Edge Intelligence Framework for Data-Driven Dynamic Priority Sensing and Transmission

Sushmita Ghosh, Swades De, Shouri Chatterjee, and Marius Portmann

***Abstract*—Owing to the limited storage capacity, battery-powered wireless sensor nodes often suffer from energy sustainability. To optimize the energy consumption of a multi-parameter sensor hub, a novel edge intelligence-based data-driven priority-aware sensing and transmission framework is proposed in this paper. The proposed framework jointly exploits the cross-correlation among the sensing parameters and temporal correlation of the individual sensing signals to find an optimal active sensor set and optimal sampling instants of the sensors in the next measurement cycle. The length of measurement cycle is dynamically decided based on the change in cross-correlation among the parameters and the system state. A discounted upper confidence bound algorithm-based optimization function is formulated to find the optimal active sensor set by solving the trade-off among cross-correlation, energy consumption, and length of measurement cycle. The proposed framework uses Gaussian process regressor-based prediction models to estimate the temporal and cross-correlated parameters of the active and inactive sensor set, respectively. The sampling interval of each active sensor is dynamically adapted based on the temporal prediction error. Extensive simulations are performed on air pollution monitoring dataset to validate the efficacy of the proposed framework in both real-time and non-real-time applications. The proposed algorithm saves up to 41% energy and 32% bandwidth with 68% data accuracy compared to the existing competitive frameworks for non-real-time systems. The proposed framework also identifies the time-critical sensing scenarios with 98% accuracy.**

***Index Terms*—Temporal correlation, cross-correlation, adaptive sensing, edge intelligence, optimal sensor set, optimal sampling instants, priority-aware sensing.**

I. Introduction

The evolving Internet-of-Things (IoT) technology is expected to connect billions of devices that are used in our daily lives. To sense the environmental conditions, IoT sensor nodes have enormous applications in today's world and it is growing at a significant rate. To monitor various parameters in the environment, sensor hubs are equipped with multiple sensors. Such a large number of sensor hubs, wirelessly connected to an edge node or a central entity (CE) constitutes a wireless sensor network (WSN). WSNs are used in industrial process monitoring, smart city, smart agriculture, hospital monitoring, border surveillance, etc. Good quality environmental sensors are highly energy consuming. In many cases, sensing energy is higher than communication energy [1]. Hence, the energy efficiency and sustainability of sensor nodes with limited battery capacity are of major concern [2], [3]. In such scenarios managing high energy demand over a massive deployment of sensor nodes is very challenging. To mitigate these problems and increase the energy efficiency of WSNs, various intelligent sensing strategies are considered in the literature. The edge intelligence-based adaptive sensing strategies can reduce the energy consumption at the sensor nodes by shifting the computation/processing complexity at the edge node [4], [5].

The sensing parameters of a multi-sensing node often possess temporal correlation and cross-correlation, which can be exploited at the node-level to predict some parameters from the other parameters. In many applications, the parameter variations are captured and stored at the individual sensor nodes, which are transmitted to the edge node at the end of a period called measurement cycle [6]. These mechanisms reduce the energy consumption in the case of non-real-time systems [7], where the data is not time-critical. However, in the case of real-time systems, the samples need to be transmitted to the edge node immediately after sensing. Hence, energy consumption in a time-critical WSN is still a subject of significant research interest.

A. Related Works

In most of the studies, sensor data collection mechanism is considered as a non-real-time process. The sensors collect samples over a cycle and transmit the stored data to the CE or edge node at the end of the cycle. The works in [1], [8]–[10] are dedicated to node-level analysis, where the sampling rate of a sensor is adapted based on the previously collected time series samples. On the other hand, the works in [4], [11], [12] are dedicated to network-level analysis, where some sensor nodes are activated to collect samples while the other nodes remain inactive in the network.

In [8], three different adaptive sampling algorithms were proposed for node-level analysis. The first one uses the Anova model and Bartlett test, the second and third methods use the Jaccard similarity function and the Euclidean distance function respectively to find the new sampling rate based on the previously collected samples. These algorithms were applied on temperature and humidity sensors independently. The adaptive sampling algorithm, proposed in [9] uses Kalman Filtering based estimation technique to adapt the sampling interval based on the estimation error.

The work in [10] proposed an adaptive sensor selection mechanism that exploits the cross-correlation among the sensing parameters of a sensor hub to find an optimal active sensor set at each measurement cycle based on the cross-correlation

S. Ghosh, S. De, and S. Chatterjee are with the UQ-IITD Academy and Department of Electrical Engineering, Indian Institute of Technology Delhi, New Delhi, India (e-mail: sushmita.ghosh@uqidar.iitd.ac.in, swadesd@ee.iitd.ac.in, shouri@ee.iitd.ac.in)

M. Portmann is with the School of IT and Electrical Engineering, University of Queensland, Brisbane, Australia (e-mail: marius@itee.uq.edu.au).

and sensing energy consumptions of the parameters; the sensors collect samples at the Nyquist sampling rate.

A sparse Bayesian learning (SBL)-based adaptive sensing strategy was proposed in [11], where some sensors in the network are selected over a measurement cycle based on the spatio-temporal correlations among the sensing signals. The sensor nodes were considered to have a single sensor to monitor a single parameter in the environment. To increase the energy efficiency of densely deployed multi-sensing networks, an adaptive sensing framework was proposed in [4]. The algorithm in [4] exploited the spatio-temporal correlation and cross-correlation among the parameters to find an optimum number of sensors to be activated in the next measurement cycle. Similarly, a Compressive sensing algorithm, proposed in [12] selects some sensor nodes from a large set of nodes in a heterogeneous sensing environment.

A Gaussian process regression (GPR)-based prediction model was proposed in [13] to predict spatio-temporal parameters of a mobile sensor network. According to [14]–[16], GPR can predict the spatio-temporally varying signals with good accuracy.

The adaptive sensing strategies discussed above considered that the sensing process is non-real-time, where the data are transmitted to the edge node at the end of the measurement cycle. However, the works, in [17], [18] considered a real-time system, where the data collected at the sensor node are transmitted immediately to the edge node to monitor the system state. Similarly, a node-level adaptation was proposed in [1] for a real-time snow monitoring application. The algorithm detects the maximum frequency of the sensing signal using fast Fourier transform of the previously collected samples and the new sampling rate is decided based on Nyquist criteria. The collected samples are transmitted to the base station without any delay. Hence, the energy consumption in real-time systems is much higher than that in non-real-time systems.

B. Research Gap and Motivation

As noted above, several studies have been reported on learning-based adaptive sensing to reduce the energy consumption in WSNs. However, in a large-scale deployment scenario, limited battery capacity of a miniature sensor hub limits the network lifetime.

A comparison of the state-of-the-art and the proposed framework is presented in Table I. The considered sampling rate adaptation algorithm in [1], [8], [9], [19] are applicable on individual sensors. The adaptive sensing algorithm in [10], exploits only cross-correlation of the parameters to find optimal active sensor set, where the sensors collect samples at a fixed Nyquist sampling rate. The studies in [11]–[13] considered densely-deployed WSN for monitoring the same parameter, where a subset of sensor nodes is activated in the network based on the spatio-temporal correlation of the sensing parameter and the remaining energy of the nodes. The above algorithms *do not exploit the cross-correlation among multiple sensing parameters in a sensor hub.*

In the SBL-based multi-sensing framework [4], the reconstruction accuracy is achieved by jointly predicting the sensing signals based on the spatio-temporal correlation and cross-correlation among the parameters. However, *it did not consider multi-parameter sensor hub in a typical sparse deployment scenario, such as in an individual home or office environment, where spatial signal information do not exist. Hence, the SBL-based framework is not suitable in such scenarios.*

TABLE I: Comparison of state-of-the-art

State-of-the-art	System model	Sensing Parameters	Correlation exploited	System state	Sampling interval
[1], [8], [9]	node level	single	not exploited	non-real-time	adaptive
[4]	network level	multiple	spatio-temporal and cross	non-real-time	fixed
[10]	node level	multiple	cross-correlation	non-real-time	fixed
[11]	network level	single	spatio-temporal	non-real-time	fixed
Proposed work	node level	multiple	temporal and cross	adaptive	adaptive

So far, none of the works in the literature have exploited both the temporal correlation and cross-correlation of a distributed multi-parameter sensor hub. Also, the existing sensing algorithms consider the systems dealing with either purely real-time applications or purely non-real-time applications. *The sensed signal aware dynamic adaptability of sensing and transmission time-criticality has not been addressed before.*

C. Contributions

We observe that, in real-life, multi-sensing optimization at the individual sensor nodes are of interest in scenarios where for lack of network connectivity and/or security/privacy sensors, such as in ambient parameter sensing in home/office spaces and medical health monitoring, network-level (inter-node) cooperation is not possible. Further, sensing and data reporting in such applications may need to be adapted from non-time-critical ('good' state) to time-critical ('bad' state) depending on the values of some sensed parameters. Thus, motivated by the need for individual node-level multi-sensing and transmission optimization, and dynamic adaptation, a learning-based data-driven priority aware sensing and transmission is proposed in this paper, which exploits the sensing process dynamics of a multi-parameter sensor hub. At every measurement cycle, an optimal set of sensors are activated by exploiting the cross-correlation among the parameters and temporal correlation of the selected sensing signals based on reinforcement learning (RL) techniques [10], [20]. In view of the non-stationary distribution of sensing parameters, discounted upper confidence bound (UCB) algorithm is adopted to find the next optimal set of sensors. The proposed framework uses GPR models to estimate the temporal and cross-correlated parameters for non-real-time and real-time modes of data handling. *To the best of our knowledge, such data-driven dynamic multi-sensing and adaptation has not been considered before in the literature.*

The key contributions and significance of the proposed approach are as follows:

1) A novel edge intelligence-based framework on dynamic priority sensing and transmission to optimize the energy

sustainability of a multi-parameter sensor hub is proposed in this paper. The algorithm exploits the cross-correlation among the parameters and temporal correlation in a selected time sequence of data to find an optimal active sensor set and optimal sampling instants at every measurement cycle.

2) Unlike the prior art, sensing and communication priority of the sensed parameters are dynamically varied from non-real-time (in 'good' state) to time-critical (real-time, in 'bad' state) depending on the value of the sensed parameters. The corresponding GPR models are also developed for the estimation of missing temporal/cross-correlated parameter samples.
3) An optimization function is formulated to find the optimal length of measurement cycle for each active sensor set by exploiting the cross-correlation of the parameters and the 'good' state probability of the sensors that belong to the inactive/sleep set in that cycle.
4) An optimization function is developed to find the optimal sensor set for the next measurement cycle by solving the trade-off among the cross-correlation, energy consumption of the sensors, and the length of measurement cycle using discounted UCB algorithm.
5) The proposed framework detects the time-critical (i.e, 'bad') states of the system with 98% accuracy in reporting the events to the edge node.
6) With $\mathcal{O}(I)$ computational complexity for predicting I samples at the sensor node, the proposed RL-aided intelligent multi-sensing algorithm saves up to 41% node-level sensing energy and 32% communication bandwidth compared to the multi-sensing framework in [4].

D. Paper Organization

Section II presents the system model and problem definition. The characterization of sensing signal dynamics is discussed in Section III, followed by the GPR-based temporal and cross-correlated parameter prediction model in Section IV. The proposed data-driven adaptive sensing framework is explained in Section V, followed by results and discussion in Section VI and conclusion in Section VII.

Notations: $\mathcal{A}$ denotes a set; $\mathbf{A} \in \mathbb{R}^{N\times 1}$ denotes a P dimensional vector, and $\underline{\mathbf{A}} \in \mathbb{R}^{N\times I}$ denotes a matrix of size $N \times I$; and the cardinality of a set $\mathcal{A}$ is denoted as $|\mathcal{A}| = A$.

II. System Model and Problem Definition

The proposed system model is presented in Section II-A, followed by the problem definition in Section II-B.

