01 Oct 2022

# An Energy Efficient Smart Metering System using Edge Computing in LoRa Network

Preti Kumari

Rahul Mishra

Hari Prabhat Gupta

Tanima Dutta

*et. al. For a complete list of authors, see https://scholarsmine.mst.edu/comsci_facwork/1216*



## Recommended Citation

P. Kumari et al., "An Energy Efficient Smart Metering System using Edge Computing in LoRa Network," *IEEE Transactions on Sustainable Computing*, vol. 7, no. 4, pp. 786 - 798, Institute of Electrical and Electronics Engineers, Oct 2022.

The definitive version is available at https://doi.org/10.1109/TSUSC.2021.3049705

# An Energy Efficient Smart Metering System Using Edge Computing in LoRa Network

Preti Kumari, Rahul Mishra, Hari Prabhat Gupta, *Senior Member, IEEE*, Tanima Dutta, *Member, IEEE*, and Sajal K. Das, *Fellow, IEEE*

**Abstract**—An important research issue in smart metering is to correctly transfer the smart meter readings from consumers to the operator within the given time period by consuming minimum energy. In this paper, we propose an energy efficient smart metering system using Edge computing in Long Range (LoRa). We assume that all appliances in a house are connected to a smart meter that is affixed with Edge device and LoRa node for processing and transferring the processed smart meter readings, respectively. The energy consumption of the appliances can be represented as an energy multivariate time series. The system first proposes a deep learning based compression-decompression model for reducing the size of the energy time series at the Edge devices. Next, it formulates an optimization problem for finding the suitable compressed energy time series to reduce the energy consumption and delay of the system. Finally, the system presents an algorithm for selecting the suitable spreading factors to transfer the compressed time series to the operator in the given time. Our simulation and prototype results demonstrate the impact of the parameters of the compression model, network, and the number of smart meters and appliances on delay, energy consumption, and accuracy of the system.

**Index Terms**—Deep learning, edge device, energy efficiency, Internet of Things, LoRa, smart meter

## 1 Introduction

Internet of Things (IoT) is envisioned to improve the quality and experience of human living. As one of the potential applications of IoT [1], smart metering enables remote monitoring of energy consumption, which is essential for both the consumers (end users) and the operator (electricity utility or supplier). The amount of energy used by an appliance in the house can be represented as Energy Time Series (ETS). An ETS consists of a large number of smart meter readings. As a consumer usually has multiple appliances, they generate Energy Multivariate time Series (EMS). Monitoring of EMS helps to identify energy inefficient appliances. It also induces energy-saving behaviour of end users, load forecasting, electricity price design, and promotions for reducing energy consumption [2], [3], [4], [5], [6]. Despite these advantages, it is hard to communicate a large volume of EMS data from the consumers to the operators. The communication of such EMS not only consumes considerable energy but also incurs significant communication delay and generates substantial network traffic.

A viable solution for smart metering is Edge computing, where the local processing and storage are available close to the leaf devices (end users) [7], [8]. Due to local processing of the tasks near to the users, Edge computing reduces communication delay and energy consumption for transmitting the EMS data. The recent development of compression-decompression techniques using Machine Learning (ML) and Deep Learning (DL) models in Edge Computing can help to reduce the size of EMS at the Edge device [9], [10]. Both ML and DL are used for features extraction, but the difference is that ML requires human intervention for selecting the features, whereas DL performs this intuitively and gives the reason to use DL for compressing EMS. Such compressed EMS consumes low energy and requires smaller delay while transferring them from consumers to the operators. Fig. 1 illustrates an example scenario of smart metering where Edge computing reduces the size of EMS of the appliances before communicating to the operators.

The selection of communication protocols is another important factor for energy consumption of Edge devices. While short-range communication protocols support low coverage range and consume low energy [11], [12], [13], [14], long-range communication protocols support a wide coverage with high transmission rates but at the cost of high power consumption [15]. The Long Range Wide Area Network (LoRaWAN), with its scalable star of stars network architecture and simple medium access mechanism, fulfills the requirement of smart metering, i.e., long-range communication with low energy consumption [16]. The LoRaWAN architecture consists of LoRa Nodes (LNs), LoRa Gateway (LG), and Network Server (NS). Each LG can be connected with a limited number of LNs on a given Spreading Factor

- *Preti Kumari, Rahul Mishra, Hari Prabhat Gupta, and Tanima Dutta are with the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, Uttar Pradesh 221005, India. E-mail: {pretikri.rs.cse17, rahulmishra.rs.cse17, hariprabhat.cse, tanima.cse}@iitbhu.ac.in.*
- *Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA. E-mail: sdas@mst.edu.*

*Manuscript received 21 Apr. 2020; revised 9 Nov. 2020; accepted 7 Dec. 2020. Date of publication 6 Jan. 2021; date of current version 8 Dec. 2022. (Corresponding author: Hari Prabhat Gupta.) Recommended for acceptance by Daniel Grosu, Jiannong Cao, and Marco Brocanelli. Digital Object Identifier no. 10.1109/TSUSC.2021.3049705*

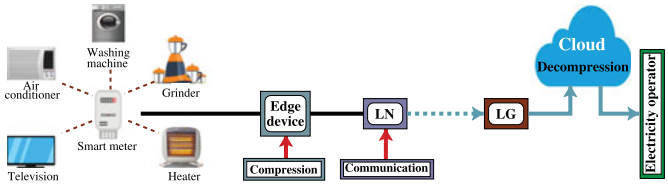



Fig. 1. Illustration of an example scenario of smart metering.

