VSF: An Energy-Efficient Sensing Framework using Virtual Sensors

Chayan Sarkar, *Graduate Student Member, IEEE*, Vijay S. Rao, *Graduate Student Member, IEEE*, R. Venkatesha Prasad, *Senior Member, IEEE*, Sankar Narayan Das, *Graduate Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Athanasios Vasilakos, *Senior Member, IEEE*

***Abstract*—In this article, we describe *Virtual Sensing Framework* (VSF), which reduces sensing and data transmission activities of nodes in a sensor network while not compromising on the sensing interval, and hence data quality. VSF creates virtual sensors at the sink to exploit the temporal and spatial correlations amongst sensed data. Using an adaptive model at every sensing iteration, the virtual sensors can predict *multiple consecutive* sensed data for all the nodes with the help of sensed data from a few active nodes. We show that even when the sensed data represents different physical parameters (e.g., temperature and humidity), our proposed technique still works making it independent of sensor type. Applying our techniques can substantially reduce data communication among the nodes leading to reduced energy consumption per node yet maintaining high accuracy of the sensed data. In particular, using VSF on the temperature data from IntelLab and GreenOrb dataset, we have reduced the total data traffic within the network up to 98% and 79% respectively, while the average root mean squared error of the predicted data per node is as low as 0.36°C and 0.71°C respectively. This work is expected to support future Internet of Things in large scale deployment.**

***Index Terms*—**

I. Introduction

Energy is a precious resource in wireless sensor nodes. Idle-listening and packet overhearing is a great source of energy drain. Thus, a number of MAC protocols [1], [2] have been developed to tackle these issues. Since data transmission and reception require higher energy compared to sensing, many efforts have also been made to reduce the overall data traffic within the network. This includes efficient clustering algorithms and in-network data aggregation [3], [4], coverage problems [5], [6], data compression [7], [8], etc. This work falls in the later category, where energy-efficiency is achieved by reducing activity of the nodes.

A. Motivation

A popular technique to reduce overall data traffic is to utilize the over-provisioning of nodes, i.e., keep only a subset of nodes active at any point of time [9]. In such cases, data from the inactive nodes are either assumed to be the same as that of active nodes or can be reproduced using the data from the active nodes. This reproduction is possible due to the fact that there is inherent spatial and temporal correlation amongst the sensors, and if two sensor nodes show very high correlation, the data from one node can be predicted accurately with the help of the other. In a wireless sensor networks (WSN), usually a node is highly correlated with many other nodes. To maximize the energy savings of the network, as many nodes as possible need to be kept in low-power sleep mode (dormant) while their data can be predicted accurately with the help of few active nodes. We term this as "maximum sleeping node policy".

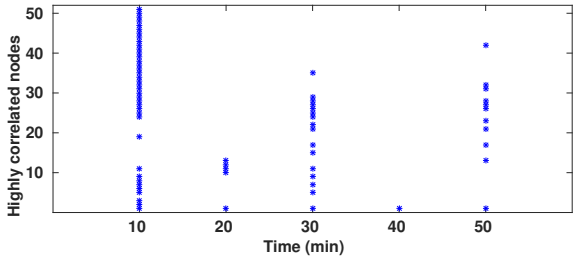


Fig. 1: Nodes that are highly correlated with Node 1 at different period are marked in the vertical lines.

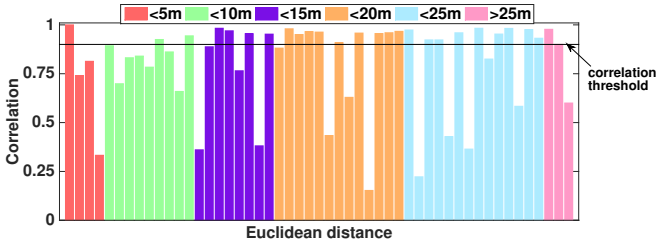


Fig. 2: Correlation among the nodes cannot be described using Euclidean distance.

Many existing correlation-based data gathering techniques assume *a priori* correlation among the sensor nodes [10], [11], [12]. However, data correlation among the sensor nodes often shows a dynamic behavior, i.e., the correlation among the nodes changes over time. Based on the temperature data of the IntelLab deployment [13], we have calculated the correlation among the nodes. Fig. 1 shows the nodes (asterisks on the vertical line) that are highly correlated with Node 1 at different time. From this figure it is clear that the correlation among the nodes varies significantly with time.

Another common assumption is that if two sensor nodes are geographically collocated, they produce highly correlated data [12], [14]. However, in many real-life deployments, two sensor nodes that are far apart can also show high data

correlation; whereas, two close-by sensors may show poor data correlation. To demonstrate our claim, we show the correlation and geographical distance between Node 1 with all other nodes in Fig. 2. According to general assumption, the correlation among nodes decline with increase in distance. However, the figure shows that this claim is not always valid. Based on these observations, we claim that any correlation-based data prediction framework should be dynamic and adaptive.

B. Contribution

We propose *Virtual Sensing Framework* (VSF), an efficient data collection framework for WSN that reduces activities of the nodes to reduce overall energy consumption of the network, and maintains sensing requirements of the deployment. It considers changes in underlying correlation structure while exploiting the data correlation among the nodes. The following are our main contributions.

1) Virtualization of sensed data: To reduce energy consumption, VSF adopts the policy of keeping as many nodes as possible in low-power sleep mode and only a few nodes as active. It also helps avoiding data transmission for the active nodes whenever possible. The fewer measurement data is complemented by accurate prediction of the sensed data exploiting correlations. The data prediction is done by the virtual sensors in successive sensing intervals with minimal involvement of the physical nodes.

2) Adaptive correlation exploitation: VSF provides an adaptive prediction scheme without considering any *a priori* correlation structure among the nodes. We consider a generic situation where geographical collocation of nodes may or may not imply higher correlation among the nodes. To the best of our knowledge, this is the first piece of work that considers this generalist view. Using an information theoretic approach, we also show that VSF can retain high data accuracy whenever there is high correlation among the nodes.

3) Active node selection: We formulate a sensor selection problem for collection of sensor data that are correlated where maximal sleeping and minimum energy consumption policy is adopted. We propose a heuristic algorithm for the selection of active nodes. The algorithm selects a new set of active nodes after a few sensing intervals to attain uniformity in energy expenditure amongst the nodes over a longer period. Using VSF on real-world datasets, we show significant amount of energy savings while maintaining the required data accuracy. This eventually leads to lifetime improvement of the deployed WSN.

The rest of the paper is organized as follows. First, we discuss some the existing energy-efficient data collection techniques for WSN in Section II. Then, we discuss the virtual sensing framework and activity reduction technique in Section III. In Section IV, we discuss the problem of active node selection and propose a heuristic algorithm. In Section V, we provide a thorough evaluation of our system. In the first part of our evaluation, we show how the estimation mechanism performs and how to set various parameters of the system. In the latter part we compare our system with some of the existing approaches. Finally, we conclude our work in Section VII.

TABLE I: Comparing VSF with energy-efficient data collection techniques.

Techniques	Data traffic reduction	Keep nodes in dormant state	Correlation-based	Remarks
Clustering	yes	no	yes/no	usually all nodes are active and send their
In-network data aggregation	yes	no	yes/no	improvements (compliments) clustering
Sensor data prediction	yes	no	yes	most comparable work with our method
Coverage problem	yes	yes	no	also comparable with our work
Compressed sensing	yes	no	no	does not exploit correlation; but the goals is to reduce data traffic

II. Related work

A large body of energy efficient data gathering techniques exists in the literature. As the data transmission consumes a large amount of energy, a common approach is to reduce the amount of traffic in the network. Clustering is an effective approach that reduces energy consumption by reducing the number of data transmissions within a network [15], [16]. A group of collocated nodes select one amongst them as a cluster-head (CH). Instead of directly reporting the sensed data to the sink node, the nodes deliver the data to the CH. The CH then takes the responsibility of delivering the data to the sink. Usually, in a large WSN, multiple cluster heads exist, and collaborate amongst themselves to deliver the data to the sink. Thus, there is less traffic and less energy expenditure for data transmission. However, every node need to be active to sense and send their data.

Another important class of data collection technique is in-network data aggregation to restrict the amount of data to be delivered to the sink [4], [17], [18]. Data aggregation can be performed either in every node or in a special node like the CH. Though the idea is to combine data from multiple sources and reduce data transmissions within the network, nodes are still required to sense and send data. In network aggregation combined with clustering can significantly reduce the overall energy consumption of the network.

