IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 21, 2019, accepted June 18, 2019, date of publication July 1, 2019, date of current version July 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2926209

RDCM: An Efficient Real-Time Data Collection Model for IoT/WSN Edge With Multivariate Sensors

NAYEF ABDULWAHAB MOHAMMED ALDUAIS[1,2], (Member, IEEE), JIWA ABDULLAH[1], AND ANSAR JAMIL[1]

[1]Wireless and Radio Science Centre, Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Batu Pahat 86400, Malaysia
[2]Department of Computer Engineering, Faculty of Computer Science and Engineering, Hodeidah University, Al Hudaydah, Yemen

Corresponding authors: Nayef Abdulwahab Mohammed Alduais (naifalduais@gmail.com) and Jiwa Abdullah (jiwa@uthm.edu.my)

This work was supported in part by the Universiti Tun Hussein Onn Malaysia under Grant H132-TIER 1, and in part by the Fundamental Research Grant Scheme (FRGS) from the Ministry of Education Malaysia under Grant 1532.

ABSTRACT In the application of the Internet of Things (IoT), a sensor board depends on a battery that has a limited lifetime to function. Furthermore, the IoT sensor board with multivariate sensors influences the battery life-time, since there are additional data transmissions that must be supported by the board causing it to drain the battery much faster than the sensor board with one sensor. The main aim of this paper is to increase the battery life of the IoT sensor node. To do so, this paper proposes an efficient real-time data collection model for multivariate sensors in IoT/WSN applications named RDCM. The general structure of RDCM is composed of two main levels: the IoT sensor board level and the fusion center level. The IoT sensor board level is implemented in real time by all the IoT sensor boards simultaneously in each cycle and fusion center level is executed by the fusion center. The IoT sensor board level includes various stages as follows: check the physical conditions of the IoT edge device (board) stage and update data strategy stage, data validation stage, and sensed data reduction stage. The average of the total percentage of energy saved by the application of RDCM to real-time data sets injected with various percentages of errors for all nodes is 98%. In summary, the RDCM has a very high performance in terms of energy consumption compared with other algorithms. This paper concludes with the limitation of the current study and some further research opportunities.

INDEX TERMS IoT, WSN, data collection, energy consumption, multivariate sensors.

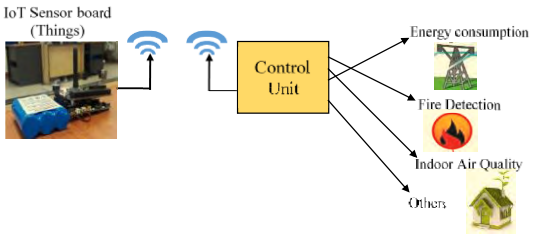
I. INTRODUCTION

A. OVERVIEW

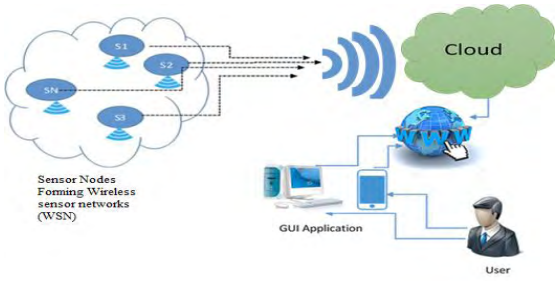
Wireless sensor network (WSN) consists of spatially distributed autonomous devices that used sensors to monitor physical or environmental conditions. It integrates a gateway that provides wireless connectivity to the internet. The Internet of Things (IoT) is a communication paradigm that envisions total connectivity with objects of everyday life and is an integral part of the Internet [1] infrastructures. Hence, the IoT concept promotes the Internet even more immersive and pervasively, enabling an easy access and interactions with a variety of devices [2]. Various practical communications models are used in IoT implementations, and each model has its own characteristics. There are three models described by the IoT architecture board which includes machine-to-machine, machine-to-cloud, and machine-to-gateway-to-cloud as shown in Figure 1(a), Figure1(b) and Figure1. (c), respectively. These models highlight the flexibility in ways that IoT devices can be connected and provide value added to the user [4]. It must be noted that in all previous models the source machine (IoT edge devices) is the backbone of the system, which is used to collect the data.

The world evolves, and so does our lifestyles, where we are more dependent upon numerous modern electronic devices. In recent years, WSN has played a vital role in IoT applications. Numerous applications are based on WSNs and IoT technologies, which have been applied in various fields. They may be in healthcare, smart homes and buildings, air pollution, military, industrial, precision farming, modern horticulture industry and many more. In the wearable medical monitoring applications [3], [4] sensors can be very useful

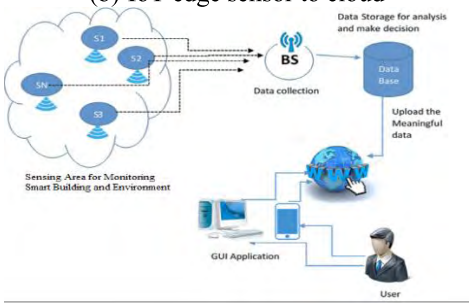
The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.



(a) IoT edge sensor to device



(b) IoT edge sensor to cloud



(c) IoT edge sensor to BS

FIGURE 1. Three architecture data collection IoT -based WSN; (a) IoT edge sensor to device, (b) IoT edge sensor to cloud and (c) IoT edge sensor to BS.

to provide accurate and reliable information about people's activities and behaviors, provide assistance to the human living environment [5], [6]. Furthermore, healthcare is not only for humans, but also includes animal care observing biological parameters such as rumination, body temperature, heart rate, ambient temperature and humidity [7]. Smart home and building applications, such as home environment monitoring [8], real-time wireless monitoring of indoor air quality [9] and energy management, all contribute to the widespread usage of WSN/IoT integration. WSN possesses several constraints such as limited energy availability, low memory size, and low processing speed, which are the principal obstacles to designing efficient management algorithms for WSNs [10], even more so if it concerns WSN/IoT integration.

B. MOTIVATION

Various data collection algorithms played a very important role in improving the efficiency of the real-time IoT/WSN applications. Nevertheless, there are still obvious challenges and development issues faced by data collection algorithms for real-time applications. Numerous researchers have addressed the issue of reducing the number of transmissions packets by the IoT/WSN sensor board with a univariate sensor. However, reducing the number of transmissions packets by IoT/WSN sensor board with multivariate sensors is one of the most important research issues. Similarly, data reduction methods based on the coding scheme in WSN/IoT still have many constraints, such as delay, solve floating point variables, and historical data. Despite all this, the problem of multivariate sensors is still a critical issue open for further research. In WSN/ IoT, sensor data consist of one attribute (univariate) or multiple attributes (multivariate). Since the sensor board is only used to collect one type of data (light/temperature or humidity), this type of data is called univariate data. Similarly, in some IoT / WSN applications, each sensor board is equipped with a multivariate sensor to support different application needs. Furthermore, the current multivariate data models used in IoT/WSN applications for the purpose of reducing or validating the sensed data during data collection process may have some challenges. For example, these models are dependent on training, which means that the accuracy of those models is declining over time due to the increment in the approximation error. This increment in the approximation error of the multivariate data model during the real-time data collection is one of the significant challenges. The standard solution to this issue is accomplished by applying an adaptive model that is able to update its reference parameters during data collection. However, the act of increasing the updating frequency of the model reference parameters will affect the energy efficiency of the sensor board due to the size of transmitted data after the updates. Normally the model reference parameters typically are larger than or equal to the payload data size without reduction. The process of determining the threshold value is a difficult task in the multivariate data models, which is why an objective solution is needed.

This paper studies the problem of how to design and improve an efcient real-time data collection model for IoT/WSN sensor board with multivariate sensors. It will be as a means to address several issues which have been described previously. The research problem as such be stated as the follows:

(i). How will the model reduce the number of transmissions packets by IoT/WSN sensor board with multivariate sensors?
(ii). What is the most beneficial method to determine the threshold value for multivariate data reduction models?
(iii). How will the proposed model avoid transmitting incorrect sensed data during the data collection for IoT/WSN sensor board with multivariate sensors?
(iv). How will the model reduce the number of bits' payload for IoT/WSN sensor board with multivariate sensors?

II. RELATED WORKS

It should be noted that some of the recent works aim to improve the network performance by means of focusing on the sensed data processing, dissemination and scheduling.

For example, in [10], adaptive data processing and dissemination for drone swarms in urban sensing named ADDSEN has been proposed. The authors focused on improving the in-network quality by means of observing the low-quality or faulty sensed data and separating it from the set of sensed data and redundant data. Similarly, many researchers focus on mobility management and flow scheduling in IoT, where the work in [11] achieved scalable mobility management and robust flow scheduling in IoT multi-networks. For example, compared with other software-defined networking (SDN) systems as presented by the authors, the throughput has been increased by 67.21%, the delay has been reduced by 72.99%, and the jitter has been improved by 69.59%. It is clear that the previous solutions benefit mobile nodes. In our work, like many previous works [13]–[48], have addressed the fixed nodes scenario. For more clarity, this work focused on payload data only. Also, there is no connection between the nodes with the assumption that each IoT sensor node is able to immediately update its sensed data to the fusion center. We are probably the first to address various issues in one model data collection for multiple sensors in IoT application. Therefore, in this section of related works, it is divided the prior works into two main parts: first, (A) evaluation of measures related to the reduction in the number of transmitted packets, which includes two subsections (i) reduce the number of transmitted normal data and (ii) reduce the number of transmitted incorrect data. In this study, normal data is correct sensed data. However, the aim of the update data strategy during sensing phase is to save the energy consumption of the IoT sensor board with multiple sensors by reducing the number of transmission packets if no significant change is reported by the payload sensing block. Similarly, the incorrect data, their values measured by the sensor board that are wrong, therefore avoiding the waste of energy consumption for incorrect data. In the second part, (B) evaluation measures related to payload data size reduction approaches are presented.

A. EVALUATION MEASURE RELATED TO REDUCTION IN THE NUMBER OF TRANSMITTED PACKETS

In order to decrease energy consumption, various methods have been proposed to reduce the number of transmitted data. On the other hand, avoiding transmitting incorrect sensed data during the data collection will contribute significantly to saving the energy of IoT Edge device.

1) REDUCE THE NUMBER OF TRANSMITTED NORMAL DATA

Packet transmission can be drastically reduced if data prediction algorithm such as time series prediction (TSP) can be utilized. TSP is a significant applied technique for commercial, inventory, weather prediction, manufacturing control and signal processing. TSP is defined as a sequence of data that is ordered by time and characterized by chronological importance. Thus, the indices of variables and the correlation between them can be used to develop mathematical models. Therefore, the main purpose of time series modeling is to collect and study historical values in order to find the appropriate models that represent the general structure of a given data [13]. The prediction of the time series based sensing data model is a conventional technique for reducing the transmission by sensor nodes, and there are several ways to use this technique, which includes: moving average (MA), exponential smoothing (ES), autoregressive (AR), autoregressive with exogenous inputs (ARX), and autoregressive integrated moving average (ARIMA). Nonetheless, these methods only support a single type of data sensor. For instance, in [14], the authors presented a prediction-based data reduction method by joining it with an adaptive sampling rate. In addition, the recent work by Tan and Wu [15] introduces a method to reduce the number of sensor node transmitted packets by applying the hierarchical Least-Mean-Square (HLMS) in the presence of adaptive filter. In the previous work by [16], the authors presented a fast and efficient dual-forecasting method to reduce the number of messages sent from the sensor board. Careful evaluation of the findings presented by [15] and [16] shows that only univariate data with a fixed threshold error were investigated. Recently, the work by [13] proposed a new method based on forecasting to reduce the number of transmitted packets. The advantage of the proposed model is its ability to evaluate the proposed model using vibration sensors datasets. However, the method only addresses the univariate data. In [17], the study shows a prediction of a light-weight model with 0.2 tolerance error and in [19], Artificial Neural Networks (ANN) was employed to predict the sensed data. It uses a Multi-Layer Perceptron (MLP) to decide on the required data samples. Collective forecast exploiting temporal-spatial correlation named CoPeST based on Least Mean Square (LMS) algorithm as reported in [19] reduces the amount of energy that is crucial for expensive transmission while maintaining the data integrity to be within the error threshold of the user. In [20], the study is a preliminary work on optimizing sensor node energy employing an efficient data collection and dissemination (EDCD) updating strategy. EDCD is a strategy to update sensed data to the fusion center, which is employed to reduce the number of transmitted packets. On the other hand, ref [21] proposed an adaptive method for data reduction (AM-DR). AM-DR method is based on a convex combination of two decoupled Least-Mean-Square (LMS) windowed filters with different sizes. AM-DR is used to reduce the number of transmission packets by predicting the current sensed data at the base station. The trends in the majority of the forecasting methods are to broadcast the original sensed data to the sink, when the predicted data error is more than the threshold value. The authors in [22] proposed an adaptive data acquisition mechanism that allows each sensor node to adjust its sampling rate according to its environmental changes while optimizing its energy consumption. In another attempt, the author [23] used a simple linear regression to save power consumed by sensor nodes. It is done by reducing data transmission. The study considered that only one attribute is related to the prediction, and only one attribute

TABLE 1. Characteristics of the prior works that address the problem of reducing the number of transmitted packets.