(SF). The SFs consume unequal energy and support uneven data transmission rate and communication range. Thus, the selection of appropriate SF for communicating the EMS data from LN to LG helps to reduce the energy consumption and delay of the smart metering architecture.

In this paper, we focus on the smart metering application of IoT, where a consumer has different types of appliances generating EMS data. Direct transferring of such large volume of EMS to the electricity operator consumes huge communication energy, network bandwidth, and increases the network delay. The main objective of the proposed work is to reduce the size of the EMS and successfully regenerate EMS at the electricity operator. Such regenerated EMS can be used for various purposes such as to detect fraud, reduce outage restoration times, adjust transformer loading, target energy efficiency marketing *etc*. For different purposes, multiple ML/DL models are required; thus, it is not feasible to push multiple ML/DL models to the Edge. Therefore, we are transmitting EMS to the electricity operator for multiple purposes. The energy efficient smart metering problem can be stated as: *How to efficiently transfer the EMS to the operator in a given time period with minimum energy consumption?*

To solve this problem, we present an Energy Efficient Smart Metering (EESM) system that first collects EMS, generates the high-level features using DL model, and transmits the features to the LG. Further, the LG transmits such features to the electricity operator via TCP/IP protocol or other cheaper and reliable public wireless network. The high order compression indeed reduces the energy consumption during data transmission, but increases energy consumption of the compression process as well as the decompression error. Therefore, while designing the EESM system, we need to address the following three challenges: i) Ensure that the data compression and decompression model provides the desired accuracy suitable for Edge computing; ii) Compress the EMS in an appropriate size such that the time duration for compressing and transmitting the EMS does not exceed the given time period and the energy consumption is minimized; and iii) Assign the SFs to the LoRa nodes in such a way that the energy consumption of all LoRa nodes are nearly equal.

### 1.1 Major Contributions

To the best of our knowledge, this is the first work to address the energy efficient smart metering problem using Edge computing. Our major contributions are as follows:

- *Processing at Edge Device:* We propose a DL based compression-decompression model for reducing the size of EMS at the Edge device. Specially, we use Long Short Term Memory (LSTM) for compression and decompression of EMS. We present an analysis for estimating a relationship between the size of compressed EMS and the required time and energy for compression.
- *Compressing EMS:* The EESM system formulates an optimization problem and uses a Semi-smooth Newton method for finding the suitable compressed size of the EMS, which provides an energy efficient smart metering system. Different from the existing work, we use combination of mean and variance error in the proposed model to improve its accuracy.
- *Energy Efficient Communication of EMS:* The EESM system uses LoRa network that provides long-range communication with low energy consumption. We present an algorithm for selecting the suitable SFs in LoRa network to communicate the compressed EMS from the consumer to the operators. The algorithm uses a minimum heap (MinHeap) data structure to improve the time complexity.
- *Simulation and Prototype Results:* We validate the proposed EESM system using Network Simulator-3 [17] that simulates a large number of LoRaWAN scenarios. We also build a prototype to demonstrate the impact of the compression model parameters, the network, the number of smart meters and appliances on the delay, energy consumption, and accuracy.

The rest of the paper is organized as follows. Section 2 reviews the literature that addresses smart metering problems and presents the motivation behind this work. Section 3 describes the preliminaries, while Section 4 analyzes the delay and energy consumption for compression and communication of EMS data. Section 5 presents the EESM system, followed by the experimental and prototype results in Sections 6 and 7, respectively. The conclusions are offered in Section 8.

## 2 Related Works

This section covers the related work mainly focusing on data compression, EMS transfer, and energy efficient communication protocols in smart metering applications.

*Data Compression in Smart Metering.* The authors in [2], [18], [19], [20], [21], [22], [23] proposed data compression techniques for smart metering data. The authors in [2] proposed a non-negative K-Single Value Decomposition-based sparse coding technique for smart meter data compression and compared the performance with wavelet reduction technique. An adaptive data reduction algorithm is proposed in [18] using compressive sampling technique such that the bandwidth requirement for smart meter data transmission is reduced with minimum loss of information. In [19], smart meter readings are compressed using burrow-wheeler transform and entropy encoding. Lightweight data compression technique using task offloading at the Edge device is proposed in [10]. The authors in [20] proposed adaptive single and multiple modality data compression schemes based on DL, whereas in [21], the authors presented a comparison of the performances of neural network and linear predictors for near-lossless compression of EEG signals. Authors in paper [22], designed a lossy data compression format to store key information of load features which is generated by using general extreme value

TABLE 1
Symbols Used in the Paper

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| $\mathcal{N}$ | Set of LNs | $p_{Q_n}$ | Compressed data size of LN $n$ |
| $N$ | Number of LNs | $T_n^{comp}$ | Compression time of LN $n$ |
| $n$ | Index of LN | $E_n^{comp}$ | Compression energy of LN $n$ |
| $\mathcal{S}$ | Set of SFs | $T_n^{comm}$ | Communication time of LN $n$ |
| $s_n$ | SF of LN $n$ | $E_n^{comm}$ | Communication energy of LN $n$ |
| $\lambda_n$ | Sampling rate of LN $n$ | $\mathbb{T}_n$ | Total time taken by LN $n$ |
| $Acc_n$ | Accuracy of LN $n$ | $\xi_n$ | Total energy consumption of LN $n$ |

distribution characteristic. In [23], authors proposed a dynamic compression ratio selection scheme for edge inference system with hard deadlines.