Most of these techniques do not consider correlations among the sensor nodes. A policy for reduction in data transmission can be formulated leveraging correlation-based estimation techniques. A data estimation model can be formulated if the estimated data does not deviate much from the sensed data. Thus transmissions can be avoided. Transmission occurs only when there is a large deviation is detected in the sensed data [19], [20], [21]. Even though energy is saved significantly by avoiding data transmission whenever possible all nodes need to sense continuously.

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*

[10], [22], [23], [24]. Then by exploiting the correlation, a significant amount of energy is saved. Correlation-based collaborative MAC exploits spatial correlation in the MAC layer to reduce the communication of redundant data [25]. On the other hand, He *et al.* [26] propose a cross-layer approach to gather correlated data. Cheng *et al.* utilize spatial correlation for data collection in the WSNs equipped with multiple sinks [27]. Further, correlation is also used for enhancing monitoring quality [28] and estimating the missing data [29].

For WSNs, periodic on-off scheduling is an effective tool to save energy of the nodes, and some important contributions from the literature are briefly described here. When multiple sensor nodes can monitor a common sub-area, data from only one node is sufficient. The rest of the nodes are kept in sleep mode [5], [9]. This is applicable only when sensor nodes within close proximity produce correlated data. Moreover, spatial correlation is assumed to be known (or modelled).

Recently, compressed sensing based data collection mechanisms have received a lot of attention from the research community [8], [30], [31], [32]. The basic idea behind compressed sensing is that if a large dataset has high sparsity, it can be compressed. As a result, data transmission becomes less expensive. More sparsity means smaller data size. Based on this principle, a few samples of sensed data (from few sensor nodes) are collected at the sink. Later, whole data set for the entire network is reconstructed from these samples. Though this technique does not assume any correlation among the nodes, the basic assumption is that the data is sparse in some domain, e.g., frequency domain.

The above mentioned works are summarized in Table I. In contrast, VSF exploits the inherent data correlation among the sensor nodes without assuming any *a priori* correlation amongst the nodes including the physical parameters these sensors measure. VSF can predict data for a large number of consecutive sensing instances with sufficiently high accuracy. Furthermore, VSF also tracks changes in correlation among the sensor nodes over time. Hence VSF can adapt to the changes in the environment and can work in a wide range of deployment.

III. Virtual Sensing Framework (VSF)

A sensor node consumes energy for each of its activity, e.g., sensing, processing, data transmission etc. VSF aims to reduce energy consumption by reducing frequency of sensing, processing and data transmissions. In a data collection WSN, each sensor node sense data in a predefined way and report the sensed value to the sink. If the sensing activity of the sensor nodes is reduced to lower the energy consumption, the purpose of the deployment cannot be fulfilled. VSF supplements the loss of sensing at the nodes by predicting their values. As a result, the energy consumption of the nodes is reduced while the application requirement is met.

A. VSF: Activity Reduction Scheme

Virtual sensors (VS) are the basic building blocks of VSF. A VS contains a prediction model for the associated physical sensor. As virtual sensors are at the sink, the reconstruction of the sensed data does not affect the deployment goal (see Fig. 3). That is any application that uses the sensed data (for monitoring and/or decision making) will always receive the sensed data – actual measurement or predicted – from all the virtual sensors.

(a) (b)

Fig. 3: (a) Data collection scenario in a WSN; (b) Data collection with virtual sensing framework.

As sensor values are usually correlated with their immediate past values, they can be predicted by exploiting its temporally correlated data. In order to increase the energy savings, a VS should predict successive values while its corresponding PS remains dormant. Therefore, changes in the physical parameter during long dormant periods might not be captured by the temporal correlation based method. In this case, prediction accuracy can be improved by exploiting cross correlations among the nodes. If two nodes have had very high correlation in the recent past, it is safe to assume that both the nodes will behave in a similar fashion for some time in the future too. Hence in VSF, we choose to exploit autocorrelation as well as cross correlations to fine-tune the prediction. The value of r ranges between [-1,+1], where +1 signifies exactly the same pattern (or highest correlation), 0 signifies absolutely no correlation, and -1 signifies exactly opposite pattern. To define high correlation between two nodes, we define **correlated node pair** as

$$corr(s^i, s^j) = \begin{cases} 1, & \text{if} \quad |r(\underline{u}^i, \underline{u}^j)| \geq th_{corr}; \\ 0, & \text{otherwise}, \end{cases} \tag{1}$$

where node s^i and s^j form a highly correlated node pair (i, j) (or highly correlated nodes), if the above condition is satisfied; $\underline{u}^i$ and $\underline{u}^j$ are the sensed data vectors for nodes i and j respectively, and th_{corr} is a predefined correlation threshold that is used to decide sufficiently high correlation as any two vectors will always have some correlation, high or low.

In a correlated node pair, one node can remain dormant for the duration of prediction and the other one can remain active to help predict the sensing data of dormant node. This active node is referred to as *active companion*. A node can be highly correlated with more than one node, e.g., node s^i is highly correlated with m different nodes. This implies that node s^i is part of m different *correlated node pairs*. This subset (C^i) of m nodes are termed as **correlated companions**, and it can be defined as

$$s^k \in C^i \iff corr(s^i, s^k) = 1. \tag{2}$$

Every node in the network has a set of correlated companions. If a node becomes active, it can act as active companion for all of its correlated companions. Thus only a few nodes need to be active within a WSN, and most of the (remaining) nodes can be kept dormant. The correlated companions of a node change with time as the data correlation change. Therefore, the companion of a dormant node is not predefined and fixated

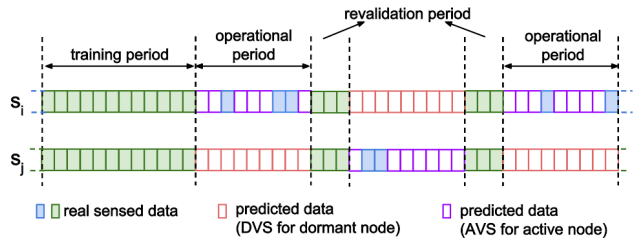


Fig. 4: Data collection phases in VSF using activity reduction scheme.

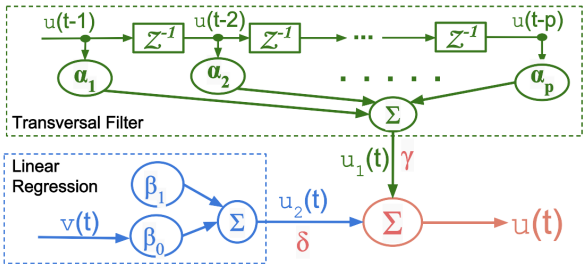


Fig. 5: The filter used for a dormant virtual sensor is a hybrid of a transversal filter (based on autocorrelation) and linear regression (based on cross correlation).

i.e., it can also change over time based on changing correlation between the sensors.

As mentioned earlier, VS uses a mix of autocorrelation and cross correlation based predictor, i.e., it utilizes the past values of itself and the currently sensed data from its active companion. This may drain energy of active nodes significantly. To circumvent this problem, VSF also conserves energy in the active nodes, whenever possible. Here, the node continues to sense the physical parameters, and also predicts the value using an autocorrelation based predictor. If the prediction error lies within a tolerable error bound, then the node does not transmit the sensed data. The data is transmitted only when the predicted data differs largely with that of sensed data. As energy spent by a nodes CPU for making the selective transmission decision is less than the energy cost of a data transmission, by withholding data transmissions, significant energy is saved even in the active sensors. When an active node does not transmit the sensed data, then the associated virtual sensor predicts the data (using autocorrelation only). It is clear that the functionality of the virtual sensors associated with a dormant and an active node are different. Thus, two different types of VSs are used. We use the terms *Dormant Virtual Sensor* (DVS) when the corresponding PS is in dormant state, and *Active Virtual Sensor* (AVS) when the corresponding PS is active.

B. VSF: Adaptive Node Correlation

It is clear that a dormant node can conserve more energy than an active node. State of the nodes switch between dormant and active modes after a certain number of time-slots to ensure equal drain of battery. As we do not assume *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 divided into three phases – training period, operational period and revalidation period as shown in Fig 4.

The sensing activity starts with training period. During this period, all the PSs collect data and transmit their data to the sink. At this point all the PSs are associated with their respective AVS. Using these training data, correlation among the sensor nodes is computed. At the end of this period, VSF marks the nodes to be either active or dormant based on their correlation. It also assigns an active companion for each of the dormant nodes. Then it creates prediction models for the dormant nodes (DVS). As mentioned earlier, a DVS uses a hybrid model of autocorrelation and cross correlation for its prediction. On the other hand, an AVS uses only an autocorrelation function. We have explained the prediction mechanism in Section III-C.

