IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 29, 2020, accepted July 13, 2020, date of publication July 23, 2020, date of current version August 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3011496

User-Driven Adaptive Sampling for Massive Internet of Things

LE KIM-HUNG, (Member, IEEE), AND QUAN LE-TRUNG
Faculty of Computer Networks and Communications, University of Information Technology, Vietnam National University Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam

Corresponding author: Quan Le-Trung (quanlt@uit.edu.vn)

This research was supported by Vingroup Innovation Foundation (VINIF) in project code VINIF.2019.DA01.

ABSTRACT Energy conservation techniques are crucial to achieving high reliability in the Internet of Things (IoT) services, especially in the Massive IoT (MIoT), which stringently requires cost-effective and low-energy consumption for battery-powered devices. Most of the proposed techniques generally assume that data acquiring and processing consume significantly lower than that of communication. Unfortunately, this assumption is incorrect in the MIoT scenario, which mostly involves the low-power wide-area network (LPWAN) and complex data sensing operations (e.g., biological and seismic sensing) using ''power-hungry'' sensors (e.g., gas sensors, seismometers). Thus, sensing actions may consume even more energy than transmission. In addition, none of them support end-users in controlling the trade-off between energy conservation and data precision. To deal with these issues, we propose an adaptive sampling algorithm that estimates the optimal sampling frequencies in real-time for IoT devices based on the changes of collected data. Given a user's saving desire, our algorithm could minimize the device's energy consumption while ensuring the precision of collected information. Practical experiments over IoT datasets have shown that our algorithm can reduce the number of acquired samples up to 20 times compared with a traditional fixed-rate approach at extremely low Normal Mean Error value around 3.45%.

INDEX TERMS Constrained devices, energy efficiency, massive internet of things, sampling algorithm.

I. INTRODUCTION

Recently we have been witnessing the explosion of the Internet of Things, aiming to bring every physical object into digital worlds [1]. Leveraging the novel advantages in microelectronic and telecommunication, the smart devices in IoT significant increase in both quantity and quality perspectives. A massive number of these devices are deployed to enhance human awareness about surrounding objects. According to Cisco, 1 trillion smart sensors are connected to the Internet by 2020, and up to 45 trillion in the next 20 years [2]. Relied on real-time analytic insights on collected data, IoT offers disruptive opportunities to maximize business profits and user experiments in several areas, from manufacturing, logistics to transportation, health care. Boston Consulting Group reported that business to business spending on IoT technologies, applications, and solutions reaches 267 billion dollars by 2020, especially in manufacturing, transportation and logistics, and utilities [3].

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenyu Zhou.

As a result of significant growth, Massive Internet of Things (MIoT) has been emerging as a novel technology referring to a large volume of constrained IoT devices that stringently require excellent coverage, cost-effective, and low-energy consumption. Among several cutting-edge connectivity solutions for MIoT, the LPWAN technologies such as Sigfox and LoRA have been considered as the most potential candidates, while cellular-based connectives such as 5G or NB-IoT are under developing and testing processes. Although these protocols support low-power transmission via the radio wave, energy consumption sources may come from other parts of the devices, such as processing operations, sensing actions. On the other hand, frequently collecting and analyzing a large amount of raw monitoring data from plugged sensors on IoT devices notably consume both resources and times due to involving expensive operations, such as powering sensing components or performing complex queries [4]. If these devices are battery-powered, such operations quickly drain their power source capacity, to the point that it may lead to suddenly interrupt all running operations and strongly impact the whole MIoT system.

Moreover, the battery has a limited energy capacity. Recharging or replacing such battery is extremely costly or even impossible because the IoT devices may be deployed in hostile environments (e.g., under the sewer networks or in the deep forest). Thus, it is no wonder why energy consumption remains a vital challenge to achieve energy-efficiency requirements in IoT, specially MIoT [5]. The MIoT scenario in our work refers to a massive number of constraint devices (battery-power, limited memory, and CPU) connecting to the Internet using lower power connections (such as LPWAN, NB-IoT, and 5G). These devices may equip ''power-hungry'' sensors and be deployed in wide geographic regions.

In general, a typical IoT device has four principal components:

- A sensing subsystem senses and collects information from the environment.
- A processing subsystem manages all device operations.
- A communication subsystem transmits the collected data.
- A power source supplies the energy needed for all device operations.

The device must have a sufficient lifetime to fulfill the application requirements [6]. Many approaches have been proposed to minimize energy consumption such as compressing data [7], [8], aggregating data before sending [9], [10], and predictive monitoring [11], [12]. They all target minimizing the radioactivity as they assume that the communication subsystem is the most consumed energy source, and the energy consumption of sensing and processing subsystems are negligible. However, in IoT devices, the sensing subsystem may consume more energy than other device components. With the exponential growth of the Internet of Things and its applications, the IoT sensors supporting environmental information perceptions increase in both quality and quantity. The sensors are more complicated and able to deeply aware of environmental information. However, these sensors require high energy consumption to accomplish these tasks, namely ''power-hungry sensors.'' For example, a catalytic gas sensor consumes massive energy (about 600 mW) to power a bridge circuit containing the wiring resistances to detect the heat, which is released by catalytic oxidation between combustible gases and a platinum treated wire coil. In contract, the transmission protocols in the MIoT context become more energy-efficiency due to the emergence of LPWAN technologies, such as Sigfox, Lora, and NB-IoT. Table 1 and 2 present the power consumption of radio chips and common sensors, respectively. We observe that sensing operations consume significantly more energy than transmission. For example, Semtech SX1272 radio chip needs 26 mW to transmit data, whereas gas sensor Hanwei-MC needs more than twenty-time energy (about 600 mW) to detect the presence of gases in an area. The consumption is caused by various factors [13]:

- Power-hungry transducers: Many sensor types use high power resources to perform sensing tasks such as multimedia sensors or chemical sensors.

TABLE 1. Power consumption of existing sensor chips [14].

Sensor Name	Sensing Type	Power Consumption
Hanwei-MC	Catalytic gas sensor	600 mW
GE-AD81	Semiconductor gas sensor	650 mW
LUC-M10	Level Sensor	300 mW
STM-TDA0161	Proximity	420 mW
Turck-AP8X	Flow Control	1250 mW

TABLE 2. Power consumption for transmission of existing Radio Chips [15].

Radio Chip	Power Consumption
Microchip RN2483	38.9 mW
Semtech SX1272	26 mW
Semtech SX1276	46 mW
Texas CC2420	35 mW
Texas C1000	42 mW

- Power-hungry Analog/Digital converter: Some sensors need converting collected data from analog to digital formats.
- Long acquisition time: Some sensing operations may require from seconds to minutes.

Therefore, saving energy in the communication subsystem is not enough. The IoT systems need to be concerned about the sensing subsystem's energy by decreasing the number of acquisitions.

In general, the energy consumption of all subsystems (including sensing, processing, and communicating) highly relates to sampling frequency either directly or indirectly. By default, battery-power devices are in sleep mode and only waked-up to collect samples via sensing subsystems. These samples are then required to transmit to gateways directly or locally process on the device after successfully collecting. A typical data collection process of a device includes three main steps: (1) the device waking up the sensing subsystem to perform environmental information perception. (2) the sensed data is then stored in temporary memory and pre-processed by the processing subsystem. (3) The communication subsystem transmits the data to the gateway or cloud server for further processing. Thus, the data collection process highly relates to all subsystems' energy consumption, including environmental information perception via sensing action. Among methods used to decrease sampling frequency, *Adaptive Sampling* is a technique for dynamically adapt the sampling rate to defined metrics [16]. When the metric stream is stable, it automatically reduces the sampling rate to conserve energy. In contrast, when the input stream's fluctuation is detected, the sampling rate is increased to collect more data about current events. Despite owning several advantages, applying existing adaptive sampling techniques into MIoT context is still a challenge, because: (1) The adaptation process requires a large volume of data [17], which are not stored or process by MIoT devices; (2) They require

data-specific thresholds [18] or fixed metric distribution [19]; (3) There is a delay in detecting fluctuations in the data [20]. Moreover, none of them supports end-users in expressing their desired level of saving energy. In other words, the end-users can not control the sampling rate. This limitation is critical in MIoT because each radio connectivity technology has a maximum number of messages per day [21]. Network providers immediately banned the devices violating this limitation. On the other hand, through the desired saving level, the end-user is able to control the trade-off between the amount of saved energy and data precision. Enabling this capability makes the adaptive sampling method suitable for various MIoT use cases, which demand different priorities for saving energy. For example, fire-forest detection applications require higher priority for data correctness than conserving energy, whereas soil monitoring applications focus more on preserving energy.