*Data Transfer in Smart Metering*. For transferring EMS data to the operators, the authors in [3], [4], [5], [6] proposed techniques to aggregate energy readings of all appliances, transfer the aggregated readings, and finally breakdown the EMS at the operators. In [3], a TreeCNN model is developed for energy breakdown on low-frequency data. The authors in [4] used Factorial Hidden Markov Model (FHMM) to breakdown EMS at operators, where each appliance is modelled as a Gaussian HMM. Sparse coding based approach [6] has been proposed to breakdown EMS at an interval of one hour.

*Communication Protocols in Smart Metering*. Recent studies on smart metering applications have shown the use of such technologies as Zigbee, Bluetooth, and WiFi for short-range [11], [12], [13], [14], and cellular networks (3G and LTE) for long-range [15] communications. For example, a ZigBee mesh network is proposed in [11] for smart metering applications with inherent redundancy, self-configuring, and self-healing capabilities. Based on Bluetooth low energy (BLE), a house energy management system is proposed in [12] which can forecast energy consumption conditions, such as predicting the house energy requirements at different times of the day. Finally, long term evolution (LTE) network is considered in [15] for communicating the smart meter reading for high throughput, low latency, and operation in plug and play mode.

### 2.1 Motivation of This Work

The work proposed in this paper is motivated by the following limitations as noted in the existing literature.

- The time complexity of the existing data compression techniques [2], [18] is very high. They require substantial processing power and energy, and hence are not suitable for Edge devices. The work in [10] assumed a fixed compression ratio. However, considering a system with a fixed compression ratio hampers efficient utilization of data transmission resources.
- Since the work in [3] considers hourly and daily smart meter readings as input and therefore, it breaks down the EMS in daily batches, which is not useful for the near real time applications. The basic premise of [5] is that common design and construction patterns for house creating a repetitive structure in their energy data. This assumption may not be practically true. Furthermore, the techniques proposed in [3], [4], [5], [6] transfer uncompressed EMS to the operators which consume huge communication energy and delay in large size energy meter readings. The considered data in the work [20] has inter-modality correlation, whereas ETS of appliances are less correlated with each other therefore, not suitable for smart metering application. In [21], authors have considered EEG signals with different sampling rates and compared the compression techniques on them separately, which is not useful for the multivariate time series data.
- The long-range communication protocols [15] suffer from colossal power consumption, while short-range communication protocols have limited coverage, and increased hardware and maintenance cost [11], [12], [13].

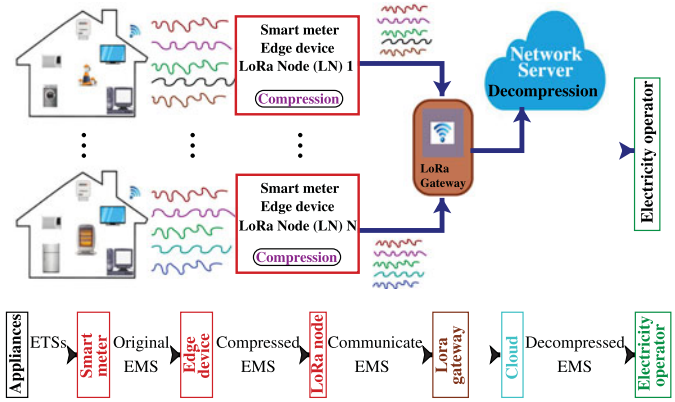



Fig. 2. An Energy Efficient Smart Metering (EESM) system.

## 3 Preliminaries

This section defines the system model, terminology and notations used in this paper. Table 1 shows the list of notations.

### 3.1 System Model

The proposed Energy Efficient Smart Metering system assumes that each house is equipped with various appliances connected with a smart meter. The smart meter works as an Edge device which collects energy readings, generates Energy Multivariate time Series, compresses EMS, and sends to the connected LG using LN. The red color rectangle box in Fig. 2 illustrates a smart meter with Edge device and LN. The EESM system uses one smart meter at each house and an apartment building of multiple houses has one LG. Since the LG is placed inside the apartment building and therefore all houses of the apartment building can easily be connected with the LG using one-hop connectivity. Recently, some industries also use LoRa for smart metering and find various advantages [24]. Let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of LNs, connected to the LG, which use the spreading factor set $\mathcal{S} = \{SF7, SF8, \ldots, SF12\}$. In this work, we use layer-based Compression-Decompression model where $Q$ is the maximum number of layers of the

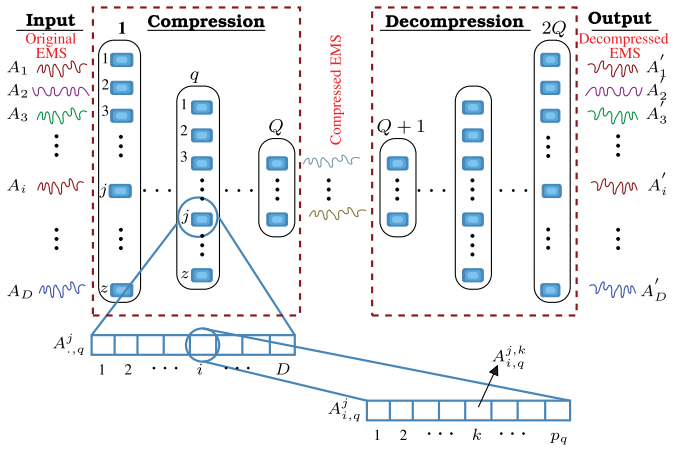



Fig. 3. Illustration of Compression-Decompression model, where symbol denotes an LSTM cell.

model for EMS compression. Let $Q_n$ denotes the layer at which LN $n$ compressed data and $p_{Q_n}$ is the compressed data size at layer $Q_n$. The data size vector $\mathbf{p}_Q$ of $N$ LNs is represented as

$$\mathbf{p}_Q = [p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}]. \tag{1}$$