The training period is followed by an operational period, where only active nodes sense. A dormant node remains in sleep mode during this period and its values are predicted by a DVS using the values from an active companion. On the other hand, an active node performs only selective data transmissions. It discards the sensed data if it identifies that the predicted data by the AVS is within a tolerable error bound. The active PS is able to calculate the prediction error, because at the beginning of an operational period, its corresponding AVS shares the prediction model with it. When the prediction error crosses certain error threshold, the active node transmits the data and updates the model parameter. When an AVS receives the sensed data, it also updates the model parameter. In this way, the model parameters remain synchronized at both the places. Method of model update is discussed in Section III-D. The selective transmission process of an active node runs in every active node. On the other hand, the activity reduction scheme of VSF runs at the sink node (Section IV). Please note, we assume that the sink node has sufficient amount of memory and energy.

The revalidation period resumes all PSs to active mode and all DVSs are switched to AVSs. A revalidation period is shorter compared to the training period. The prediction models for the dormant nodes (in the last operational period) are validated. At the end of this period, the correlation pattern among the sensor nodes is updated, a new set of active and dormant nodes are selected and an active companion is assigned to each of the dormant node. Then, the operational period resumes. If a significant change in the correlation pattern is identified, instead of resuming an operational period, another training period may be started.

C. Prediction models for Virtual Sensors

As mentioned earlier, a dormant virtual sensor exploits exploits autocorrelation as well as cross-correlation to predict the data. Fig. 5 shows the hybrid filter used for a DVS, which is a combination of an autoregression function (transversal filter) and linear regression function. On the other hand, an active virtual sensor exploits only autocorrelation of the sensed

data; thus, only the autoregression function is used for it. An autocorrelation based transversal filter can be described as following –

$$u(t) = \sum_{i=1}^{p} u(t-i)\alpha(i), \tag{3}$$

where $u(t)$ is the predicted time-series value at time instance t; $[u(t-1), u(t-2), ..., u(t-p)]$ are the previous p values of time-series; $[\alpha(1), \alpha(2), ..., \alpha(p)]$ are the filter coefficients; and p is the order of the autocorrelation model. Once the filter coefficients are determined, the current value of the series can be estimated using the past values. To find the filter coefficients, a set of training data is required from the time series. Suppose, there are T_p training data $(u(1), u(2), ..., u(T_p))$ available from the series. Then, the filter coefficients can be found by solving the following linear equations,

$$U\underline{\alpha} = \underline{u}, \tag{4}$$

where,

$$U = \begin{bmatrix} u(p) & u(p-1) & \cdots & u(1) \\ u(p+1) & u(p) & \cdots & u(2) \\ \vdots & \vdots & \ddots & \vdots \\ u(T_p-1) & u(T_p-2) & \cdots & u(T_p-p) \end{bmatrix},$$

$$\underline{\alpha} = [\alpha(1), \alpha(2), ..., \alpha(p)]^T,$$
$$\underline{u} = [u(p+1), u(p+2), ..., u(T_p)]^T.$$

This is a overdetermined system of linear equations, as the number of equations $(T_p - p)$ are more than the number of variables (p). As U is not a square matrix, an unique solution is not possible, and an approximate solution can be found using least-square method. The least-square solution of the Eq. 4 can be found using the following equation –

$$\underline{\alpha} = (U^T U)^{-1} U^T \underline{u}. \tag{5}$$

The linear regression model can be represented using the following equation,

$$u(t) = \beta(0) + v(t)\beta(1), \tag{6}$$

where $u(t)$ is the predicted time-series value at time instance t; $v(t)$ is the another time-series (correlated) value at time t; and $[\beta(0), \beta(1)]$ are the coefficients of the linear regression model. Similar to the autoregression model, the coefficients of the linear regression can also be found by solving the following linear equations,

$$V\underline{\beta} = \underline{u},$$

where,

$$V = \begin{bmatrix} 1 & v(p+1) \\ 1 & v(p+2) \\ \vdots & \vdots \\ 1 & v(T_p) \end{bmatrix},$$

$$\underline{\beta} = [\beta(0), \beta(1)]^T,$$
$$\underline{u} = [u(p+1), u(p+2), ..., u(T_p)]^T.$$

The final prediction for a DVS is achieved by combining the outputs of both the regressor. Based on the accuracy of the models, the final value is calculated as a weighted sum of the predicted values,

$$\hat{u}(t) = \frac{(\gamma \cdot u_1(t) + \delta \cdot u_2(t))}{(\gamma + \delta)}. \tag{7}$$

The accuracy of the models are calculated using *Chi-squared statistics*, which is a well-known method to test goodness of fit [33]. 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 value is estimated using both the models ($u_1(t)$ and $u_2(t)$ in Fig. 5). Then, the Chi-squared statistics can be obtained by taking normalized sum of the squared-errors. Chi-squared statistic is calculated as,

$$\chi_1^2 = \sum_{t=p+1}^{T_p} \frac{(u(t) - u_1(t))^2}{\sigma^2}, \tag{8}$$

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 accuracy of the autoregression model (the goodness of fit), represented as γ, is given by,

$$\gamma = 1 - \frac{\chi_1^2}{\nu}, \tag{9}$$

where, ν, the degrees of freedom is equivalent to the number of samples $(T_p - p - 1)$. γ lies between $(0, 1)$, where 0 implies complete failure of capturing the system behaviour and 1 implies complete resemblance of the system behaviour by the model parameters. Similarly, the accuracy of the linear regression model (δ) can also be calculated.

D. Model parameter update

If the correlation structure among the nodes is known *a priori* and remain constant, a *Wiener filter* can be developed, which is said to be the *optimum in the mean-square error sense*. As the correlation is unknown and changes significantly with time, the filter coefficients can become outdated and may result in erroneous prediction. Thus, VSF uses an adaptive filtering technique to update the filter coefficients.

As mentioned earlier, a sensor node transmits the sensed data only when the error in prediction crosses a certain threshold value. To reduce the prediction error in future instances, the model parameters need to be updated. To update the model parameters, we have used an least-mean-square based adaptive filtering technique. First, the prediction error is calculated using,

$$e_1(t) = u(t) - u_1(t),$$

where $u(t)$ and $u_1(t)$ are the actual and predicted sensor values respectively. Then the filter coefficients are updated using,

$$\underline{\alpha} = \underline{\alpha} + \mu \cdot \underline{u} \cdot e_1(t),$$

where $\underline{u} = [u(t-1), u(t-2), ..., u(t-p)]^T$ is the input vector of the filter, and μ is the learning rate of the adaptive algorithm. Procedure to set μ can be found in [20]. Similarly, cross-correlation among the nodes are updated after each training and revalidation period.

E. Heterogeneous Virtual Sensor (HVS)

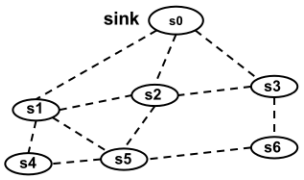
As discussed previously, VSF can accurately predict the sensed data for a dormant (or semi-dormant) sensor with the help of an active sensor. Here, the inherent assumption is that both the sensor nodes sense the same physical parameter. We extend this work to heterogeneous sensing, i.e., predicting the sensed data even if the two sensor nodes sense two different physical parameters, e.g., a temperature sensor data can be used to predict humidity sensor data, and vice-versa, a light sensor data can be used to predict temperature sensor data, and vice-versa. The mechanism for a heterogeneous virtual sensor (HVS) is just like a DVS. HVS works because VSF considers only data correlation among two sensor nodes. However, for a HVS, some contextual information is required. For example, the error thresholds might be different for two different physical parameters. Similarly, heterogeneous virtual sensing can be applied for a particular time frame, e.g., light and temperature data show correlation only during the daytime.

IV. Active node Selection

As described in Section III, if a node has multiple correlated companions, it can become an active companion for all these nodes, i.e., sensed data only from this active node is sufficient to predict data for all other nodes that are highly correlated with it. However this raises a few questions such as, (i) wheather a node should be active or dormant? (ii) how many nodes should be active? To answer these questions, in this section, we describe how nodes are assigned to either of these sets - active and dormant. The process is divided into two steps: (i) finding correlation structure of nodes; and (ii) assigning roles for nodes.