To address the mentioned limitations, we propose an *User-driven Adaptive Sampling Algorithm* (UDASA) that estimates in real-time the optimal sampling frequency for IoT devices based on the changes of data in history. By taking into account a user-desired saving level, our algorithm ensures output frequency in a corresponding range, while maximizing the correctness of collected information. As a result, UDASA enables end-users to control the trade-off between conserving energy and data precision. Practical experiments on NOAA and IoT datasets have shown that UDASA reduces the number of acquired samples up to twenty times compared to a traditional fixed-rate approach at low error around 6.4% and 3.45%, respectively. These error values mean that reconstructed data from samples are 93.6% and 96.55% similar to the original. Our key contributions are summarized below:

- An UDASA for prolonging device lifetime in Massive IoT context is presented. The proposed algorithm could estimate the optimal sampling frequency in real-time following a given saving level from end-users.
- The robustness and applicability of our proposal are validated over real datasets with respect to various parameter settings. Enabling end-users to control the saving level makes the algorithm more applicable to several user-cases in the MIoT context.
- We compare the superiority of UDASA with the existing adaptive sampling algorithms in terms of energy conservation and data precision.

The remainder of the paper is organized as follows. In Section II, we formalize the problem of adaptive sampling with user-desired saving levels and related definitions. The UDASA algorithms are presented in Section III. Section IV reports the evaluation of our method through real datasets. Section V discusses related work, and conclusions are reported in Section VI.

II. PRELIMINARIES

A. RELATED DEFINITIONS AND CONCEPTS

Definition 1 (Median Absolute Deviation): Given dataset X including n data points $X_1, X_2, \ldots, X_n$, the Median Absolute Deviation (MAD) of data point $X_i \in X$ collected at time i, denoted as $\triangle\theta(X_i)$, which is defined that:

$$\triangle\theta(X_i) = median(|X_i - median(X)|) \tag{1}$$

TABLE 3. Table of notations.

Notation	Description
X_i	A data point is sampled from a data stream at timestamp t_i
T_{i+1}	Sampling interval at time $i+1$, where X_i is the last collected data point. T_{i+1} is limited in a specific range of values from T_{min} to T_{max}
T_{base}	Minimum sampling interval is supported by MIoT connectivity
n	Saving levels is desired by end-users
W_N	A last data window includes N data points.
$\overline{W_N}$	Median Absolute Deviation value of W_N.
$\triangle\theta(X_i)$	Median Absolute Deviation value of data point X_i

Definition 2 (Linear Interpolation): In mathematics, linear interpolation is a curve fitting method. It is used to compute new data points within given data points by using linear polynomials. For example, given 2 data points (x_0, y_0) and (x_1, y_1), a new data point (x, y) with $x \in (x_0, x_1)$ is interpolated from:

$$\frac{y - y_0}{x - x_0} = \frac{y_1 - y_0}{x_1 - x_0} \tag{2}$$

Definition 3 (Sigmoid Function): Sigmoid function refers to a continuous, monotonically increasing function that is the special case of the logistic function. It has a characteristic S-shaped curve, which is defined that:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

In general, the sigmoid function is bounded by a pair of horizontal asymptotes $0 < S(x) < 1$ when $x \to \infty$.

B. ADAPTIVE SAMPLING PROBLEM DEFINITION

Denote a data stream X, *periodic sampling* is a T interval process (every T time units) of collecting a data instance from X. The T value is also known as the sampling rate. Thus, data point X_i is collected at time $t_i = i * T$. For example, if $T = 1$ seconds, 20^{th} data point is collected at $t_{20} = 20$ seconds. The simplicity of periodic sampling is well-fitted with constrained devices, which have limited computational resources. For this reason, it is widely used for IoT devices. However, selecting an effective T value is very challenging because it is data-specific. Considering the data stream illustrated in the top plot of Figure 1, if the sampling rate (T value) is low, the IoT device generates and transfers a massive volume of data over the network. This behavior seriously damages the device battery and cloud resources required to process such data. In case the T value is high, the notable events may be ignored. For example, as shown in the second plot of Figure 1, the events occurred at T_{20}, T_{60}, and T_{120} are undetected due to increasing T value from

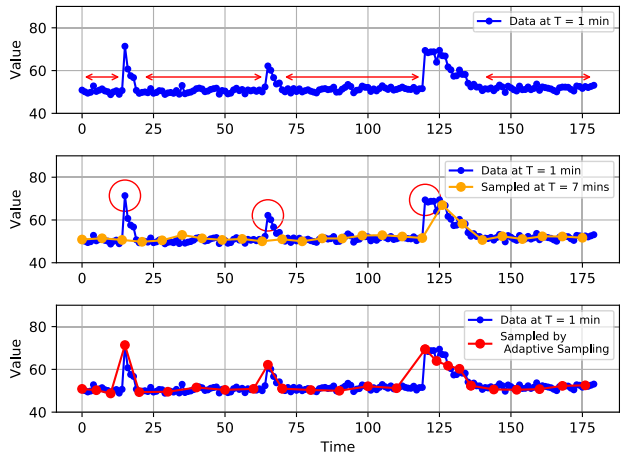

FIGURE 1. Periodic and adaptive sampling.

1 minute to 7 minutes. Therefore, periodic sampling is ineffective for saving energy and guaranteeing data precision.

Adaptive sampling is a potential approach to mitigate periodic sampling problems. It is defined as a collecting process using dynamical sampling interval $T \in [T_{min}, T_{max}]$, which is calculated from data stream X based on estimation models, denoted as $f(X)$. Suppose X_i is the latest sample from X and X_* is a reconstructed data stream of M after performing adaptive sampling. The difference of X and X_* is denoted as ERR, which evaluates the accuracy of the estimation model. The goal of adaptive sampling is to provide a maximum T value to collect sample X_{i+1} from X based on the estimation model $f(X)$, while minimizing ERR and guaranteeing $T \in [T_{min}, T_{max}]$. Thus, the problem is formalized into the following equation:

$$T = \underset{T}{argmax} \{ f(X_i, X, ERR) \mid argmin\{ERR\}, \quad T \in [T_{min}, T_{max}]\} \tag{4}$$

As we can see from (4), $ERR \rightarrow 0$ may lead sampling interval $T \rightarrow T_{min}$ that reduces energy-saving performance. Thus, an effective adaptive sampling algorithm is capable of balance the data accuracy and preserving energy to fulfill given user-desired saving level.

III. USER-DRIVEN ADAPTIVE SAMPLING ALGORITHM

To mitigate the discussed limitations, we propose a light-weight user-driven adaptive sampling algorithm to improve energy-efficiency, while maintaining the high accuracy of collected information. In this section, we first present the overall of our proposal. Then, we discuss each step along with related definitions.

The general idea of UDASA is to dynamically adapt the sampling frequency to the changes in sensed data in history. Intuitively, a higher frequency is preferred to fully aware of the events when there are significant changes (high variance). For example, in forest fire warning services, the sudden increases of current temperature are considered as notable events (e.g., forest fire). Increasing the device frequency to collect more data may help to deeper investigating such events. In contrast, if the observed values are hardly fluctuated, decreasing the frequency is demanded to reduce energy consumption in sensing, processing, and transmitting operations. To actualize the above idea, we exploit an enhanced sigmoid function to quickly adapt the sampling frequency to the sudden changes in collected data D following given user-desired saving level n. In theory, the valid value of n is from 1 to ∞. However, from the experimental evaluation on real datasets, we recommend that the n value should be from 1 to 20.