Ref	Applications	Network topology	Implementation	Dataset used	Type of sensors	Tolerance Threshold	Transmitted Reduction
[13]	Industrial application	Star	Simulation MATLAB & R language	Dataset collected by the authors at factory	Vibration data	0.8%	73%
[14]	Environmental monitoring	Star	Simulation MATLAB	Dataset collected by the authors at laboratory	Temperature Humidity Infrared	+-0.1 +-0.1 +-1	98% 39% 45%
[15]	Environmental monitoring	Star	Simulation MATLAB	IBRL Dataset	Temperature	0.3◦C	95%
[16]	Industrial application	Star	Sensor node level: Testbed (TelosBs and TinyOS) At Gateway level MATLAB	Dataset collected by the authors at factory	Temperature Friction	NA	74% 60%
[17]	Environmental/ Building Application	Star	Simulation MATLAB & WISPES W24th node prototype	Dataset collected by the authors at building	Temperature	0.2	96%
[18]	Environmental	Short distance Node send the data directly to the sink	Simulation NA	Dataset meteorological and collected by the authors	Temperature Humidity	0.1°C 0.3%,	60% 60%
[19]	Environmental	Cluster topology	Simulation MATLAB	Dataset NA	Temperature	0.6°C	65%
[20]	Environmental	Start	Simulation MATLAB	IBRL dataset Air quality data set	Temperature Humidity	0.5% 0.5%	74% 80%
[21]	Environmental	Star	Simulation NA	IBRL Dataset	Temperature	+-0.5	95%
[22]	Industrial Process Monitoring	Cluster	Java simulator & Testbed (TelosBs)	IBRL Dataset	Temperature Humidity	NA	80% 80%
[23]	Environmental	Cluster	Simulation NA	NA	Temperature	10%	67%

is used to predict the dependent variable. Time characteristics are not the most relevant variables compared to other features such as lighting, temperature and humidity, which makes the predictions used by the solution inaccurate [24].

As a key assessment, most of the methods currently used to reduce the number of transmitting packets in WSN/IoT cover only univariate data, except for the work involving multivariate data in [20]. Table 1 illustrated the characteristics of the prior works that address the problem of reducing the number of transmitted packets. The authors applied separately these algorithms to each type of sensors listed in Table 1.

2) REDUCE THE NUMBER OF TRANSMITTED INCORRECT DATA

In WSN/IoT, the sensing data error detection approaches can be divided into two types, namely: centralized error discovery method and distributed error discovery method. Most existing error detection approaches use periodic batch testing at a central location, possibly a cluster head or a fusion center [25]. A useful background overview of the current outlier detection methods for WSN can be found in [25]–[28]. In addition, recent work [25] introduced a novel mathematical model for assessing the impact of different data verification systems on energy dissipation in the edge device. The One

TABLE 2. Characteristics of the prior works that address the problem of abnormal data.

Ref	Applications	Network topology	Implementation	Dataset used	Type of sensors	Accuracy*
[30]	Environmental monitoring	Cluster Sensor Node Level	Simulation MATLAB	IBRL, GSB and LUCE	T,A.T , RH and S.T	97.23%
[32]	Environmental monitoring	Cluster Sensor Node Level	Simulation MATLAB	Networked Aquatic Microbial Observing System (NAMOS) IBRL and LUCE	T,H and V	94.8%
[34]	Environmental monitoring	Star Sensor Node Level	Simulation (NA) and Realistic testbed	Dataset collected by the authors	carbon-dioxide (CO2), carbon-monoxide (CO), and H	NA
[36]	Environmental monitoring	Cluster Sensor Node Level	Simulation(MATLAB) and Realistic testbed	IBRL and Dataset collected by authors	T and H	97%

*Accuracy is the average of all authors results

Class Quarter Sphere Support Vector Machine (OCSVM) was used in [29] to create an anomaly discovery algorithm. The authors in [30] proposed a method for observing outliers by using a kernel principal component analysis (KPCA) based on the Mahalanobis kernel. The idea behind is to isolate the anomaly from the normal data distribution pattern. However, this work was executed at the CH level and only supports a single sensor (single variable). The previous work [31] reported a qualified study of strategic detection of abnormal sensed data in the smart city applications based on WSN. In [32], the study presented an adaptive One Class Principal Component Classifier model to detect the outliers in real-time. The problem in the proposed work, which was how to detect outliers on training samples, was not solved. Therefore, in [33], the authors proposed a statistical training sensed data removing approach for PCA-based chiller sensor fault discovery, diagnosis, and data reconstruction technique. The study discussed the discovery and the elimination of outliers from the original training sensed samples. In [34], the authors proposed a data validation algorithm for detecting different types of faults. Its evaluation used data samples of WSN's prototype for environment monitoring injected with different types of faults. The Modified-Z score method [35] was used to detect outliers. Similarly, in [36], they proposed a new real-time algorithm for observational verification of sensor data at the node level, which is named Validity of the measuring sensor reading at node level (VSNL). VSNL is a sensor data verification algorithm based on an adaptive threshold. VSNL considers detecting various types of errors in the sensed data and proposes a simple mechanism to classify errors and events. Sensor anomaly detection system for distinguishing between real and false alarms has been provided in [37] for healthcare applications. Table 2 illustrates the characteristics of the previous works that address the problem of abnormal data.

B. EVALUATION MEASURES RELATED TO PAYLOAD DATA SIZE REDUCTION APPROACHES

In the previous section, some of the work related to reducing the number of packets transmitted in WSN/IoT were explored. However, this section provides a thorough discussion on the latest work on the method of reducing payload data size through the transmission of sensed data from the IoT edge device to the FC. In a report presented by [38], the authors propose a coding scheme to reduce the size of the payload data sent by the cluster head node. Similarly, the work of [39] aims to improve the accuracy of the data received by the fusion center. The proposed coding scheme is based on relative differences and precision factors rather than the absolute variation method used in [38]. These tasks are beneficial for cluster head nodes with univariate data. Principal Component Analysis (PCA) is one of the most widely used methods for multivariate data reduction. Various types of PCA-based data reduction models are reported in [40]–[44]. Due to limited resources of the sensor board, the original version from PCA is not suitable for WSN/IoT edge level. Therefore, a lightweight version of PCA called Candid Covariance-free Incremental PCA (CCIPCA) was proposed in [45]. The previous work in [46] used CCIPCA as multivariate data reduction in WSN with a fixed threshold and two Principal Component (PC). In addition, the recent work [47] proposed two methods for multivariate data reduction for adaptive threshold known as PCA-B and MLR-B. PCA-B is a multivariate data reduction that used CCIPCA with adaptive threshold and set the number of PC to one in order to achieve a high reduction level. MLR-B is a multivariate data reduction utilizing Multiple Linear Regression model (MLR) with an adaptive threshold. According to the work of [47], the size of transmitted data after updating the model reference parameters which are larger or equal to the payload data size without reduction. It means that the sensor board

TABLE 3. Characteristics of the prior works that address the problem of reduction the size of payload data.

Ref	Applications	Network topology	Implementation	Dataset used	Type of sensors	Payload Data Reduction
[24]	Environmental monitoring	Cluster Sensor Node Level	Simulation MATLAB	IBRL	T,H and L	50%
[41][46]	Environmental monitoring	Cluster Sensor Node Level	Simulation MATLAB	IBRL	T,H, Voltage(V) and L	50%
				GSB, LUCE	A.T , RH and S.T	33%
[47]	Environmental monitoring	Cluster Sensor Node Level	Simulation MATLAB	LUCE	A.T , RH and S.T	66.6%
[48]	Environmental monitoring	Start Sensor Node Level	Simulation MATLAB	IBRL, GSB and LUCE	T, H, V and L A.T , RH and S.T	95%
[38]	Environmental monitoring	Cluster topology CH Node Level	Simulation MATLAB	Data collected usingMTS420CA	T	25%
[39]	Environmental monitoring	Cluster topology CH Node Level	Simulation MATLAB	IBRL	T,H	NA

requires more energy in the updating stage than the reduction stage. The study recommended the frequency of updating the model reference parameters during data collection be used as a new metric to evaluate the performance of the multivariate data reduction models. More detail regarding the data reduction methods has been described in recent work [48]. Additionally, that work proposed a new simple mechanism called the Adaptive Real-time Payload Data Reduction Scheme (APRS) for energy-efficiency purpose in IoT/WSN sensor board with multivariate sensors. APRS aims to reduce the transmitted packet size for each sensed payload. Table 3 illustrates the characteristics of the previous works that address the problem of reducing the size of payload data. In addition, Table 4 and Table 5 show the summary of comparison of the related works for each issue and the summary of their limitations, respectively.

III. MATHEMATICAL MODEL OF ENERGY CONSUMPATION FOR REAL-TIME DATA COLLECTION SCHEMES IN IoT/WSN EDGE DEVICE LEVEL

In this section a mathematical model of energy consumption to evaluate the real-time data collection schemes for IoT/WSN edge device level is introduced. The model solves several problems related to the energy consumption of IoT sensor nodes. It addresses the issues related to reduction of transmission packets when using multiple sensor IoT board. In this model, incorrect data transmission is avoided and also it reduces the amount of payload bits before transmitting it to FC. The proposed model can be used for numerical analysis of energy consumption in different highlighted issues.

A. CONSTRAINTS

Let us consider that an IoT sensor board battery life-time $\mathcal{L}$ is defined as Eq. (1).

$$\mathcal{L} = (b_{max} \times \mathcal{E}_{bit}) \tag{1}$$

Thus, in this work the problem of data collection formulae is defined as

$$f_c\,(\mathcal{E}_{total}, \mathcal{R}) \leq \mathcal{L};$$
$$\mathcal{R} = \{r_1, r_2 \ldots r_N\}\,; c = 1, 2, \ldots C \tag{2}$$

where b_{max} is the maximum number of bits that could be transmitted and received during a period time, $\mathcal{E}_{bit}$ the energy cost of transmitting or receiving one-bit, R the measured data , N is the number of samples and C is the number of constraints, $\mathcal{E}_{total} = \mathcal{E}_{total}(\boldsymbol{u}, \boldsymbol{d}, \boldsymbol{v})$

Subject to

$$\textit{Minimize} \rightarrow \{f(\boldsymbol{u}), f(\boldsymbol{d}), f(\boldsymbol{v})\}$$

where $f\ (\boldsymbol{u})$ is the function referring to reducing the number updating times during data collection issue, $f\ (\boldsymbol{d})$ is the function referring to reducing the number of transmitted bits issue when it is necessary to update the IoT sensor board sensed data to the FC/cloud and $f\ (\boldsymbol{v})$ is function referring to reducing the cost of data validation as well as avoiding sending incorrect data issues. It should be noted that another reason for the loss of sensor node energy is data processing. In this paper, the transmitted data during the data collection phase constitute a fundamental component of energy consumption. This is because the energy consumed in sending one bit via sensor board is higher than running many microcontroller instructions [49]. Hence, in wireless devices, the energy consumed by transceiver accounts for 80% of the overall energy consumption of the node [50].

This study highlights that incorrect data is one of the reasons for wasting battery energy. This is because the transmission of erroneous data requires the same amount of energy as transmission of normal data. In addition, at FC this data will be removed from the dataset after applying a data validation algorithm, which means we avoid wasting some energy by not transferring the incorrect data. Therefore, applying a simple solution at the IoT sensor node level to avoid transmitting incorrect data will help in saving the energy of the IoT sensor node.

B. NUMERICAL EXAMPLE

Consider that a battery is used to equip an IoT sensor node for a specific application. The maximum number of samples that can be sent to the FC is N = 1000 samples when the sensor node is in active mode/ RF(on) with energy consumption

TABLE 4. Summary of comparison the related works.