1) Finding correlation structure of nodes: Based on the sensed data, the correlation amongst the nodes can be calculated at the end of the training period. A threshold is set to define highly correlated nodes. As mentioned earlier (Section III-A), the value of correlation coefficient is 1 when two time series shows identical trend over time. Thus, correlation coefficient close to 1 signifies a similar trend between two time series if not exactly identical. As a result, sufficiently high correlation based on high correlation threshold (close to 1) between nodes means their sensed data follow a similar pattern and the prediction is expected to be highly accurate. As mentioned earlier, every node in a WSN has its *correlated companions*. As an illustration, Fig. 6a shows a small WSN with six sensing nodes. The idea is to find the correlation between all pairs of nodes and listing those which are above a certain threshold as shown in Fig. 6b. From this table, we can find the corresponding correlated companions.

2) Assigning roles to nodes: If a node is selected as an active node, all its correlated companions can be kept in dormant state. The goal is to select a minimal number of active



(a)

Node	Correlated Companions
s1	s1,s2,s3,s6
s2	s1,s2,s3
s3	s1,s2,s3,s5
s4	s4,s6
s5	s3,s5,s6
s6	s1,s4,s5,s6

(b)

Fig. 6: (a) A small WSN with six sensing nodes and a sink node; (b) correlated companions of each node.

nodes (maximal sleeping node policy) such that the union of the correlated companions of all the active nodes contains all the nodes in a WSN. For simplicity, we consider that all the nodes consists homogeneous hardware so that for any node, energy consumption in active mode is equivalent. As a result, a minimum number of active nodes can ensure minimum overall energy consumption by the WSN.

The problem is that if a random node is selected as active, it may not be able to communicate directly or through multihop path with the sink. Therefore, it is required to select a node as active, only if, it can send its sensed data directly or through another active node. The best selection is to find a set of minimum number of active nodes while satisfying the following conditions: (a) they form a connected graph, specifically, all of them can reach the sink node; (b) every dormant node has at least one active companion so that its sensed data can be predicted with a tolerable error bound. Once a node is selected as active, it consumes more energy than a dormant node. Naturally, it should not operate as an active node for a long time; otherwise it will drain out all of its energy quickly. To ensure fairness in energy consumption, a new set of active (and dormant) nodes are selected before every operational period.

A. Active node selection problem: combinatorial optimization

We represent the network as a node- and link-weighted, undirected graph $G = (S, L)$, where S is the set of all nodes in the network and L is the set of links in the network. A link between a nodes i and j exists if they are within the transmission range of each other. The cost of this link, denoted by $e_l(i, j)$ or $e_l(j, i)$, is equal to the energy consumption for transmitting a packet from one node to the other. Each node is also associated with a non-zero cost (node-weight), which is equal to the energy consumption based on its CPU and sensing activity. For a node i, let $e_a(i)$ be the amount of energy consumed in active state and $e_d(i)$ be the amount of energy consumed in dormant state. We now formally define the problem as follows.

Definition 1. *Active node selection: Given a node- and link-weighted undirected graph $G = (S, L)$ with n nodes and a collection of n subsets $\{S_1, ..., S_n\}$ such that S_i is the correlated companions of node i and $\bigcup_{i=1:n} S_i = S$, find the minimum cost subgraph $G' = (A, L')$, where $A \subseteq S$ and $L' \subseteq L$. Here cost of G' is defined as, $\sum_{i \in A} e_a(i) + \sum_{(i,j) \in L'} e_l(i, j)$. The minimum cost subgraph, G, has to satisfy the following*

conditions – (i) the union of the correlated companions of the nodes in A is S, i.e., $\bigcup_{k\in A} S_k = S$, and (ii) G' is a connected graph.

Theorem 1. *Active node selection (decision) problem is NP-complete. The optimization version of the problem is NP-hard.*

Proof. It is easy to show that active node selection $\in$ NP, since a nondeterministic algorithm only needs to find a subgraph A and then verify in polynomial time that (i) the union of the correlated companions of the nodes in A is equal to S (complexity is O(n)), and (ii) the nodes in A are connected (complexity of depth first search (DFS) is O(n)). Therefore, active node selection (decision) problem is NP. The problem can be reduced to the set-cover problem. Without the additional constraint that the minimal subset of nodes should form a connected subgraph, the problem is same as the set-cover problem. This means the problem is as hard as the set cover problem, if not harder. As the set-cover problem is known to be NP-hard [34], the Active node selection problem is also NP-hard. Because of this direct equivalence with the Set cover problem, we have omitted detailed proof. □

The primary goal is to follow a maximal sleeping node policy to reduce energy expenditure of a WSN. This leads to a compromise in sensed data accuracy. Since we exploit the high data correlation feature of the nodes, the information provided by VSF can be close enough to the ground truth. By information, we refer to the whole data set, which constitutes the actual sensed values from the active nodes and predicted values from the dormant nodes. The following proposition tries to address the deviation of the predicted data compared to the ground truth *vis-à-vis* the correlation amongst the sensors.

Proposition 1. *The total distortion D of information, due to selection of G' instead of G, is a function of $|A|$ and the correlation among the sensed data of the nodes. For high correlation, D is small.*

Proof. Let us assume Δ is the total information communicated by all the nodes of S. Δ can be estimated from the joint entropy of the observations of the nodes and is expressed as,

$$\begin{aligned}\Delta &= H(S) \qquad (10)\\ &= H(S_1, S_2, ..., S_n) \quad | \quad \bigcup_{i=1:n} S_i = S\\ &= H(S_1) + H(S_2) + ... + H(S_k) \quad | \quad \bigcup_{k\in A} S_k = S.\end{aligned}$$

S_k represents the set of correlated companions of node $k \in A$. The observation from all the nodes in S_k can be represented by a random variable Y_k. The information obtained by observing Y_k can be measured using the entropy of Y_k, i.e., $H(Y_k)$, further, $H(S_k) = H(Y_k)$. However, only node k is selected as an active node from S_k, and the distribution, sampled by node k, can be described by another random variable X_k. $H(Y_k)$ can be expressed with respect to X_k as follows [35].

$$H(Y_k) = I(Y_k, X_k) + H(Y_k|X_k). \qquad (11)$$

The mutual information $I(Y_k, X_k)$ measures the gain of information about Y_k by observing X_k. On the other hand, the conditional entropy $H(Y_k|X_k)$ denotes the extra information required to estimate $H(Y_k)$. In this context, the conditional entropy $H(Y_k|X_k)$ is the estimation of distortion D_k for observing X_k instead of Y_k. The distortion D_k can be calculated from the correlation of nodes of S_k. The correlation ρ_k can be expressed as follows [35],

$$\rho_k = I(Y_k, X_k)/H(Y_k). \qquad (12)$$

From the equations (11), and (12), the distortion can be calculated as,

$$\begin{aligned}D_k &= H(Y_k|X_k) = H(Y_k) - I(Y_k, X_k) \qquad (13)\\ &= H(Y_k) - \rho_k H(Y_k)\\ &= H(Y_k)(1 - \rho_k)\end{aligned}$$

It can be observed that, from Eq.(13), for high value of ρ_k, D_k is small. □

The total distortion D can be calculated as,

$$D = \sum_{k=1}^{|A|} D_k = \sum_{k=1}^{|A|} H(Y_k)(1 - \rho_k). \qquad (14)$$

So, the total distortion D depends on the size of A or $|A|$, and very small for high ρ_k, $\forall k \in A$.

B. A heuristic algorithm for active node selection

As explained in Section IV-A, the Problem 1 on hand is NP-hard. An optimal solution is not guaranteed to be found in polynomial time. To evolve a polynomial time solution for the active node selection problem, we propose a heuristic, called *active node selection heuristic* (ANSH). Apart from the connectivity constraint, an incautious active node selection may render some of the nodes to be active more often than the rest. Consequently this leads to run down of energy of these nodes faster than the rest. This might cause a disconnected network and reduces the lifetime of the network. Therefore, a careful active node selection should not only consider minimal energy consumption for a WSN, but also balance the energy expenditure across the nodes.

We separate the active node selection from the route finding process. To find the shortest path from each node to the sink node, we depend on the underlying routing protocol. The shortest path tree helps to derive the parent-child relationships for the nodes in the network. Now, if we ensure that the parent node of every active node is also selected as active, the connectivity can be ensured. Note that the sink node is always selected to be active.