Let $s_n \in \mathcal{S}$ denote the SF used by LN $n$ to transmit $p_{Q_n}$ data to the LG. The SF vector $\mathbf{s}$ for $N$ LNs is given by

$$\mathbf{s} = [s_1, s_2, \ldots, s_N]. \tag{2}$$

### 3.2 Definitions

Let $\mathscr{A} \in \mathbb{R}^{D \times Z}$ denote an EMS of size $D \times Z$, where $D$ and $Z$ are respectively the number of time series generated by the connected $D$ appliances to an LN, and the total number of events in each time series. The $D$ time series in $\mathscr{A}$ are denoted by $\mathscr{A} = \mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_D$. The Energy Time Series $\mathscr{A}_i$ consists zero and non-zero smart meter readings when the appliance $i$ is OFF and ON, respectively, where $1 \leq i \leq D$. Fig. 11 illustrates the ETSs of appliances.

**Definition 1 (Energy Multivariate time Series, or EMS).** *A smart meter reading corresponding to an ordered sequence of data points taken at a specific sampling rate in real-time is called ETS. If the reading is generated through multiple appliances, then it is called Energy Multivariate time Series.*

**Definition 2 (Accuracy).** *The accuracy of the compression-decompression model is defined as the ratio of correctly decompressed energy time series to the total energy time series generated by the appliances. Let $D_n$ denote the number of appliances connected to an LN $n$ generate energy time series. Then the accuracy of LN $n$ is defined as*

$$Acc_n = \frac{1}{D_n} \sum_{i=1}^{D_n} \mathbf{1}(x_i == \bar{x}_i), \tag{3}$$

*where $x_i$ and $\bar{x}_i$ denote a smart meter reading before and after decompression, respectively, and the function $\mathbf{1}(\cdot)$ is given as*

$$\mathbf{1}(F) = \begin{cases} 1, & \text{if } F \text{ is true,} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

### 3.3 Long Short Term Memory (LSTM) Model

The LSTM is a deep learning model that learns long term dependencies in the input sequence and extracts the temporal features. The objective of the proposed work is to regenerate the same EMS whereas the low-pass information filters out the high frequency signals and not able to regenerates the same EMS. The input vector $\mathbf{x}$ to the LSTM is a sequence of events, $\mathbf{x} = \{x_1, x_2, \ldots, x_t\}$, such as smart meter readings of an appliance at timestamp $t$. Let $\mathscr{S}(\cdot,\cdot,\cdot)$ denotes a function that combines all the LSTM operations [25]. The update of each LSTM cell with parameter $\phi$ is given by

$$\mathbf{h}_t = \mathscr{S}(\mathbf{h}_{t-1}, x_t, \phi). \tag{5}$$

## 4 Delay and Energy Analysis in EESM

In this section we first develop a deep learning based Compression-Decompression model. Next, we present the parameters learning for compression and decompression of EMS generated by appliances. Finally, we estimate the required energy and delay for compressing and communicating the compressed EMS to the Lora gateway.

### 4.1 Compression-Decompression Model

Fig. 3 illustrates the proposed Compression-Decompression model, where the compression maps a higher-dimensional point in the EMS to a lower-dimensional point. On the contrary, the Decompression maps a lower-dimensional point to the higher-dimension by arranging the layer in reverse order to that of Compression.

#### 4.1.1 Compression

We use LSTM model with $Q$ layers as Compression (see Fig. 3) for mapping input sequences $\mathscr{A} \in \mathbb{R}^{D \times Z}$ to a reduced-dimensional sequences $\mathscr{A}' \in \mathbb{R}^{D \times Z'}$, where $Z' \leq Z$. The compression divides each time series $A_i \in \mathscr{A}$ into fixed $p$ length windows, where the number of windows $z = \lceil Z/p \rceil$ and $1 \leq i \leq D$. A $j$th window of $A_i$ is denoted by $A_i^j$, where $1 \leq j \leq z$. The input $\mathscr{A}$ can be represented as the set of windows $\{\{A_1^1, \ldots, A_1^z\}, \{A_2^1, \ldots, A_2^z\}, \ldots, \{A_D^1, \ldots, A_D^z\}\}$. The $k$th event of $A_i^j$ window is denoted by $A_i^{(j,k)}$, where $1 \leq k \leq p$. The window $A_i^j$ and its length at $q$th layer of Q layers Compression model are respectively denoted by $A_{(i,q)}^j$ and $p_q$, where $1 \leq q \leq Q$, the input layer $A_{(i,1)}^j = A_i^j$, and $p_1 = p$. The index terms $\{i, j, k, q\}$ represent the time series, window, event, and layer of the Compression model, respectively. Similarly, the event $A_i^{(j,k)}$ at layer $q$ is denoted by $A_{(i,q)}^{(j,k)}$ as shown in Fig. 3. The compression uses the following mathematical expression to generate the events of window $A_{(i,q)}^j$ at LSTM layer $q$.

