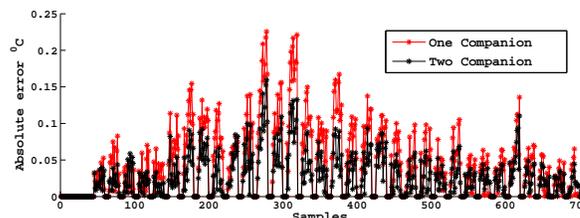


Fig. 7: Prediction error for a Type-I VS, when one and two active nodes are used as companion.

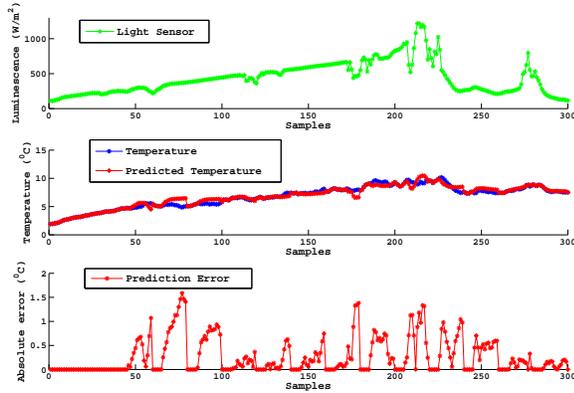


Fig. 8: Temperature sensor values are predicted using a light sensor as its companion.

of heterogeneous virtual sensing. However, there are some issues to be addressed such as light sensor data can change quickly over a short period of time, while temperature changes gradually. This affects accuracy of predictions. Fine tuning our technique to adapt to the situation based on estimated prediction errors may help in increasing the reliability and usability of VSF in large scale deployment. Further, VSF could also be used in case some type of sensors is not available at a location. To achieve better error bounds, more investigations are required with respect to heterogeneous virtual sensing.

V. CONCLUSIONS

In this article we introduced virtual sensing framework, which predicts multiple consecutive sensor data while the physical sensor remains dormant. We have utilized the inherent spatio-temporal correlation amongst the sensor data without having any *a priori* knowledge about the statistics of the data, location of the sensor nodes and type of observed physical parameters. A case in point is predicting temperature with a light sensor with a tolerable error bound. The proposed prediction scheme adapts to the changes in the sensor data. Using our technique, we have achieved a significant improvement on energy saving as compared to other methods while maintaining high accuracy of the sensor data. We have reported around 4% of error in data. We believe that our technique will be useful when large number of sensors are deployed and with the advent of Internet of Things (IoT) paradigm. One criticism of this work, as it stands now, is that we have data from both sensors in our VSF. However, the idea is to make sensors sleep. When some sensors sleep, a method to estimate the error between predicted value and what measurement those sensors would have thrown is needed. This is a hard problem but is crucial to achieve higher savings in energy and higher network lifetime.

ACKNOWLEDGMENT

This work is supported by an EU FP7 project, called iCore (<http://www.iot-icore.eu/>), contract number: 287708.

REFERENCES

[1] Nrel: Lowry range solar station, colorado state land board [online]. <http://www.nrel.gov/midc/>.

[2] Sensorscope: Lausanne urban canopy experiment deployment [online]. <http://lcav.epfl.ch/cms/lang/en/pid/86035>.

[3] M. Amiri. Measurements of energy consumption and execution time of different operations on tmote sky sensor nodes. 2010.

[4] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. In *ICDE'06.*, pages 48–48. IEEE, 2006.

[5] R. Cristescu and M. Vetterli. On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 21. IEEE Press, 2005.

[6] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 588–599. VLDB Endowment, 2004.

[7] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, IPSN '04, pages 1–10, New York, NY, USA, 2004. ACM.

[8] H. Gupta, V. Navda, S. Das, and V. Chowdhary. Efficient gathering of correlated data in sensor networks. *ACM Trans. Sen. Netw.*, 4(1):4:1–4:31, Feb. 2008.

[9] S. S. Haykin. *Adaptive Filter Theory, 4/e.* Pearson Education India, 2005.

[10] A. Jain, E. Y. Chang, and Y.-F. Wang. Adaptive stream resource management using kalman filters. In *Proceedings of the 2004 ACM SIGMOD*, pages 11–22. ACM, 2004.

[11] Y. Li, C. Ai, W. Deshmukh, and Y. Wu. Data estimation in sensor networks using physical and statistical methodologies. In *Distributed Computing Systems, 2008. ICDCS '08. The 28th International Conference on*, pages 538–545, june 2008.

[12] D. C. Montgomery, E. A. Peck, and G. G. Vining. *Introduction to linear regression analysis*, volume 821. Wiley, 2012.

[13] L. Pan. K-Nearest Neighbor Based Missing Data Estimation Algorithm in Wireless Sensor Networks. *Wireless Sensor Network*, 02(02):115–122, 2010.

[14] Z. Quan, W. J. Kaiser, and A. H. Sayed. A spatial sampling scheme based on innovations diffusion in sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 323–330, New York, NY, USA, 2007. ACM.

[15] S. Santini and K. Romer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd International Conference on Networked Sensing Systems (INSS 2006)*, pages 29–36, 2006.

[16] J. R. Taylor. *An introduction to error analysis: the study of uncertainties in physical measurements.* University science books, 1997.

[17] T. Venkata Prabhakar. An empirical approach towards zero energy networks (zen). 2012.