The active node selection process, described in Algorithm 2, marks each node for either of the roles - active ($state = 1$) or dormant ($state$ = -1). Initially, all the nodes are marked as undecided ($state = 0$). Then, the algorithm selects a node as active that has maximum residual energy among all the nodes. This selected node is then marked as active. To ensure connectivity, its parent node (also the parent's parent) needs to be marked as active. A recursive function is called to mark a node and its parent node as active (*markAsActive*). When a node is marked active, all its correlated companions

Input: Residual energy $(e^1_{res}, ..., e^n_{res})$, parent node $(p^1, ..., p^n)$, and the correlated companions $(C^1, ..., C^n)$ of every node.
Output: Nodes are split into two subsets: Active $(state = 1)$ and Dormant $(state = -1)$.

```
state[1..n] ← 0; cover ← 0;
while (cover < n) do
    e^i_max ← 0;  i_max ← 0;
    for (i = 1 : n) do
        if (e_max < e^i_res) then
            e_max ← e^i_res;  i_max ← i;
        end
    end
    cover ← markAsActive(i_max);
end
if (state[i] == 0) then
    cover ← cover + 1;
end
state[i] ← 1;
if (p^i ≠ sink AND state[p^i] ≠ 1) then
    cover ← markAsActive(p^i);
end
for (k = 1 : n) do
    if (state[k] == 0 AND s^k ∈ C^i) then
        state[k] ← −1;
        cover ← cover + 1;
    end
end
return cover;
```

Algorithm 2: *ANSH:* Algorithm for active node selection.

can be marked dormant. However, some of these correlated companions might be marked active previously. Thus, only the undecided correlated companions of an active node are marked as dormant. After this, next active node is selected based on the maximum residual energy among nodes that are still undecided. The node selection process continues, until all the nodes are marked either active or dormant. Note that if a node is marked dormant, one of its highly correlated nodes is guaranteed to be marked as active.

To equalize the energy expenditure among the nodes, we mark nodes active based on maximum residual energy. If a node is marked active, its residual energy will be lesser in the next round of selection as compared to the dormant nodes. As a result, a new set of nodes will be marked as active. This ensures fair energy expenditure among the nodes in the network. If multiple nodes have the same residual energy, one random node among them is marked active.

V. Evaluation

We have evaluated virtual sensing framework based on the temperature and humidity data sets collected from the Intel Lab deployment [13] and the GreenOrb deployment [36] along with some computer generated data sets. In this section, we present some of the results based on the temperature dataset obtained from these two real-world deployments. The summary of the deployments are shown in Table II. Note that we consider only those nodes from these deployments that have sufficient data points.

TABLE II: Deployment summary.

deployment	nodes[1]	data points/node	sensing interval	type
Intel Lab [13]	51	5000	31 sec	Indoor (office)
GreenOrb [36]	220	432	10 min	Outdoor (woods)

TABLE III: Current and energy consumption by a tmote sky node for its various activity. Energy consumption is calculated based on 31s sensing interval.

(a) Current consumption

Activity	Current
CPU active	1800 μA
CPU sleep	5.1 μA
Radio idle	20 μA
Radio sleep	1 μA
Radio transmit	17.4 mA
Radio receive	19.7 mA

(b) Energy consumption

Activity	Energy
CPU active	9.33 mJ
CPU sleep	0.57 mJ
Packet TX	4.74 mJ
Packet RX	5.23 mJ
Temp. sensing	0.33 mJ

We divide the evaluation into two parts. In the first part, we compare performance of the VSF, and how it is affected by changing the various parameters. Since choice of a proper set of values for the parameters is highly dependent on the data set, we provide an indication on how to set those parameters. In the second part, we compare performance of VSF with respect to some of the existing data collection techniques. Though the results described in this article are produced using a Matlab implementation, we have developed a working prototype of VSF using Java.

We used three metrics to evaluate performance of VSF – (i) the number of data packet transmission, (ii) accuracy of prediction, and (iii) energy expenditure by the nodes. We use root mean squared error (RMSE) to measure accuracy of the prediction. To compute energy expenditure by the nodes during the simulation period, we have considered the Tmote-sky [37] for energy consumption measurement. Various current and energy consumption values are summarized in Table III.

A. Performance of VSF

In VSF, nodes either remain active or dormant based on their assigned roles. As mentioned earlier, a node's role (active or dormant) changes over time. Fig. 7 shows the predicted data using VSF along with the sensed data for a total 5000 data points for Node 1 of the IntelLab deployment (upper plot). During this period, the node was assigned both the roles for some time. From this figure, it is clear that the predicted data closely follows the actual sensed data. The lower part of the plot shows the prediction error (absolute error values). Most of the time, the prediction error is within the limit of 1°C. There exists some occasional outliers, but the overall VSF error bound is suitable for many WSN applications. Similar

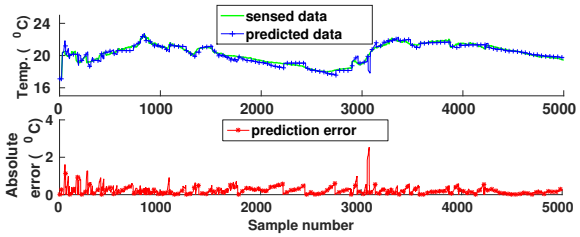


Fig. 7: Sensed and predicted data of Node 1 over a period, where it was assigned both active and dormant role some time. The length of training, operational, and revalidation periods are set to 40, 20, and 10 data points repectively.

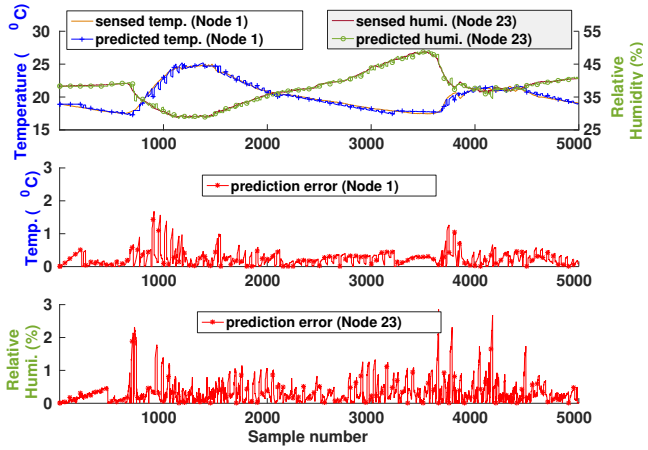


Fig. 8: Temperature data from Node 1 is used to predict humidity data from Node 23 and vice-versa.

results have also been observed for all the remaining nodes of both the deployments.

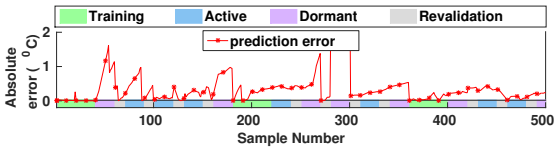
B. Heterogeneous Virtual Sensing

As mentioned earlier, VSF framework is not only applicable when all nodes have homogeneous sensor; it can be applied to any type of sensor data that show high correlation irrespective of data type. But it cannot be applied to any such sensed data that can change abruptly without showing any trend over time, e.g., indoor light (artificial light) can be completely different with a small time interval.

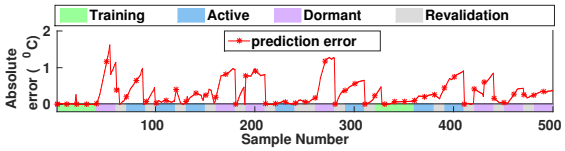
We have applied VSF on two heterogeneous data types. As an illustration, we have used temperature data from Node 1 and humidity data from Node 23 (see Fig. 8). We have preselected them, as we know that they have a very high data correlation. During the process, either of the node is selected as active. That means, temperature data is predicted based on the humidity data and vice-versa.

C. Adaptation to data correlation change

In VSF, all nodes go through the three operating modes, i.e., training, operational (active/dormant) and revalidation. As mentioned earlier, the operating mode of the nodes are decided based on the data correlation among the nodes. A

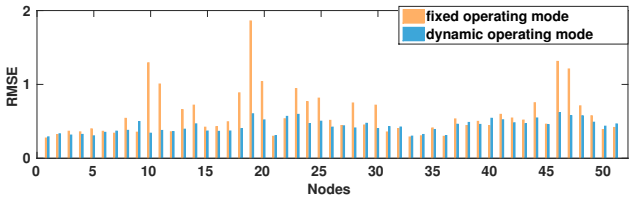


(a) Re-training at a fixed interval (after 5 revalidation period).

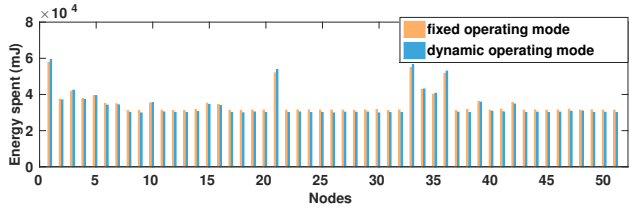


(b) Decision about re-training is taken dynamically based on data correlation.