A. System Model

As shown in Fig. 1, a multi-parameter sensor hub consists of P sensors to monitor P different parameters in the environment. The sensor node is wirelessly connected to an edge node to report the sensed data. The edge node further transmits the data to the cloud. An intelligence algorithm is executed at the edge node to find the optimal parameters of the system based on the data received from the individual sensor nodes

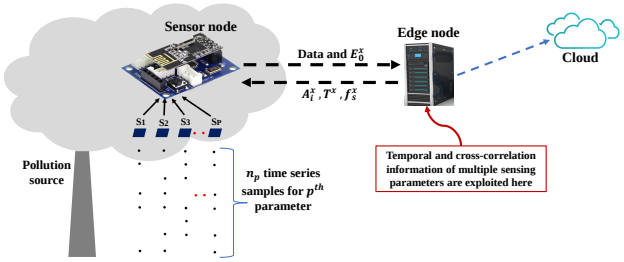


Fig. 1: Smart sensing system model.

and the information is transmitted back to the sensor nodes, as depicted in Fig. 1. Consider a set $\mathcal{P} = \{P_p; 1 \le p \le P\}$ that contains all the parameters. The data sensed by the p^{th} parameter at i^{th} sampling instant of the x^{th} measurement cycle is denoted as $z_p^x(i)$. If N_p^x is the number of samples collected for p^{th} parameter in x^{th} measurement cycle, $\mathbf{Z}_p^x \in \mathbb{R}^{N_p^x\times 1}$ contains the temporal measurements of the p^{th} parameter, which can be expressed as [21]

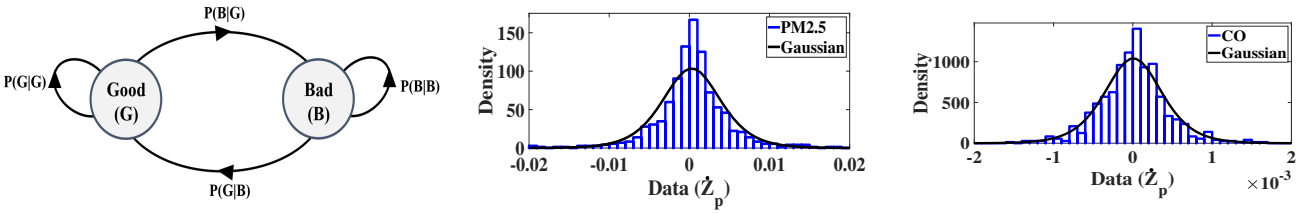
$$\mathbf{Z}_p^x = \mathbf{Y}_p^x + \boldsymbol{\eta}_p^x, \tag{1}$$

where $\mathbf{Y}_p^x \in \mathbb{R}^{N_p^x\times 1}$ is the true signal vector and $\boldsymbol{\eta}_p^x \in \mathbb{R}^{N_p^x\times 1}$ is the identically and independently distributed measurement noise vector associated with the p^{th} sensor, where $\eta \sim N(0, \sigma)$ for all the parameters.

B. Problem Definition

In the system model, depicted in Section II-A, the sensor node consists of P sensors. Consider that the p^{th} sensor collects data at a fixed Nyquist sampling interval t_p and transmits the samples to the edge node which is placed at a fixed point nearby. If the sensing signals remain within the satisfactory level/threshold, the system is considered to be in 'good' state and behaves as a non-real-time system, where the data can be sent to the edge node at the end of the measurement cycle. If any signal crosses the threshold, the system is considered to be in 'bad' state and acts as a real-time system, wherein and the data is sent to the edge node immediately with higher priority. When the system acts on non-real-time basis, the interval between two data transmissions or the length of measurement cycle depends on the process dynamics. Let τ^x be the length of x^{th} measurement cycle, where $\tau^x = N_p^x t_p$. To reduce the complexity, t_p is considered to be fixed in every measurement cycle. Since the time series data exhibits a strong temporal correlation, the signals can be reconstructed with $M_p^x \ll N_p^x$ number of samples, where M_p^x is the number of optimal sampling instants for p^{th} parameter at the x^{th} measurement cycle. Let $\mathbf{N}^x \in \mathbb{R}^{1\times P}$ and $\mathbf{M}^x \in \mathbb{R}^{1\times P}$ be the vectors containing the total sampling instants and the optimal sampling instants of P parameters, respectively, such that $N^x(p) = N_p^x$ and $M^x(p) = M_p^x$.

As observed in [4], the sensing signals exhibit cross-correlation, hence all the sensors need not to be activated all the time. $N' = (2^P - 2)$ number of sensor sets can be created Using P sensors, excluding the null set and the set containing all parameters. Let $\mathcal{A}_i^x = \{P_{\mathcal{A}_i^x,m}; 1 \le m \le A_i^x\}$



(a) (b) (c)

Fig. 2: (a) Two-state Markov model of sensing signal, (b) distribution of 1^{st} order time derivative of $PM_{2.5}$, and (c) distribution of 1^{st} order time derivative of CO.

be the i^{th} active sensor set in the x^{th} measurement cycle, then $\mathcal{B}_i^x = \mathcal{P} - \mathcal{A}_i^x$ contains the inactive/sleeping sensors. Hence, i^{th} sleep set in the x^{th} measurement cycle is $\mathcal{B}_i^x = \{P_{\mathcal{B}_i^x,k}; 1 \leq k \leq B_i^x\}$. $\mathcal{S}^x = \{(\mathcal{A}_i^x, \mathcal{B}_i^x); 1 \leq i \leq N'\}$ contains the active-sleep pair of sets.

The sensor node collects samples based on the optimal active sensor set and optimal sampling instants of the active sensors. The data collected by the sensor node is transmitted to the edge node as per non-real-time/real-time priority. Section III describes the process of deciding the non-real-time/real-time nature of the system based on the sensed signal values. If any of the sensing signals crosses above a user-defined satisfactory level, the system operates in real-time mode, else it operates in non-real-time mode. The missing samples are estimated at the edge node using GPR-based prediction models, as described in Section IV. At the end of every measurement cycle, the edge node exploits cross-correlation among the parameters and temporal correlation of the sensing signals to find an optimal sensor set and the optimal sampling instants for the next measurement cycle based on the adaptive sensing algorithm proposed in Section V-D and the information is transmitted to the sensor node. By optimizing the number of samples to be collected at each measurement cycle, the proposed framework achieves energy saving as well as bandwidth saving.

III. Characterization of Sensing Signal Dynamics

This section describes the characterization of sensing signal dynamics. The sensing signals are usually non-stationary [22]. If any signal remains within its satisfactory level it is said to be in 'good' state. Otherwise, the signal is considered to be in 'bad' state. Thus, according to [23], [24], a two-state Markov model can be created for each signal; shown in Fig. 2(a). In the proposed model, the system is statically defined. The system acts on non-real-time basis when all the parameters are in 'good' state. If any parameter enters into the 'bad' state, the system acts as a real-time system.

Let $Z_p(t)$ be the amplitude of p^{th} parameter signal at time t. The signal is said to be in 'bad' state if $Z_p(t) > Z_{p,th}$, else it is in 'good' state. If $Z_p(t) = Z_p^0$ at the present time slot, then its amplitude after time T_p can be expressed as, $Z_p(t + \Delta_1) \approx Z_p^0 + \dot{Z}_p\Delta_1$, where $\Delta_1 << 1$s is the slot duration. Let $Z_p^1 = \dot{Z}_p\Delta_1$, similarly after j slots $Z_p^j = \dot{Z}_p j\Delta_1$.

To find the distribution of time derivative of sensing signals, an air-pollution monitoring dataset with parameters $PM_{2.5}$, CO, NH_3, NO_2, Ozone, SO_2 are studied. It has been observed that the distribution of sensing parameters are non-stationary, however, the time derivative of these signals follow Gaussian distribution, as shown in Fig. 2(b) and Fig. 2(c). Thus, $\dot{Z}_p = \frac{dZ_p}{dt}$ is a zero mean Gaussian random variable (RV) i.e, $\dot{Z}_p \sim N(0, \sigma_{\dot{Z}_p})$, where $\sigma_{\dot{Z}_p}$ is the standard deviation. Following the study in [25], Z_p^j is a truncated Gaussian RV and its probability density function is given by,

$$f_{Z_p^j}(Z_p^j) = \begin{cases} \frac{1}{\sqrt{2\pi}\dot{\sigma}_{Z_p^j}(1-\Phi(-\frac{Z_p^0}{\dot{\sigma}_{Z_p^j}}))} exp(-\frac{Z_p^{j\,2}}{2\dot{\sigma}^2_{Z_p^j}}), & Z_p^j \geq -Z_p^0 \\ 0, & \text{elsewhere,} \end{cases}$$

where $\sigma^2_{\dot{Z}_p^j} = j\sigma^2_{\dot{Z}_p^1}$ and $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} exp(-\frac{t^2}{2})dt$. If $Z_p(t) = Z_0 (< Z_{p_{th}})$, the Gaussian mixture distribution of the temporal variation of the signal over the next ζ slots is:

$$f_\Gamma(\gamma | Z_p(t) = Z_0, \zeta) = \begin{cases} \frac{1}{\zeta}\sum_{j=1}^{\zeta} \frac{exp(-\frac{\gamma^2}{2\dot{\sigma}^2_{Z_p^j}})}{\sqrt{2\pi}\dot{\sigma}_{Z_p^j}(1-\Phi(\frac{-Z_0}{\dot{\sigma}_{Z_p^j}}))}, \gamma \geq -Z_0 \\ 0, \quad \text{elsewhere.} \end{cases}$$

The probability that the p^{th} parameter will remain below threshold up to ζ slots is given by,

$$P_p^g(Z_p(t) = Z_0, \zeta) = Pr(Z_p(t) < Z_{p_{th}})$$
$$= \frac{1}{\zeta}\sum_{j=1}^{\zeta} \frac{1}{(1-\Phi(\frac{-Z_0}{\dot{\sigma}_{Z_p^j}}))} \int_0^{Z_{p_{th}}} \frac{1}{\sqrt{2\pi}\dot{\sigma}_{Z_p^j}} exp(-\frac{(\gamma - Z_0)^2}{2\dot{\sigma}^2_{Z_p^j}}). \quad (2)$$

The probability that the p^{th} parameter will be above threshold up to ζ slots is:

$$P_p^b(Z_p(t) = Z_0, \zeta) = Pr(Z_p(t) > Z_{p_{th}})$$
$$= 1 - P_p^g(Z_p(t) = Z_0, \zeta). \quad (3)$$

Let τ^x be the length of x^{th} measurement cycle which can be divided into J slots such that $\tau^x = \Delta_1 J$. Then $P_p^{g,x} = P_p^g(Z_p(t) = Z_p^0, J)$ and $P_p^{b,x} = P_p^b(Z_p(t) = Z_p^0, J)$ are the 'good' and 'bad' state probabilities of the p^{th} parameter in the x^{th} measurement cycle, which will be subsequently used in Section V-A to find the optimal length of measurement cycle.

IV. GPR-based Temporal and Cross-correlated Parameter Prediction Model

This section describes the GPR-based signal recovery models to predict the sensing signals. As stated in Section I-A,

GPR is widely used to predict sensing parameters in WSNs due to its efficient signal recovery performance. GPR exploits all possible functions, drawn from a Gaussian distribution passing through the points rather than assuming a fixed parametric model of the data and finds a suitable function that fits the data well with ± 1 standard deviation [26]–[28]. The proposed framework uses GPR model to predict the temporal and cross-correlated parameters, as it performs better than the existing signal recovery methods [13]–[15].

In the proposed approach, two GPR models are used. GPR_1 model consists of P regressors to predict the time series data from their previous samples based on the temporal correlation of the individual parameters. The P regressor models are implemented in both the edge node and the sensor node. The sensor node predicts the next sample from the previous samples of each parameter belongs to $\mathcal{A}_i^x$. Based on the predicted data, the node decides the next state of the parameter. If the parameter goes to the 'bad' state the sensor node collects the next sample for that parameter and verify the state of the signal. If it still remains in the 'bad' state, the sensor node immediately transmits the data to the edge node. GPR_2 is used to predict the parameters of $\mathcal{B}_i^x$ from the cross-correlated parameters of $\mathcal{A}_i^x$. The GPR_2 model consists of N' sub-models for N' active-sleep pairs of $\mathcal{S}$. If $\mathcal{A}_i^x \subset \mathcal{P}$ is activated in the x^{th} measurement cycle, the i^{th} sub-model is selected to predict the parameters of $\mathcal{B}_i^x$. Intuitively, the correlated parameters have more significance in prediction. GPR_1 and GPR_2 models are briefly described in Section IV-A, and IV-B, respectively.

A. GPR Model for Time Series Prediction

The GPR model used to predict the temporal values of the parameters, named as GPR_1 is placed at both the edge node and the end/sensor node. The GPR_1 model consists of P regressors to predict the temporal samples of P parameters. Let $p \in \mathcal{P}$ is selected to activate in the x^{th} measurement cycle. The j^{th} temporal sample of p^{th} parameter can be predicted from previous l_p samples, where l_p is the lag value. Thus, the true signal value is $Y_p^x(j) = f(Z_p^x(j-1), Z_p^x(j-2), \cdots, Z_p^x(j-l_p))$.