Ref.	Sensor board		Methodology	Issues addressed					Complexity			Threshold	
	univariate sensor	Multivariate Sensors		Single errors	Data validation Multiple errors	Reducing number of Transmissions data	Reducing Size of Transmissions data	IoT Prototype	Training needed	No training	Adaptive- estimate by the model	Non-Adaptive-estimate by the model	Fixed -Select by FC
[14]	Yes	×	Data prediction & adaptive Sampling	×	×	Yes	×	×	Yes	×	×	×	Yes
[15]	Yes	×	LMS prediction	×	×	Yes	×	×	Yes	×	×	×	Yes
[16]	Yes	×	Fast and efficient dual-forecasting	×	×	Yes	×	×	Yes	×	×	×	Yes
[13]	Yes	×	Dual-forecasting	×	×	Yes	×	×	Yes	×	×	×	Yes
[17]	Yes	×	light-weight forecasting	×	×	Yes	×	×	Yes	×	×	×	Yes
[18]	Yes	×	A neural data-driven	×	×	Yes	×	×	Yes	×	×	×	Yes
[19]	Yes	×	Prediction (CoPeST)	×	×	Yes	×	×	Yes	×	×	×	Yes
[20]	×	Yes	EDCD - Relative change/ Relative difference	×	×	Yes	×	×	×	Yes	×	×	Yes
[21]	Yes	×	Dual prediction scheme using a convex combi×tion of two LMS adaptive filters	×	×	Yes	×	×	Yes	×	×	×	Yes
[22]	Yes	×	Adaptive Sampling	×	×	Yes	×	Yes	Yes	×	×	×	Yes
[23]	Yes	×	based SLR	×	×	Yes	×	×	Yes	×	×	×	Yes
[24]	×	Yes	based MLR	×	×	×	Yes	×	Yes	×	×	×	Yes
[38][39]	Yes	×	Based coding scheme for cluster head nodes level	×	×	×	Yes	×	×	Yes	×	×	×
[48]	×	Yes	Based coding scheme for sensor node level	×	×	Yes	Yes	Yes	×	Yes	×	×	×
[40-45]	×	Yes	based on PCA	×	×	×	Yes	×	Yes	×	×	×	Yes
[46]	×	Yes	based on CCIPCA	×	×	×	Yes	×	Yes	×	×	×	Yes
[47]	×	Yes	Based on CCIPCA ×med PCA-B	×	×	×	Yes	×	Yes	×	Yes	×	×
[47]	×	Yes	Based on MLR ×med MLR-B	×	×	×	Yes	×	Yes	×	Yes	×	×
[29]	×	Yes	Based on OCSVM	Yes	×	×	×	×	Yes	×	×	×	Yes
[30]	×	Yes	Based on KPCA	Yes	×	×	×	×	×	×	×	×	×
[32]	×	Yes	Based on OCPCC	Yes	×	×	×	×	Yes	×	×	×	Yes
[34]	Yes	×	Based on MZ-score	×	Yes	×	×	Yes	×	Yes	×	×	Yes
[36]	×	Yes	Based on absolute change	×	Yes	×	×	Yes	×	×	Yes	×	×

$\mathcal{E}_{byet} = 52.92\mu$J/byte and for simplicity, we assumed that the consumption in sleep mode/RF(Off) is $\mathcal{E}_{byet} = 0$uJ. The number of sensors in the same node is n = 3 and each sensor needs 4 bytes. The number of incorrect data is Er = 100 samples. The energy consumption for each scenario is as follows:

- *Secnario1*: *Transmit all data.*

$$Energy = (1000 \times 4 \times 3 \times 52.92\mu\text{J}) = 635040\mu\text{J}$$

- *Secnario2: Transmit correct data only.*

$$\begin{aligned} Energy &= ((1000 - 100) \times 4 \times 3 \times 52.92\mu\text{J}) \\ &= 571536\mu\text{J} \end{aligned}$$

C. DISCUSSION

From the above numerical analysis, we can prove that avoiding the transmission of incorrect data leads to minimizing the total energy consumption $\mathcal{E}_{total}$.

According to [25], the IoT sensor node energy consumption will be affected by a mechanism that is used to detect and remove the incorrect data during data collection for IoT real-time application. The cost of the error detection and transmissions $E_{DV-Phase}$ during the validation phase based on the approaches applied is defined as in Eq. (3).

$$\begin{aligned} &\mathcal{E}_{DV-Phase}(\mathcal{P}_{\text{time}}) \\ &= \mathcal{E}_{\text{Tr}}(\mathcal{P}_{\text{time}}) \\ &\quad + \mathcal{E}_{\text{NN}}(\mathcal{P}_{\text{time}}) + \mathcal{E}_{\text{SD}}(\mathcal{P}_{\text{time}}) \\ &= \sum_{\mathcal{P}_{\text{time}=1}}^{\mathcal{NP}} ((\mathcal{H}_{Tbits} \times \varepsilon_{Tr}) + \mathcal{E}_{\mathcal{NN}}(\mathcal{P}_{\text{time}}) + \mathcal{E}_{SD}(\mathcal{P}_{\text{time}})) \end{aligned} \tag{3}$$

where ε_{Tr} is the energy consumption for transmission of normal data, $\mathcal{E}_{\text{NN}}$ is the energy dissipation to receive data from various nearest neighbor nodes, $\mathcal{H}_{Tbits}$ is the size of transmitted data, $\mathcal{P}_{\text{time}}$ is the current time and $\mathcal{E}_{\text{SD}}$ is the

TABLE 5. Summary the current data collection limitations.

Issue	Methods	Limitations
Reducing number of transmissions by Edge device with multiple sensors	Reducing the number of transmissions by applying forecasting / prediction methods	• Univariate data only address • Fixed threshold • Number of update models effect the energy consumption • Based on the above issues it could be difficult to apply those methods for edge device equipped with multiple sensors.
Data validation	Based on Statistical methods , classification and PCA.	• Statistical method simple but is of no benefit for real-time applications. • The tolerance error / threshold is fixed • Training data need extra energy. • Updating the reference model is not based on approximation error, in this case the model not accurate enough.
Multivariate data reduction	MLR /SLR, PCA and, Data coding	• Extra energy for Training is needed. • Number of retraining times negatively effect the energy consumption. • Heavy code book and historical data are needed and support only univariate data

energy dissipation to read/write data from SD-card memory during data collection. *For example*, if the used algorithm to observe an incorrect data during data collection does not need to build a historical data or receive data from nearest neighbor nodes, the $\mathcal{E}_{\mathrm{SD}}$ and $\mathcal{E}_{\mathrm{NN}}$ is equal to zero. Therefore, our proposed model is able to observe the incorrect data without the need to build historical data or receive data from the nearest neighbor nodes (See in Algorithm 4).

According to Eq. (3), we can infer the following: (i) Increase in the value of $\mathcal{E}_{SD}$ negatively affects the energy dissipation to detect the incorrect data. This is because the error observation approach is unable to check whether the data is being sensed directly in real-time, but it needs to collect enough number of samples N, and save in a memory, thus creating a historical data. (ii) Similarly, increase in the value of $\mathcal{E}_{\mathrm{NN}}$ negatively affects the energy dissipation to detect the incorrect data. This is because the error observation approach is unable to check the condition of the sensing data in real-time directly, but it needs to receive the neighbor's sensed data to verify its validity. This mechanism is thus totally dependent on the spatial-temporal correlation among neighbor's edge devices. Regardless of the percentage of accuracy, its disadvantage is in the energy consumption of error observation. (iii) the use of online/real-time approach is the best way to observe incorrect data with the lowest energy consumption. The key point of this situation is that the error observation method can check the sensor data in real time without delay, or need to construct historical data or bring neighbor data for data validity verification.

The energy dissipation to receive data from various nearest neighbor nodes $\mathcal{E}_{\mathrm{NN}}$ defined in Eq. (4). From the equation, it is clear that increasing the number of nearest neighbor nodes will increase the cost of detecting the data error at the edge device. This is because the cost of detecting the status of the sensed data (normal/abnormal) is higher than the cost of transmitting the sensed data itself.

$$\mathcal{E}_{\mathcal{NN}}\left(\mathcal{P}_{\mathrm{time}}\right) = \sum_{i=1}^{\mathcal{N}_{\mathrm{nib}}} \left(\mathcal{H}_{\mathrm{Rbits}} \times \mathcal{E}_{\mathrm{r}}\right) \tag{4}$$

where $\mathcal{H}_{Rbits}$ is the size of received data, $\mathcal{E}_{\mathrm{r}}$ is the energy dissipation to receive one-bit and $\mathcal{N}_{\mathrm{nib}}$ is the number of nearest neighbor nodes.

Eq.(5) defined the energy dissipation for reading/writing samples from memory $\mathcal{E}_{\mathrm{SD}}$ during data collection which used for checking the validity measured data. The number of samples and its sizes affect the cost of the observed error.

$$\mathcal{E}_{\mathrm{SD}} = \left(\sum_{i=1}^{\mathcal{H}} \left(\mathcal{R}_{\mathrm{bits}} \times \mathcal{R}_{\mathrm{Ecost}}\right)\right) + \left(\sum_{i=1}^{\mathcal{H}} \left(\mathcal{W}_{\mathrm{bits}} \times \mathcal{W}_{\mathrm{Bcost}}\right)\right) \tag{5}$$

where $\mathcal{R}_{bits}$, $\mathcal{W}_{bits}$ the number of read and write bits to and from the memory, respectively. $\mathcal{R}_{\mathrm{Ecost}}$, $\mathcal{W}_{\mathrm{Ecost}}$ represent the cost of energy dissipation to read and write bits from the memory, respectively. $\mathcal{H}$ is the number of samples that have been collected before checking validity of the current data.

The energy consumption during the reduction phase $\mathcal{E}_{\mathrm{RD-Phase}}$ depends on the methods used as defined as follows. Equation Eq.(6) is used to calculate the energy consumption during the reduction phase $\mathcal{E}_{\mathrm{RD-Phase}}$ as in [48]. $\mathcal{E}_{\mathrm{RD-Phase}}$ is divided into three parts as follows (i) Reduction Mode (RM), (ii) Non-Reduction Mode (N-RM) and (v) Retraining Mode (RTM).

$$\mathcal{E}_{\mathrm{RD-Phase}} = \left(\sum_{\mathrm{i}=1}^{\mathcal{H}_1} \mathcal{E}_{\mathrm{N-RM(i)}}\right) + \left(\sum_{i=1}^{\mathcal{H}_2} \mathcal{E}_{\mathrm{RM(i)}}\right) + \left(\sum_{i=1}^{\mathcal{H}_3} \mathcal{E}_{\mathrm{RT-M(i)}}\right) \tag{6}$$

where $\mathcal{E}_{\mathrm{N-RM}}$, $\mathcal{E}_{\mathrm{RM}}$ and $\mathcal{E}_{\mathrm{RT-M}}$ are the energy consumption per sample in N-RM, RM and RTM, respectively. $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$ are the number of forwarded samples through data collection in N-RM, RM and RTM, respectively.

N-RM is a common mechanism for sending payload data from the sensor node to the FC without reducing its size. The energy consumption per sample in N-RM is defined as in Eq. (7).

$$\mathcal{E}_{\mathrm{N-RM}} = \left(O\mathcal{R}_{\mathrm{Lenght}} \times \mathcal{E}_{\mathrm{T}}\right) \tag{7}$$

RM is a mechanism for sending payload data from the sensor node to the FC with reducing its size by applying a benefit algorithm. The energy consumption per sample in RM is defined as in Eq. (8).

$$\varepsilon_{\mathrm{RM}} = \left(\varepsilon_{\mathrm{N-RM}} \times \left(1 - \frac{\mathcal{RD}_{\mathrm{Lenght}}}{\mathcal{OR}_{\mathrm{Lenght}}}\right)\right) \tag{8}$$

The efficiency of the data reduction models that are dependent on training declines over time due to the increase in the approximation error. The retraining process aims to update the reference parameters to represent the new dynamic changes in the sensed data [46]. Therefore, the sensor node needs to transmit a copy from the reference parameters to the FC. The energy consumption per sample in RTM is defined as in Eq. (9).

$$\varepsilon_{\mathrm{RT-M}} = \left(\mathcal{RF}_{\mathrm{Length}} \times \varepsilon_{\mathrm{T}}\right) + \varepsilon_{\mathrm{RM}} \tag{9}$$

where $\mathcal{OR}_{\mathrm{Length}}$, $\mathcal{RD}_{\mathrm{Length}}$ and $\mathcal{RF}_{\mathrm{Length}}$ is the original length of payload, reduced data and the model reference parameters per sample, respectively.

D. DISCUSSION

From Eq. (6) it is clear that the energy consumption of the sensor node is affected by the type of the algorithm used to reduce the sensed data during data collection. For example, if the data reduction model is based on training such as PCA-B and MLR-B, which means that increase in the value of $\mathcal{E}_{\mathrm{N-RM}}$, $\mathcal{E}_{\mathrm{RM}}$ and $\mathcal{E}_{\mathrm{RT-M}}$ negatively impacts the battery life-time. Therefore, in our proposed model we are taking this issue into account, where the cost of $\mathcal{E}_{\mathrm{N-RM}}$ is the same energy consumption for transmission of one sample where $\mathcal{H}_1 = 1$ only (See in algorithm2). In addition, there is no effect for $\mathcal{E}_{\mathrm{RT-M}}$ because in the proposed model, there is no need to retrain or transmit any reference parameters to the FC, which means $\mathcal{E}_{\mathrm{RT-M}} = 0$ due to $\mathcal{H}_3 = 0$.

The total energy consumption is calculated by combining equation (3) and equation (6) as defined in Eq. (10).

$$\mathcal{E}_{Total} = \mathcal{E}_{RD-Phase} + \mathcal{E}_{DV-Phase} \tag{10}$$

It should be noted that Eq. (10) highlighted numerous issues impact on energy consumption. In summary, avoiding transmitting of incorrect data helps in reducing energy consumption, thus selecting an appropriate approach for that purpose is very important. This is because the value of energy consumed by applying an approach to check the validity of the sensed data is higher than the energy consumption if it is forwarded to FC (See in Eq. (3)). Similarly, reducing the size of payload data in the sensor board with multivariate sensors will help in saving the energy consumption. Nevertheless, as is clear from Eq. (6), design of an efficient model for that aim is a vital issue, as previously discussed. Accordingly, the proposed RDCM model addressed different issues that help to save energy such as reducing the number of transmission packets by IoT sensor board with multiple sensors, avoiding transmission error measured and reducing the number of payload bits. More details about the RDCM model is presented in the following section.