Let T_{i+1} denote the sampling interval at time $i+1$, where X_i is the last collected data point. Given the minimal sampling interval of MIoT system is T_{base} (For example, T_{base} of Sigfox device is 10 minutes. It is about 144 messages per day), saving level n, we calculate:

$$T_{i+1} = f_{change} * T_{base} \tag{5}$$

where

$$f_{change} = n + \frac{1-n}{1+e^{-n*D}} \tag{6}$$

with:

$$D = \triangle\theta_i(X_i) - \frac{n+1}{2} * (\overline{W_N}) \tag{7}$$

In (7), the D value is computed by comparing the change of the Median Absolute Deviation between incoming data X_i and the mean of such deviation over last N data points (denoted by $\overline{W_N}$).

$$\overline{W_N} = \frac{1}{N} \sum_{j=i-N}^{i} \triangle\theta(X_j) \tag{8}$$

In the enhanced sigmoid function defined in (6), the saving level n acts as an upper asymptote to ensure the next sampling interval lower than n times minimal sampling interval (T_{base}). In detail, if D value is sufficiently large (the current change of MAD is overwhelming their recent changes in last N data points), the value of f_{change} converges to 1. As a result, the next sampling interval is minimal to collect more data. In turn, if the change of incoming data is minor, next sampling interval converges to n times T_{base}, with n is the given saving level. Thus, in theory, the valid value of level n is from 1 to infinity. With $n = 1$, the output sampling interval equals T_{base}. In other words, there is no energy saving. Increasing n values are preferred in cases end users desire more saving. Note that, if the level n value is notably high, the upper asymptote is negligible. The next sampling interval only depends on the changes of incoming data (reflected by D value).

$$\lim_{D \to \infty} f_{change} = 1 \Rightarrow T_{min} \longrightarrow T_{base}$$
$$\lim_{D \to -\infty} f_{change} = n \Rightarrow T_{max} \longrightarrow n * T_{base}$$

The value $\frac{n+1}{2}$ in (7), known as "equity factor", is used to ensure that the change of current value $\triangle\theta_i(X_i)$ comparing

with $\overline{W_N}$ equal the change sampling interval T_{i+1} comparing with T_{base}. For example, if the change of current value equals $\frac{n+1}{2}$ times mean of this change over last N data points, the D value in (7) is 0. For this reason, the f_{change} value in (6) is $\frac{n+1}{2}$. This means the next sampling interval T_{i+1} equals $\frac{n+1}{2}$ times T_{base}.

The enhanced sigmoid function aims to minimize the consecutive sensed data with similar values by decreasing the sampling frequency when detecting the stability in collected data (the changes are minor). In contrast, increasing the sampling frequency is preferred to conserve data accuracy when there are significant changes. The natural boundaries (lower bound and upper bound) of the sigmoid function ensure the sampling frequency changes under control. The output frequency of our proposal is always higher than a base frequency (T_{base}). Thus, the accuracy of the collected data is highly preserved. The key challenge is to detect the changes quickly and robust to the outlier.

To solve this challenge, we exploit the Median Absolute Deviation (MAD) stable with outliers than the standard deviation (STD) that squares the distance from the mean. Thus, fluctuated data values strongly impact STD values, whereas the deviations of outliers are irrelevant to MAD values. As a result, the D function defined in (7) not only quickly detects the changes in data but is also robust to the outlier. On the other hand, as a property of sigmoid function, the upper and lower asymptotes of f_{change} function are n and 1, respectively. Thus, regardless of the change of incoming data reflected by D value, the next interval is in a specific range from $1 \rightarrow n$ times T_{base}. The other superiority of f_{change} function is that its converging rate is managed via n value. Figure 2 illustrates the f_{change} function with different saving levels n (from 2 to 5). The x and y axes are $n * D$ and f_{change} values, respectively. As the figure shown, the f_{change} values with low n values converge to their boundaries slower than the ones have high n values. This demonstrates that by configuring n value, the end-user is able to fully control the sampling frequency conforming with several IoT application contexts.

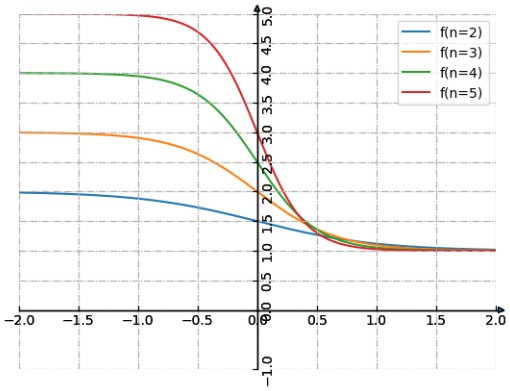

FIGURE 2. The illustration of proposed enhanced sigmoid function (f_{change}).

Our proposed algorithm exploits the extended sigmoid function corresponding with a natural sampling process, which is robust with faulty in sensed data. This scheme ensures that the new frequency is calculated not only based on the latest sensed data but also historical data. In more detail, an anomaly value in sensor reading, which may be significantly higher than the average change recently, does not strongly impact on the next sampling frequency. The frequency only changes when there are consecutive changes in sensed data. As a result, our approach effectively determines whether the device energy is either consumed or conserved based on the trend of the sensed data rather than uncertain changes on the last value. The pseudo-code for implementing our proposal is presented as below:

Algorithm 1 Adaptive Frequency for of Newest Data Point X_i

Input: Dataset X, timestamp i, window size N, and saving level n

Output: Sampling interval T_{i+1}

1. Initializing: Window $W_N = \{X_{i-N}, X_{i-N+1}, \ldots, X_i\}$
2. Calculating: $\overline{W_N} = \frac{1}{N}\sum_{j=i-N}^{i} \triangle\theta(X_j)$
3. Calculating changing degree:
   $D = \triangle\theta(X_i) - \frac{n+1}{2} * \overline{W_N}$
4. Calculating the next sampling interval:
   $T_{i+1} = (n + \frac{1-n}{1+e^{-n*D}}) * T_{base}$

return T_{i+1}

IV. EXPERIMENTAL EVALUATION

A. METRICS OF MEASUREMENT

To evaluate the efficiency of our proposal, we use two metrics:

- *Normalized Mean Error* (NME) indicates the overall goodness of fit after normalizing between the original signal and reconstructed signal from sampled data. This factor is defined as:

$$NME = \frac{1}{n}\sum_{i=1}^{n} |\hat{x}_i - x_i| * 100\% \tag{9}$$

  with $\hat{x}_i$ denotes the normalized i^{th} data in the reconstructed signal, x_i represents the normalized ith data in the original signal and n is the size of signal. Due to the different numeric ranges of collected data, normalization step is necessary before calculating the goodness of fit to rescale the data distributions so that the overall mean and standard deviation are 0 and 1, respectively. For example, the temperature in house is typically in range from 10 to 25, whereas the pH values of water is from 0 to 14. A value x_i in dataset X is normalized as follows:

$$x_i = \frac{x_i - \max(X)}{\max(X) - \min(X)} \tag{10}$$

- *Sampling fraction* (SF) indicates the conserved resources based on the reduction in transmitted messages. It is defined as the number of samples collected

by UDASA over the original data size.

$$SF = \frac{\hat{m}}{n} \tag{11}$$

with $\hat{m}$ and n are the sizes of sampled data according to UDASA and original data, respectively.

B. BENCHMARK DATASETS

1) NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION (NOAA) DATASETS

NOAA provides a set of real-time data about water-quality from a place named "Jamestown". To reasonably compare with state-of-the-art algorithms, we choose the same datasets and monitoring durations with them, which are turbidity and DO from 15 December 2016 to 15 March 2017. As shown in the bottom plots of Figure 3, these datasets contain about 2179 data points with 1h interval sampling.

2) IoT DATASETS

These datasets are about temperature collected from two IoT devices deployed in our working space. They are illustrated in the top plots of Figure 3. The first device produces a dataset about 7683 data points, namely "IoTdevice1", which has a sampling interval of 10 minutes for a sample. The dataset acquired from the second device is named "IoTdevice2", which is about 1731 data points with 1-hour sampling interval.

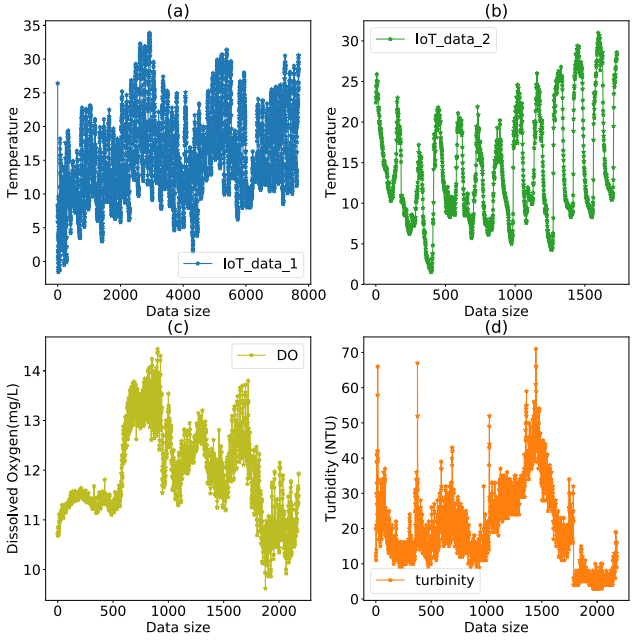

FIGURE 3. The benchmark datasets: (a) IoTdevice1 dataset; (b) IoTdevice2 dataset; (c) DO dataset; (d) Turbidity dataset.