The training matrix and target vector for p^{th} regressor model are respectively $\underline{\mathbf{Z}}_p \in \mathbb{R}^{n \times l_p}$ and $\mathbf{Z}_p \in \mathbb{R}^{n \times 1}$, where n is the number of training samples. To predict $Y_p^x(*)$ in the x^{th} measurement cycle, GPR finds the underlying function f_p as,

$$Y_p^x(*) = f_p(\mathbf{Z}_p^x(*)) = f_{p,*}^x, \tag{4}$$

where $f_p \sim N(0, K_{n\times n})$. $K_{n\times n} = [k(\mathbf{Z}_p(r), \mathbf{Z}_p(s))]_{n\times n}$, where $k(\mathbf{Z}_p(r), \mathbf{Z}_p(s))$ is the covariance function between the r^{th} and s^{th} instant of design/training matrix $\underline{\mathbf{Z}}_p$. According to [29, Ch. 2], the squared exponential covariance function for the p^{th} regressor of GPR_1 model is given by,

$$k(\mathbf{Z}_p(r), \mathbf{Z}_p(s)) = exp\{-\frac{1}{2l'^2}\sum_{q=1}^{l_p}|Z_p(r,q) - Z_p(s,q)|^2\} + \sigma^2\delta_{rs}. \tag{5}$$

The p^{th} prediction model of GPR_1 is given by, $f_{p,*}^x|\underline{\mathbf{Z}}_p, \mathbf{Z}_p, \mathbf{Z}_p^x(*) \sim N(\overline{f_{p,*}^x}, \text{Cov}(f_{p,*}^x))$, where $\overline{f_{p,*}^x}$ and $\text{Cov}(f_{p,*}^x)$ are the mean and covariance function, respectively, which can be derived from (A.1) and (A.2) of Appendix A. From [29, Ch. 2], $\overline{f_{p,*}^x}$ and $\text{Cov}(f_{p,*}^x)$ are expressed as,

$$\overline{f_{p,*}^x} = K(\underline{\mathbf{Z}}_p, \mathbf{Z}_p^x(*))[K(\underline{\mathbf{Z}}_p, \underline{\mathbf{Z}}_p) + \sigma^2 I]^{-1}\mathbf{Z}_p \tag{6}$$

$$\text{Cov}(f_{p,*}^x) = k(\mathbf{Z}_p^x(*), \mathbf{Z}_p^x(*)) + K(\mathbf{Z}_p^x(*), \underline{\mathbf{Z}}_p)[K(\underline{\mathbf{Z}}_p, \underline{\mathbf{Z}}_p) + \sigma^2 I]^{-1}K(\underline{\mathbf{Z}}_p, \mathbf{Z}_p^x(*)). \tag{7}$$

B. GPR Model for Cross-correlated Parameter Prediction

The GPR model used to predict cross-correlated parameters, named as GPR_2 is placed at the edge node. Let $\mathcal{A}_i^x$ be the optimal sensor set selected to activate in the x^{th} measurement cycle and the parameters of $\mathcal{B}_i^x$ need to be predicted from the cross-correlated parameters of $\mathcal{A}_i^x$ using the i^{th} sub-model of GPR_2. The i^{th} sub-model contains B_i^x number of regressors, each predicts one parameter of $\mathcal{B}_i^x$. The training matrix of the k^{th} regressor of the i^{th} sub-model of GPR_2 is $\underline{\mathbf{Z}}_{\mathcal{A}_i} \in \mathbb{R}^{n \times A_i}$ and the target vector is $\mathbf{Z}_{\mathcal{B}_i^x}(k) \in \mathbb{R}^{n\times 1}$. Each row of $\underline{\mathbf{Z}}_{\mathcal{A}_i}$ corresponds to one feature vector. The j^{th} row of $\underline{\mathbf{Z}}_{\mathcal{A}_i}$ is expressed as $\mathbf{Z}_{\mathcal{A}_i} = \{Z_{\mathcal{A}_i}(j,1), Z_{\mathcal{A}_i}(j,2), \cdots, Z_{\mathcal{A}_i}(j,A_i)\}$. During prediction in the x^{th} measurement cycle, the measurement vectors of $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ for the j^{th} sampling instant are expressed as $\mathbf{Z}_{\mathcal{A}_i}^x(j) = \{Z_{\mathcal{A}_i}^x(j,1), Z_{\mathcal{A}_i}^x(j,2), \cdots, Z_{\mathcal{A}_i}^x(j,A_i^x)\} \in \mathbb{R}^{1\times A_i^x}$ and $\mathbf{Z}_{\mathcal{B}_i}^x(j) = \{Z_{\mathcal{B}_i}^x(j,1), Z_{\mathcal{B}_i}^x(j,2), \cdots, Z_{\mathcal{B}_i}^x(j,A_i^x)\} \in \mathbb{R}^{1\times B_i^x}$, respectively.

The equation (1) for $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ can be written as, $\mathbf{Z}_{\mathcal{A}_i}^x(j) = \mathbf{Y}_{\mathcal{A}_i}^x(j) + \boldsymbol{\eta}_{\mathcal{A}_i}^x(j)$ and $\mathbf{Z}_{\mathcal{B}_i}^x(j) = \mathbf{Y}_{\mathcal{B}_i}^x(j) + \boldsymbol{\eta}_{\mathcal{B}_i}^x(j)$, respectively. For a test vector $\mathbf{Z}_{\mathcal{A}_i}^x(*)$, GPR finds the underlying function $f_{i,*}^{x,k}$ for the k^{th} regressor as,

$$Y_{\mathcal{B}_i}^x(*,k) = f_k(\mathbf{Z}_{\mathcal{A}_i}^x(*)) = f_{i,*}^{x,k} \in \mathbb{R}\forall k \in \mathbf{Y}_{\mathcal{B}_i}^x(*), \tag{8}$$

where $f_i^k \sim N(0, K_{n\times n})$. $K_{n\times n}$ is the covariance matrix whose elements are derived from the squared exponential covariance function, as described in Section IV-A.

The prediction model for GPR_2 is given by, $f_{i,*}^{x,k}|\underline{\mathbf{Z}}_{\mathcal{A}_i}, \mathbf{Z}_{\mathcal{B}_i}(k), \mathbf{Z}_{\mathcal{A}_i}^x(*) \sim N(\overline{f_{i,*}^{x,k}}, \text{Cov}(f_{i,*}^{x,k}))$. Thus, the mean function and the covariance function, derived from (A.1) and (A.2) of Appendix A are expressed as,

$$\overline{f_{i,*}^{x,k}} = K(\underline{\mathbf{Z}}_{\mathcal{A}_i}, \mathbf{Z}_{\mathcal{A}_i}^x(*))[K(\underline{\mathbf{Z}}_{\mathcal{A}_i}, \underline{\mathbf{Z}}_{\mathcal{A}_i}) + \sigma^2 I]^{-1}\mathbf{Z}_{\mathcal{B}_i}(k) \tag{9}$$

$$\text{Cov}(f_{i,*}^{x,k}) = k(\mathbf{Z}_{\mathcal{A}_i}^x(*), \mathbf{Z}_{\mathcal{A}_i}^x(*)) + K(\mathbf{Z}_{\mathcal{A}_i}^x(*), \underline{\mathbf{Z}}_{\mathcal{A}_i})[K(\underline{\mathbf{Z}}_{\mathcal{A}_i}, \underline{\mathbf{Z}}_{\mathcal{A}_i}) + \sigma^2 I]^{-1}K(\underline{\mathbf{Z}}_{\mathcal{A}_i}, \mathbf{Z}_{\mathcal{A}_i}^x(*)). \tag{10}$$

These prediction models will be used in Section V to estimated temporal values of the parameters at the edge/end node and cross-correlated parameter values at the edge node.

V. Proposed Framework

This section describes the proposed data-driven dynamic priority sensing and transmission framework. The formulation of optimization function to find the optimal length of measurement cycle is presented in Section V-A, followed by the method to find optimal lead values of the sensing parameters in Section V-B. Section V-C presents the formulation of UCB-based optimization function to find optimal sensor set,

followed by the algorithms applied at the edge/sensor node in Section V-D and complexity of the framework in Section V-E.

As discussed in Section I, a sensor node has limited energy availability, which needs to be used optimally to enhance the network lifetime. The adaptive sensing algorithm, running at the sensor node increases the processing energy consumption of the node. To reduce the complexity and hence energy requirement of the sensor node, an edge intelligence-based adaptive sensing framework is proposed in this section.

Let a and b are two sensing parameters having cross-correlation coefficient $c(a,b)$. a and b are said to be correlated if $|c(a,b)| \geq c_{th}$, where c_{th} is the cross-correlation threshold. As discussed in Section II, $2^P - 2$ active-sleep pair of subsets can be created from P sensors excluding the set having all the parameters and the null set. The i^{th} sensor set is said to be correlated if all the parameters of $\mathcal{B}_i^x$ are correlated with at least one parameter of $\mathcal{A}_i^x$. The strength of cross-correlation between $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ is defined by the cross-correlation factor C_i^x. If $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ are cross-correlated, C_i^x is given by,

$$C_i^x = \frac{1}{A_i^x B_i^x} \sum_{k=1}^{B_i^x} \sum_{m=1}^{A_i^x} |c^x(m,k)|; \forall k \in \mathcal{B}_i^x \text{ and } m \in A_i^x. \quad (11)$$

If $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ are not cross-correlated, $C_i^x = 0$ for the i^{th} active-sleep sensor set.

The performance of the proposed algorithm is measured in terms of reconstruction error of the sensing parameters at the edge node. If Pe_p^x is the reconstruction error of the p^{th} parameter in the x^{th} measurement cycle, the average error is:

$$PE_i^x = \frac{1}{P} \sum_{p=1}^{P} Pe_p^x; \forall p \in \mathcal{P}. \quad (12)$$

A. Optimal Length of Measurement Cycle

Let, $\mathbf{A}^T \in \mathbb{R}^{P \times 1}$ be the binary sensing vector such that $\mathbf{A}(p) = 1$ if $p \in \mathcal{P}$ is active. A set $Q^x = \{\mathbf{A}_i^x; 1 \leq i \leq N'\}$ is the set of vectors created from N' active sets in $\mathcal{S}^x$. A correlation matrix $\underline{\mathbf{C}}^x \in \mathbb{R}^{P \times P}$ contains the cross-correlation coefficients of the parameters such that $\underline{\mathbf{C}}^x(p,q) = |c(p,q)|; \forall p,q \in \mathcal{P}$, where $c(p,q)$ is a function of the length of measurement cycle τ. If τ increases the signal estimation error increases due to the large variation in environmental conditions between two cycles. Thus $c(p,q)$ decreases with the increase in τ. Moreover, the $P_k^g; \forall k \in \mathcal{B}_i^x$ should be higher than the threshold P_{th}^g to ensure that the inactive sensors will not go to 'bad' state over the whole measurement cycle.

If any signal enters into 'bad' state, the data stored in the sensor node need to be transmitted immediately. Let, Ec be the total communication energy required for one transmission. $Ec = Ew + Et$, where Ew and Et are the wake up and data transmission energy, respectively. Ew is assumed to be fixed and Et increases with the increase in the number of samples to be transmitted. Hence, the transmission energy consumption of i^{th} active sensor set in the x^{th} measurement cycle is:

$$Et_i^x = Et1 \sum_{m=1}^{A_i^x} M_m^x; \forall m \in \mathcal{A}_i^x. \quad (13)$$

$Et1$ is the transmission energy consumed by the radio module to transmit one sample. Since Ew is the initial energy required to turn on the communication module, it is much higher than Et. Although data transmission energy increases with the length of measurement cycle, total communication energy over a large period decreases due to the fewer number of activations of the radio module. Therefore, an optimal length of measurement cycle τ has to find by solving the trade-off among $\underline{\mathbf{C}}^x$, $P_k^g; \forall k \in \mathcal{B}_i^x$, and Ew. Due to the unavailability of data for the $(x+1)^{th}$ cycle, the cross-correlation factor is calculated from the data received at the x^{th} cycle. However, the 'good' state probabilities of the parameters are estimated for $(x+1)^{th}$ cycle using (III). It is assumed that the sensing signals are slowly varying [11]. Hence, the cross-correlation among the parameters does not change drastically. Thus, $\underline{\mathbf{C}}^x$ can be used to find the length of $(x+1)^{th}$ cycle. τ is a function of $\mathcal{A}_i^x$. For each active set $\mathcal{A}_i^x$, the optimal value of $\tau_{\mathbf{A}_i}^x$ is:

$$\begin{aligned} \tau_{\mathbf{A}_i}^x = \quad & \max\{\zeta^c\} \\ \text{s.t.} \quad & \frac{1}{A_i^x B_i^x} |\mathbf{A}_i^x \underline{\mathbf{C}}_1^x (\mathbf{U}^T - \mathbf{A}_i^{xT}) - \mathbf{A}_i^x \underline{\mathbf{C}}_{\zeta^c}^x (\mathbf{U}^T - \mathbf{A}_i^{xT})| < \delta \\ & P_k^g(Z_k(t) = Z_0, \frac{\Delta_2}{\Delta_1}\zeta^c) > P_{th}^g \forall k \in \mathcal{B}_i^x \\ & \zeta^c \in \{\zeta_{min}^c, \zeta_{max}^c\}. \end{aligned} \quad (14)$$

$\underline{\mathbf{C}}_1^x$ is the cross-correlation matrix obtained from the most recent samples of the x^{th} measurement cycle and $\underline{\mathbf{C}}_{\zeta^c}^x$ is obtained by shifting the data window up to past ζ^c slots, where the duration of each slot is Δ_2. Δ_2 should be very small and it is assumed that the process dynamics are similar over Δ_2.