$$A_{(i,q)}^j = f_e(\mathbf{W}_{iq-1} \odot \mathbf{A}_{(i,q-1)} + U_{iq}^{j-1} h_{(i,q)}^{j-1} + b_{iq-1}), \tag{6}$$

where $f_e(.)$ denotes the activation function and $\odot$ the dot product. $\mathbf{W}_{iq-1} \in \mathbb{R}^{p_q \times p_{q-1}}$ and $U_{iq}^{j-1}$ are used to denote the weight metric for input of all windows i.e., $\mathbf{A}_{(i,q-1)}$ and previous window output state i.e., $h_{(i,q)}^{j-1}$, respectively, and $b_{iq-1} \in \mathbb{R}^{p_q}$ denotes the bias. The output of the last layer of compression is the compressed events $\mathscr{A}$ given by,

$$\mathscr{A}' = \{\{A_{(1,Q)}^1, \ldots, A_{(1,Q)}^z\}, \ldots, \{A_{(D,Q)}^1, \ldots, A_{(D,Q)}^z\}\},$$

where

$$A^{j}_{(i,Q)} = \{A^{(j,1)}_{(i,Q)}, A^{(j,2)}_{(i,Q)}, \ldots, A^{(j,p_Q)}_{(i,Q)}\}. \quad (7)$$

#### 4.1.2 Decompression

The Decompression gradually transfers $\mathscr{A}'$ to produce an estimate $\mathscr{A} \in \mathbb{R}^{D \times Z}$. The decompression uses the fully-connected $Q$ layers to map to each window from $\mathbb{R}^{p_Q}$ back to $\mathbb{R}^{p}$. Similar to compression, the events window $A^{j}_{(i,q)}$ is produced at layer $q$ by the following expression.

$$A^{j}_{(i,q)} = f_d(\mathbf{W'}_{iq-1} \odot \mathbf{A}_{(i,q-1)} + U'^{j-1}_{iq} h^{j-1}_{(i,q)} + b'_{iq-1}), \quad (8)$$

where $f_d(.)$ denotes the activation function, $\odot$ the dot product. $\mathbf{W'}_{iq-1} \in \mathbb{R}^{p_q \times p_{q-1}}$ and $U'^{j-1}_{iq}$ are used to denote the weight metric for input of all windows i.e., $\mathbf{A}_{(i,q-1)}$ and previous window output state i.e., $h^{j-1}_{(i,q)}$, respectively, and the bias $b'_{iq-1} \in \mathbb{R}^{p_q}$, where $Q+1 \leq q \leq 2Q$. The output of the last layer of decompression is the events $\mathscr{A}'$, i.e.,

$$\mathscr{A} = \{\{A^{1}_{(1,2Q)}, \ldots, A^{z}_{(1,2Q)}\}, \ldots, \{A^{1}_{(D,2Q)}, \ldots, A^{z}_{(D,2Q)}\}\},$$

where

$$A^{j}_{(i,2Q)} = \{A^{(j,1)}_{(i,2Q)}, A^{(j,2)}_{(i,2Q)}, \ldots, A^{(j,p_{2Q})}_{(i,2Q)}\}. \quad (9)$$

#### 4.1.3 Compression-Decompression Parameters Learning

The compression and decompression are non-linear mapping of $\mathscr{A} \rightarrow \mathscr{A}'$ and $\mathscr{A}' \rightarrow \bar{\mathscr{A}}$, respectively. By using Eqs. (5), (7) and (9), the compressed of a window $A^{j}_{(i,1)}$ at layer $Q$ and decompressed of a window $A^{j}_{(i,Q)}$ at layer $2Q$ are updated as given by

$$A^{j}_{(i,Q)} = \mathscr{S}_e(A^{j}_{(i,1)}, h^{j-1}_{i,1}, \chi), \quad (10)$$

$$A^{j}_{(i,2Q)} = \mathscr{S}_d(A^{j}_{(i,Q)}, h^{j-1}_{i,Q}, \chi'), \quad (11)$$

where $\chi = \{\mathbf{W}, \mathbf{U}, \mathbf{b}\}$ and $\chi' = \{\mathbf{W}', \mathbf{U}', \mathbf{b}'\}$ are the weight and bias parameters of compression and decompression, respectively. Using L2-norm, the mean error of the Compression-Decompression model of $A_i \in \mathscr{A}$ time series is given by

$$Err_{\bar{x}}(A_i) = \frac{1}{zp} \sum_{j=1}^{z} \sum_{k=1}^{p} \left\| A^{(j,k)}_{(i,2Q)} - A^{(j,k)}_{(i,1)} \right\|_2^2. \quad (12)$$

Next, we consider variance information of $\mathscr{A}$ and $\bar{\mathscr{A}}$ for computing the variance error [26] that is given by

$$\begin{aligned} V_j(A_{(i,1)}) &= \frac{1}{p-1} \sum_{k=1}^{p} \left\| A^{(j,k)}_{(i,1)} - \frac{1}{p} \sum_{k=1}^{p} A^{(j,k)}_{(i,1)} \right\|_2^2, \\ V_j(A_{(i,2Q)}) &= \frac{1}{p-1} \sum_{k=1}^{p} \left\| A^{(j,k)}_{(i,2Q)} - \frac{1}{p} \sum_{k=1}^{p} A^{(j,k)}_{(i,2Q)} \right\|_2^2, \\ Err_v(A_i) &= \frac{1}{z} \sum_{j=1}^{z} \left\| V_j(A_{(i,1)}) - V_j(A_{(i,2Q)}) \right\|_2^2. \end{aligned} \quad (13)$$

The objective function of the Compression-Decompression model can be expressed as the minimization of the sum of loss and regularization terms [27], and is given as

$$\begin{aligned} \mathcal{L} &= \frac{1}{D} \sum_{i=1}^{D} (Err_{\bar{x}}(A_i) + \Upsilon Err_v(A_i)), \\ \mathbf{Y}^{*} &= \arg \min_{Y = \{\mathbf{W}, \mathbf{W}', U, U'\}} \mathcal{L}. \end{aligned} \quad (14)$$

By solving Eq. (14), the Compression-Decompression model learns the weight matrices $(\mathbf{W}, \mathbf{W}', U, U')$ and the corresponding bias vectors $(\mathbf{b}, \mathbf{b}')$. The minimization of loss reciprocates to the maximization of accuracy. The accuracy of the Compression-Decompression model can be written as

$$Acc = 1 - \mathcal{L} = 1 - \frac{1}{D} \sum_{i=1}^{D} (Err_{\bar{x}}(A_i) + \Upsilon Err_v(A_i)). \quad (15)$$

Using Eq. (15), we can compute the accuracy $Acc_n$ of generated $D_n$ time series of LN $n$ as defined in Definition 2.