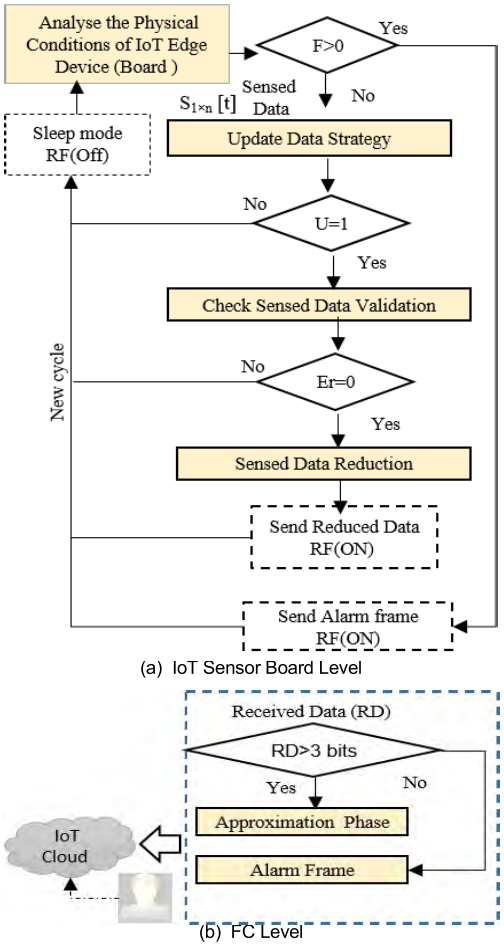


FIGURE 2. RDCM block diagram; (a) IoT sensor board level and (b) FC level.

IV. PROPOSED RDCM

In this section, a detailed description about the proposed RDCM is provided. Figure 2 illustrates the block diagram of the RDCM in a general structure, composed of two main levels; IoT sensor board level and fusion center level. The IoT sensor board level is implemented in real-time by all IoT sensor boards simultaneously in each cycle and fusion center level is executed by the fusion center. IoT sensor board level includes various stages; (i) analyze the physical conditions of

the IoT edge device; (ii) update data strategy stage (iii), data validation and (iv) sensed data reduction.

IoT edge device phase dealt with the physical state of the IoT edge device. The quality of the sensed data is a correlation with the state of the edge device, such as the temperature and the battery level of the board. Therefore, it is important to check the physical state of the IoT edge device before start sensing. If the edge device is not in good condition, an alarm is send to the fusion center informing that the device needs maintenance or change. In this paper, a simple algorithm to check the physical state for the IoT edge device has been proposed (See in Algorithm3). If the edge device is in good condition, the board will start sensing. After that, the sensed data is passed to the update data phase. Updating data phase makes a decision to transmit the sensed data or not depends on the percentage of the different between the current sensed data and the last transmitted sensed data. If the sensed data must be transmission to the fusion center, the model will pass the sensed data to the data validation phase. Data validation phase able the algorithm to decide the state of the sensed data are correct or error event. If the sensed data either it is incorrect/error the model will define it as unreliable data it must be discard and increment the error counter by one, after which it reads a new sample again. In another case, when the sensed data is correct the model will forward the sensed data to payload data reduction phase. Payload Data Reduction Phase dealt with reducing the payload size for the edge device with multiple sensors. The key point for this phase is that it only uses one variable $D[t]$ to represent the multiple sensors measured data $S_{1\times n}[t]$ (n-variables) based on the relative difference between the current measured data $S_{1\times n}[t]$ and last transmitted measured data $S_{1\times n}[t-1]$ to the fusion center for all sensors. At the fusion center, the RDCM is able to execute reconstruction of the original real-time sensed data $\hat{S}_{1\times n}[t]$ (n-variables) from $D[t]$.

A. RDCM-IoT SENSOR BOARD LEVEL

IoT sensor board level is the main phase in the RDCM model, which is implemented in real-time at the edge board with multiple sensors.

1) RDCM - INITIAL PHASE

This phase includes the following steps (i) calculate the prediction model threshold that will be used to check the measured data validity only during data collection (ii) sensor board transmits only one sample without any reduction in the payload packet size.

This paper proposed the calculation of the model threshold during the initial phase. Accordingly, the minimum residual errors between the training data and approximated data occurred during the initial phase [47]. This is because the purpose of training any model is to get the benefit of reference parameters / weights which will be used later to enable prediction of one attribute from various attributes based on simple linear regression. The proposed threshold is adaptive such that the value of the threshold changes during data collection as in Algorithm 1.

Definition 1 (Reference Parameters (RFP) Function): we compute the prediction model references $\mathbb{r}$ by applying Eq. (11) as follows;

$$r = \left(S_I^T \times S_I\right)^{-1} \times S_I^T \times S_p \tag{11}$$

After careful study of the correlation between the multiple sensors on the same sensor board, the independence sensor S_I and dependence sensor S_p are selected, where $S_p \leftarrow S_{w\times p}[T]$ the Predicted sensor and the other multiple sensors $S_{w\times i}[T]$ where $i \neq pi = 1, ..n\ \forall S_i, S_p \in R^{1\times W}$. The Predicted sensor in this study is a sensor having high correlation with all sensors in the same board.// $S_I = [ones, S_{w\times i}]$.

Definition 2 (Predicted Sensor Value(PSV) Function): we calculate the predicted sensor $\bar{S}_p$ by applying Eq. (12)

$$\bar{S}_p(j) = r_0 + r_i \times S_i(j) + \ldots r_n \times S_n(j) \tag{12}$$

where $i \neq p, i = 1, ..n$ and $j = 1, 2 \ldots, w$ is the number of collected samples.

In order to determine the value of the threshold, we first calculate the approximation error between the training data S_p [T] and Predicted data $\bar{S}_p$ [T] which is defined in Eq. (13). After that, estimate the threshold value by selecting maximum approximation error value for S_p.

$$AE\ (j) = \left|\bar{S}_p\ (j) - S_p(j)\right|, \quad Er \in R^{1\times W} \tag{13}$$

According to [47], [48], increase the number of updating frequency metric (UFM) for data reduction model effect of the energy consumption. This is because the size of transmitted data after updating the model reference parameters which is larger or equal to the payload data size without reduction. It means that the sensor board requires more energy in updating stage than the reduction stage. The UFM values in the case of the non- adaptive threshold is larger than the adaptive one. The reason for that, the model based on non-adaptive threshold is entirely dependent on the value of threshold that has been calculated during the training phase and is used in reduction phase without any change in the value of that threshold. Furthermore, the probability that the value of the threshold to be small for the first time. In this case, the model will still be retrained as the dynamic data will change in most of the cases leading to the production of error that is larger than the threshold. Conversely, the adaptive threshold changes its value every time the reference parameters need updating. Therefore, this study updated the model reference parameters at the node level without send a copy from the reference parameters to the FC. This is because this study used the prediction model only to check the validity of the sensed data (See algorithm4) at the sensor node level. More detail about determining the threshold and step phase steps is presented in the following pseudocodes for algorithm1 and algorithm 2, respectively.

It should be noted that RDCM-Initial phase is run only once during data collection. The detailed description of this phase is stated in the following pseudo-code.

Algorithm 1 Adaptive Threshold (ATR)

1) **Input:** $p, w, n, \boldsymbol{S_{w\times n}[T]} \in R^{w\times n}$
2) **Output:** $Thr, \mathbb{r}$
3) **Begin:**
4) **Set** $S_p \leftarrow S_{w\times p}[T]$, $S_p \leftarrow S_{w\times I}[T]$,
5) $\mathbb{r} \leftarrow \boldsymbol{RFP}(S_I, S_p), \mathbb{r} \in R^{1\times n}$
6) **For** j = 1 to w **do**
7) $\bar{S}_p\,(j) = \boldsymbol{PSV}(\mathbb{r}, S_I)$
8) AE (j) = **ABS**($\bar{S}_p$ (j) − S_p(j))
9) **End for**
10) $Thr \leftarrow \max\{\mathbb{AE}\}$

Algorithm 2 Setup Phase

1) Compute threshold Call ATR // algorithm1
2) Read $S_{1\times n}[t]$ // n is the number of sensors
3) Send $S_{1\times n}\,[t]$
4) $S_{1\times n}\,[t-1] \leftarrow S_{1\times n}\,[t]$
5) End

TABLE 6. Classification of physical state of IoT sensor boards.

B_L	T_B	CMS	Description	Classification
1	0	0	Battery level problem	Bad conditions
0	1	0	Problem in the temperature of the board	Bad conditions
0	0	1	The sensors measurement are not accurate	Bad conditions
0	0	0	No problem	Good conditions

2) RDCM – SENSING PHASE

In this phase, conditions of the IoT edge sensor board such as battery level, board temperature and confidence level measured by the sensors is a very important issue, since bad conditions will reduce the accuracy of measured data. For example, in order to read the temperature sensor, the study in [51] recommended to use the sensor board with battery level that should be greater than or equal to a specified threshold.

Definition 3 (Confidence Level Measure for the Sensor (CMS) Function): CMS is a strategy to measure that amount of acceptance of the readings obtained from the sensor board.

$$\boldsymbol{CMS} = \left(\mathbf{1} - \frac{\boldsymbol{E}}{(\boldsymbol{E}+\boldsymbol{C})}\right) \times \mathbf{100} \tag{14}$$

where E is the number of measured errors and C is the number of correct measured data. Eq. (14) is used to evaluate how reliable the IoT sensor board is by dividing the number of sensor error readings to the total number of sensor readings. Furthermore, standing IoT/ WSN sensor boards are more reliable when the error rate (CMS) is close to zero and vice versa. Table 6 shows the classification of physical state of IoT sensor boards.

If the IoT edge device is in poor condition, an alert should be sent to the fusion center to inform the device that it needs to be maintained or replaced. In this study, a simple algorithm has been proposed and described to check the physical state of IoT edge devices. The following pseudo code details the implementation steps of the RDCM-Sensing phase.

Algorithm 3 Physical Conditions of IoT Edge (PCIE)

1) **Input:** B_L, T_B, CMS
2) **Output:** $\mathbb{F} \in R^{1\times 3}$
3) **Begin:**
4) **If** $B_L < B_{Ld}$
5) $F(1) \leftarrow 1$; else $F(1) \leftarrow 0$
6) **End if**
7) **If** $T_B >= T_{Bd}$
8) $F(2) \leftarrow 1$; else $F(2) \leftarrow 0$
9) **End if**
10) **If** $CMS >= CMS_d$
11) $F(3) \leftarrow 1$; else $F(3) \leftarrow 0$
12) **End if**

3) RDCM –DATA UPDATING PHASE

The aim of this phase is to save the energy consumption of the IoT sensor board with multiple sensors by reducing the number of transmission packets if no significant change is reported by the payload sensing block.

Definition 4 (Relative Difference (RTV) Function): we calculate the relative difference vector $\mathbb{RD}$ between the current sensed data $S_{1\times n}[t]$ and last transmitted data $S_{1\times n}[t-1]$ by applying the Eq.(15).

$$\boldsymbol{RD_i} = \left\| \frac{\left(\boldsymbol{S_i[t]} - \boldsymbol{S_i[t-1]}\right)}{\left(\boldsymbol{s_i[t]} + \boldsymbol{S_i[t-1]}\right) * \mathbf{0.5}} \times \mathbf{10^2} \right\|" \tag{15}$$

where $i = 1, 2, .. n$ and n is the number of sensors on the same board.

Decision: If there is no significant change in the sensed data (for more detail, see *algorithm5*), then set RF (Off), otherwise check the validity of the current sensed data.

4) RDCM – VALIDATION PHASE

The aim of this phase is to avoid transmitting any incorrect data, which will contribute to saving in energy consumption as well as increase the system accuracy. The following pseudo code details the implementation steps of the RDCM-Validation phase. In this study, the types of error are range error (RE), constant error(CE) outlier error (OE) and event value (EV).

- *Definition RE:* The invalid reading sensor value detects when the value is outside the visible measuring range, where S ∈ [MinValue, MaxValue]. For example, regarding the features of the MCP9700A temperature sensor, the sensor measurement range is [−40°C, +125°C].
- *Definition CE:* fixed measured fault occurs when a sensor appears as a fixed value for an enormous number of continuous samples.
- *Definition OE:* Let us consider that all the samples that have been measured are within the range as

in " *Definition RE*", but some samples lie outside, either smaller or larger than most of the other values in a set of samples, so those values are denoted as outliers. For more explain, consider that $\mathbb{B} = \{S(t-1), S(t-2), \ldots S(t-w)\}$ is the past sensed data read by the sensor S_p in the setup phase and w is the number of samples. The array of measuring differences $\mathbb{D} = \{d_1, d_2, ..d_w\}$, can be defined as follows:

$$d(i) = \left|S_p(t) - B(i)\right| \quad i = 1, 2, \ldots, w \tag{16}$$

This work considers the current measured value $S_p(t)$ is an outlier fault if the maximum difference value is higher than the threshold value, otherwise, the sensed value is normal. In addition if, the sum of matrix $\{\mathbb{D}\}$ is zero, in this case $S_p(t)$ is an constant fault. Table 4 shows the transmission decisions based on the status of the current sensed data.