B. Optimal Lead Values of the Sensing Parameters

The GPR_1 model, discussed in Section IV-A is used to predict the consecutive samples of the active sensors from their previous samples. If the prediction error remains below the threshold, these samples need not be collected by the sensors, which reduces the sensing energy further.

Let, $ct_p^x(n)$ be the temporal correlation between $Z_p^x(i)$ and $Z_p^x(i-n)$. If $ct_p^x(n) \geq ct_{p,th}$, n consecutive samples can be predicted from their previous samples with prediction error below et_{th}. The maximum value of n is the lead value of the p^{th} parameter in the x^{th} cycle; defined as lead value ${\zeta^t}_p^x$. At the edge node, the optimal value of ${\zeta^t}_p^x$ is given by,

$$\begin{aligned} {\zeta^t}_p^x = \max \quad & n \\ \text{s.t.} \quad & ct_p^x(n) \geq ct_{p,th}. \end{aligned} \quad (15)$$

Thus, $M_p^x = \frac{N_p^x}{({\zeta^t}_p^x + 1)} = \frac{\mathcal{T}^x}{t_p({\zeta^t}_p^x + 1)}$ is the number samples need to be collected for the p^{th} parameter in the x^{th} measurement cycle. Here t_p is the interval between two collected/predicted samples, obtained using Nyquist sampling rate. For simplicity of the algorithm t_p is not computed at every cycle. ${\zeta^t}_p^x$, obtained form (15) is used to estimate M_p^x which is further used to find the optimal active sensor set for $(x+1)^{th}$ measurement cycle.

If $et_p(n)$ is the temporal prediction error of the p^{th} parameter for the n^{th} sample, $et_p(n)$ can be represented as,

$$et_p(n) = \alpha_p n^{\beta_p}. \quad (16)$$

$$\alpha_p, \beta_p = \min_{\alpha,\beta} \quad \{\sum_{n=1}^{n_p} (et_p(n) - \alpha n^{\beta})^2\}. \tag{17}$$

The optimal values of α_p and β_p are estimated using (17).

After receiving $\mathcal{A}_i^{x+1}$, the sensor node updates the lead values of the active sensors to dynamically adapt their sampling intervals based on the temporal prediction error. If $et_{p,0}$ is the prediction error at the $n_{p,0}^{th}$ slot in the sensor node, the new ${\zeta_p}^{x+1}$ in the $x+1^{th}$ measurement cycle at the sensor node is given by,

$${\zeta_p}^{x+1} = \min_{n} \quad (et_{p,0} + \alpha_p(n^{\beta} - n_{p,0}^{\beta}) - et_{th})^2. \tag{18}$$

The deviation of temporal prediction error from its threshold value decides the new value of ${\zeta_p}^{x+1}$. After ${\zeta_p}^{x+1}$ slots the sensor node collects l_p samples for the p^{th} active parameter to predict the next consecutive samples.

C. Discounted UCB-based Optimization Function for Sensor Selection

The optimization functions formulated to find the optimal sensor set to be activated in the next measurement cycle are discussed in this subsection. (19) presents the optimization function used to find the optimal sensor set for the next cycle from the data collected at the present cycle. To make the system more predictive, an optimization function is formulated in (21), that finds the optimal sensor set for the next cycle from the data collected at the present and past cycles using discounted UCB algorithm.

Let $\underline{\mathbf{Es}}$ be a diagonal matrix whose diagonal elements contain the energy consumed by the sensors to collect one sample. Thus, $Es(p,p) = Es1_p$, where $Es1_p$ is the energy consumption of p^h sensor to collect one sample. Hence, the total sensing energy consumption of the i^{th} active set in the x^{th} measurement cycle is $Es_i^x = \mathbf{A}_i^x \underline{\mathbf{Es}} \mathbf{M}^x$, where $\mathbf{M}^x$ contains the optimal sampling instants of the parameters obtained using (15) and $\mathbf{A}_i^x$ is the binary sensing vector. Let $\mathbf{U}^T \in \mathbb{R}^{P\times 1}$ be a unity vector such that $\mathbf{U}(i) = 1; 1 \leq i \leq P$. The cross-correlation factor of the i^{th} active set for the first slot ($\zeta^c = 1$) is given by, $C_i^x(1) = C_i^x = \frac{1}{A_i^x B_i^x}[\mathbf{A}_i^x \underline{\mathbf{C}}_1^x (\mathbf{U}^T - {\mathbf{A}_i^x}^T)]$. Higher value of C_i^x gives better sensing quality. Let $\mathbf{G}_i^x \in \mathbb{R}^{P\times 1}$ be a vector such that $\mathbf{G}_i^x = \underline{\mathbf{C}}_1^x {\mathbf{A}_i^x}^T$. $\mathcal{A}_i^x$ and $\mathcal{B}_i^x$ are correlated if $G_i^x(k) > c_{th}; \forall k \in \mathcal{B}_i^x$.

To incorporate the energy awareness, the residual energy is considered as another performance parameter for finding optimal sensor set. Let E_{batt} be the battery capacity of the sensor node and E_0^x be its available energy in x^{th} measurement cycle. $\lambda^x \triangleq \frac{E_0^x}{E_{batt}}$ is the normalized energy available at the sensor node. The objective is to maximize C_i^x, $\tau_{\mathbf{A}_i}^x$ and λ^x and minimize Es_i^x.

The optimization function formulated to find $\mathcal{A}_i^{x+1}$ by solving the trade-off among the cross-correlation, energy consumption, residual energy of the sensor node, and length of measurement cycle is given by,

$$\begin{aligned} \max_{\mathbf{A}_i^x \in \mathcal{Q}^x} \quad & \frac{\lambda^x C_i^x \tau_{\mathbf{A}_i}^x}{Es_i^x} \\ \text{s.t.} \quad & G_i^x(k) > c_{th}; \forall k \in \mathcal{B}_i^x \\ & Es_i^x < E_0^x. \end{aligned} \tag{19}$$

The above optimization function explores all the binary sensing vectors of $\mathcal{Q}^x$ and selects an optimal sensing vector $\mathbf{A}_i^x$ for the next cycle. The above function is a concave function, whose proof is given in the Appendix B. The first constraint ensures that the selected active and sleep sensor sets are correlated and each parameter of $\mathcal{B}_i^x$ can be predicted from at least one parameter of $\mathcal{A}_i^x$. The second constraint ensures that the required sensing energy for the selected active sensor set is available at the sensor node.

The optimization function, given in (19) can find an optimal sensor set for the next measurement cycle based on the present state of the system, without having any knowledge about the previous states. Since the environment is non-stationary, predicting the next state from only the present state may not give the optimal solution. The system becomes more predictive when it includes past experiences along with the present state. If a sensor set obtains higher rewards in the past cycles, its probability of being optimal in the next measurement cycle is higher than the other sets. Considering a sensor hub collecting samples of various parameters from an environment is a random process, an RL model can be created where the learner can learn the system with time, based on previous experiences. To this end, an RL algorithm such as UCB algorithm is used in the proposed framework to select an optimal sensor set to be activated in the next measurement cycles based on the present and previous rewards of the sensor set. UCB is widely used to select optimal arm in the multi-armed bandit problems. Finding an optimal arm in multi-armed bandit problems is analogous to finding an optimal sensor set in the proposed adaptive sensor selection problem. Let $R_i^x = \frac{\lambda^x C_i^x \tau_{\mathbf{A}_i}^x}{\nu^x Es_i^x}$ be the reward obtained for the i^{th} sensor set in the x^{th} measurement cycle, where $\nu^x = \max_{i\in\mathcal{Q}^x} \frac{C_i^x \tau_{\mathbf{A}_i}^x}{Es_i^x}$. By putting ν in the denominator, the reward is bounded to $[0,1]$. An optimal active sensor set can be chosen for the $(x+1)^{th}$ measurement cycle based on the rewards of the sensor sets obtained in the present and past cycles. It has been observed that the reward distribution changes abruptly in a non-stationary environment. In such cases, the most recently collected samples are more responsible to decide the next state compared to the old samples. Since the sensing parameters follow non-stationary distribution, discounted UCB algorithm is more suitable in the proposed framework. The rewards are discounted using $\gamma \in (0,1)$ to give more weightage to the recent samples [30]. In the proposed discounted UCB-based sensor selection framework, the reward is computed for all the sensor sets at every measurement cycle.

Let $R_i^1, R_i^2, R_i^3, \cdots, R_i^x$ be the rewards for the i^{th} active-sleep pair of sensor set up to x^{th} measurement cycle. The rewards are identically and independently distributed Gaussian random variables. Thus, the empirical mean of the distribution of R_i is given by,

$$\widehat{\mu}_i^x = \frac{1}{\sum_{t=1}^x \gamma^{x-t}} \sum_{t=1}^{x} \gamma^{x-t} R_i^t = \frac{1}{\sum_{t=1}^x \gamma^{x-t}} \sum_{t=1}^{x} \gamma^{x-t} \frac{\lambda^t C_i^t \tau_{\mathbf{A}_i}^t}{\nu^t Es_i^t}. \tag{20}$$

The modified optimization function to find an optimal

sensor set in the $(x+1)^{th}$ measurement cycle is given by,

$$\max_{\mathbf{A}_i^x \in \mathcal{Q}^x} \frac{1}{\sum_{t=1}^{x} \gamma^{x-t}} \sum_{t=1}^{x} \gamma^{x-t} \frac{\lambda^t C_i^t \tau_{\mathbf{A}_i}^t}{\nu^t E s_i^t} + \sqrt{\frac{\alpha'}{T_i^x} log(\sum_{i-1}^{N'} T_i^x)}$$
$$\text{s.t.} \quad G_i^x(k) > c_{th}; \forall k \in \mathcal{B}_i^x$$
$$Es_i^x < E_0^x. \tag{21}$$

In (21), the first term is the empirical mean of the distribution of reward while the second term provides the confidence bound, where $T_i^x = \sum_{t=1}^{x} \gamma^{x-t} \mathbb{1}_{[\mathbf{A}^t = \mathbf{A}_i]}$. A higher value of T_i^x makes the learner more confident about the distribution of reward R_i^x [10]. At the end of every measurement cycle the optimization function explores all the sensor sets and selects an optimal sensor set that satisfies the constraints given in (21). If sufficient energy available at the node, the optimization function always returns an optimal sensor set, as proven in the Appendix B. If $\mathcal{A}_i^{x+1}$ is selected to activate in the $(x+1)^{th}$ cycle, the length of $(x+1)^{th}$ measurement cycle is $\tau^{x+1} = \tau_{\mathbf{A}_i}^x$.

D. Data-Driven Dynamic Priority Sensing and Transmission Mechanism

As discussed in the problem definition in Section II, the sensor node transmits data to the edge node at the end of every measurement cycle when the system is in 'good' state and it transmits data immediately when the system is in 'bad' state. When the system is in 'good' state the data can be predicted with lesser accuracy, however, during the 'bad' state the data should be predicted with good accuracy. Hence, three different error thresholds are defined in the proposed algorithm. When the system is in 'bad' ('good') state, the temporal prediction error of GPR_1 model should be below et_{th}^b (et_{th}^g). The overall error, which is the reconstruction error of the GPR_2 model is considered as e_{th}. e_{th} is the average reconstruction error of all the parameters over a measurement cycle. If a parameter is in 'bad' state, its temporal prediction error is below et_{th}^b. However, the overall error can be above et_{th}^b.