C. EVALUATIVE SIMULATION

To assess the performance of our proposal, we simulate the evaluation process closed to real deployments on IoT devices. The original dataset acts as an environmental context providing sensing values for the sampling algorithm. The detail of the evaluation process is presented in the Algorithm 2. In the first step, the sampled dataset contained sample data by our proposed algorithm is initiated from the first N data points of the original dataset (line 1). In the next step, the sampling process is performed repeatedly until the next sample exceeded the original dataset. In detail, sampling intervals are real-time calculated using our proposed algorithm described in Algorithm 1 (line 2.1), and their sensing values are derived from the original dataset by using an interpolant method defined in (2) (line 2.2). Then, the sampled values are added to the sampled dataset (Line 2.3). In the final step, since indicating the goodness of fit of the sample dataset and the original dataset require equal data size, we perform upsampling of the sampled dataset to the size of the original dataset using simple interpolation [22] (line 3) before calculating the evaluation metrics (NME and SF). This simulation process is implemented by using Python programming language, and performed on-device has the processor: Intel(R) Core(TM) i5-6200U CPU @ 2.30 GHz, 2401 MHz, 2 Core(s), 4 Logical Processor(s), 4 GB of RAM.

Algorithm 2 Evaluation Process

Input: Dataset X, Window size N, Saving level n
Output: NME, FS
1. $Y = [X_0, X_1, \ldots, X_{i-1}]$, i = N
2. **While** $i < size(X)$ **do**
   1. $T_i \leftarrow \text{UDASA}(X_{i-1}, N, n)$
   2. $X_i \leftarrow \text{Interpolation}(X, T_i)$
   3. $Y = Y \cup \{X_i\}$
   4. $i = i + T_i$
3. $\hat{X} \leftarrow UpSample(Y)$

Return $NME(X,\hat{X}), SF(Y, X)$

D. SIMULATION RESULTS

The simulation results are separated into two parts. We first evaluate UDASA performance (indicating via Sampling Fraction and Normalized Mean Error) over real datasets with various saving levels and window size settings. Then, a comparison between our algorithm with baselines is presented. These results demonstrate that UDASA both effectively reduces the number of samples (low SF) and maintains the high-level goodness of fit between the original signal and reconstructed signal from sampled data (low NME).

1) NOAA DATASETS

We first evaluate the proposed algorithm over the NOAA datasets, including DO and Turbidity datasets, with different parameters. Figure 4 and Figure 5 illustrate simulation results at several saving levels over the DO and Turbidity datasets, respectively. In this experiment, the window size (N) is set to 30. With $n = 1$, shown in Figure 4(a), there is no reduction comparing with the original dataset. It means the sampling frequency and sample size are unchanged. As a result, the sample fraction value equals 1. Figure 4(b)-(d) present

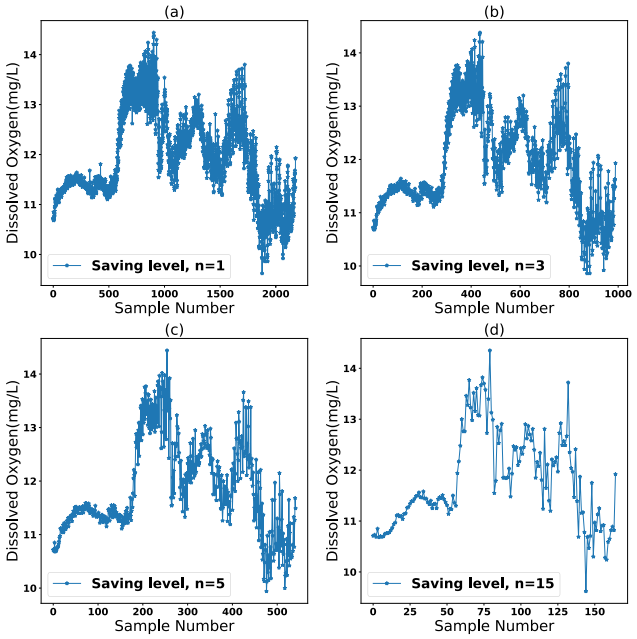

FIGURE 4. DO dataset with different saving levels: (a) Saving level 1 (no saving); (b) Saving level 3; (c) Saving level 5; (d) Saving level 15.

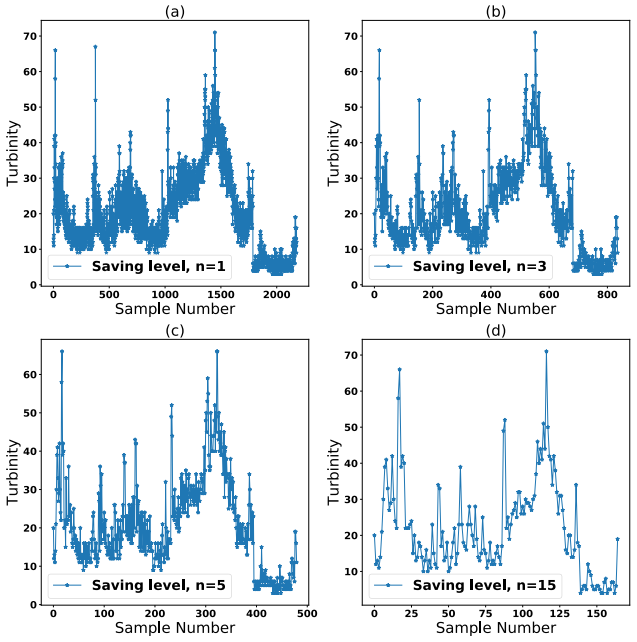

FIGURE 5. Turbidity dataset with different saving levels: (a) Saving level 1 (no saving); (b) Saving level 3; (c) Saving level 5; (d) Saving level 15.

results with saving levels equal 3, 6, and 15, respectively. We can see that increasing the saving level could significantly decrease the SF values since UDASA acquires fewer samples. For example, tripling the saving level from 1 to 3 over the DO dataset reduces the sample size from 2179 to 992 samples and the SF value is reported at 0.45. Increasing the saving level to 5, the obtained data size is significant decrease to 541 points corresponding with the SF values is 0.25. Similar simulation results are also found over the Turbidity dataset in Figure 5.

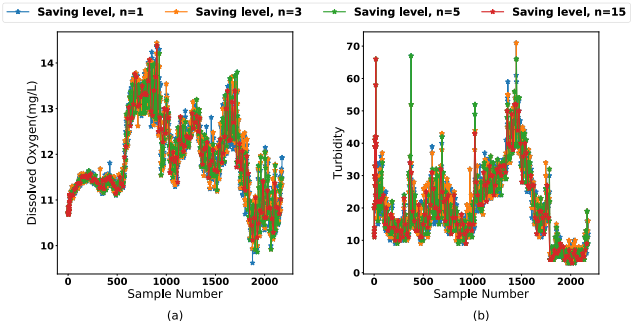

FIGURE 6. The similarity among data trends with different saving levels: (a) Turbidity dataset; (b) DO dataset.

Although collecting fewer samples, the trend of data is highly preserved regardless of saving levels. Figure 6 provides an intuitive view about this preservation in the NOAA datasets by combining the outcome of subplots in Figure 4 and Figure 5. Similar results are also presented in Table 4. The NME values indicating the overall fit of reconstructed data from sampled data and original data is extremely low. For example, given the saving level 3 over the DO dataset, NME value is about 2.21%. In other words, the difference between corresponding values of the reconstructed data (from around 45% original data size) and the original dataset is only 2.21%. The NME value slightly increases to 4.38% when saving level equals 5. This increase is a result of the notable drop-off in the number of collected data, which is reduced about 4 times. Increasing the saving level to 20, the corresponding NME is reported about 7.09%, even though UDASA only collects a tiny number of samples roughly equaling 6% original data size. Performing the same simulation over the Turbidity dataset even provides better results. The NME values are 2.42%, 3.52%, and 5.51% for saving levels at 3, 5, and 20, respectively.

TABLE 4. Simulation results with different saving levels over NOAA datasets.

Datasets	Metrics	Saving Levels					
		1	3	5	10	15	20
DO	# of Samples	2179	992	541	248	175	139
	SF	1	0.45	0.25	0.11	0.08	0.06
	NME	0	2.21	4.38	6.38	6.64	7.29
Turbinity	# of Samples	2179	834	480	248	174	138
	SF	1	0.38	0.22	0.11	0.07	0.06
	NME	0	2.42	3.52	5.35	5.44	5.51

2) IoT DATASETS

Next, we evaluate the performance of UDASA over IoT datasets with various saving level configurations and report the results in Table 5. Similar to the simulation results of

TABLE 5. Simulation results with different saving levels over IoT datasets.