Algorithm 4 Sensed Data Validation (SDV)

Inputs: $\mathbf{S_p(t)}$, $\mathbb{B}$, *Thr*, **w**
Output: ***Err***, ***EV***
Begin:
If $S_P(t) > MaxV$ OR $S_P(t) < MinV$
  $RE \leftarrow 1$
Else
  For i = 1 to w do
    $d(i) =$ **ABS**$(S_P(t) - B(i))$
  End for
  If SUM $\{\mathbb{D}\} == 0$
    $CE \leftarrow 1$
  Else
    If Max$\{\mathbb{D}\} > Thr$
      $OE \leftarrow 1$
    Else
      $\bar{S}_p(t) =$ ***PRV***$(r, S_I(t))$
      AE (t) = ***ABS***$(\bar{S}_p(t) - S_p(t))$
      If AE (t) $> Thr$
        Update threshold // Call algorithm1
        $\bar{S}_p(t) =$ ***PRV***$(r, S_I(t))$
        AE (t) = ABS$(\bar{S}_p(t) - S_p(t))$
        If AE (t) $> Thr$
          $EV \leftarrow 1$
        End if
      End if
  End if
End if
$Err \leftarrow$**SUM** $\{RE, CE, OE\}$

5) RDCM – REDUCTION PHASE

The main aim of this phase is to reduce the transmitted packet size for each sensed payload, which will help in saving the energy of the IoT sensor board as in APRS [48].

The following pseudo code details the implementation steps of the RDCM-Reduction phase.

Algorithm 5 Multivariate Data Reduction (MDR)

Inpu t: $\mathbb{RD}$, n
Outpu t: $\boldsymbol{D[t]}$
Begin:
$\boldsymbol{m} \leftarrow \lfloor \boldsymbol{log2(Max)}\{\boldsymbol{ABS}(\mathbb{RD})\} \rfloor + \mathbf{1}$
$\boldsymbol{L} \leftarrow \boldsymbol{m} + \mathbf{1}$
For i = 1 to n do
  If RD (i) $\geq \mathbf{0}$
    Sb(i) $\leftarrow \mathbf{2}^{\wedge((\boldsymbol{i}\times\boldsymbol{L})-\mathbf{1})}$
  Else
    Sb(i) $\leftarrow \mathbf{0}$
  End if
  $\boldsymbol{X(i)} \leftarrow \boldsymbol{ABS}(\mathbf{RD\ (i)}) \times \mathbf{2}\wedge^{((i\times L)-\mathbf{L})} + \mathbf{Sb(i)}$
End for
$\boldsymbol{D[t]} \leftarrow$ **SUM**$\{\mathbb{X}\}$
Send $\boldsymbol{D[t]}$ to FC
$\hat{S}_{1\times n}[t] \leftarrow$ ***Approx***$(\mathbb{RD}, S_{1\times n}[t-1])$
$S_{1\times n}[t-1] \leftarrow \hat{S}_{1\times n}[t]$

First, calculate the required number of bits to represent $|\boldsymbol{RD_i}|$ as the following

$$\boldsymbol{m} = \lfloor \log 2\,(\boldsymbol{Max}\,(|\boldsymbol{RD_{1\times n}}|)) \rfloor + 1 \tag{17}$$

where m is the maximum number of bits.

Calculate the total number of bits (L) required to represent relative difference $\pm RD_i$ and defined as

$$\boldsymbol{L} = m + 1 \tag{18}$$

Definition 5: In order to manage negative and non-negative ***RD*** tests, Eq.(19) is applied

$$Sb_i = \begin{cases} 2((i \times L) - 1) & \textit{for positive change } (+\boldsymbol{RD_i}) \\ 0 & \textit{for negative change } (-RD_i) \end{cases} \tag{19}$$

Definition 6: calculate the representation of the sensed data $D[t]$ in real time [t] as defined in Eq. (20).

$$\boldsymbol{D[t]} = \sum_{i=1}^{n} |RD_i| \times 2^{((i\times L)-L)} + sb_i) \tag{20}$$

Definition 7: Approximated data *(Approx) Function*), we calculate the approximated of the sensed data $\hat{S}_{1\times n}[t]$ at current time t as the following

$$\widehat{\boldsymbol{S}}_{\mathbf{1\times n}}[\boldsymbol{t}] = \left(S_{1\times n}[t-1] \times \left(RD_{1\times n} \times \mathbf{10^{-2}}\right)\right) + S_{1\times n}[t-1] \tag{21}$$

The following pseudocode details the implementation steps of the RDCM- IoT edge device Level. IoT sensor board level includes various stages including analyse of the physical conditions of the IoT edge device, updating data strategy stage, sensed data validation and sensed data reduction.

B. RDCM- FUSION CENTER LEVEL

It should be noted that the FC receives data from the IoT sensor nodes and is able to identify each IoT sensor node by its ID, where the sensor node ID is the name of the node.

Algorithm 6 IoT Edge Device Level

1) Input: Thr, r, $S_{1\times n}[t]$ and $S_{1\times n}[t-1]$, β,n
2) Output:
3) Begin:
4) $\mathbb{F} \leftarrow$ **PCIE**(B_L, T_B, CMS)// algorithm
5) **If** SUM$(\{\mathbb{F}\}) = 0$
6) $\mathbb{RD} \leftarrow$***RTD***$(S_{1\times n}[t], S_{1\times n}[t-1])$
7) **If** Max $\{ABS(\mathbb{RD})\} > \beta$
8) $\{\boldsymbol{Err}, EV\} \leftarrow$ ***SDV***$(\boldsymbol{S_p}(t), \mathbb{B}, Thr, \mathbf{w})$
9) **If** $\boldsymbol{Err} = \mathbf{0}$ or $EV = 1$
10) Call MDR//
11) C $\leftarrow$ C + 1
12) **Else**
13) $E \leftarrow E + 1$
14) **End if**
15) **End if**
16) **Else**
17) $\boldsymbol{D}[t] \leftarrow$ BinToDec(F)
18) Send $\boldsymbol{D}[t]$ to FC
19) **End if**

Algorithm 7 IoT FC Level

1) **Input:** $\boldsymbol{D}[t]$, $S_{1\times n}[t-1]$, n
2) **Output:** $\hat{S}_{1\times n}[t]$
3) **Begin:**
4) L1 $\leftarrow \lfloor log2(\text{D[t]}))\rfloor + 1$
5) **If** L1 > 3
6) $\boldsymbol{m} \leftarrow \|\boldsymbol{L1/n}\|$
7) $\mathbb{D}$b $\leftarrow$**DecToBin**$(\boldsymbol{D}[t], m \times n)$
8) **For** i = 1 to n **do**
9) X(i) $\leftarrow Db((m \times i) + 1 - m : m \times i))$
10) D(i) $\leftarrow$ **BinToDec**$(X$ (i))
11) **If** D (i) $> 2^{\wedge m-1}$
12) RD (i) $\leftarrow$ D (i) $- 2^{\wedge m-1}$
13) **Else**
14) RD (i) $\leftarrow$ D (i) $\times -1$
15) **End if**
16) **End for**
17) $\hat{S}_{\mathbf{1\times n}}[\mathbf{t}] \leftarrow$***Approx***$(\mathbb{RD}, \mathbf{S_{1\times n}}[\mathbf{t-1}])$
18) $\mathbf{S_{1\times n}}[\mathbf{t-1}] \leftarrow \hat{S}_{\mathbf{1\times n}}[\mathbf{t}]$
19) **End if**

After the FC receives the reduced data D [t] from the IoT sensor node, we determine the total number of bits of the received data D[t] by applying Eq. (22).

$$\boldsymbol{L1} = \lfloor \log 2(\text{D[t]}))\rfloor + 1 \tag{22}$$

If the size of the received data is 3 bits, which means that the IoT sensor board is not in a good condition, do the action based on the frame information as shown in TABLE 6. Otherwise, the received data will pass through approximation phase as follows (See in algorithm7). *First*, we estimate the number of bits for each sensor by applying $\boldsymbol{m} = \|\boldsymbol{L1/n}\|$, $\|\quad\|$ denotes the nearest integer to m. *Next*, we convert $D[t]$ from decimal to binary based on BCD code $Db = Dec2bin(D\,[t]\,, m \times n)$, where $(m \times n)$ is the number of bits. Then, we predict the relative difference for each sensor RD_i by taking m-bits from right to left, Db is stated as the following

$$D_i = Db[(m \times i) + 1 - m : m \times i)] \tag{23}$$

After that, we convert D_i from binary to decimal as follows;

$$RD_i = \begin{cases} \text{Bin2Dec}(D_i) - 2^{m-1}, & \text{Bin2Dec}(D_i) > 2^{m-1} \\ \text{Bin2Dec}(D_i) \times -1, & \text{Bin2Dec}(D_i) < 2^{m-1} \end{cases} \tag{24}$$

Finally, we predict the $\hat{S}_{1\times n}\,[t]$ sensed data $\hat{S}_{1\times n}\,[t]$ at the time [t] at the IoT sensor node level by applying Eq.(21). For the next cycle, we **Set** $S_{1\times n}\,[t-1] = \hat{S}_{1\times n}\,[t]$. More detail is shown in the following pseudocode.

V. IMPLEMENTATION AND PERFORMANCE EVALUATION

Performance evaluations of the proposed RDCM model are done using different real-time data sets as follows:

TABLE 7. Transmission decisions based on the status of the current sensed data.

RE	*CSE*	*OTE*	*EV*	*Status*	*Decision*
0	0	0	0	Normal –State	Transmitted (ON) Update Buffer
1	0	0	0	Out-Rang data	Transmitted (Off)
0	1	0	0	Frozen data	Transmitted (Off)
0	0	1	0	Event –State	Transmitted (ON) Update Buffer
0	0	1	1	Outlier-data	Transmitted (Off)
0	0	0	1	Error-Data	Transmitted (Off)

(i) "Intel Berkeley Research Lab dataset (IBRL). IBRL wireless network recorded various types of sensed data as follows; air temperature, air humidity, light and voltage" [52]; (ii) "Grand St. Bernard dataset (GSB). GSB network used sensor nodes to measure the metrological characteristics of the environment which are ambient temperature, surface temperature and relative humidity " [53]; (iii) "Lausanne Urban Canopy Experiment dataset (LUCE). LUCE measure critical environment quantities which are ambient temperature, surface temperature and relative humidity" [54]; (iv) UTHM_LAB measure air quality which are temperature and humidity [36]. MATLAB is used to simulate the algorithms effect in the performances of IoT edge node. The proposed RDCM model is evaluated using different benchmark real-time datasets as shown in Table 7 and Table 9. These datasets and network structure (See in Figure 3) are commonly used to evaluate the performance of some existing approaches in WSN (See in Tables 2–5). The assumptions of the simulation system model are summarized as follows [20], [36], [47], [48]:

i Each IoT sensor board has different sensors as shown in Figure 3, $S = \{S_1, S_2, \ldots, S_n\}$, S_i i-th sensor, $= \{1, 2, \ldots, n$ and (n) is the number of sensors.

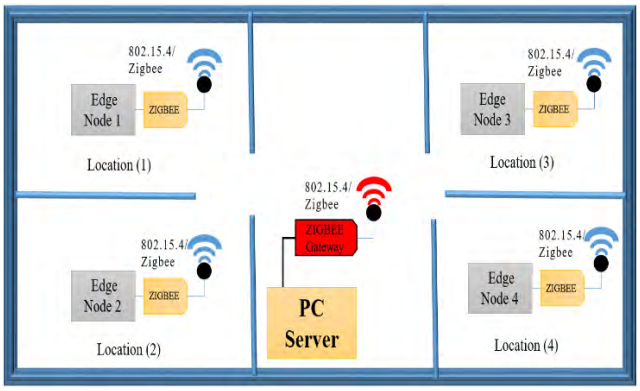


FIGURE 3. Network structure.

For example, the humidity, temperature, atmospheric pressure, carbon dioxide and some other sensors are supported on IoT *Lobelia Waspmote Gases board*.

ii The IoT sensor board must update its data to the fusion center continuously at a specific interval time.

iii Each sensor board is able to directly update its measured data to the fusion center. In other words, the sensor node does not need to use two or more wireless hops to convey information from its location to the fusion center.

iv The energy consumption for transmission of one byte is $52.92\mu J$ in calculated for MICA2Dot mote.

v The energy consumption in the case of no transmission (Off) is $0\mu J$.

Real-Time Definition: In general, real-time data (RTD) is data that is provided directly after aggregation. The sensor node transmits the measured data to the fusion center without any delay. The simple meaning of real-time sensed data is that it is information that is not saved or stored, instead, it is provided to the end-user/gateway as it is collected. The RTD does not actually mean that the data will reach the end-user immediately as there may be presence of bottlenecks correlated to the data collection structure, bandwidth between numerous events, or slowness of the computer of the end-user. Unfortunately, the RTD does not promise sensed information within a certain number of microseconds. It only means that the sensed data is not planned to be kept back from its eventual use after it is collected [48]. The authors declared that, in this paper, the word "real-time" refers to the real dataset and, also to show that the proposed model has been applied for the sensed data after sensing immediately at the sensor node level. Figure 4 shows some samples of real-time dataset versus some samples of real-time datasets have been injected by random errors. In this study, the real-time data set is original datasets that have been collected by sensor nodes without any change in its values. The injected dataset is a real-time dataset after injected with some artificial errors.