Initially, the sensor node collects n samples for all the parameters at their respective Nyquist rates and transmits them to the edge node. The edge node receives the samples and reconstructs the sensing signals for all the parameters and sample data at a high sampling rate (maximum of all the Nyquist rates of the parameters) from all the sensing signals. As described in Algorithm 2, the GPR_2 model is trained using the samples drawn from the reconstructed signals at a fixed sampling rate. The P regressors of GPR_1 model are trained using the data received from the sensor node based on their respective Nyquist rate. An identical GPR_1 model is created at the sensor node to predict temporal samples. The sensor node transmits only the collected samples to the edge node. The samples predicted by the GPR_1 model at the sensor node are not transmitted to the edge node, as these samples are predicted at the edge node using the same GPR_1 model at both 'good' and 'bad' states. The hyper-parameters of the algorithm such as, lag values, α, β, c_{th}, ζ_{max}^c, ζ_{min}^c, $\mathbf{ct}_{th}$, and δ are estimated during training/retraining the models. At every x^{th}

Algorithm 1: Data-driven dynamic priority sensing algorithm at the sensor node

Initialization: et_{th}^g, et_{th}^b, c_{th}, $\mathbf{Z}_{th}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $x = 1$, create GPR_1 model for time series prediction, find $\boldsymbol{\zeta}^{\mathbf{t}^x}$.
Input: Receive $\mathcal{A}_i^x$, τ^x, $\mathbf{N}^x$, $\mathbf{t}^x$ from edge node.
if *($e = 1$)* **then**
  Activate all sensors and collect samples at Nyquist sampling interval $\mathbf{t}^x$ untill $Time > \tau^x$
  Retrain and test GPR_1 model for P parameters with recently collected samples.
else
  for all $P_p \in \mathcal{A}_i^x$ **do**
  Activate sensors belongs to $\mathcal{A}_i^x$
  Collect l_p samples $\forall P_p \in \mathcal{A}_i^x$
  Set $j = l_p + 1$ and $n_{p,0} = 1$
  while *($j < N_p^x$)* **do**
    Predict next sample $\tilde{Z}_p^x(j)$ using GPR_1 model; Collect next sample $Z_p^x(j)$
    if *($(Z_p^x(j) < Z_{p,th})$)* **then**
      Find $et_{p,0} = |Z_p^x(j) - \tilde{Z}_p^x(j)|$,
      Find new $\zeta_p^{t^x}$ by putting $et_{th} = et_{th}^g$ in (18)
      Set $k = \zeta_p^{t^x}$
      while *($k > 0$ && $\tilde{Z}_p^x(j) < Z_{p,th}$)* **do**
        Predict next sample $\tilde{Z}_p^x(j)$
        Set $k = k - 1$, $j = j + 1$, and $n_{p,0} = \zeta_p^{t^x} - k$
      end
    else
      Collect w consecutive samples and find average $\bar{Z}_p$
      Set $j = j + w$ and $n_{p,0} = \zeta_p^{t^x} - k + w$
      while *($\bar{Z}_p > Z_{p,th}$)* **do**
        Set $em_p = 1$; Send $em_p = 1$ to the edge node
        Transmit the stored samples of all the parameters of $\mathcal{A}_i^x$
        while *($em_p = 1$)* **do**
          Predict next sample $\tilde{Z}_p^x(j)$; Collect next sample $Z_p^x(j)$ and transmit to edge node
          if *($(Z_p^x(j) > Z_{p,th})$)* **then**
            Find $e_{p,0}^t = |Z_p^x(j) - \tilde{Z}_p^x(j)|$
            Find new $\zeta_p^{t^x}$ by putting $et_{th} = et_{th}^b$ in (18)
            Set $k = \zeta_p^{t^x}$
            while *($k > 0$ & $\tilde{Z}_p^x(j) > Z_{p,th}$)* **do**
              Predict next sample $\tilde{Z}_p^x(j)$ using GPR_1 model
              Set $k = k - 1$ and $j = j + 1$
            end
          else
            Set $em_p = 0$; Send $em_p = 0$ to the edge node
          end
        end
      end
    end
  end
end
Set $x = x + 1$
Output: Transmit data and E_0^x to the edge node.

Algorithm 2: Data-driven dynamic priority sensing algorithm at the edge node

Initialization: et^g_{th}, et^b_{th}, e_{th}, c_{th}, $\mathbf{ct}^b_{th}$, $\mathbf{Z}_{th}$, $x = 1$, $e = 0$, find $\mathbf{t}^x$ and $\mathbf{N}^x$ using FFT, create GPR_1 and GPR_2 models using recently collected samples.
Input: Receive data and E^x_0 from sensor node.
if *($e = 1$)* **then**
  Find $\mathbf{t}^x$ and $\mathbf{N}^x$ for P parameters using FFT
  Retrain and test GPR_1 model for P parameters with recently collected samples.
  Retrain and test GPR_2 model for N' active-sleep sensor sets with recently collected samples.
  Set $e = 0$
else
  Predict the missing samples $\forall P_p \in \mathcal{A}^x_i$ using GPR_1 model (Section IV-A)
  Reconstruct the signals $\forall P_p \in \mathcal{A}^x_i$
  Draw samples from the reconstructed signals at interval $t = min\{t^x_k; 1 \leq k \leq B^x_i\}$
  Predict the cross-correlated parameters $\forall P_p \in \mathcal{B}^x_i$ using GPR_2 model (Section IV-B)
  $\mathbf{t}^{x+1} = \mathbf{t}^x$ and $\mathbf{N}^{x+1} = \mathbf{N}^x$
end
Set $x = x + 1$
Find $ct^{x-1}_p; \forall p \in \mathcal{P}$
if *($ct^{x-1}_p \not\geq ct^b_{p,th}; \forall p \in \mathcal{P}$)* **then**
  Set $e = 1$
  $\mathcal{A}^x_i = \mathcal{P}$, $\tau^x = n \times max(\mathbf{t}^x)$
else
  if *($C^x_i \leq c_{th}; \forall i \in \mathcal{S}^x || P^g_p(Z_p, \zeta^c_{min}) \leq P^g_{th}; \forall p \in \mathcal{P}$)* **then**
    $\mathcal{A}^x_i = \mathcal{P}$, $\tau^x = \tau^{x-1}$
  else
    Find $\zeta^{t^x}_p \forall p \in \mathcal{P}$, τ^x, and $\mathcal{A}^x_i$ using (15), (14), and (21), respectively
  end
end
Transmit $\mathbf{t}^x$, $\mathbf{N}^x$, $\mathcal{A}^x_i$, τ^x, and e to the sensor node
Start listen mode
Set $Time = 1$
for all $P_p \in \mathcal{A}^x_i$ **do**
while *($Time < \tau^x$)* **do**
  while *($em_p = 1$)* **do**
    Send ACK= 1 to sensor node
    Receive the stored samples of all the parameters of $\mathcal{A}^x_i$
    while *($em_p = 1$)* **do**
      if *(no sample received)* **then**
        Predict the missing sample and store
      else
        Store the received sample
      end
    end
  end
end
Receive remaining samples

measurement cycle, the edge node selects the active set $\mathcal{A}^x_i$ using (21) and transmits $\mathcal{A}^x_i$, τ^x, $\mathbf{t}^x$, $\mathbf{N}^x$, and status flag e to the sensor node. Based on the received information, the sensor node collects samples and transmits the collected samples and E^x_0 to the edge node. When the temporal correlation of the parameters falls below $\mathbf{ct}^b_{th}$, all the prediction models are retrained. Algorithm1 is programmed at the sensor node and Algorithm2 is programmed at the edge node.

TABLE II: Specification of sensors

Sensing parameter	Energy consumption (J)	Threshold ($Z_{p_{th}}$)
$PM_{2.5}$	29.55	$60\mu g/m^3$
CO	26m	$1mg/m^3$
NH_3	1.1	$30\mu g/m^3$
NO_2	20m	$38\mu g/m^3$
Ozone	50m	$50\mu g/m^3$
SO_2	26m	$40\mu g/m^3$

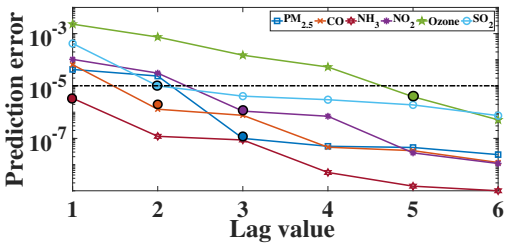


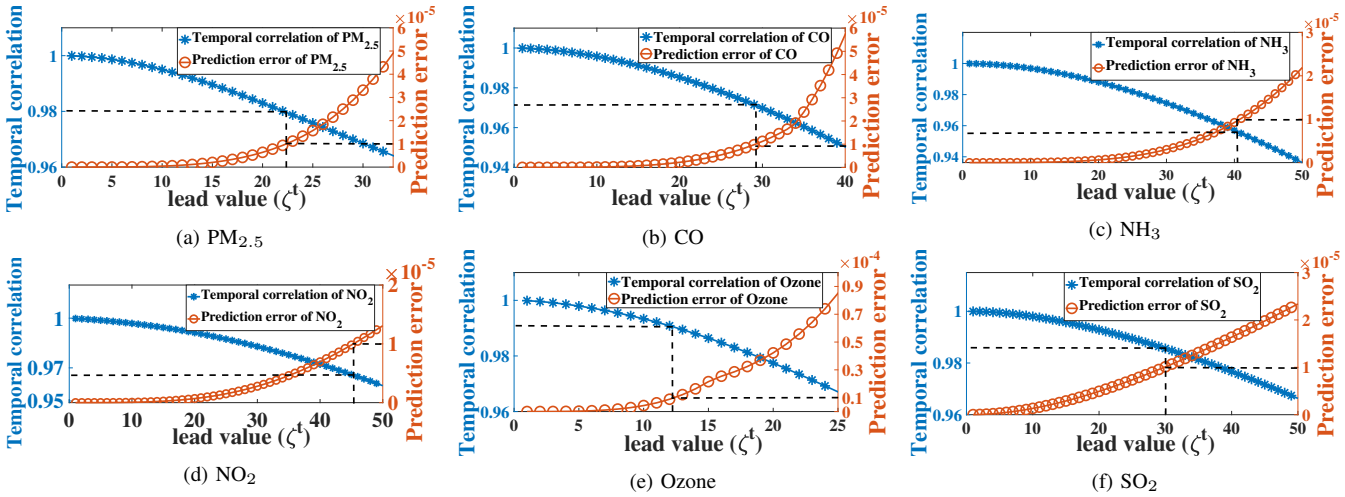
Fig. 3: Variation of temporal prediction error with l_p.

E. Computational Complexity of the Proposed Framework

The proposed framework is mainly edge intelligence-based, where GPR models for time series prediction and cross-correlated parameter prediction are developed at the edge node. For n training samples the computational complexity of GPR is $\mathcal{O}(n^3)$ in training. However, the complexity in prediction is $\mathcal{O}(I^x)$, where I^x is the total number of samples predicted in the x^{th} measurement cycle. c_{th}, l_p, and $\mathbf{ct}_{th}$ are estimated during training/retraining with complexity $\mathcal{O}(P)$. The complexity in finding τ is $\mathcal{O}(P\zeta_{max}) + \mathcal{O}(P^2\zeta_{max}) + \mathcal{O}(N')$, where $\mathcal{O}(P\zeta_{max})$ and $\mathcal{O}(P^2\zeta_{max})$ are the complexity in finding P^g_p and $\underline{\mathbf{C}}$ for P parameters over ζ^c_{max} slots and $\mathcal{O}(N')$ is the complexity in finding τ for N' sensor sets. Similarly, ζ^t for P parameters are estimated with complexity $\mathcal{O}(P)$ and the optimal active sensor set is selected with complexity $\mathcal{O}(N')$. Thus, the total complexity in training/retraining and estimating the hyper parameters is $(N' + P)\mathcal{O}(n^3) + \mathcal{O}(P)$. The computational complexity at the edge node in the x^{th} measurement cycle is $\mathcal{O}(I^x) + \mathcal{O}(P\zeta_{max}) + \mathcal{O}(P^2\zeta_{max}) + 2\mathcal{O}(N') + \mathcal{O}(P)$. The sensor node is also programmed with GPR_1 model to predict the temporal samples of the sensing parameters. Hence, the complexity of the sensor node during training/retraining is $P\mathcal{O}(n^3)$ and prediction in the x^{th} cycle is $\mathcal{O}(I^x)$.

VI. Results and Discussion

The simulation results of the proposed framework, described in Section V are discussed in this section. An air pollution monitoring setup, consisting of six sensors has been deployed in the campus to capture the variation of six parameters ($PM_{2.5}$, CO, NH_3, NO_2, Ozone, SO_2) in the environment and store the data in the memory [31]. The dataset, retrieved from the sensor node is used to simulate the proposed algorithm in MATLAB. From [31] and [4], the energy consumption of the six sensors for monitoring six parameters are listed in Table II. To decide the 'good' and 'bad' states of the parameters, the respective threshold values considered for the parameters are also listed in Table II. The threshold values are chosen based on the dataset and the satisfactory level of the parameters. The



(a) $PM_{2.5}$ (b) CO (c) NH_3

(d) NO_2 (e) Ozone (f) SO_2

Fig. 4: Variation of temporal prediction error and temporal correlation with lead value ζ^t.

parameters are assumed to be in 'bad' state when the signals exceed their respective threshold values.

A. Finding Optimum Values of Hyper-Parameters

In the proposed framework, the performance of the prediction models are evaluated by choosing appropriate values of temporal correlation threshold $\mathbf{ct}_{th} \in \mathbb{R}^{1\times P}$ for P parameters and cross-correlation threshold c_{th}. The sensing error for each parameter is calculated in terms of normalized mean squared error (nMSE) using (22).