Fig. 9: Various operating modes of Node 1 in VSF for the first 500 data points. During the operational period, a node can be either active or dormant.



(a) Prediction error for every node



(b) Energy spent by each node

Fig. 10: A comparison between fixed and dynamic re-training schedule. Average prediction error for a node is 0.59 °C and 0.43 °C for the fixed and dynamic schedules respectively. Similarly, average energy consumption is 34.64 J and 33.90 J respectively.

small snapshot (500 data points) of the operating modes for Node 1 is shown in Fig. 9b. At the beginning, Node 1 enters the training period like all other nodes in the deployment and they are associated with a low threshold active virtual sensor (AVS). The length of the training period is set to 40 data points. At the end of this period, the operational period starts (20 data points). During this operational period, Node 1 is assigned dormant. Thus, it is associated with a dormant virtual sensor (DVS).

After the operational period, Node 1 enters the revalidation period as all other nodes in the deployment. Similar to the training period, nodes are also associated with an AVS during this period. After the revalidation period, usually another operational period starts with new role for the nodes (a different set of active and dormant nodes). In this case, Node 1 is assigned as active and is associated with an AVS. Presuming an operational period depends on the prevalence of correlation

pattern among the nodes as it were during the last training or revalidation period. If a smaller fluctuation is noticed, a revalidation period can proceed to another revalidation period (after data point 250). In case of a huge data correlation deflection, a new training period is started (after data point 320).

The dynamic adaptation of the operating modes lead to a better prediction accuracy and lesser energy consumption by the nodes, as compared to a fixed schedule for the operating modes (Fig. 9b). The lengths of the operating modes are same in both the cases, i.e., 40, 20 and 10 data points for the training, operational, and revalidation period respectively. But, for the fixed schedule case, a revalidation period is always followed by a operational period, and after 5 operational period (and 4 revalidation period in-between) the training period is initiated. In this case the revalidation period is used only for selecting a set of active node for the next operational period.

The dynamic adaptation of the operating modes is a better choice than the fixed schedule as it is clear from Fig. 10. Using 5000 sensed data from the 51 nodes of the IntelLab deployment, we found that the prediction error (RMSE) for most of the nodes is higher for fixed schedule case than the dynamic adaptation case (Fig. 10a). At the same time, energy consumption by most of the nodes for the fixed schedule case is also higher than the dynamic case (Fig. 10b). The energy consumption for the fixed schedule case can be reduced by increasing the interval between two training periods. But, this would only increase the prediction error. As a result, the dynamic scheduling of the operating modes can tackle the dynamic data correlation pattern among the nodes (over time).

D. Deciding the lengths of the operating periods

The lengths of the operating periods play a role in prediction error, and number of data transmissions (or energy consumption by the nodes) within the network. To reduce energy consumption, as many nodes as possible need to be kept in dormant mode for the maximal amount of time. Thus, a longer operational period can lead to a overall reduction in energy consumption as most of the nodes are kept dormant. However, longer operational period can lead to higher prediction error for the dormant nodes. These can be complemented by longer revalidation period. But, again, longer revalidation period leads to higher energy consumption by the nodes as all the nodes are accompanied by active virtual sensor (AVS) during this period.

Fig. 11 shows the average prediction error (RMSE) and average energy spent by a node for 5000 sensing intervals (data points). Four different lengths (20, 40, 60 and 80 data points) of training periods (T_p) are used. For each length of the training period, four different operational periods (O_p) of 10, 20, 30 and 40 data points are used. The revalidation periods (R_p) is kept fixed to 10 data points. From the figure, it is clear that energy expenditure is lower for the smaller training period and higher operational period, but it lacks prediction accuracy of the sensed values. On the other hand, prediction accuracy increases with the cost of more energy expenditure, i.e., longer training period and smaller operational period (more data transmissions within the network).

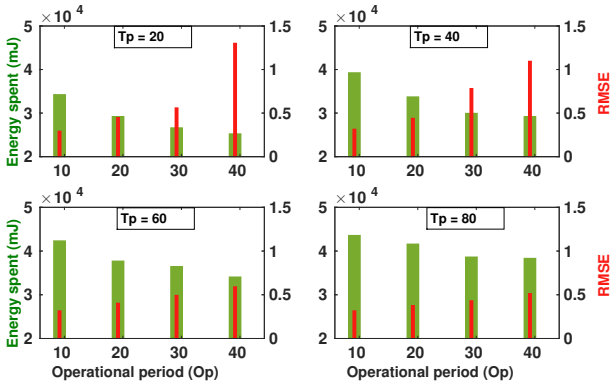


Fig. 11: Average prediction error and energy consumption per node for various lengths of the operating periods based on the IntelLab deployment data.

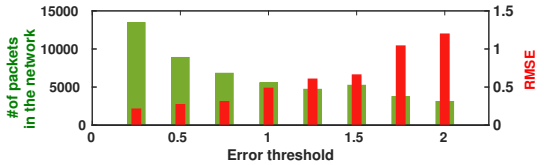


Fig. 12: Tolerable prediction error has an immediate impact on the accuracy of VSF mechanism, as well the on the energy consumption of the nodes. Setting a tolerance level (threshold) depends on the application requirement.

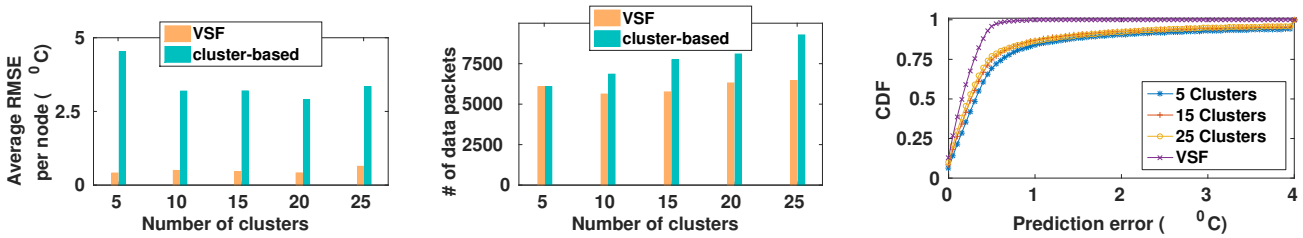
The length of these periods can be dependent on the deployment and their sensing intervals. Though the tendency should be larger operational period and smaller revalidation period, a suitable length of these periods can be selected in such a way that the operational periods are small enough to contain any possible drift of the data correlation among the nodes. To maintain minimal energy expenditure and high data accuracy, for the IntelLab deployment, a suitable $< T_p, O_p, R_p >$ combination is 40, 20, and 10 data points respectively (Fig. 11).

E. Effect of the error threshold

In an active node, not all the sensed data is transmitted. If the prediction error is going to be within a predefined error threshold (tolerable error), the node avoids transmitting the data. If this threshold is set to a higher value, the significant number of data transmission can be avoided. But, this can decrease the accuracy of the overall data set. Fig. 12 shows how the prediction error increases when the error threshold increases. On the other hand, when the error threshold is lower, the data accuracy increases; but this also increases the number of data packets within the network, as well as energy consumption of the nodes. So, setting up this threshold is highly dependent on the application requirement.

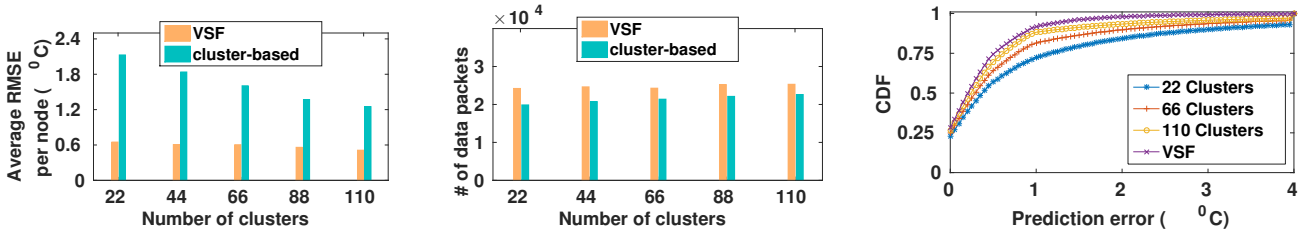
F. Effectiveness of correlation based node grouping rather than collocation

We have argued for a system that node correlation is calculated based on their sensed data only, irrespective of



(a) average RMSE per node (b) total number of data packet within the network (c) CDF of prediction error

Fig. 13: IntelLab dataset: Comparison of VSF with geographical collocation-based clustering.