Datasets	Metrics	Saving Levels					
		1	3	5	10	15	20
IoT device1	# of Samples	7683	3372	1587	797	541	413
	SF	1	0.43	0.2	0.1	0.07	0.05
	NME	0	0.75	1.34	1.59	2.01	2.59
IoT device2	# of Samples	1731	778	371	201	144	116
	SF	1	0.44	0.22	0.11	0.08	0.06
	NME	0	0.1	1.41	2.21	3.20	4.31

NOAA datasets, increasing saving levels could significantly reduce the number of samples. For example, we can decrease the sample size of IoTdevice1 datasets from 7683 samples to 1587 samples or 541 samples when increasing the saving level from 1 to 5 or 15, respectively. We also note in Table 5 that given a saving level, the SF values of two datasets are nearly equal. For example, the SF values of IoTdevice1 and IoTdevice2 datasets at saving level 3 are 0.43 and 0.44, respectively. This can be explained that these datasets have similar data distribution (trend, seasonality, and data range). As a result, their D values calculated from Median Absolute Deviation are the same. Remarkably, our proposal is highly effective over IoT datasets. At the same saving level, the NME values of simulation results over IoT datasets are significantly lower than ones over NOAA datasets. For example, at saving level 3, the NME value of the IoTdevice1 dataset is reported about 0.75%, whereas this value of the DO dataset is 2.21%.

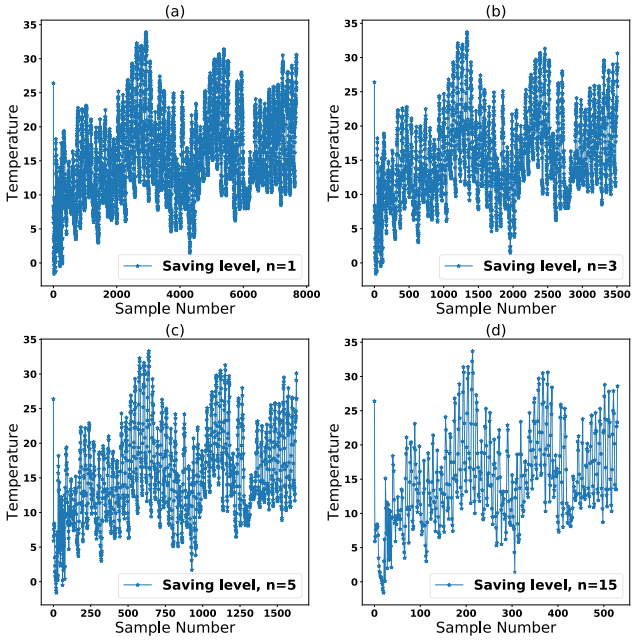

FIGURE 7. IoTdevice1 dataset with different saving levels: (a) Saving level 1 (no saving); (b) Saving level 3; (c) Saving level 5; (d) Saving level 15.

Figure 7 and 8 illustrate the sample data of IoTdevice1 and IoTdevice2 datasets at various saving levels. As the

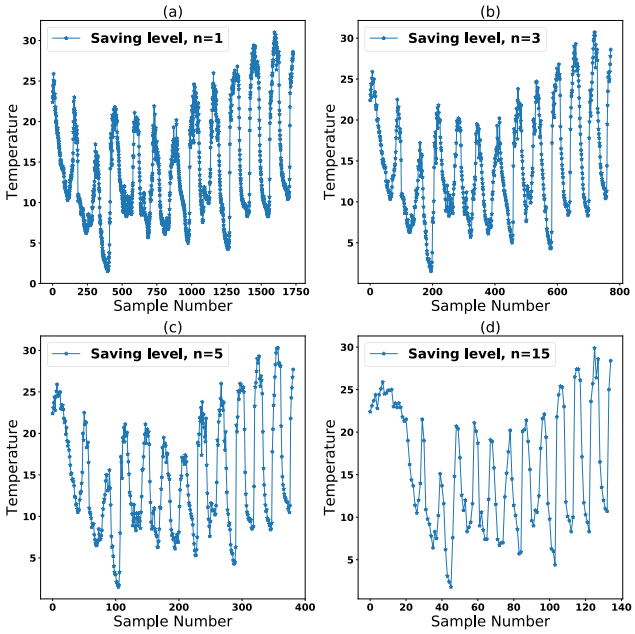

FIGURE 8. IoTdevice2 dataset with different saving levels: (a) Saving level 1 (no saving); (b) Saving level 3; (c) Saving level 5; (d) Saving level 15.

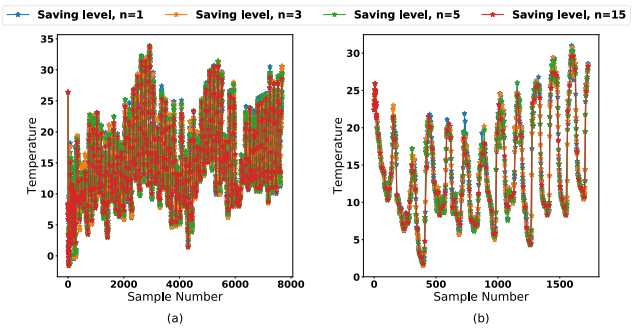

FIGURE 9. The similarity among data trend for different saving levels: (a) IoTdevice1 dataset; (b) IoTdevice2 dataset.

figures showed, the data are very similar in trend regardless of data size. This similarity is not only intuitive (shown in Figure 9) but also represented by low NME values (shown in Table 5). For example, given saving level 5, the NME values of IoTdevice1 and IoTdevice2 datasets are 1.34% and 1.41%, respectively. These values mean that the reconstructed data from samples (sample size equals around 22% original data size) and original data are nearly 99% similar. These results again demonstrate the superiority of our proposal about effectively reduce the number of sample while maximizing the precision of collected data.

3) VARYING WINDOW SIZE

To assess the impact of window size (N value) setting on UDASA performance, we report simulation results about SF and NME values when varying the window size settings for different saving levels in Figure 10 and Figure 11, respectively. The window size values are simulated from 20 to 100. As our expectation, this parameter does not strongly affect

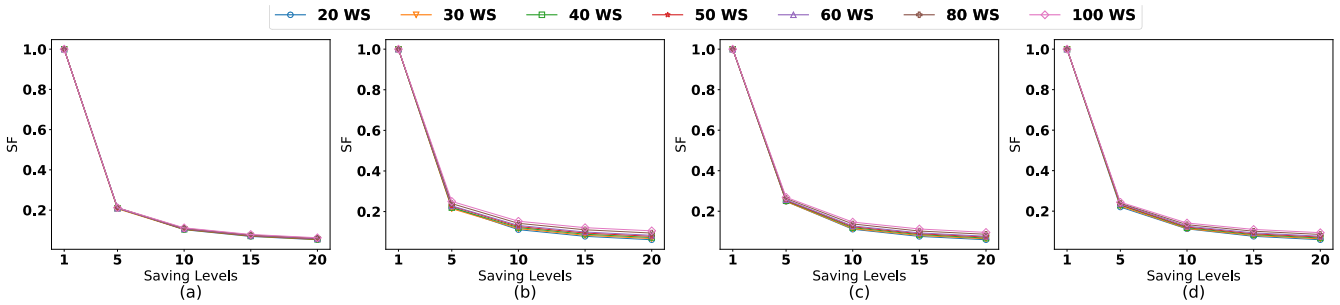

FIGURE 10. The changes of SF values when varying window size parameters over: (a) IoTdevice1 dataset; (b) IoTdevice2 dataset; (c) DO dataset; (d) Turbidity dataset.