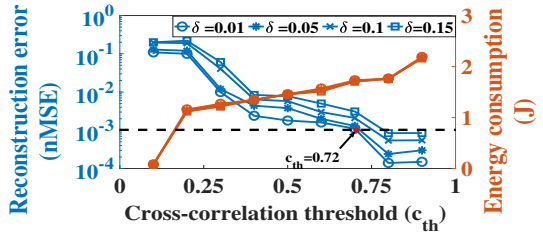
If $\mathbf{Z}_p^x$ and $\widehat{\mathbf{Z}}_p^x$ are respectively the measurement vectors of the actual and predicted/reconstructed data sequence of the p^{th} parameter in x^{th} measurement cycle, the nMSE is given by,

$$nMSE_p^x = \frac{||\mathbf{Z}_p^x - \widehat{\mathbf{Z}}_p^x||}{||\mathbf{Z}_p^x||}. \tag{22}$$

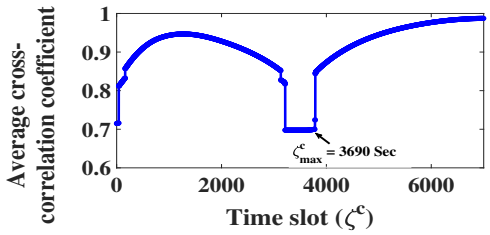
Initially, the temporal prediction error of each parameter is observed for different lag values. In [4], the nMSE achieved is on the order of 10^{-5}. Hence, nMSE $= 10^{-5}$ is considered as the temporal prediction error threshold et_{th}^b when the parameters are in 'bad' state. It ensures good accuracy of the recovered signal at the edge node during 'bad' state. On the other hand, nMSE $= 10^{-4}$ is chosen as temporal prediction error threshold et_{th}^g in 'good' state. Since the signals need not be predicted accurately in 'good' state, energy consumption gets more priority than accuracy. However, the optimal lag value for each parameter of the temporal prediction model GPR_1 is chosen to achieve error below 10^{-5}. As shown in Fig. 3, the lag values for the parameter set {PM, CO, NH_3, NO_2, Ozone, SO_2} are set as $\{l_p, 1 \le l \le P\} = \{3, 2, 1, 3, 5, 2\}$. When the p^{th} sensor is turned on, it collects l_p consecutive samples. The next consecutive samples are predicted from previous l_p collected/predicted samples until the prediction error exceeds the threshold at that state. To reduce complexity at the sensor node, Nyquist sampling interval t_p for each parameter is computed at the beginning while training GPR_1 model and is not updated at every cycle. The values are recomputed at the time of retraining the GPR_1 model. The sampling interval set, obtained from the simulation is $\{21, 65, 22, 75, 58, 41\}$.

The variation of temporal prediction error and temporal correlation with the increase in lead value ζ^t for each parameter is shown in Fig. 4. As discussed in Section V-B, the relation between temporal prediction error and ζ^t is established as, $et_p(\zeta^t) = \alpha_p \zeta^{t\beta_p}$, which is validated by Fig. 4. α_p and β_p for p^{th} parameter is obtained from the initial training samples using (17). Since the temporal correlation decreases and prediction error increases with ζ^t, a temporal correlation threshold $ct_{p,th}^g$ ($ct_{p,th}^b$) is set for p^{th} parameter, such that the prediction error remains below et_{th}^g (et_{th}^b). Thus, from Fig. 4, $\mathbf{ct}_{th}^g$ and $\mathbf{ct}_{th}^b$ for P parameters are respectively set as $\{0.96, 0.95, 0.93, 0.95, 0.96, 0.96\}$ and $\{0.98, 0.97, 0.955, 0.965, 0.99, 0.985\}$. If $P_P^g \ge P_{th}^g$, $ct_{p,th}^g$ is used to solve (15), else $ct_{p,th}^b$ is used to find ζ_p^t for p^{th} parameter in (15).

The 'bad' state parameters of the active set are predicted with an error below 10^{-5}, whereas the 'good' state parameters of the active set are predicted with an error below 10^{-4} using GPR_1 at both the sensor node and the edge node. The parameters of the inactive set are predicted from the parameters of the active set using GPR_2, as discussed in Section V-D. After predicting all parameters of the inactive set, the signals are reconstructed at the edge node and the reconstruction error is calculated in terms of nMSE. It is observed that the multi-sensing strategy, proposed in [4], achieved nMSE on the order of 10^{-5} due to joint prediction at the network level, where the spatial samples of the parameters are were also available. However, the proposed model in [4] is completely dedicated to the node-level analysis of a multi-parameter sensor hub. Applying the adaptive multi-sensing strategy of [4] in the proposed model, the average signal reconstruction error achieved is 2.1×10^{-3}. Hence, the reconstruction error (nMSE) threshold for GPR_2 is set at 10^{-3} to find optimum values of c_{th} and δ. As shown in Fig. 5a, the reconstruction error decreases and energy consumption increases with c_{th}. Thus, $c_{th} = 0.72$ is set as the optimum value. It is observed in Fig. 5a that the error increases with δ, whereas $\delta = 0.01$ results in a small length of measurement cycle which in turn increase the communication energy. Hence, an optimum value

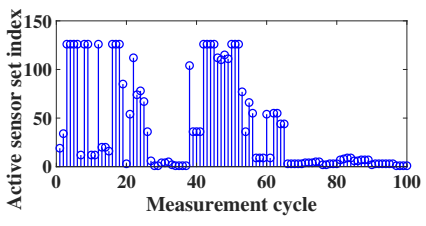


(a)

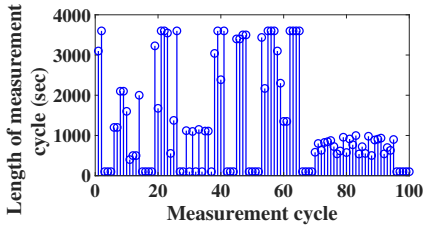


(b)

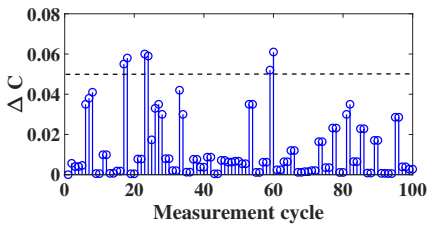
Fig. 5: (a) Average energy consumption and reconstruction error versus cross-correlation threshold and (b) Variation of cross-correlation coefficient with ζ^c.



(a)



(b)



(c)

Fig. 6: (a) Optimal active sensor set, (b) adaptive length of measurement cycle, and (c) deviation of average cross-correlation coefficient of the active set.

of δ is set as 0.05.

As discussed in Section V-A, the length of measurement cycle for each sensor set is obtained using (14), where the time slot ζ^c is bounded to $[\zeta^c_{min}, \zeta^c_{max}]$. As shown in Fig. 5b, the average cross-correlation factor of N' sets is changing with time. With slot duration $\Delta_2 = 1$ sec, ζ^c_{max} is set as 3690. Hence the maximum length of measurement cycle is 3690 sec, whereas the minimum length of measurement cycle is the set as the minimum value of all the Nyquist sampling intervals of the parameters i,e. 21 sec. The noise variance is considered as $\sigma^2 = 10^{-6}$ for all the parameters [11].

B. Validation of the Proposed Algorithm

The proposed algorithm is simulated over 100 measurement cycles to validate the efficacy of the algorithm. Fig. 6(a) captures the variation of optimal active sensor set at different measurement cycles. The variation of length of measurement cycle is shown in Fig. 6(b), which shows that the τ remains within the bound $[21, 3200]$ sec. ΔC = is the change in cross-correlation factor of the selective sensor set over the cycle. It can be observed that ΔC remains below $\delta = 0.05$ in almost every cycle. Although it exceeds 0.05 in some cycles, the deviation is very small. The average of ΔC is $0.018 < 0.05$, which validates the efficiency of the algorithm. Fig. 6 shows that the proposed algorithm is adaptive to the process dynamics.

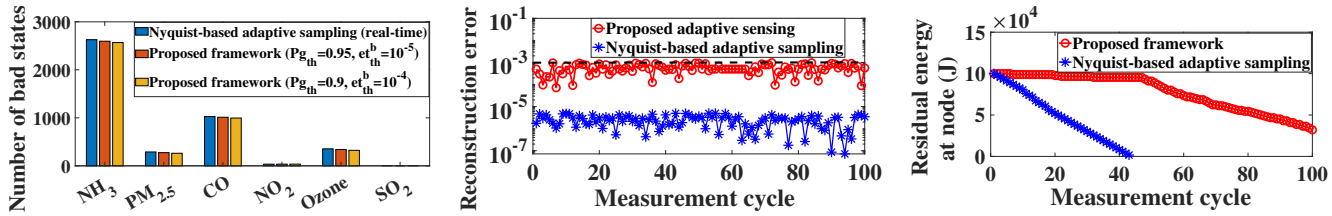
C. 'bad' states Detection Accuracy

As discussed in Section III, the proposed system model is statically defined. If any parameter crosses above the 'bad' threshold value, the system responds on real-time basis. Otherwise, the system responds on non-real-time basis. To detect the state of the system, the parameter values are either predicted or collected at the sensor node at their individual Nyquist sampling rate. To find the accuracy of the proposed algorithm in detecting 'bad' states of the system, it is compared with the Nyquist-based adaptive sampling algorithm proposed in [1]. In the Nyquist-based algorithm, all the sensors periodically collect data at their individual sampling intervals and transmit sporadically to the edge node. However, in the proposed model the data is transmitted sporadically during 'bad' state. In Fig. 7(a), it has been observed that the 'bad' states detection accuracy increases with higher P^g_{th} and lower et^b_{th}. Considering 100% 'bad' states detected in Nyquist-based model, the proposed model detects 98% 'bad' states with $P^g_{th} = 0.95$ and $et^b_{th} = 10^{-5}$. When $P^g_{th} = 0.9$ and $et^b_{th} = 10^{-4}$, the 'bad' state detection accuracy is 96%. Hence, required accuracy can be achieved by by adjusting P^g_{th} and et^b_{th}. A comparison of reconstruction error and residual energy between the proposed framework and Nyquist-based model are respectively shown in Fig. 7(b) and Fig. 7(c) considering $P^g_{th} = 0.95$ and $et^b_{th} = 10^{-5}$. It has been observed that the signal reconstruction error (nMSE) in the proposed framework is much higher than the Nyquist-based framework due to prediction errors of GPR_1 and GPR_2, however, it is below the threshold (10^{-3}). Fig. 7(c) shows that the proposed framework is more energy-efficient compared to [1].

TABLE III: Comparison with state-of-the-art

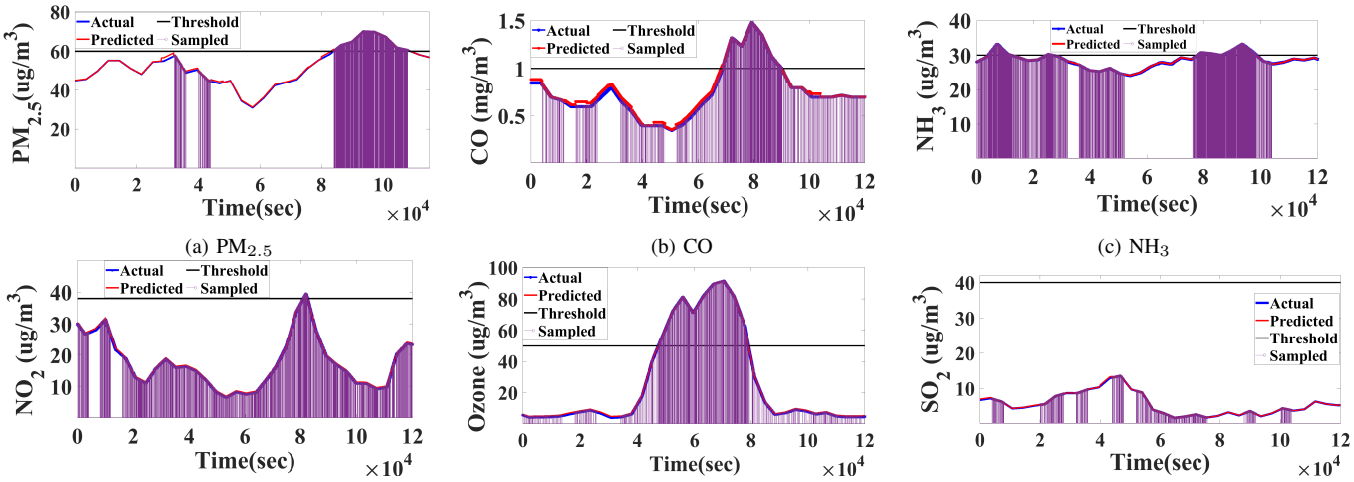
	Proposed framework	Nyquist-based framework
Sensing energy (J)	264.3	2210
Communication energy (J)	0.0055	0.0998
Number of samples	68	810
Reconstruction error	7×10^{-4}	2.5×10^{-6}
'bad' states detection efficiency	98%	100%

A performance comparison of the proposed framework with



(a) (b) (c)

Fig. 7: Comparison of (a) number of 'bad' state detected, (b) residual energy at the sensor node, and (c) reconstruction error of the proposed framework with Nyquist-based framework [1].