TABLE 8. Summary of the characterises of real-time dataset used in this work.

Dataset	No.Sensors	Sensors	Time to Update data	Selected nodes
IBRL	4	T,RH,L and V	31s	N1, N2, N3, N4, N7, N8 ,N9,N33 and N35
LUCE	3	A.T , RH and S.T	45s	N10
GSB	3	A.T , RH and S.T	45s	N9 , N11
UTHM	2	T and H	30s	N1

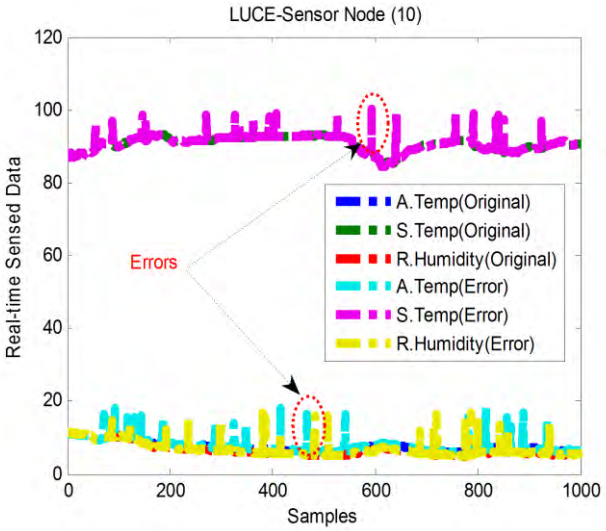


FIGURE 4. Real-time dataset (original) vs injection real-time dataset with random errors.

A. THE EFFECT OF β (TOLERANCE UPDATE DATA) TO RDCM ALGORITHM

In the related works to reduce the number of transmitted packets is supported by a single variable data. These methods require different thresholds for sensor boards with multiple sensors. Therefore, this work proposes a simple and effective updating data strategy. The proposed method aims to reduce the number of messages transmitted by a sensor board with multiple sensors based on the relative difference between the current and last sensor measurements transmitted. The advantage of this solution is that it prevents any transmission if the payload sensing block does not report a significant change. The proposed method uses only one threshold for multiple sensors on the same board (See in the *Algorithm5* step 7).

This section examines the effect of β on the performance of the proposed model. The value of β is set to 0%, 1%, 2%, 3%, 4% and 5%. RDCM applies for different real-time data sets and various nodes. This study used real-time data set with

TABLE 9. Summary of the type of sensors for all selected nodes.

NO	Sensor board	Equipped sensors					
		T	RH	L	V	AT	ST
N1	LUCE_N10	×	√	×	×	√	√
N2	GSB_N9	×	√	×	×	√	√
N3	GSB_N11	×	√	×	×	√	√
N4	IBRL-N33	√	√	√	√	×	×
N5	IBRL-N2	√	√	√	√	×	×
N6	IBRL-N35	√	√	√	√	×	×
N7	IBRL-N10	√	√	√	√	×	×
N8	IBRL-N7	√	√	√	√	×	×
N9	IBRL-N8	√	√	√	√	×	×
N10	IBRL-N9	√	√	√	√	×	×
N11	IBRL-N1	√	√	√	√	×	×
N12	IBRL-N4	√	√	√	√	×	×
N13	UTHM-N1	√	√	×	×	×	×

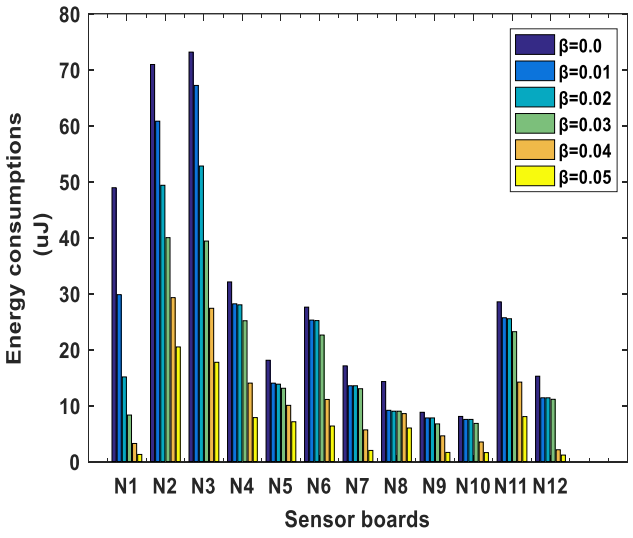


FIGURE 5. Average of Energy consumption by applying RDCM with various β values.

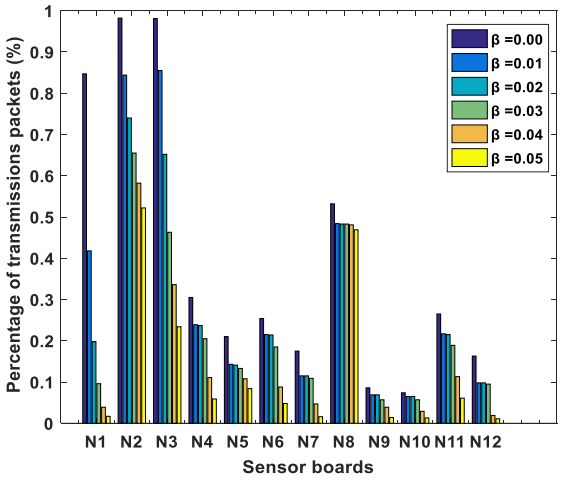


FIGURE 6. Percentage of transmissions packets (%) by applying RDCM with various β values.

no change in its content (no injection error). The results of the study in this section is shown in Figure 5 and Figure 6, respectively. The results show that energy consumption is reduced by applying RDCM with $\beta = 5\%$, which is better as compared to other β values. From the results, a significant increase in β will reduce the energy consumption of the IoT sensor. The reason is that the fusion center can only be updated if the difference between the current sensor data value and the previously sent data is lower than β. Although the increase in β reduces the number of transmitted packets and thus saves energy, as the results show, some nodes only sent 17 of 1000 samples. This will inadvertently affect the system accuracy. The advantages of the proposed model are that its β value can be easily adjusted depending on the application used. In order to obtain the highest possible accuracy in this study, the β value was set to 0%. Hence, RDCM and APRS can reduce the amount of bit payload sensing data before sending it to the fusion center.

B. THE ANALYSES OF RDCM-VALIDATION PHASE

In our work [47] we suggested a novel method for estimating the error threshold value during the training phase. The outcome showed that the adaptive threshold is better than the non-adaptive threshold with respect to decreasing the number of times the model required updating of its reference parameters, which positively affected prolonging the IoT sensor node lifetime. Moreover, adapting the threshold produced more accurate results. Therefore, the proposed threshold for the models that are being used to observe outliers or recover the sensed data in IoT/WSN real-time application would help in increasing the accuracy of these systems. The RDCM verifies the validity of the current sensed data before sending it to the fusion center.

C. PERFORMANCE OF THE RDCM-VSNL METHOD

In RDCM – validation phase as shown in the section that described RDCM model, sensed data validation checks for only one attribute from multiple attributes. The attribute (sensor) is denoted as $S_p(t)$ and it has a high correlation with other sensors on the same IoT sensor board. The RDCM- VSNL method is used for this purpose in order to examine performance of the RDCM-VSNL method during RDCM–validation phase. In this subsection, RACAD_UTHM and IRBL-Intel datasets have been injected randomly with 10% errors of 40627 and 1000 samples, respectively.

From the simulation results as shown in Table 10, it is clear that the RDCM- VSNL is able to observe the sensed data errors in the real-time during data collection with high performance and the average of accuracy for all examined sensors is around 97 %.

TABLE 10. Results of apply RDCM- VSNL for different data sets.

Real-Time Datasets	Sensors	Errors (10%)	Detect	Accuracy	Samples
RACAD_UTHM	Temp	406	405	0.9975	40627
	Humdity	406	405	0.9975	
IRBL-Intel	Temp	10	10	1	1000
	Humdity	10	10	1	
	Voltage	10	9	0.9	
Avarage				0.97	

D. PERFORMANCE OF RDCM-PREDICTED METHOD FOR ADAPTIVE THRESHOLD (thrd) WITH VARIOUS BUFFER SIZES (W)

This section investigates the effect of buffer size W on the adaptive threshold in Predicted model performance. The value of w is set to 5, 15, 25, 35, 45 and 55. RDCM-Predicted method applies for real-time data sets and various number of samples are 500, 1000, 2000 and 6000. This study uses the real-time data set with no change in its content. RDCM-Predicted model is a simple Machin learning approach called linear regression with multiple variables.

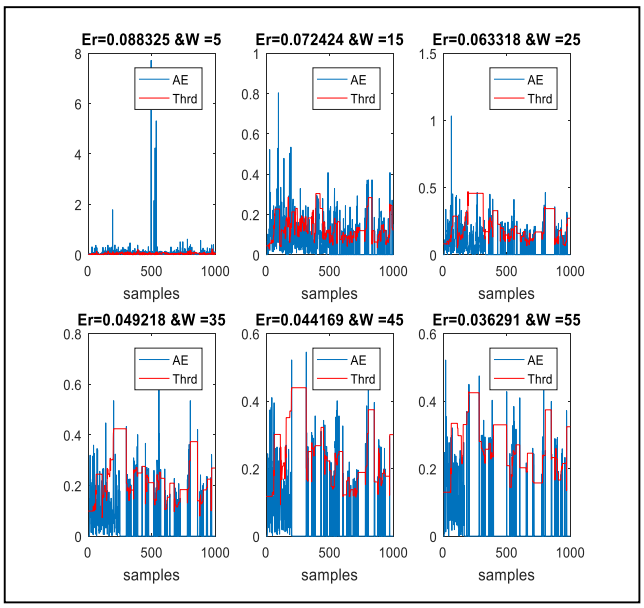


FIGURE 7. Absolute Error (AE) vs adaptive threshold (Thrd) for real-time dataset 1000 samples with various buffer size W.

Figure 7 to Figure 11 show the simulation results of study of RDCM-Predicted model for real-time data set with 500, 1000, 2000 and 6000 samples and various buffer size W. The maximum average absolute error occurs when W was 5, and no slight change in the adaptive threshold during

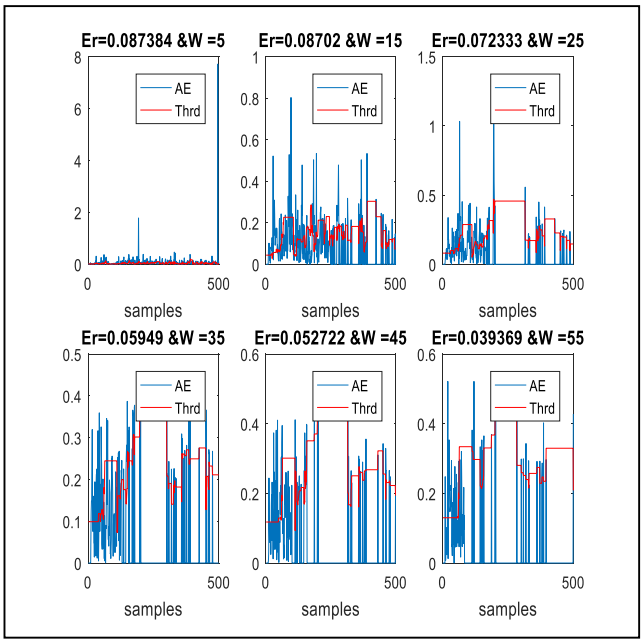


FIGURE 8. Absolute Error (AE) vs adaptive threshold (Thrd) for real-time dataset 500 samples with various buffer size W.