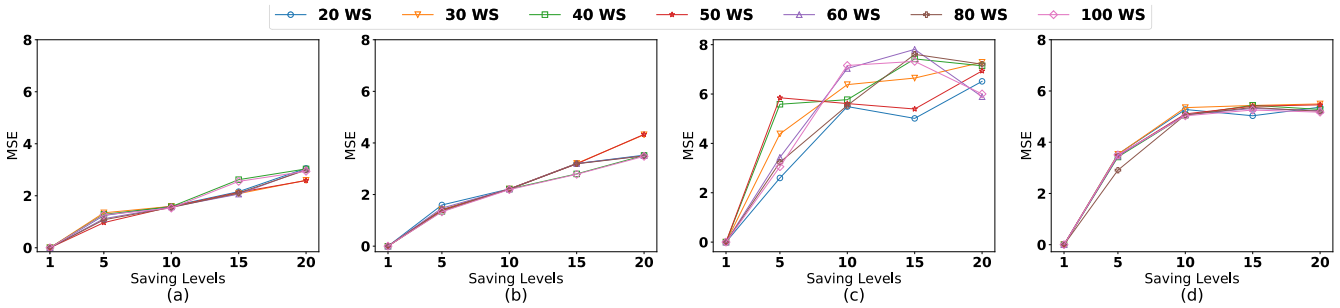

FIGURE 11. The changes of NME values when varying window size parameters over: (a) IoTdevice1 dataset; (b) IoTdevice2 dataset; (c) DO dataset; (d) Turbidity dataset.

TABLE 6. Comparing NME values of UDASA with baselines over the DO dataset.

	ASAP	DDASA (t=0.03)	DDASA (t=0.02)	AM-DR	DDASA (t=0.015)	ASA	ASDR	DDASA (t=0.01)	Fixed Rate Sampling
Number of Samples	200	297	421	502	548	637	917	1064	2179
Competitor's NME	11.57 %	9.99 %	8.43 %	6.11 %	5.31 %	5.52 %	4.7 %	1.62 %	0
UDASA NME	7.61 %	6.04 %	5.2 %	4.72 %	4.35 %	3.92 %	3.21 %	2.07 %	0

sampling performance, especially the SF value. In more detail, plots in Figure 10 show that given a specific saving level, SF values of all benchmark datasets are stable around a constant value when changing the window size. Similarly, the NME values of IoTdevice1, IoTdevice2, and Turbidity datasets shown in Figure 11 hardly fluctuate under different window size settings (Evaluated window size values are 20, 30, 40, 50, 60, 80, and 100). For example, these experiment results of the IoTdevice1 dataset at saving level 5 shows that the variance of NME values when increasing the window size value from 20 to 100 is about 0.02%. This value of the IoTdevice2 dataset is reported at 0.019%. However, there are slight changes in NME values of the DO dataset between different window size settings. By manually analyzing such NME values, we notice that an abrupt disruption (at around the 600^{th} data instance) separates the dataset into two parts with different trends, causing such changes of NME values. These evaluation results again consolidate the effectiveness and consistency of our proposal.

E. COMPARISON OF RESULTS

We do state-of-the-art common adaptive sampling algorithms for energy-efficiency purposes, but evaluating all of them is extremely heavy. Therefore, we compare our proposal with DDASA [23] with different thresholds t, ASA [14], ASAP [24], AMDR [25], and ASDR [26] which are the most related to our proposal. Because the codes of DDASA and ASA algorithms are unavailable, their evaluation results are derived from the original papers.

As shown in Table 6, we compare the NME value of UDASA over the DO dataset with the ones of our competitors in different sample sizes. The better algorithm has lower NME. At the sample size equals 1064, the NME value of UDASA is 2.07% and slightly higher than the one of DDASA (t=0.01) reported about 1.62%. However, our proposal shows superiority over the competitors in all remaining cases. For example, reducing the original dataset (2179 data points) to 297 and 421 samples, the NME values of our approach are about 6.04% and 5.2% in comparison with 9.99% and 8.43% of DDASA, respectively. Similarly, our proposal outperforms ASAP, AM-DR, and ASDR. Decreasing to 200, 502, and 917 samples, the NME values of our algorithm are 7.61%, 4.72%, and 3.21%, respectively. These values of ASAP, AM-DR, and ASDR are 11.57%, 6.11%, and 4.7% This means that the reconstructed data from our selected points is more likely to original data than others. On the other hand, our

approach is more consistent and robust than the competitors in high power saving scenarios. This is demonstrated by the minor changes of NME when significantly reducing the number of samples. The variance of DDASA's NME values is around 2.91 in comparison with 13.6 of DDASA when decreasing the sample size from 1064 to 297. In summary, comparing with other approaches, our algorithm has better performance demonstrated by lower NME, and it is robust with various parameter configurations.

V. RELATED WORK

Energy efficiency for constraint devices is a highly interesting topic in the context of the Internet of Things since power management may involve several device operations, such as sensing, processing, and transmitting. For instance, in the agriculture scenario where IoT devices are widely distributed in a large region, optimizing the energy consumption may extend the device life cycle and significantly reduce the maintenance cost. In what follows, a series of techniques and frameworks to achieve energy-efficiency are presented.

A. ADAPTIVE SAMPLING

Considering that waking-up, collecting, and pre-processing operations consume a similar proportion of energy compared with transmitting, the fundamental idea of adaptive sampling technique is to adapt the sampling rate to the observation changes based on specific criteria ensuring the precision of outcome information. We catalog the adaptive sampling approaches based on such criteria.

Send-on-delta sampling: is the most commonly used in wireless networks. The original of such approach is the level-crossing sampling at late 1950s based on the idea "the most suitable sampling is by transmission of only significant data, as the new value obtained when the signal is changed by a given increment" [27]. Due to its popularity, there are various temps expressing this strategy such as event-based sampling [28], magnitude-driven sampling [28] or deadbands [29]. Formally, given threshold δ, a message has value y_i at t_i is sent if and only if

$$(y_i - y_k) > \delta \tag{12}$$

With y_k is the last message sent at t_k. To prevent babbling-idiot failure on such approach, the min and max sending time, denoted by T_L and T_H, respectively, are defined as the boundary of sampling interval ($T_L \leq t_i - t_l \leq T_H$).

Integral Sampling uses the concept of integral or energy of the error to deal with small oscillations in the signal. The message is sent if the accumulated error of sampling, denoted by CES, is greater than a pre-defined threshold ξ. The min and max-send-time are also applied. The CES value of a signal $x(t)$ is the difference between $x(t)$ and accumulated value from the most recent sample $x(t_k)$.

$$CES_{x_t} = \int_{t_i}^{t_{i-1}} [x(t) - x(t_{t-1})]^2 dt \tag{13}$$

where $i = 1, 2, \ldots, n$ is the number of sample taken from t_0 to t_n [30].

Predictor-based sampling uses a model to predict the next measure based on past values. The message $x(t)$ is sent if it significantly differs with the predicted value $\hat{x}(t)$. The criterion of the difference may reuse either sen-on-data or interval sampling. The model is built from a simplified statistic using linear extrapolation [31]. To maintain the high information quality, the predictor is used in the receivers to extrapolate the signal value until receiving the new message. However, updating the receiver predictor requires at least two samples, which reduces the efficiency of this approach.

Gradient-based integral sampling is an extension of the integral sampling approach with the optimization of wake-up energy consumption—this method based on the fact that the waking-up device consumes considerably larger than collecting message. Hence, the next wake-up time is automatically adjusted to the current gradient of the signal [32]. A max-sleep-time is defined to avoid the signal gradient reducing to zero.

Sigmoid-based sampling uses a sigmoid function to estimate the changes of sampling rate based on the variance of the last windowed signal [14], [23]. Let denote the last message be $x(t)$ belonging a signal window size W, the variance is the absolute difference of between $x(t)$ and $x(t-1)$ over the average value of W. Next, such variance is compared with a predetermined threshold before calculating the new sampling rate is the multiplication of current race and the sigmoid function of such variance. The new rate is limited from 0 to 2 as a result of sigmoid function properties.

Other methods apply adaptive sampling to reduce device energy consumption for disseminating monitoring information. Demistris *et al.* [19] propose an energy efficiency framework for IoT devices, namely ADMin, supporting adaptive monitoring rate based on variability and seasonal behavior of current monitoring stream. To achieve this, ADMin exploits a probabilistic learning algorithm to build a model used to estimate the next stage of the metric stream. Instead of transmitting all monitoring information, IoT devices only send updates to the estimation model. Similarly, the authors in [25] and [26] introduce a prediction-based method exploiting least mean squares adaptive filters to reduce transmission data. In [24], the authors provide a method to reduce local variance while preserving data deviations in time series. In turn, LANCE [33] reduces energy consumption by only sending the average of windowed values. Based on the user-defined policies, the receiver decides summarized values are useful. In [34], the authors introduce G-SIP, an energy-efficiency IoT framework that updates only when the values are changed in a way that can not be estimated by using historical values. G-SIP uses an exponential weighted average to adapt the update rate to the changes in collected values. LANCE and G-SIP's common disadvantages are slow to adapt to sudden changes and using static thresholds that are data-specific.