(a) $PM_{2.5}$ (b) CO (c) NH_3

(d) NO_2 (e) Ozone (f) SO_2

Fig. 8: Comparison of actual and reconstructed sensing signals.

the Nyquist-based framework in terms of average sensing energy consumption, average communication energy consumption, average number of samples, average reconstruction error, and 'bad' states detection accuracy is shown in Table III.

Remark 1: *The proposed framework is* 88% *more energy-efficient and* 91% *more bandwidth-efficient compared to the Nyquist-based adaptive sampling algorithm [1] with moderate complexity, while maintaining a sensing error below* 10^{-3} *and detecting the system states with* 98% *accuracy.*

D. Reconstruction Performance Evaluation

The comparison of actual and reconstructed sensing signals at both 'good' and 'bad' states is presented in this subsection. As shown in Fig. 8, the signals are sampled at a faster rate when they exceed the threshold. The sensors are not collecting samples when they are inactive.

The reconstruction error (nMSE) of all the six parameter are presented in Fig. 9. It is observed that the error is below 10^{-5} when the parameters are in 'bad' state. However, the error is bounded to $[10^{-5}, 10^{-3}]$ when the parameters are in 'good' state. Fig. 8 and Fig. 9 validates the efficiency of the GPR-based prediction models, discussed in Section IV.

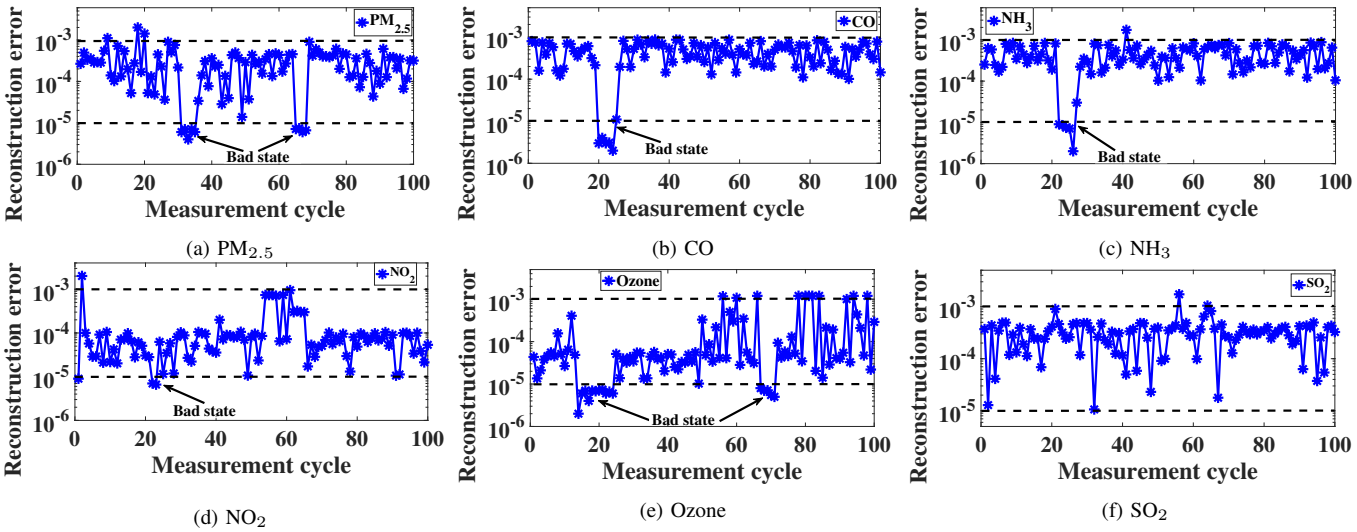
E. Performance Comparison with State-of-the-Art

In the proposed framework, both temporal- and cross-correlation of the parameters are exploited. A performance comparison in terms of error and energy efficiency between the proposed signal recovery and the SBL-based approach, proposed in [4], is studied in this subsection.

Since the SBL-based prediction framework in [4] considers the system as non-real-time, it is unable to detect the time-criticality of the system. On the other hand, the proposed framework detects the time-criticality of the system with 98% accuracy. It is assumed that the active sensors in [4], collect samples at Nyquist sampling rate. However, due to time series prediction model at the edge node and the end node in the proposed framework, the active sensors collect only a few samples, which gives a huge reduction in energy consumption and bandwidth utility. Considering NB-IoT as the radio module, the communication energy ($Ec = Ew + Et$) is computed using (13), where each sample is represented in 32 bits [31], [32].

As shown in Fig. 10(a), GPR-based prediction models performs better than SBL-based model in terms of accuracy. Although the proposed framework consumes more energy and bandwidth in 'bad' states compared to 'good' states, Figs. 10(b)-(e) depict that the proposed framework is more energy and bandwidth-efficient compared to SBL-based approach [4]. The average sensing error in the proposed and SBL-based framework are respectively 7×10^{-4} and 2.1×10^{-3}.

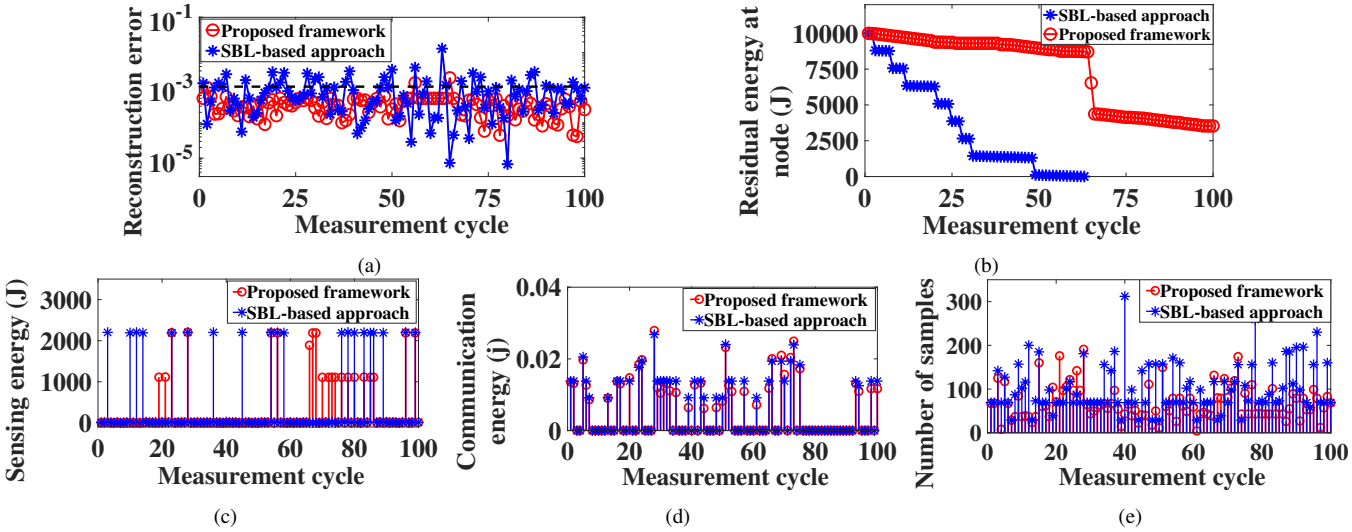
Remark 2: *The proposed framework saves* 41% *sensing energy,* 67% *communication energy, and* 32% *bandwidth compared to the framework in [4] with* 68% *reduction in sensing error, while detecting the system states with* 98% *accuracy.*



(a) $PM_{2.5}$ (b) CO (c) NH_3

(d) NO_2 (e) Ozone (f) SO_2

Fig. 9: Reconstruction error of the parameters at the edge node.



(a) (b)

(c) (d) (e)

Fig. 10: Performance comparison of the proposed framework with SBL-based approach [4].

TABLE IV: Performance comparison of the proposed framework with and without RL

	Without RL	with RL (UCB)	with RL (Q-learning)
Average energy consumption	340.35 J	264.3 J	264.2 J
Average reconstruction error	9×10^{-4}	7×10^{-4}	6.4×10^{-4}
Computational complexity	$\mathcal{O}(N)$	$\mathcal{O}(N)$	$\mathcal{O}(NI_{itr})$

F. Sensitivity of the Proposed Framework to RL-based Optimization Functions

To analyze the efficiency of the proposed discounted UCB-based adaptive sensor selection algorithm, the performance matrices are compared using RL-based optimization function (21) and without RL-based optimization function (19). As discussed in Section V-C, the proposed framework uses discounted UCB algorithm to find optimal active sensor set for the next measurement cycle, where γ and α' in (21) are respectively set as 0.9 and 0.95 [30]. However, Q-learning algorithm is also used widely to solve RL problems. A performance comparison of discounted UCB and Q-learning-based optimization function is shown in Table IV. Table IV also lists the performance matrices without using any RL algorithm in the optimization function as given in (19). Here the Q-learning is used as a stateless variant with ϵ-greedy method, as discussed in [33]. The values of ϵ, learning rate, and decay are respectively set as 1, 0.9, and 0.95. In Q-learning, the Q-value of a sensor set is estimated iteratively. If I_{itr} is the number of iterations, the complexity to find optimal sensor set out of N' sets is $\mathcal{O}(N'I_{itr})$.

Remark 3: *The energy consumption and error in RL-based methods are much lower compared to without RL-based method. Although Q-learning performs slightly better, the computational complexity of Q-learning is much higher than that of discounted UCB.*

VII. Conclusion

A novel data-driven edge intelligence-based priority aware sensing and transmission framework has been presented in this paper. The cross-correlation among the parameters and temporal correlation of the sensing signals have been exploited

to find the optimal active sensor set and optimal sampling instants of the sensors. The optimal active set has been estimated by solving the trade-off among performance parameters using discounted UCB-based optimization function. Extensive simulation studies on air-pollution monitoring parameters have validated the efficacy of the proposed framework. The GPR-based prediction models perform better in the proposed framework compared to the SBL-based prediction model. The application of reinforcement learning in finding optimal sensor set in a non-stationary environment increases the efficacy of the proposed framework. The proposed framework is both energy and bandwidth-efficient compared to the existing real-time and non-real-time adaptive sensing algorithms, while detecting the system states with close to 100% accuracy.

As an immediate extension, the proposed framework will be implemented in a real-life sensor network to validate its efficiency. The sensor node will be wirelessly connected to an edge node to transmit data and receive instructions based on the intelligence algorithm programmed in the edge node. It will be interesting to study the energy efficiency of the proposed framework in real operation scenario applications, as there is a clear tradeoff of sensing and communication energy saving versus computational energy overhead.

APPENDIX A

Let $f(\mathbf{Z}) \sim GP(m(\mathbf{Z}), k(\mathbf{Z}, \mathbf{Z}'))\ \forall\ \mathbf{Z}, \mathbf{Z}'$ is a Gaussian process, where $m(\mathbf{Z}) = E[f(\mathbf{Z})]$ is the mean function and $k(\mathbf{Z}, \mathbf{Z}') = E[(f(\mathbf{Z}) - m(\mathbf{Z}))(f(\mathbf{Z}') - m(\mathbf{Z}'))]$ is the covariance function. If $\underline{\mathbf{Z}}$ is the training matrix with n feature vectors, $\mathbf{Y}$ is the target vector, and $\mathbf{W}$ is the weight vector, the prior and posterior distributions of the underlying function f for a test point $\mathbf{Z}_*$ are respectively given by,

$$Pr(f(\mathbf{Z}_*)) = \int_{\mathbf{W}} Pr(f|\mathbf{W}, \mathbf{Z}_*) Pr(\mathbf{W}) d\mathbf{W} \tag{A.1}$$

$$Pr(f(\mathbf{Z}_*)|\underline{\mathbf{Z}}, \mathbf{Y}) = \int_{\mathbf{W}} Pr(f|\mathbf{W}, \mathbf{Z}_*) Pr(\mathbf{W}|\underline{\mathbf{Z}}, \mathbf{Y}) d\mathbf{W}. \tag{A.2}$$

Assuming $Pr(\mathbf{W})$ is Gaussian and $Pr(f|\mathbf{W}, \mathbf{Z}_*)$ is deterministic, $Pr(f(\mathbf{Z}_*))$ and $Pr(f(\mathbf{Z}_*)|\underline{\mathbf{Z}}, \mathbf{Y})$ are also Gaussian [29, Ch. 2]. Equation (A.1) and (A.2) are used in Section IV to find the mean and covariance functions of GPR-based temporal and cross-correlated parameter prediction models.