**Example 1.** Let an EESM system consists of an LN connected to $D$ = 24 appliances, similar as in Reference Energy Disaggregation Dataset (REDD) [28]. REDD consists of device specific electricity consumption of 10 real houses over a time period of 119 days. For each monitored house, it consists of the records up to 24 individual devices in the house (one reading consists of 24 bits of information on current and voltage). The smart meter captured the energy consumption at the interval of 3 seconds and transmitted to the electricity operator after every 12 hours. The example instance of EMS of the appliances is $\mathscr{A} \in \mathbb{R}^{24 \times 14400} = \{\{0, 0, 2, 4, 3, \ldots, 5, 3, 4, 0, 0\}, \ldots, \{0, 0, 6, 3, 2, \ldots, 7, 2, 7, 4, 3\}\}$, where zero and non-zero values show that the appliances are OFF and ON, respectively. Using the compression model, the compressed EMS is $\mathscr{A}' \in \mathbb{R}^{24 \times 5760}$ and the decompressed EMS is $\mathscr{A} \in \mathbb{R}^{24 \times 14400} = \{\{0, 0, 2, 4, 3, \ldots, 4, 3, 4, 0, 0\}, \ldots, \{\{0, 0, 6, 3, 1, \ldots, 7, 2, 7, 4, 3\}\}$. This example shows that the EESM system compressed $346K$ smart meter readings to $138K$ and therefore it needs 60 percent less communication energy.

### 4.2 Estimation of Delay and Energy Consumption

Our Compression-Decompression model is trained on a high-end machine only once to obtain an optimal configuration of weights and biases. During testing, each LN attached with the smart meter collects and compresses the EMS by using the proposed Compression model and communicates to the LG. The LG, in turn, forwards it to the wireless base station where the compressed EMS is decompressed for future processing.

#### 4.2.1 Compression

The following lemma estimates the compression delay and energy consumption at LN $n$ connected with the smart meter.

**Lemma 1.** *Given a LoRa network with $Q_n$ layers of an LN $n$, where $p_q$ is the length of the window at layer $q$. Let $\mathscr{A} \in \mathbb{R}^{D_n \times Z}$ be an Energy Multivariate Time Series of length $Z$ with $D_n$ dimensions, generated by appliances with a sampling rate of $\lambda_n$. Then the predicted delay for compressing*

*the data is given by*

$$T_n^{comp} = \sum_{i=1}^{D_n} \sum_{q=1}^{Q_n-1} \frac{\lambda_n^i \tau l_{q+1}(2p_q + \eta)}{l}. \quad (16)$$

*Let $E_c$ be the energy consumption per floating point operation, then the total energy consumption for compression is given by*

$$E_n^{comp} = \sum_{i=1}^{D_n} \sum_{q=1}^{Q_n-1} \frac{\lambda_n^i \tau l_{q+1}(2p_q + \eta)}{l} E_c. \quad (17)$$

**Proof.** To estimate the runtime, we count the required FLOating Point operations (FLOPs). Eq. (16) shows that an event window of length $l$ requires $p_q \times (2p_{q-1})$ FLOPs between layers $q-1$ and $q$ for dot products and addition of bias. The total required FLOPs per window for $Q_n$ layers compression is given by $\sum_{q=1}^{Q_n-1} 2p_q p_{q+1}$. Such compression also uses $\sum_{q=1}^{Q_n-1} p_{q+1}$ activation functions per window, requiring $\sum_{q=1}^{Q_n-1} p_{q+1}\eta$ FLOPs, where $\eta$ is the distinct operation in the activation function. The total runtime of compression per window is $\sum_{q=1}^{Q_n-1} 2p_q l_{q+1} + \sum_{q=1}^{Q_n-1} l_{q+1}\eta = \sum_{q=1}^{Q_n-1} l_{q+1}(2p_q + \eta)$. The $i$th ETS of an LN $n$ with $\lambda_n^i$ sampling rate consists of $\lambda_n^i \tau / l$ windows during $\tau$ time duration, where $1 \leq i \leq D_n$. The total FLOPs for $\mathscr{A}$ is the sum of the required FLOPs for all $D_n$ time series and given in Eq. (16). Next, the total energy consumption for compression of $\mathscr{A} \rightarrow \mathscr{A}'$ is the product of compression time and energy per FLOP operation i.e., $T_n^{comp} \times E_c$ and given in Eq. (17). □