B. OPTIMUM TASK SCHEDULING

Several works argue that minimizing energy consumption can be achieved by optimizing task scheduling policy. In [35], Zhou *et al.* address a joint route planning and task assignment problem to optimize energy for unmanned aerial vehicles (UAVs). The joint optimization problem is formulated into a two-side two-state matching problem. In the first stage, dynamic programming or genetic algorithms are exploited to solve the route planning problem. Then, the authors use the Gale-Shapley algorithm to solve the second-stage task assignment problem. Its effectiveness is demonstrated by theoretical analysis and practical evaluation results. Similar works for UAVs, [36] aims to minimize energy consumption by combining dynamic programming, auction theory, and matching theory. The authors formulate a minimization problem as a joint optimization involving both large-scale and small-scale optimization. In terms of energy consumption, trajectory scheduling, velocity control, frequency regulation, relay selection, and power allocation algorithm are optimized. The authors in [37] demonstrate that the optimization of task scheduling in IoT devices is an NP-hard problem in a pseudo-polynomial time and propose a dynamic programming algorithm to cope with this problem. Their algorithm prototype is evaluated on three common IoT platforms (TMote, Raspberry PI, and Arduino), effectively adopting the system parameters with different device battery conditions. [38] proposes an enhanced version of the earliest deadline first (MEDF) algorithm to reduce energy consumption while maintaining low delay missing rate. A new task is scheduled based on the current device energy level in the super-capacitor and system timing constant. In a similar approach, [39] uses an energy-aware lazy scheduling algorithm to optimize task scheduling. Its effectiveness is evaluated on both low-cost and commercial devices in practical scenarios. In [40], the authors introduce a dynamic optimization model based on Markov Decision Process (MDP). The proposed model considers the priorities and deadlines of the tasks and battery capacity to derive optimum scheduler. However, this MDP-based scheduler is insufficient for real-time tasks, so the authors propose a greedy version of this scheduler for real devices. In [41], the task scheduling problem of monitoring applications is tackled with a Q-learning-based algorithm. The algorithm applies an annealing strategy to update the scheduling policy to maintain a sufficient battery level for the device's functional jobs. In addition, it also defines the constraint for sending the state values to reduce exchanged information between nodes in Wireless Sensor Network.

C. OPTIMIZED NETWORKING FUNCTIONS

Network operations are recognized as a significant energy consumption source. Many existing works have studied network optimization to prolong device battery. The authors in [42] introduce a stochastic network optimization algorithm for access control and resource allocation by exploiting Lyapunov optimization. Without requiring the channel state information, the proposed algorithm aims to optimize sensing rate control, power allocation, and channel selection in both application and physical layer. Under various simulation results, its ability to execute online with low complexity is testified. Similarly, [43] studies the node transmission behavior to build the model, which provides a joint optimization of the data queue. The model can predict both long and short terms. The work in [44] considers the optimization of channel selection for resource-limited machine-types devices (MTDs) under the missing of global state information. The optimization problem is formulated as a one-to-one matching between MTDs and channels. Combining machine learning, Lyapunov optimization, and matching theory, the authors propose a matching-learning based channel selection framework. The effectiveness and reliability of the proposed framework are validated under various simulated scenarios and parameter settings.

[45] presents a variable sample scheme, which supports receivers adapting the sampling period to the current energy state and the incoming packet length. A sampling information bit is added to the request message header to indicate the complete samples of the network packet. In addition, the authors use the Markov decision process to model the transmission system to minimize power transmission. To enhance efficiency, an optimum re-transmission scheme is also presented. In [46], the optimal deployment of data routing devices in WSN is formulated as a multi-constraint mixed-integer linear program (MILP). Due to the high complexity and strongly effected by network size, a lightweight k-connected greedy solution is also developed. [47] introduces an energy-efficient IoT network for a 5G system leveraging RF energy from the cellular traffic to transmit data. Through experiments in a realistic traffic model, the authors prove that more energy is saved when network utilization is increased.

D. OPTIMUM SENSING

Several works are proposed to optimize sensing operations on devices powered by limited battery resources. The authors in [48] exploit Mean Squared Error estimation (MSE) to express the correlations in energy consumption between sensing and transmitting operations. Through theoretical analysis, they show that adjusting the sampling rate can achieve optimum energy consumption. [49] introduces an asymptotically optimal solution using power-law decaying covariance for random processes. Assuming the energy consumption distribution follows a Poisson process, the authors discover a Markovian property could be used to identify a lower bound of sensing MSE. Typically, they split the time into fixed time windows and estimate optimum sensing performance based on the current battery level. In their experiments, the proposed solution's MSE values are lower than the uniform sampling policy (fixed sampling rate). [50] presents optimum random sensing policies, which randomly choose a set of candidate sensing instants. This work addresses the sampling of a band-unlimited continuous-time random process. At a given candidate sensing instant, the sensor only performs sensing operations only if the battery level is sufficient. The authors

demonstrate that minimum MSE can be achieved by adjusting the sampling period and sampling energy.

VI. CONCLUSION

In this work, a user-driven adaptive sampling algorithm namely UDASA was presented. Given user-desired saving levels, our proposed algorithm could minimize device energy consumption while ensuring the accuracy of collected data. Our practical experiments have shown that UDASA is robust for different parameter values, and effectively decrease the sample size with low error in reconstructed data. Applying this algorithm to NOAA and IoT datasets could significantly reduce the number of acquired samples up to 20 times in comparison with a traditional fixed-rate approach at extremely low error around 6.4% and 3.45%, respectively. Comparing with the existing adaptive sampling algorithm, UDASA is superior in terms of the accuracy of the collected data. Therefore, the energy-efficiency goal for constrained devices is archived by our proposal. Also, its light-weight and ability to control the energy-saving level make UDASA widely used in various use-cases, especially in the MIoT context.

In the future, we will make UDASA become a non-parametric algorithm by automatically estimate the window size parameter following the properties of input data. Moreover, we will perform additional experiments on embedded devices with various communication protocols to verify the amount of saved energy. Combining UDASA with existing compression algorithms to enhance energy conservation could be a potential approach.

REFERENCES

[1] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. M. A. Kazmi, and C. S. Hong, "Internet of Things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Gener. Comput. Syst.*, vol. 92, pp. 265–275, Mar. 2019.

[2] V. Afshar, *Cisco: Enterprises are Leading the Internet of Things Innovation*. New York, NY, USA: Huffpost, 2017.

[3] L. Columbus, "Internet of Things market to reach $267 B by 2020," *Forbes.com*, 2017. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Search+Results+Web+results++Internet+Of+Things+Market+To+Reach+%24267B+By+2020+&btnG=

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[5] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging Internet of Things marketplace from an industrial perspective: A survey," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 4, pp. 585–598, Dec. 2015.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient IoT solutions," in *Proc. 25th Telecommun. Forum (TELFOR)*, Nov. 2017, pp. 1–4.

[8] C. J. Deepu, C.-H. Heng, and Y. Lian, "A hybrid data compression scheme for power reduction in wireless sensors for IoT," *IEEE Trans. Biomed. Circuits Syst.*, vol. 11, no. 2, pp. 245–254, Apr. 2017.

[9] F. Xiao, G. Ge, L. Sun, and R. Wang, "An energy-efficient data gathering method based on compressive sensing for pervasive sensor networks," *Pervas. Mobile Comput.*, vol. 41, pp. 343–353, Oct. 2017.

[10] P. Rajpoot, "Data aggregation and distance based approach to boost life span of wsn," *Int. J. Eng. Technol. Sci. Res., IJETSR*, vol. 4, no. 11, pp. 1–8, 2017.

[11] D. Ventura, D. Casado-Mansilla, J. López-de-Armentia, P. Garaizar, D. López-de-Ipiña, and V. Catania, "ARIIMA: A real IoT implementation of a machine-learning architecture for reducing energy consumption," in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, R. Hervás, S. Lee, C. Nugent, and J. Bravo, Eds. Cham, Switzerland: Springer, 2014, pp. 444–451.

[12] J. Lee and J. Lee, "Prediction-based energy saving mechanism in 3GPP NB-IoT networks," *Sensors*, vol. 17, no. 9, p. 2008, Sep. 2017.

[13] V. Raghunathan, S. Ganeriwal, and M. Srivastava, "Emerging techniques for long lived wireless sensor networks," *IEEE Commun. Mag.*, vol. 44, no. 4, pp. 108–114, Apr. 2006.

[14] 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, Feb. 2010.