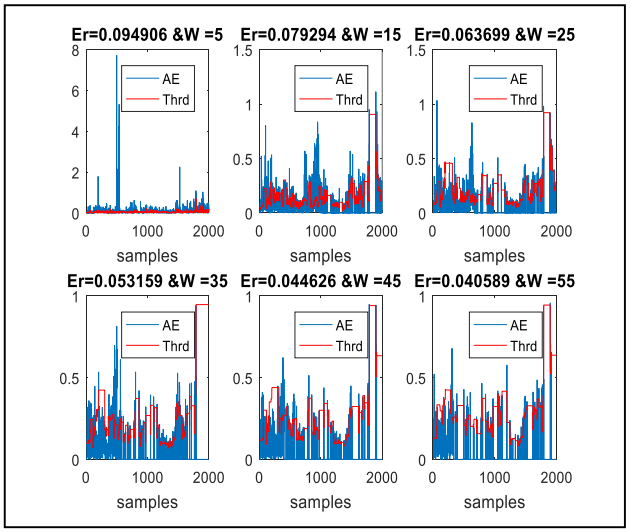


FIGURE 9. Absolute Error (AE) vs adaptive threshold (Thrd) for real-time dataset 2000 samples with various buffer size W.

data collecting. This is because the small size of W effects the accuracy of determining the model parameters. In contrast, using a large size training model W effects the efficiency of the IoT sensor node due to the resource constraints. Therefore, in this study, a small value of w was chosen, ranging from 5 to 55. This is to obtain acceptable performance considering the IoT sensor node component constraints. From the results it could be estimated that the model shows a better performance when the value of W is more than 15 samples. In addition, as long as the model RDCM - the Predicted model needs to update its reference parameters, the value of the adaptive threshold changes dynamically. The importance of changing the threshold according to the training of the model

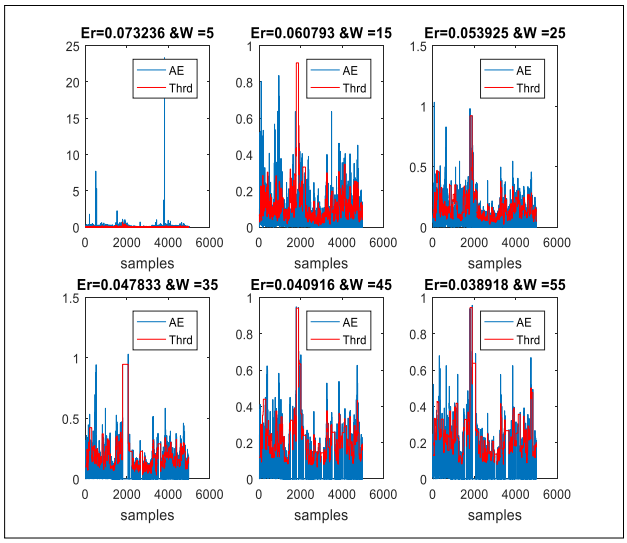


FIGURE 10. Absolute Error (AE) vs adaptive threshold (Thrd) for real-time dataset 6000 samples with various buffer size W.

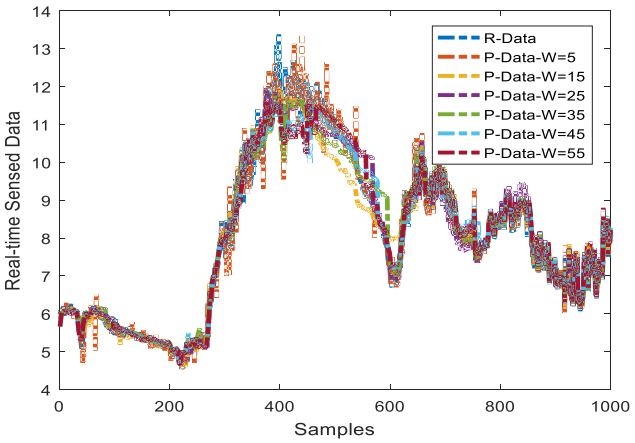


FIGURE 11. Predicted data (P-Data) vs real-time data(R-Data) with various buffer size W.

is very important because the model here is used to detect irregular data. The lack of accuracy in the model reduces the performance of the pattern in detecting errors or events during real-time data collection.

E. PERFORMANCE COMPARISON RDCM WITH DIFFERENT ALGORITHMS

In this section the performance of various algorithms in terms of energy consumption is investigated. It should be noted that this study used the original datasets with no change in its content values. In order to analyze the performance of RDCM, EDCD2 and APRS algorithms with real-time datasets that have some errors, the original real-time datasets are randomly injected with different percentages of errors. The percentage of errors is set to 1%, 2% 3%, 4%, 5%, 6%, 7%, 8%, 9% and 10%. The number of samples is 1000.

Figure 12, Figure 13, and Figure 14 show the energy consumption (μJ) results for APRS, EDCD2, and RDCM

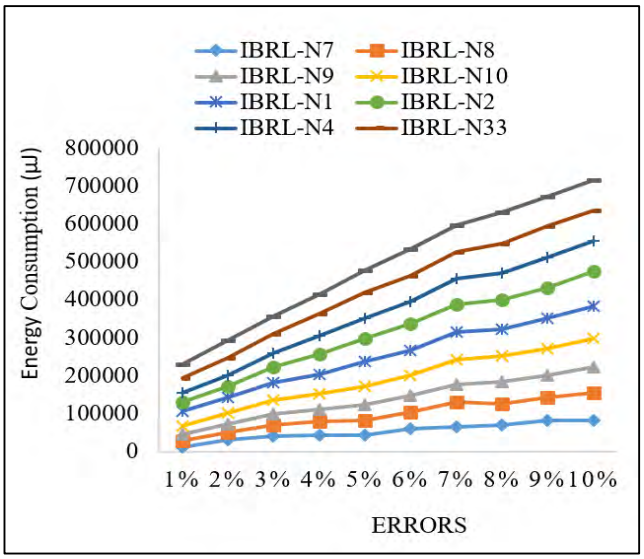


FIGURE 12. Energy consumption (μJ) by applying APRS algorithm for IBRL – sensor nodes with different percentages of errors.

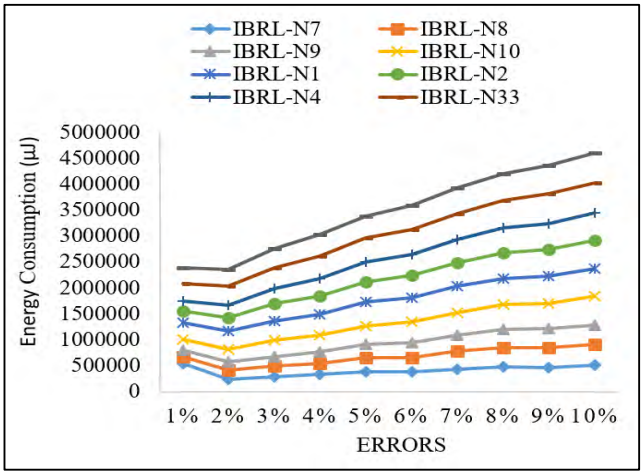


FIGURE 13. Energy consumption (μJ) by applying EDCD2 algorithm for IBRL – sensor nodes with different percentages of errors.

algorithms applied to IBRL sensor nodes with different error percentages, respectively. Obviously, increasing the number of incorrect data transmissions will affect the energy of the IoT sensor board because the sensor board wastes its energy sending incorrect data that will be omitted at the fusion center. However, if the fusion center cannot detect the erroneous data received, then those errors will affect the accuracy of the whole system.

RDCM shows better performance than other algorithms, as shown in Figure 15. This is because RDCM can detect errors and ignore them during real-time data collection of IoT/WSN applications. In addition, RDCM can reduce the number of transmission packets and reduce the number of transmission bits of payload data. The average of the energy saving ratio for the algorithms RDCM, APRS, and

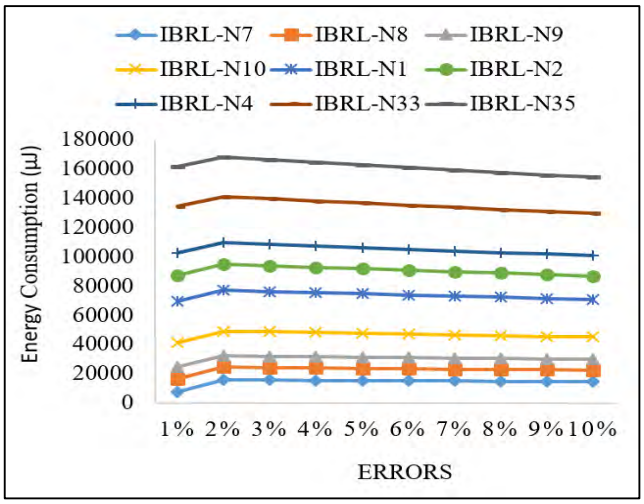


FIGURE 14. Energy consumption (μJ) by applying RDCM algorithm for IBRL – sensor nodes with different percentages of errors.

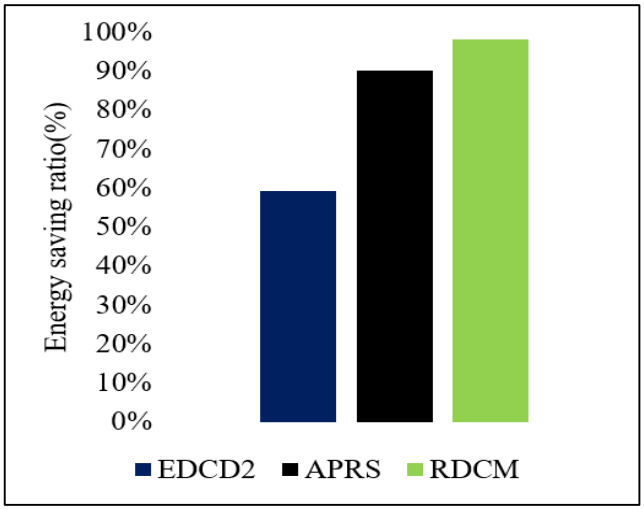


FIGURE 15. The average of the energy saving ratio for all IBRL-Sensor nodes with various errors (1%-10%).

EDCD2 applied to all IBRL-Sensor nodes with various errors (1%-10%) is 98%, 90% and 58%, respectively.

F. ANALYSIS THE ENERGY CONSUMPTION MODEL BASED ON RDCM

Table 11 shows the qualitative comparison of the proposed algorithms in energy saving. Compared with other solutions, RDCM has the advantage of saving energy because it solves most of the problems of wasting IoT board energy during data collection. Figure 16 shows the total energy consumption for applied EDCD2, VSNL, APRS, PCA-B, MLR-B, RDCM and Direct to real-time data LUCE– sensor node (N10) with its measured value injected with 2% errors of 1000 samples. The results show that sending the sensing data directly (without any algorithm) has the worst performance. The applied VSNL, PCA-B, MLR-B and EDCD2 show different performance because each save the energy of the IoT sensor board

TABLE 11. A qualitative comparison of the presented algorithms in this paper in terms of energy saving.

Method	Issues		
	Reducing the number of transmitted-Normal data	Reducing the number of transmitted Error Data -	Reducing -Size of transmitted data
EDCD2	√	×	×
VSNL	×	√	×
APRS	√	×	√
PCA-B	×	×	√
MLR-B	×	×	√
RDCM	√	√	√

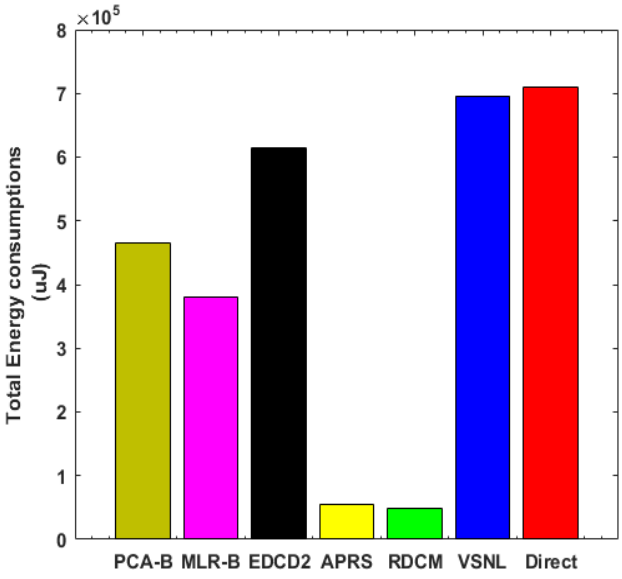


FIGURE 16. Total energy consumption for applied different algorithms to Real-Time Datasets LUCE –N10 (1000 samples) injected with 2% error.

by addressing only one issue. For example, the PCA-B and MLR-B algorithms can only reduce the size of the transmission. However, they cannot reduce the number of transmissions or observe errors. VSNL can only reduce incorrect data transmission. The EDCD2 can reduce the number of transmitted packets only if the current sensed data does not change significantly compared to the last transmitted data. APRS and RDCM show high performance because it solves several problems, as shown in Table 11.

VI. CONCLUSION

This paper introduces a new model designed to save the energy consumption of IoT sensor board, which is denoted as RDCM. RDCM in a form of general structure is composed of two main levels; IoT sensor board level and fusion center level. IoT sensor board level is implemented in real-time by all IoT sensor boards simultaneously in each cycle and fusion center level is executed by the fusion center. IoT sensor board level includes various stages as follows; (i) check the physical conditions of the IoT edge device (board) stage, (ii) update data strategy stage (iii), data validation stage and (iv) sensed

data reduction stage. The average of the total percentage of energy saved by applied RDCM to real-time data sets injected with various percentage of errors for all nodes is 98%. In summary, RDCM has a very high performance in terms of energy consumption compared to other algorithms.

The research stated in this paper reveals some possible further research opportunities as follows:

i This work proposes solution to reduce size of payload data only from whole of packet during transmission phase. It is recommended to propose a new scheme to reduce whole packet size.

ii This work assumes that the sensor node is able to send the data directly (One hop) to the FC/BS. It is recommended to design a new data collection model for multivariate sensors in IoT applications with consideration of the multi-hop network.

iii The algorithms EDCD, VSNL, ARPS, MLR-B, PCA-B and RDCM discussed in this paper are analyzed through environment dataset for smart and green blinding application. It is recommended to analyze those algorithms with vibration dataset for industrial application. In addition, apply those algorithms for wearable health-care application and logistic application.

iv The MRL-B and PCA-B models cannot reduce the number of transmitted packets. It is recommended to design a hybrid model involving those models with the EDCD algorithm.

v Similarly, the PCA-B is based on a lightweight version from PCA. It is recommended to use the adaptive threshold which was proposed in this paper with PCA-B for anomaly detection at the cloud/FC level.