#### 4.2.2 Communication

After successful compression of the data stored at the LNs, the compressed data is further transferred to the LG. The time required for transmission of data from an LN $n$ to the LG is known as communication time, denoted as $T_n^{comm}$. It depends on the SF onto which the LN transfers the data to the LG. Let an LN $n \in \mathcal{N}$ use SF $s_n$ and bandwidth $b$ for transmitting data to the LG with coding rate $c$. The transmission rate from LN $n$ to LG can be obtained from [16], which is given as

$$r_n = b \times \frac{s_n}{2^{s_n}} \times \frac{4}{4+c}. \quad (18)$$

Now, let $E_o$ denotes the energy consumed per unit data for communication in the EESM system. The communication delay and energy consumption for transmitting the compressed $p_{Q_n}$ data of LN $n$ are given by

$$T_n^{comm} = \frac{p_{Q_n}}{r_n}, \quad (19)$$

$$E_n^{comm} = \frac{p_{Q_n}}{r_n} E_o. \quad (20)$$

After compressing data, the LNs try to transmit data at the LG on the selected SF. Data is transmitted immediately if the selected SF is free; otherwise, the LN has to wait for getting its turn. Let LN $n$ select SF $s_n$ for transmitting data at the LG. Then in the worst case, LN $n$ has to wait $T_n^{wait} = \sum_{i=1, s_n == s_i}^{N} T_i^{comm}$ time for getting its turn. $T_n^{wait} = 0$ for LN $n$, if selected SF $s_n$ is free. Therefore the total delay of LN $n$ is the sum of compression, waiting, and communication delay, given as

$$\mathbb{T}_n = T_n^{comp} + T_n^{wait} + T_n^{comm}. \quad (21)$$

Similarly, the energy consumption of LN $n$ is the sum of the energy consumption during compression and communication which are functions of the data size $p_{Q_n}$ and SF $s_n$. It is calculated from Eqs. (17) and (20). That is,

$$\xi_n = E_n^{comp} + E_n^{comm}. \quad (22)$$

## 5 Energy Efficient Smart Metering System

The objective of the EESM system is to minimize the energy consumption for successfully transferring the data of appliances to the electricity operator using LNs and LG within the given time delay. In this section, we formulate an optimization problem for the EESM system and solve it using Semi-smooth Newton method.

### 5.1 Problem Formulation

Let LN $n \in \mathcal{N}$ uses SF $s_n$ for transmitting the compressed $p_{Q_n}$ data to the LG, where $1 \leq n \leq N$, $1 \leq Q_n \leq Q$, and $s_n \in \mathcal{S}$. Let us define *EESM Problem*

$$\min_{\mathbf{p}_Q, \mathbf{s}} \sum_{n=1}^{N} \xi_n, \quad (23a)$$

$$\text{s.t. } C1: \ \xi_n \leq E_1^{th}, \quad (23b)$$

$$C2: \ ||\xi_n - \xi_j|| \leq E_2^{th}, \quad (23c)$$

$$C3: \ Acc_n \geq \mathbb{A}^{th}, \quad (23d)$$

$$C4: \ p_n^{min} \leq p_{Q_n} \leq p_1, \quad (23e)$$

$$C5: \ \mathbb{T}_n \leq \mathbb{D}_n^{th}, \quad (23f)$$

where $\xi_n$ can be obtained from Eq. (22), $\forall n \in \mathcal{N}, j \in \mathcal{N}/n$, and $\{s_n, s_j\} \in \mathcal{S}$.

*Constants.* The input to the *EESM Problem* includes the set of LNs ($\mathcal{N}$), their connected appliances sampling rate ($\lambda_n^i$ for $i$th appliance of $n$th LN), and maximum time $\mathbb{D}_n^{th}$ within which an LN $n$ transmits compressed data to the LG.

*Variables.* The energy consumption $\xi_n$ of an LN $n$ comprises $E_n^{comp}$ and $E_n^{comm}$ which depends on the datasize $p_{Q_n}$ and SF $s_n$. Therefore, the total energy in the LoRa network depends on vector $\mathbf{p}_Q = \{p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}\}$ and SF vector $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ of all LNs.

*Objective Function.* The objective function of *EESM Problem*, denoted as $\sum_{n=1}^{N} \xi_n$, is the total energy consumed by the LoRa network of $N$ LNs for transmitting the compressed data $\mathbf{p}_Q$ on SF $\mathbf{s}$ to the LG.

*Constraints.* Constraint $C1$ indicates that the energy consumed by any LN does not exceed a threshold $E_1^{th}$. Constraint $C2$ tries to equalize the energy level of all LNs, thus increasing the lifetime of the entire network. The thresholds $E_1^{th}$ and $E_2^{th}$ are determined experimentally. Constraint $C3$ ensures that the Compression-Decompression model maintains the accuracy of LoRa network greater than a threshold. Constraint $C4$ tries to compute minimum energy

consumption among the reduced data size within the range of $p_n^{min}$ and $p_1$. Constraint $C5$ ensures that each LN scheduled on an SF transmits its data within a specified deadline.

### 5.2 Solution to the EESM Problem

The solution mechanism of the *EESM Problem* involves two steps. Step 1 aims to find out the optimal data size vector for a fixed SF of $N$ LNs by using the Semi-Smooth Newton method. In Step 2, we repeat Step 1 for all SFs to find out the optimal set of SFs i.e., $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ to transmit data to the LG of optimal size, i.e., $\mathbf{p}_Q = \{p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}\}$.

#### 5.2.1 Step 1: Solve the EESM Problem for Fixed SFs

We convert the problem into a Quadratic Programming Problem (QPP), define the Karush Kuhn Tucker (KKT) conditions, and use Semi-Smooth Newton method for searching the optimal value of $\mathbf{p}_Q = \{p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}\}$ for fixed SFs.