[15] J. Polastre, J. Hui, P. Levis, J. Zhao, D. Culler, S. Shenker, and I. Stoica, "A unifying link abstraction for wireless sensor networks," in *Proc. 3rd Int. Conf. Embedded Netw. Sensor Syst. SenSys*, 2005, pp. 76–89.

[16] D. Trihinas, G. Pallis, and M. D. Dikaiakos, "AdaM: An adaptive monitoring framework for sampling and filtering on IoT devices," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Oct. 2015, pp. 717–726.

[17] D. Trihinas, G. Pallis, and M. Dikaiakos, "Low-cost adaptive monitoring techniques for the Internet of Things," *IEEE Trans. Services Comput.*, early access, Feb. 23, 2018, doi: 10.1109/TSC.2018.2808956.

[18] E. B. Priyanka, C. Maheswari, and S. Thangavel, "IoT based field parameters monitoring and control in press shop assembly," *Internet Things*, vols. 3–4, pp. 1–11, Oct. 2018.

[19] D. Trihinas, G. Pallis, and M. D. Dikaiakos, "ADMin: Adaptive monitoring dissemination for the Internet of Things," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, May 2017, pp. 1–9.

[20] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: Real-time data aggregation with adaptive-event differential privacy for fog computing," *Wireless Commun. Mobile Comput.*, vol. 2018, Jul. 2018, Art. no. 6285719.

[21] H. Wang and A. O. Fapojuwo, "A survey of enabling technologies of low power and long range Machine-to-Machine communications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2621–2639, 2017.

[22] T. Blu, P. Thevenaz, and M. Unser, "Linear interpolation revitalized," *IEEE Trans. Image Process.*, vol. 13, no. 5, pp. 710–719, May 2004.

[23] T. Shu, M. Xia, J. Chen, and C. de Silva, "An energy efficient adaptive sampling algorithm in a sensor network for automated water quality monitoring," *Sensors*, vol. 17, no. 11, p. 2551, Nov. 2017.

[24] K. Rong and P. Bailis, "ASAP: Prioritizing attention via time series smoothing," 2017, *arXiv:1703.00983*. [Online]. Available: http://arxiv.org/abs/1703.00983

[25] Y. Fathy, P. Barnaghi, and R. Tafazolli, "An adaptive method for data reduction in the Internet of Things," in *Proc. IEEE 4th World Forum Internet Things (WF-IoT)*, Feb. 2018, pp. 729–735.

[26] S. Santini and K. Romer, "An adaptive strategy for quality-based data reduction in wireless sensor networks," in *Proc. 3rd Int. Conf. Netw. Sens. Syst. (INSS)*TRF Chicago, IL, USA, Jun. 2006, pp. 29–36.

[27] P. Ellis, "Extension of phase plane analysis to quantized systems," *IRE Trans. Autom. Control*, vol. 4, no. 2, pp. 43–54, Nov. 1959.

[28] N. Persson and F. Gustafsson, "Event based sampling with application to vibration analysis in pneumatic tires," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2001, pp. 3885–3888.

[29] P. G. Otanez, J. R. Moyne, and D. M. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proc. Amer. Control Conf.*, May 2002, pp. 3015–3020.

[30] M. Miskowicz, "Sampling of signals in energy domain," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Sep. 2005, p. 4.

[31] Y. Suh, "Send-on-delta sensor data transmission with a linear predictor," *Sensors*, vol. 7, no. 4, pp. 537–547, Apr. 2007.

[32] J. Ploennigs, V. Vasyutynskyy, and K. Kabitzsch, "Comparison of energy-efficient sampling methods for WSNs in building automation scenarios," in *Proc. IEEE Conf. Emerg. Technol. Factory Autom.*, Sep. 2009, pp. 1–8.

[33] G. Werner-Allen, S. Dawson-Haggerty, and M. Welsh, "Lance: Optimizing high-resolution signal collection in wireless sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. sensor Syst. SenSys*, 2008, pp. 169–182.

[34] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3816–3825, Oct. 2013.

[35] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5526–5538, Nov. 2018.
[36] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial Internet of UAVs for power line inspection in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2705–2714, Jun. 2018.
[37] A. Caruso, S. Chessa, S. Escolar, X. del Toro, and J. C. Lopez, "A dynamic programming algorithm for high-level task scheduling in energy harvesting IoT," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2234–2248, Jun. 2018.
[38] H. Yang and Y. Zhang, "A task scheduling algorithm based on supercapacitor charge redistribution and energy harvesting for wireless sensor nodes," *J. Energy Storage*, vol. 6, pp. 186–194, May 2016.
[39] M. Severini, S. Squartini, F. Piazza, and M. Conti, "Energy-aware task scheduler for self-powered sensor nodes: From model to firmware," *Ad Hoc Netw.*, vol. 24, pp. 73–91, Jan. 2015.
[40] V. S. Rao, R. V. Prasad, and I. G. M. M. Niemegeers, "Optimal task scheduling policy in energy harvesting wireless sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2015, pp. 1030–1035.
[41] Z. Wei, Y. Zhang, X. Xu, L. Shi, and L. Feng, "A task scheduling algorithm based on Q-learning and shared value function for WSNs," *Comput. Netw.*, vol. 126, pp. 141–149, Oct. 2017.
[42] Z. Zhou, Y. Guo, Y. He, X. Zhao, and W. M. Bazzi, "Access control and resource allocation for M2M communications in industrial automation," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3093–3103, May 2019.
[43] N. Ashraf, A. Hasan, H. K. Qureshi, and M. Lestas, "Combined data rate and energy management in harvesting enabled tactile IoT sensing devices," *IEEE Trans. Ind. Informat.*, vol. 15, no. 5, pp. 3006–3015, May 2019.
[44] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Mumtaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, "Learning-based context-aware resource allocation for Edge-Computing-Empowered industrial IoT," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4260–4277, May 2020.
[45] A. Yadav, M. Goonewardena, W. Ajib, O. A. Dobre, and H. Elbiaze, "Energy management for energy harvesting wireless sensors with adaptive retransmission," *IEEE Trans. Commun.*, vol. 65, no. 12, pp. 5487–5498, Dec. 2017.
[46] N. Mehajabin, M. A. Razzaque, M. M. Hassan, A. Almogren, and A. Alamri, "Energy-sustainable relay node deployment in wireless sensor networks," *Comput. Netw.*, vol. 104, pp. 108–121, Jul. 2016.
[47] A. O. Ercan, M. O. Sunay, and I. F. Akyildiz, "RF energy harvesting and transfer for spectrum sharing cellular IoT communications in 5G systems," *IEEE Trans. Mobile Comput.*, vol. 17, no. 7, pp. 1680–1694, Jul. 2018.
[48] J. Wu, I. Akingeneye, and J. Yang, "Optimum sensing of a time-varying random event with energy harvesting power sources," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2015, pp. 1134–1138.
[49] J. Yang and J. Wu, "Optimal sampling of random processes under stochastic energy constraints," in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 3377–3382.
[50] J. Wu, I. Akingeneye, and J. Yang, "Energy efficient optimum sensing with energy harvesting power sources," *IEEE Access*, vol. 3, pp. 989–997, 2015.

LE KIM-HUNG (Member, IEEE) was born in Ho Chi Minh City, Vietnam, in 1990. He received the B.S. degree in networking engineering from the University of Information Technology—Vietnam National University, Ho Chi Minh, Vietnam, in 2013, the M.S. degree in network security from Telecom ParisTech, France, in 2016, and the Ph.D. degree in the Internet of Things from Sorbonne University, France, in 2019.

Since 2019, he has been an Lecturer with the Data communication and Networking Department, University of Information Technology—Vietnam National University. His research interests include applying deep learning and artificial intelligence into the Internet of Things, data cleaning, and energy-efficiency methods for embedded devices.

QUAN LE-TRUNG received the B.Eng. degree from the Bach-Khoa University of Technology (BKU), Vietnam, in 1998, the M.Eng. degree from the Asian Institute of Technology, Thailand, in 2002, and the Dr.Techn. degree from the Department of Telecooperation, Johannes Kepler University-Linz, Austria, in 2007. Then, he has spent nine months postdoc there, and another 24 months postdoc at the Networks and Distributed Systems Group, Department of Informatics, University of Oslo, Norway. He is currently an Associate Professor with the Department of Computer Networks, University of Information Technology—VNUHCM. His research interests span a variety of areas in networking with current focus on the wireless embedded Internet, cyber physical systems, the Internet of Things, cloud/edge computing, and smart cities.