ACKNOWLEDGMENT

The authors would like to express their thanks to Universiti Tun Hussein Onn Malaysia (UTHM) for support.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[3] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors J.*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.

[4] M. M. Rodgers, V. M. Pai, and R. S. Conroy, "Recent advances in wearable sensors for health monitoring," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3119–3126, Jun. 2015.

[5] H. Ghayvat, J. Liu, S. C. Mukhopadhyay, and X. Gui, "Wellness sensor networks: A proposal and implementation for smart home for assisted living," *IEEE Sensors J.*, vol. 15, no. 12, pp. 7341–7348, Dec. 2015.

[6] P. Gope and T. Hwang, "BSN-care: A secure IoT-based modern healthcare system using body sensor network," *IEEE Sensors J.*, vol. 16, no. 5, pp. 1368–1376, Mar. 2016.

[7] A. Kumar and G. P. Hancke, "A Zigbee-based animal health monitoring system," *IEEE Sensors J.*, vol. 15, no. 1, pp. 610–617, Jan. 2015.

[8] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3846–3853, Oct. 2013.

[9] S. V. Girish, R. Prakash, and A. B. Ganesh, "Real-time remote monitoring of indoor air quality using Internet of Things (IoT) and GSM connectivity," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. New Delhi, India: Springer, 2016, pp. 527–533.

[10] D. Wu, D. I. Arkhipov, M. Kim, C. L. Talcott, A. C. Regan, J. A. McCann, and N. Venkatasubramanian, "ADDSEN: Adaptive data processing and dissemination for drone swarms in urban sensing," *IEEE Trans. Comput.*, vol. 66, no. 2, pp. 183–198, Feb. 2017.

[11] D. Wu, X. Nie, E. Asmare, D. Arkhipov, Z. Qin, R. Li, J. McCann, and K. Li, "Towards distributed SDN: Mobility management and flow scheduling in software defined urban IoT," *IEEE Trans. Parallel Distrib. Syst.*, to be published.

[12] N. A. M. Alduais, A. Jamil, and J. Abdullah, "Performance evaluation of different logical topologies and their respective protocols for wireless sensor networks," *ARPN J. Eng. Appl. Sci.* vol. 10, Jan. 2015, Art. no. 862534.

[13] I. B. Arbi, F. Derbel, and F. Strakosch, "Forecasting methods to reduce energy consumption in WSN," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2017, pp. 1–6.

[14] G. B. Tayeh, A. Makhoul, D. Laiymani, and J. Demerjian, "A distributed real-time data prediction and adaptive sensing approach for wireless sensor networks," *Pervasive Mobile Comput.*, vol. 49, pp. 62–75, Sep. 2018.

[15] L. Tan and M. Wu, "Data reduction in wireless sensor networks: A hierarchical LMS prediction approach," *IEEE Sensors J.*, vol. 16, no. 6, pp. 1708–1715, Mar. 2016.

[16] F. Strakosch and F. Derbel, "Fast and efficient dual-forecasting algorithm for wireless sensor networks," in *Proc. Sensor*, 2015, pp. 859–863.

[17] F. A. Aderohunmu, G. Paci, D. Brunelli, J. D. Deng, and L. Benini, "Prolonging the lifetime of wireless sensor networks using light-weight forecasting algorithms," in *Proc. IEEE 8th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, Melbourne, VIC, Australia, Apr. 2013, pp. 461–466.

[18] L. Mesin, S. Aram, and E. Pasero, "A neural data-driven approach to increase Wireless Sensor Networks' lifetime," in *Proc. World Symp. Comput. Appl. Res. (WSCAR)*, Jan. 2014, pp. 1–3.

[19] M. Arunraja and V. Malathi, "Collective prediction exploiting spatio temporal correlation (CoPeST) for energy efficient wireless sensor networks," *KSII Trans. Internet Inf. Syst.*, vol. 9, no. 7, pp. 2488–2511, Jul. 2015.

[20] N. A. M. Alduais, J. Abdullah, A. Jamil, and L. Audah, "An efficient data collection and dissemination for IOT based WSN," in *Proc. IEEE 7th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Vancouver, BC, Canada, Oct. 2016, pp. 1–6.

[21] 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*, Feb. 2018, pp. 729–735.

[22] 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, Feb. 2018.

[23] T. B. Matos, A. Brayner, and J. E. B. Maia, "Towards in-network data prediction in wireless sensor networks," in *Proc. ACM Symp. Appl. Comput.*, Sierre, Switzerland, Mar. 2010, pp. 592–596.

[24] C. Carvalho, D. G. Gomes, N. Agoulmine, and J. N. de Souza, "Improving prediction accuracy for WSN data reduction by applying multivariate Spatio-temporal correlation," *Sensors*, vol. 11, no. 11, pp. 10010–10037, Oct. 2011.

[25] N. A. M. Alduais, J. Abdullah, A. Jamil, L. Audah, and R. Alias, "Effect of data validation schemes on the energy consumptions of edge device in IoT /WSN," in *Proc. 14th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2018, pp. 77–81.

[26] M. A. Rassam, A. Zainal, and M. A. Maarof, "Advancements of data anomaly detection research in wireless sensor networks: A survey and open issues," *Sensors*, vol. 13, no. 8, pp. 10087–10122, Aug. 2013.

[27] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 159–170, Apr. 2010.

[28] A. Ayadi, O. Ghorbel, A. M. Obeid, and M. Abid, "Outlier detection approaches for wireless sensor networks: A survey," *Comput. Netw.*, vol. 129, pp. 319–333, Dec. 2017.

[29] Y. Zhang, N. Meratnia, and P. Havinga, "Adaptive and Online one-class support vector machine-based outlier detection techniques for wireless sensor networks," in *Proc. Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2009, pp. 990–995.

[30] O. Ghorbel, W. Ayedi, H. Snoussi, and M. Abid, "Fast and efficient outlier detection method in wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 6, pp. 3403–3411, Jun. 2015.

[31] V. Garcia-font, C. Garrigues, and H. Rifà-Pous, "A comparative study of anomaly detection techniques for smart city wireless sensor networks," *Sensors*, vol. 16, no. 6, p. 868, Jun. 2016.

[32] M. A. Rassam, M. A. Maarof, and A. Zainal, "Adaptive and Online data anomaly detection for wireless sensor systems," *Knowl.-Based Syst.*, vol. 60, pp. 44–57, Apr. 2014.
[33] Y. Hu, H. Chen, G. Li, H. Li, R. Xu, and J. Li, "A statistical training data cleaning strategy for the PCA-based chiller sensor fault detection, diagnosis and data reconstruction method," *Energy Buildings*, vol. 112, pp. 270–278, Jan. 2016.
[34] J. Ravichandran and A. I. Arulappan, "Data validation algorithm for wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 9, no. 12, Dec. 2013, Art. no. 634278.
[35] F. E. Grubbs, "Sample criteria for testing outlying observations," *Ann. Math. Statist.*, vol. 21, no. 1, pp. 27–58, 1950.
[36] N. A. M. Alduais, J. Abdullah, A. Jamil, L. Audah, and R. Alias, "Sensor node data validation techniques for realtime IoT/WSN application," in *Proc. 14th Int. Multi-Conf. Syst., Signals Devices (SSD)*, Marrakech, Morocco, Mar. 2017, pp. 760–765.
[37] S. A. Haque, M. Rahman, and S. M. Aziz, "Sensor anomaly detection in wireless sensor networks for healthcare," *Sensors*, vol. 15, no. 4, pp. 8764–8786, Apr. 2015.
[38] R. N. Enam and R. Qureshi, "An adaptive data aggregation technique for dynamic cluster based wireless sensor networks," in *Proc. 23rd Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2014, pp. 1–7.
[39] N. A. M. Alduais and J. Abdullah, and A. Jamil, "Enhanced payload data reduction approach for cluster head (CH) nodes," *Telkomnika*, vol. 15, no. 3, pp. 1477–1484, 2017.
[40] L. Mesin, S. Aram, and E. Pasero, "A neural data-driven approach to increase wireless sensor networks' lifetime," in *Proc. World Symp. Comput. Appl. Res. (WSCAR)*, Jan. 2014, pp. 1–3.
[41] M. A. Rassam, A. Zainal, and M. A. Maarof, "Principal component analysis-based data reduction model for wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 18, nos. 1–2, pp. 85–101, 2015.
[42] S. Raybaud, G. Bontempi, M. L. Group, and B. Triomphe, "Distributed principal component analysis for wireless sensor networks," *Sensors*, vol. 8, no. 8, pp. 4821–4847, Aug. 2008.
[43] A. L. L. de Aquino, "A framework for sensor stream reduction in wireless sensor networks," in *Proc. 5th Int. Conf. Sensor Technol. Appl.*, Paris, France, Aug. 2011, pp. 30–35.
[44] F. Chen, F. Wen, and H. Jia, "Algorithm of data compression based on multiple principal component analysis over the WSN," in *Proc. 6th Int. Conf. Wireless Commun. Netw. Mobile Comput. (WiCOM)*, Sep. 2010, pp. 1–4.
[45] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.
[46] M. A. Rassam, A. Zainal, and M. A. Maarof, "An adaptive and efficient dimension reduction model for multivariate wireless sensor networks applications," *Appl. Soft Comput.*, vol. 13, no. 4, pp. 1978–1996, Apr. 2013.
[47] N. A. M. Alduais, J. Abdullah, A. Jamil, and H. Heidari, "Performance evaluation of real-time multivariate data reduction models for adaptive-threshold in wireless sensor networks," *IEEE Sensors Lett.*, vol. 1, no. 6, Dec. 2017, Art. no. 7501204.
[48] N. A. M. Alduais, J. Abdullah, A. Jamil, and H. Heidari, "APRS: Adaptive real-time payload data reduction scheme for IoT/WSN sensor board with multivariate sensors," *Int. J. Sensor Netw.*, vol. 28, no. 4, pp. 211–229, 2018.
[49] K. A. Bispo, N. S. Rosa, and P. R. F. Cunha, "A semantic solution for saving energy in wireless sensor networks," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2012, pp. 000492–000499.
[50] L. Liu, S. Men, M. Liu, and B. Zhou, "An energy saving solution for wireless communication equipment," in *Proc. IEEE 36th Int. Telecommun. Energy Conf. (INTELEC)*, Sep./Oct. 2014, pp. 1–3.
[51] T. Miyazaki, P. Li, S. Guo, J. Kitamichi, T. Hayashi, and T. Tsukahara, "On-demand customizable wireless sensor network," *Procedia Comput. Sci.*, vol. 52, pp. 302–309, Jan. 2015.
[52] Dataset. (2004). *Intel Berkely Research Lab*. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[53] Dataset. (2007). *Grand Saint Bernard*. [Online] Available: http://sensorscope.epfl.ch/index.php/%0AEnvironmntal Data%3E
[54] (2007). *Lausanne Urban Canopy Experiment*. [Online]. Available: http://lcav.epfl.ch/cms/lang/en/pid/

NAYEF ABDULWAHAB MOHAMMED ALDUAIS received the B.Eng. degree in computer engineering from Hodeidah University, in 2007, and the Ms.Eng. degree from Universiti Tun Hussein Onn Malaysia (UTHM), in 2015. He is currently a Ph.D. Researcher in the Internet of Things (IoT) and wireless sensor networks (WSN) with the Faculty of Electrical and Electronic Engineering, UTHM, having previously worked as an Assistant Lecturer with the Faculty of Computer Science and Engineering, Hodeidah University, Yemen, from 2007 to 2013. He has authored numerous papers in journals and conference proceedings. His research interests include WSN and the IoT. He received numerous medals and scientific excellence certificates.

JIWA ABDULLAH received the B.Eng. degree in electronic engineering from Liverpool University, U.K., and the M.Sc. and Ph.D. degrees from Loughborough University, in 1990 and 2007, U.K. He is currently with the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia (UTHM), Malaysia. He has authored more than 60 publications in journals, conferences, and book chapters. He is a member of the Board of Engineers Malaysia. His main research interests include wireless sensor networks, underwater WSNs, mobile ad hoc networks, wireless communications, networking, and application of computational intelligence to communication systems, integration of WSN and the IoT, and also in the area of engineering educations.

ANSAR JAMIL received the B.Eng. and M.Eng. degrees in electronics and telecommunications from Universiti Teknologi Malaysia (UTM), Malaysia, in 2005 and 2009, respectively, and the Ph.D. degree in electronic, electrical, and systems engineering from Loughborough University, in 2015. He is currently a Lecturer with the Department of Communication Engineering, Faculty of Electric and Electronic Engineering, UTHM. He is also an Active Researcher with the Wireless and Radio Science (WARAS) center. He has two years of experience as an Electronic Engineer with the Research and Development Department of Motorola, before joining UTHM as a Tutor for eight years. His current research interests include wireless sensor networks and the IoT.