(a) average RMSE per node (b) total number of data packet within the network (c) CDF of prediction error

Fig. 14: GreenOrb dataset: Comparison of VSF with geographical collocation-based clustering.

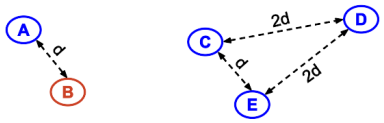


Fig. 15: Node correlation based on data and geographical collocation.

their geographical collocation. This strategy provides a better accuracy for data estimation.

Proposition 2. *Correlation based on sensed data is superior than the geographical collocation based, when data estimation is done exploiting the correlation.*

Proof. Suppose, the data correlation among the nodes is fixed and known a priori. To maintain high accuracy of the predicted data using VSF, member nodes in a particular group need to be highly correlated, i.e., their pairwise correlation is above certain correlation threshold (described as correlated companions in Section III-A). Now, the nodes grouped together based on their geographical collocation. As shown in Fig. 2, nodes in close proximity need not always show high data correlation. If two nodes are assumed to be highly correlated just because they are geographically collocated, but not in reality, the prediction accuracy will be less.

Let us take an example. In a WSN, consider nodes A and B are located very closed to each other, but have very poor data correlation, and nodes C and D have very high data correlation even though they are positioned apart from each other (Fig. 15). If geographical collocation based correlation is assumed, node A can act as an active companion of node B. But, the resulting data prediction would have lower accuracy. On the other hand, nodes C and D cannot be each other's active companion. Now, VSF would not assume A as an active companion of B, thus, lower prediction accuracy can be avoided. Also, VSF can assign C as an active companion of D, thus, significant energy savings can be achieved by keeping node D in dormant state, while its data can be predicted with high accuracy. In case, two close by nodes are highly correlated (nodes C and E), both the methods will achieve higher prediction accuracy as well as save some energy. Thus, we can conclude that correlation based on sensed data is superior than the geographical collocation based correlation when data estimation is done based on the correlation. □

To further prove our claim, we applied experimental comparison based on the IntelLab and GreenOrb datasets. The node location for both the deployments are known. We create cluster of nodes based on their geographical collocation, where nodes within a cluster are assumed to be highly correlated. The clusters are formed based on their Euclidean distance with each other. Instead of considering a fixed number of clusters within the network, we use different number of clusters. Thus, the number of nodes per cluster (cluster size) is varied. We use five different cases where the cluster sizes are equal to 10%, 20%, 30%, 40%, and 50% of the total nodes in the network. Thus, the number of clusters in Intellab deployment are 5, 10, 15, 20, and 25, respectively. Similarly, in GreenOrb deployment, the number of clusters are 22, 44, 66, 88, and 110, respectively.

From Fig. 13a and Fig. 14a, we can conclude that the prediction accuracy for VSF is much higher than the geographical collocation based clustering method, as the average RMSE per node is much lower in VSF than the cluster-based method. The cluster-based method seems to be equivalent (or superior) as compared to VSF in terms of the number of data packet within the network (Fig. 13b and Fig. 14b). The reason behind this is that cluster-based method blindly selects an active node from a cluster and keeps the remaining node in dormant state. So, the number of active nodes are fixed and there are only n

TABLE IV: Performance comparison between VFS and LMS-based method [20].

	IntelLab			GreenOrb		
Method	data packets	tx reduction	error (°C)	data packets	tx reduction	error (°C)
LMS-based	75408	71%	0.5321	29673	69%	0.4669
VSF	4947	98%	0.3593	19547	80%	0.7107

active nodes if there are n clusters. On the other hand, VSF assigns a node dormant if any of its correlated companion is assigned active. As the node correlation changes over time, the number of active nodes can also change over time. As a result, there can be more active nodes during a certain operational period, and thus, more number of data packets within the network. Naturally, this ensures high accuracy of predicted data for the dormant nodes. Please note that VSF also uses Autocorrelation to reduce the number of data transmission. Thus, over a long period the total number of data packets within a network using VSF is equivalent to the geographical collocation-based clustering (if not significantly less).

To further illustrate the data accuracy of VSF prediction mechanism, we show cumulative distribution function of the prediction error in Fig. 13c and Fig. 14c. From these figures, it is clear that VSF not only achieves lower prediction error on average, but it has a very fewer instances of high prediction error. The prediction error is bound by a lower value, where there is a few outliers, i.e., 95% cases prediction error is within 0.5°C. On the other hand, the geographical collocation based clustering and prediction mechanism has a significantly high number of instances with higher prediction error (error within 0.5°C for less than 75% of cases).

G. Comparison with LMS-based method

In this section, we compare VSF with the LMS-based sensor data estimation method by Santini *et. al* [20]. The proposed technique utilizes the inherent autocorrelation property of the sensed data by a node. Each node possess an autocorrelation-based predictor, and makes a decision about transmitting the sensed data if the predictor is not able to predict the sensed value within a tolerable error. Though this method reduces the data transmission significantly while ensuring a high accuracy of the estimated data, it does not utilize the cross-correlation among the nodes. On the other hand, by utilizing cross-correlation among the nodes along with the autocorrelation, VSF achieves a higher reduction of data transmission within the network. The summary of comparison between VSF and LMS-based method is shown in Table IV.

The results establishes the fact that VSF can outperform LMS-based method in terms of data traffic reduction, and thus, energy consumption by the nodes. Though the prediction accuracy for the GreenOrb deployment using VSF is a bit lower than the LMS-based method, the prediction error is within a tolerable error bound (1°C). A further inspection of RMSE at the individual node level establishes the fact that the prediction accuracy achieved by VSF is comparable with the LMS-based method.

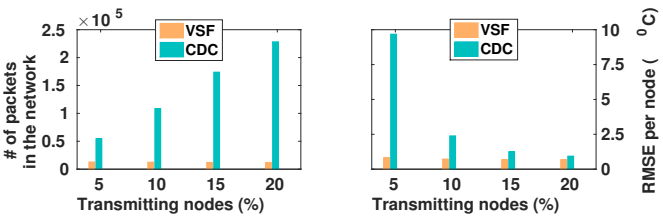


Fig. 16: IntelLab dataset: Total number of data packet within the network for VSF as compared to CDC.

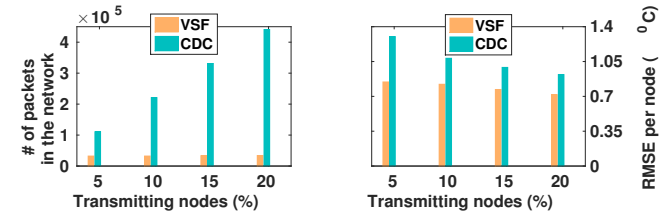


Fig. 17: GreenOrb dataset: Total number of data packet within the network for VSF as compared to CDC.

The improvement of VSF is in terms of energy consumption by the nodes as it was already evident from the data transmission reduction (Table IV). VSF conserves energy by not only reducing the data transmission, but it achieves higher energy savings by keeping the nodes in deep sleep for higher amount of time.

H. Comparison with compressed sensing techniques

Compressed sensing based data collection techniques utilizes the sparsity in the sensed data to reduce the amount of data transmission within the network. As opposed to our method, they do not consider correlation among the nodes. However, compressed sensing based methods also adopt a selective data transmission by the node, thus, they resemblance similarity with VSF.

Liu *et. al* proposed a compressed sensing based sensor data collection method, called CDC [8]. In this method, each node transmits its sensed data with a predefined probability p. As a result, the sink node would receive np packets (on an average) in each sensing rounds. Finally, the whole data set for all the nodes is reconstructed based on the partial data set using compressed sensing. The method works based on the assumption that the sensed data have a sparse representation when the data collected from all the nodes are converted to some other domain (e.g. frequency domain). If the data sparsity is very high, highly accurate reconstruction can be achieved with fewer samples. In that case the transmission probability can be set to a lower value. On the other hand, if the data sparsity is less, data reconstruction from a few samples can lead to larger inaccuracy. Naturally, a larger transmission probability is required so that sufficiently large samples can be collected and highly accurate reconstruction is achieved.

We compared the performance of VSF with CDC for various values of transmission probability (p). We tried 4 different values of p such that the average number of samples (or active nodes in case of VSF) are 5%, 10%, 15%, and 20% of the

total number of nodes in the deployment. From Fig. 16 and Fig. 17, it is clear that the prediction error for CDC is much higher higher when there is only a few sample are used for the reconstruction. As the number of samples increases, the prediction accuracy for CDC also improves and goes closer to that of VSF. But, this increases the number of data packet transmission within the network (Fig. 16 and Fig. 17), which will also increase the energy consumption of the network.