APPENDIX B

The optimization function to select an optimal sensor set $\mathcal{A}$ in the $(x+1)^{th}$ measurement cycle is given by,

$$F^{x+1}(\mathbf{A}) = \sum_{t=1}^{x} \gamma^{x-t} \frac{\frac{{E_0}^t}{E_{batt}} \frac{1}{AB} [\mathbf{A}\underline{\mathbf{C}}_1^t (\mathbf{U}^T - \mathbf{A}^T)] \tau_{\mathbf{A}}^t}{\nu^t \mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}^t} + \sqrt{\frac{\alpha'}{T_{\mathbf{A}}^x} log(\sum_{i-1}^{N'} T_{\mathbf{A}}^x)}. \tag{B.1}$$

In (B) $\nu^t = \max_{\mathbf{A} \in \mathcal{Q}^t} \frac{\frac{1}{AB} [\mathbf{A}\underline{\mathbf{C}}_1^t (\mathbf{U}^T - \mathbf{A}^T)] \tau_{\mathbf{A}}^t}{\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}^t}$ is constant for a measurement cycle and independent of $\mathbf{A}^t$. Similarly, $\frac{{E_0}^t}{E_{batt}}$ and $\tau_{\mathbf{A}_i}^t$ are also independent of $\mathbf{A}^t$. Although $T_{\mathbf{A}}^x$ in (B) is a function of $\mathbf{A}^x$, it is the number of times vector $\mathbf{A}$ is selected to activate up to x measurement cycles. Thus, (B) can be rewritten by replacing the constant terms with c_1 and c_2 as, $F^{x+1}(\mathbf{A}) = \sum_{t=1}^{x} c_1^t F_1^t(\mathbf{A}) + c_2^x$, where $F_1(\mathbf{A}) = \frac{\mathbf{A}\underline{\mathbf{C}}_1(\mathbf{U}^T - \mathbf{A}^T)}{\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}}$. If $F_1(\mathbf{A})$ is concave, $F(\mathbf{A})$ is also concave. The Frechet derivative of $F_1(\mathbf{A})$ is given by [34],

$$\begin{aligned}
F_1(\mathbf{A}+\mathbf{V}) - F_1(\mathbf{A}) &= \frac{(\mathbf{A}+\mathbf{V})\underline{\mathbf{C}}_1(\mathbf{U}^T - (\mathbf{A}+\mathbf{V})^T)}{(\mathbf{A}+\mathbf{V})\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}} - \frac{\mathbf{A}\underline{\mathbf{C}}_1(\mathbf{U}^T - \mathbf{A}^T)}{\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}} \\
&= \frac{(\mathbf{A}\underline{\mathbf{C}}_1\mathbf{U}^T + \mathbf{V}\underline{\mathbf{C}}_1\mathbf{U}^T - \mathbf{A}\underline{\mathbf{C}}_1\mathbf{A}^T - \mathbf{A}\underline{\mathbf{C}}_1\mathbf{V}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{A}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T)(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&\quad - \frac{(\mathbf{A}\underline{\mathbf{C}}_1(\mathbf{U}^T - \mathbf{A}^T))((\mathbf{A}+\mathbf{V})\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&= \frac{(\mathbf{V}\underline{\mathbf{C}}_1\mathbf{U}^T - \mathbf{A}\underline{\mathbf{C}}_1\mathbf{V}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{A}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T)(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&\quad - \frac{(\mathbf{A}\underline{\mathbf{C}}_1\mathbf{U}^T - \mathbf{A}\underline{\mathbf{C}}_1\mathbf{A}^T)(\mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&= \frac{(\mathbf{V}\underline{\mathbf{C}}_1\mathbf{U}^T - \mathbf{A}\underline{\mathbf{C}}_1\mathbf{V}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{A}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T)(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&\quad - \frac{(\mathbf{V}\underline{\mathbf{C}}_1\mathbf{U}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{A}^T)(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} \\
&= \frac{(-\mathbf{A}\underline{\mathbf{C}}_1\mathbf{V}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T)(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})}{(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M} + \mathbf{V}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})(\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M})} = \frac{(-\mathbf{A}\underline{\mathbf{C}}_1\mathbf{V}^T - \mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T)}{(\mathbf{A}+\mathbf{V})\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}}.
\end{aligned} \tag{B.2}$$

Putting $\mathbf{V}\underline{\mathbf{C}}_1\mathbf{V}^T = 0$ and $(\mathbf{A}+\mathbf{V}) \approx \mathbf{A}$, we get the Frechet derivative of $F_1(\mathbf{A})$ as, $\nabla F_1(\mathbf{A}) = \frac{-\mathbf{A}\underline{\mathbf{C}}_1}{\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}}$. Similarly, the Hessian of $F_1(\mathbf{A})$ is given by,

$$\nabla^2 F_1(\mathbf{A}) = \frac{0}{\mathbf{A}\underline{\mathbf{E}}\underline{\mathbf{s}}\mathbf{M}} = 0 \tag{B.3}$$

According to [34], a twice differentiable function $F(\mathbf{A})$ is concave if and only if $\nabla^2 F(\mathbf{A}) \leq 0 \forall \mathbf{A} \in \mathbb{R}^{P \times 1}$. Thus, the optimization functions in (19) and (21) are concave, which ensures a single maximum point exists. Since the domain of $\mathbf{A}$ is discrete, it makes a non-convex set. It can be converted to a convex set by considering the convex hull of $\mathcal{Q}^x$.

REFERENCES

[1] C. Alippi, G. Anastasi, M. Di Francesco, and M. Roveri, "An adaptive sampling algorithm for effective energy management in wireless sensor networks with energy-hungry sensors," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 2, pp. 335–344, 2009.

[2] V. Gupta, S. Tripathi, and S. De, "Green sensing and communication: A step towards sustainable IoT systems," *J. Indian Inst. Sc.*, pp. 1–16, 2020.

[3] H. Al-Tous and I. Barhumi, "Differential game for resource allocation in energy harvesting wireless sensor networks," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 4, pp. 1165–1173, 2020.

[4] V. Gupta and S. De, "Collaborative multi-sensing in energy harvesting wireless sensor networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 426–441, 2020.

[5] D. Rajavel, A. Chakraborty, and S. Misra, "QoS-aware sensor virtualization for provisioning green sensors-as-a-service," *IEEE Trans. Green Commun. Netw.*, pp. 1–1, 2021.

[6] S. Ghosh, S. De, S. Chatterjee, and M. Portmann, "Learning-based smart sensing for energy-sustainable WSN," in *proc. IEEE ICC, Montreal, Canada*, 2021, pp. 1–6.

[7] S. Tripathi and S. De, "Data-driven optimizations in IoT: A new frontier of challenges and opportunities," *Springer CSI Trans. ICT*, vol. 7, no. 1, pp. 35–43, 2019.

[8] H. Harb and A. Makhoul, "Energy-efficient sensor data collection approach for industrial process monitoring," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 661–672, 2017.

[9] A. Jain and E. Y. Chang, "Adaptive sampling for sensor networks," in *Proc. intl. wksp on Data manag. sensor networks: In conjunction with VLDB, Toronto, Canada*, 2004, pp. 10–16.

[10] S. Ghosh, S. De, S. Chatterjee, and M. Portmann, "Learning-based adaptive sensor selection framework for multi-sensing WSN," *IEEE Sensors J.*, pp. 1–1, 2021.

[11] V. Gupta and S. De, "SBL-based adaptive sensing framework for WSN-assisted IoT applications," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4598–4612, 2018.

[12] S. Hwang, R. Ran, J. Yang, and D. K. Kim, "Multivariated Bayesian compressive sensing in wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2196–2206, 2015.

[13] Y. Xu, J. Choi, S. Dass, and T. Maiti, "Sequential Bayesian prediction and adaptive sampling algorithms for mobile sensor networks," *IEEE Trans. Autom. Control*, vol. 57, no. 8, pp. 2078–2084, 2011.

[14] M. Jadaliha, Y. Xu, J. Choi, N. S. Johnson, and W. Li, "Gaussian process regression for sensor networks under localization uncertainty," *IEEE Trans. Signal Process.*, vol. 61, no. 2, pp. 223–237, 2012.

[15] D. Gu and H. Hu, "Spatial Gaussian process regression with mobile sensor networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1279–1290, 2012.

[16] T. Zheng, M. H. Bergin, R. Sutaria, S. N. Tripathi, R. Caldow, and D. E. Carlson, "Gaussian process regression model for dynamically calibrating and surveilling a wireless low-cost particulate matter sensor network in Delhi." *Atmospheric Measurement Techniques*, vol. 12, no. 9, 2019.

[17] S. Tripathi and S. De, "Dynamic prediction of powerline frequency for wide area monitoring and control," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 2837–2846, 2017.

[18] H. Harb, A. Mansour, A. Nasser, E. M. Cruz, and I. de la Torre Diez, "A sensor-based data analytics for patient monitoring in connected healthcare applications," *IEEE Sensors J.*, vol. 21, no. 2, pp. 974–984, 2020.

[19] M. V. Ramesh and V. P. Rangan, "Data reduction and energy sustenance in multisensor networks for landslide monitoring," *IEEE Sensors J.*, vol. 14, no. 5, pp. 1555–1563, 2014.

[20] K. Li, W. Ni, M. Abolhasan, and E. Tovar, "Reinforcement learning for scheduling wireless powered sensor communications," *IEEE Trans. Green Commun. and Netw.*, vol. 3, no. 2, pp. 264–274, 2019.

[21] R. Tan, G. Xing, X. Liu, J. Yao, and Z. Yuan, "Adaptive calibration for fusion-based wireless sensor networks," in *Proc. IEEE INFOCOM, San Diego, CA, USA*, 2010, pp. 1–9.

[22] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *Proc. IEEE INFOCOM, Orlando, FL, USA*, 2012, pp. 603–611.

[23] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Trans. Veh. Technol.*, vol. 46, no. 2, pp. 445–455, 1997.

[24] R. E. Caraka, R. C. Chen, T. Toharudin, B. Pardamean, H. Yasin, and S. H. Wu, "Prediction of status particulate matter 2.5 using state Markov chain stochastic process and hybrid var-nn-PSO," *IEEE Access*, vol. 7, pp. 161 654–161 665, 2019.

[25] P. Mukherjee, D. Mishra, and S. De, "Gaussian mixture based context-aware short-term characterization of wireless channels," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 26–40, 2019.

[26] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling," *Philosophical Trans. Royal Soc. A: Mathematical, Physical and Engineering Sciences*, vol. 371, 2013.

[27] C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford University press, 1995.

[28] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as Gaussian Processes," *arXiv preprint arXiv:1711.00165*, 2017.

[29] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT Press: Cambridge, MA, 2006.

[30] T. Lattimore and C. Szepesvári, *Bandit algorithms*. Cambridge University Press, 2020.

[31] P. Das, S. Ghosh, S. Chatterjee, and S. De, "Energy harvesting-enabled 5G advanced air pollution monitoring device," in *Proc. IEEE 3rd 5G World Forum, Bangalore, India*, 2020, pp. 218–223.

[32] Q. W. S. Co. Lte bc66 nb-iot module. [Online]. Available: https://www.quectel.com/product/bc66.htm

[33] T. B. de Oliveira, A. L. Bazzan, B. C. da Silva, and R. Grunitzki, "Comparing multi-armed bandit algorithms and Q-learning for multi-agent action selection: a case study in route choice," in *Proc. IJCNN, Rio de Janeiro, Brazil*, 2018.

[34] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

Sushmita Ghosh received the B.Tech degree in Electronics and Communication Engineering in 2016 and M.Tech degree in VLSI Design in 2019 from National Institute of Technology Agartala, Tripura, India. She is currently pursuing the Ph.D. degree from the UQ-IITD Academy of Research, Indian Institute of Technology Delhi, New Delhi, India. Her current research interest includes design of energy-efficient IoT communication systems and networks.

Swades De (S'02-M'04-SM'14) is a Professor with the Department of Electrical Engineering, IIT Delhi. His research interests are broadly in communication networks, with emphasis on performance modeling and analysis. Current directions include energy harvesting sensor networks, broadband wireless access and routing, cognitive/whitespace access networks, smart grid networks, and IoT communications.

Shouri Chatterjee (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from IIT Madras, Chennai, India, in 2000, and the M.S. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2002 and 2005, respectively. From 2005 to 2006, he was a Design Engineer with the Wireless Division, Silicon Laboratories Inc., NJ, USA. Since November 2006, he has been with the faculty of the Department of Electrical Engineering, IIT Delhi, New Delhi, India, where he is currently a Professor and NXP/Philips Chair Professor. His primary area of work is in integrated circuit design, circuits, and systems. His research interests include filter design, ultra-low power circuits for energy scavenging, and wireless front-ends for millimeter-wave radios.

Marius Portmann received his Ph.D. degree in Electrical Engineering from the Swiss Federal Institute of Technology (ETH), Zurich, in 2002. He is currently an Associate Professor with the University of Queensland, Australia. His research interests include general networking, pervasive computing, IoT and cyber security