VI. Discussion

We briefly discuss some of the trade-off while using VSF. Since we are focusing on sensed data and predicting the data from sensor nodes using correlations, VSF can only be applicable for a WSN. Note that we are able predict sensed data of any two nodes that are correlated irrespective of their geographical location. However, this assumption holds for nodes that are part of the same system.

Even though VSF tries to capture the dynamics of data correlation, the prediction accuracy can drastically be affected when the data correlation is highly dynamic. In those cases, a smaller operational period may be more suitable. Note that we have defined operational period in terms of number of sensing interval (time difference between two successive sensing) rather than absolute time. If the monitored region in a WSN is highly dynamic, the nodes are more likely to sense more frequently. As a result, within a small time period many sensing events will occur and various operating states (training, operational, and revalidation) would also change quickly. Thus, quick correlation change would have lesser effect on the prediction accuracy, if the correlation among the nodes withhold during the operational period.

The suboptimal solution, using the heuristic, for selecting the active nodes may result in choosing more than one active companion for a particular dormant node. In such a case selection of a particular active companion node may improve the prediction accuracy but this needs an exhaustive search. Furthermore, utilizing the sensed data from multiple active companions can also improve the prediction accuracy however, this will increase the complexity of the algorithm. Also, it is not always guaranteed to yield higher prediction accuracy. Thus, we adopted a simplistic approach, where only one active companion is chosen for each dormant node at the beginning of a operational period.

VII. Conclusions

There are many schemes and protocols to increase the lifetime of a WSN. In this article, we introduced virtual sensing framework (VSF), which predicts multiple consecutive sensor data while some of the sensors remain dormant. We have utilized the inherent correlation amongst the sensor data without having: (i) any *a priori* knowledge of the statistics of the data; (ii) location of the sensor nodes and, (iii) type of the physical parameters observed. A case in point is predicting temperature with a light sensor within a tolerable error bound.

The prediction technique of the virtual sensors adapts to the changes in the sensor data. Using VSF activity reduction technique, we have achieved a significant improvement in energy savings compared to other similar techniques while maintaining sufficiently high accuracy of the sensor data. Our maximal sleeping node policy can reduce the overall energy consumption of a WSN. However, the formulated minimum active node selection problem is shown to be an NP-hard problem. Thus, we provided a heuristic algorithm to find minimum number of active nodes at any instance. We have reported around 98% and 79% of data traffic reduction when VSF activity reduction scheme is used on the IntelLab and GreenOrb datasets, respectively. Our technique will be useful when large number of sensors are deployed in near future with the advent of Internet of Things (IoT) paradigm.

References

[1] D. Moss and P. Levis, "Box-macs: Exploiting physical and link layer boundaries in low-power networking," *Computer Systems Laboratory Stanford University*, pp. 116–119, 2008.

[2] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.

[3] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *Commun. Lett., IEEE*, vol. 12, no. 6, pp. 411–413, 2008.

[4] M. Shan, G. Chen, D. Luo, X. Zhu, and X. Wu, "Building maximum lifetime shortest path data aggregation trees in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 11, no. 1, Article 11, 2014.

[5] Q. Zhao and M. Gurusamy, "Lifetime maximization for connected target coverage in wireless sensor networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 16, no. 6, pp. 1378–1391, 2008.

[6] J. A. Torkestani, "An adaptive energy-efficient area coverage algorithm for wireless sensor networks," *Ad Hoc Networks*, vol. 11, pp. 1655–1666, 2013.

[7] L. Xiang, J. Luo, and A. Vasilakos, "Compressed data aggregation for energy efficient wireless sensor networks," in *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2011 8th Annual IEEE Communications Society Conference on*. IEEE, 2011, pp. 46–54.

[8] X.-Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. Vasilakos, and M.-Y. Wu, "Cdc: Compressive data collection for wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.

[9] B. Wang, "Coverage problems in sensor networks: A survey," *ACM Computing Surveys (CSUR)*, vol. 43, no. 4, p. 32, 2011.

[10] H. Gupta, V. Navda, S. Das, and V. Chowdhary, "Efficient gathering of correlated data in sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 1, pp. 4:1–4:31, Feb. 2008. [Online]. Available: http://doi.acm.org/10.1145/1325651.1325655

[11] S. He, J. Chen, D. Yau, and Y. Sun, "Cross-layer optimization of correlated data gathering in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 11, pp. 1678–1691, Nov 2012.

[12] E. Karasabun, I. Korpeoglu, and C. Aykanat, "Active node determination for correlated data gathering in wireless sensor networks," *Computer Networks*, vol. 57, no. 5, pp. 1124–1138, 2013.

[13] S. Madden, "Intel berkeley research lab data," http://db.csail.mit.edu/labdata/labdata.html, 2004, [accessed 30-Nov-2013].

[14] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," *Cybernetics, IEEE Transactions on*, vol. 43, no. 5, pp. 1473–1483, 2013.

[15] C.-T. Cheng, C. K. Tse, and F. C. Lau, "A clustering algorithm for wireless sensor networks based on social insect colonies," *Sensors Journal, IEEE*, vol. 11, no. 3, pp. 711–721, 2011.

[16] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mob. Comp., IEEE Trans. on*, vol. 3, no. 4, pp. 366–379, 2004.

[17] J. Crowcroft, M. Segal, and L. Levin, "Improved structures for data collection in wireless sensor networks," in *Proceedings of INFOCOM. Toronto, Canada*, 2014, pp. 1375–1383.

[18] C. Joo and N. B. Shroff, "On the delay performance of in-network aggregation in lossy wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 22, no. 2, pp. 662–673, 2014.

[19] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden, "Distributed regression: an efficient framework for modeling sensor network data," in *IPSN '04*. New York, NY, USA: ACM, 2004, pp. 1–10. [Online]. Available: http://doi.acm.org/10.1145/984622.984624

[20] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *INSS 2006*, 2006, pp. 29–36.
[21] H. Jiang, S. Jin, and C. Wang, "Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks," *Parallel and Distributed Systems, IEEE Trans. on*, vol. 22, no. 6, pp. 1064–1071, 2011.
[22] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong, "Approximate data collection in sensor networks using probabilistic models," in *IEEE Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*. IEEE, 2006, pp. 48–48.
[23] R. Cristescu and M. Vetterli, "On the optimal density for real-time data gathering of spatio-temporal processes in sensor networks," in *IPSN'05*. IEEE, 2005, p. 21.
[24] Z. Quan, W. J. Kaiser, and A. H. Sayed, "A spatial sampling scheme based on innovations diffusion in sensor networks," in *IPSN '07*. New York, NY, USA: ACM, 2007, pp. 323–330. [Online]. Available: http://doi.acm.org/10.1145/1236360.1236402
[25] M. C. Vuran and I. F. Akyildiz, "Spatial correlation-based collaborative medium access control in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 316–329, April 2006.
[26] S. He, J. Chen, D. K. Yau, and Y. Sun, "Cross-layer optimization of correlated data gathering in wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 11, pp. 1678–1691, 2012.
[27] B. Cheng, Z. Xu, C. Chen, and X. Guan, "Spatial correlated data collection in wireless sensor networks with multiple sinks," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2011, pp. 578–583.
[28] W. Liang, X. Ren, X. Jia, and X. Xu, "Monitoring quality maximization through fair rate allocation in harvesting sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1827–1840, 2013.
[29] Y. Li and L. E. Parker, "Nearest neighbor imputation using spatial-temporal correlations in wireless sensor networks," *Information Fusion*, vol. 15, pp. 64–79, 2014.
[30] C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in *Proceedings of the 15th annual international conference on Mobile computing and networking*. ACM, 2009, pp. 145–156.
[31] L. Xiang, J. Luo, and C. Rosenberg, "Compressed data aggregation: energy-efficient and high-fidelity data collection," *Networking, IEEE/ACM Transactions on*, vol. 21, no. 6, pp. 1722–1735, 2013.
[32] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos, "Hierarchical data aggregation using compressive sensing (hdacs) in wsns," *ACM Transactions on Sensor Networks (TOSN)*, vol. 11, no. 3, p. 45, 2015.
[33] J. R. Taylor, "An introduction to error analysis: The study of uncertainties in physical measurements author: John r. taylor, publisher," 1996.
[34] R. M. Karp, *Reducibility among combinatorial problems*. Springer, 1972.
[35] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*, 2nd ed. John Wiley & Sons, 2006.
[36] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? a measurement study on greenorbs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 24, no. 10, pp. 1983–1993, 2013.
[37] T. S. D. Sheet, "Moteiv, san francisco, ca, 2006," 2004.