*Convert EESM Problem into QPP:* For this purpose, we convert the problem in Eq. (23) into QPP. In the worst case, the event is generated at the end of the event window; then the length $l$ at layer $q+1$ is the same as $p_{q+1}$, i.e., $l_{q+1} = p_{q+1}$. The energy consumption of LN $n$ for every time series data is given as follows.

$$\xi_n = \sum_{q=1}^{Q_n - 1} p_{q+1}(a_1 p_q + a_2) + a_3 p_{Q_n}, \tag{24}$$

where $a_1 = \frac{2D_n \lambda_n \tau E_c}{l}$, $a_2 = \frac{\eta D_n \lambda_n \tau E_c}{l}$, and $a_3 = \frac{E_o}{r_n}$. $Q_n$ is the number of layers needed for compressing $p_{Q_n}$ data of LN $n$. Assume that the data on each layer of the compression is $x\%$ more than the data on that layer after compression. In other words, the data at layer $q$ is $x$ % more from the data at layer $q+1$, i.e., $p_q = p_{Q_n}(x')^{Q_n - q}$, where $x' = (1 + \frac{x}{100})$ and $1 \leq q \leq Q_n$. So, Eq. (24) can be rewritten as

$$\xi_n = \underbrace{a_1 \frac{1 - (x'^2)^{2Q-3}}{1 - x'^2}}_{u_n} p_{Q_n}^2 + \underbrace{\left(a_2 \frac{1 - (x')^{Q-2}}{1 - x'} + a_3\right)}_{v_n} p_{Q_n}. \tag{25}$$

*KKT Conditions for the EESM Problem.*We use the KKT condition to solve the *EESM Problem*, where the inequality constraints are converted into equality constraints by adding slack variables, i.e.,

$$\xi_n = E_1^{th} - y_{1n}, \forall n \in \mathcal{N}, s_n \in \mathcal{S}, \tag{26}$$

$$||\xi_n - \xi_j|| = E_2^{th} - y_{2n}, \forall \{n, j\} \in \mathcal{N}, \{s_n, s_j\} \in \mathcal{S}, n \neq j, \tag{27}$$

$$Acc_n = \mathbb{A}^{th} + y_{3n}, \forall n \in \mathcal{N}, \tag{28}$$

$$p_{Q_n} = p_n^{min} + y_{4n}, \forall n \in \mathcal{N}, \tag{29}$$

$$p_{Q_n} = p_1 - y_{5n}, \forall n \in \mathcal{N}, \tag{30}$$

where $\{y_{in}\}_{i=1}^5$ are slack variables, $\forall n \in \mathcal{N}$. Let $\{\mathbf{A}_i\}_{i=1}^5$ and $\{\mathbf{b}_i\}_{i=1}^5$ are $N \times 5$ and $N \times 1$ real valued matrix, respectively, $\mathbf{u}$ and $\mathbf{v} \in \mathbb{R}^n$. Thus, the *EESM Problem* is

$$\begin{aligned} \min \quad & \mathbf{u}^T \mathbf{p}_Q^2 + \mathbf{v}^T \mathbf{p}_Q \\ s.t. \quad & \mathbf{A}\mathbf{p}_Q = \mathbf{B}, \end{aligned} \tag{31}$$

where $\mathbf{u} = \{u_1, u_2, \ldots, u_N\}, \mathbf{v} = \{v_1, v_2, \ldots, v_N\}$. Assuming $\mathbf{Acc} = \{Acc_1, Acc_2, \ldots, Acc_N\}$,

$$\mathbf{A} = \begin{Bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \\ \mathbf{A}_5 \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}^T \mathbf{p}_Q + \mathbf{v}^T \\ \mathbf{u}^T \mathbf{p}_Q + \mathbf{v}^T - \xi_j / \mathbf{p}_Q \\ \mathbf{Acc} / \mathbf{p}_Q \\ 1 \\ 1 \end{Bmatrix}, \text{and}$$

$$B = \begin{Bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{Bmatrix} = \begin{Bmatrix} E_1^{th} - y_{1n} \\ E_2^{th} - y_{2n} \\ \mathbb{A}^{th} + y_{3n} \\ p^{min} + y_{4n} \\ p_1 - y_{5n} \end{Bmatrix}.$$

Next, we introduce Lagrangian multipliers $\alpha = \{\alpha_n\}_{n=1}^N$, $\beta = \{\beta_n\}_{n=1}^N$, $\gamma = \{\gamma_n\}_{n=1}^N$, $\Gamma = \{\Gamma_n\}_{n=1}^N$, and $\zeta = \{\zeta_n\}_{n=1}^N$ one for each constraint. The Lagrangian form of Eq. (31) is as follows.

$$\begin{aligned} \mathcal{L}(\mathbf{p}_Q, \alpha, \beta, \gamma, \Gamma, \zeta) = {} & \mathbf{u}^T \mathbf{p}_Q^2 + \mathbf{v}^T \mathbf{p}_Q - \sum_{n=1}^{N} \alpha_n (\mathbf{A}_1 \mathbf{p}_Q - b_1) \\ & - \sum_{n=1}^{N} \sum_{j=1, j \neq n}^{N} \beta_{nj} (\mathbf{A}_2 \mathbf{p}_Q - b_2) + \sum_{n=1}^{N} \gamma_n (\mathbf{A}_3 \mathbf{p}_Q - b_3) \\ & + \sum_{n=1}^{N} \Gamma_n (\mathbf{A}_4 \mathbf{p}_Q - b_4) - \sum_{n=1}^{N} \zeta_n (\mathbf{A}_5 \mathbf{p}_Q - b_5). \end{aligned} \tag{32}$$

Differentiating Eq. (32) *w.r.t.*, $p_{Q_n}$, we obtain

$$p_{Q_n} = \frac{\gamma_n Acc_n + \Gamma_n - \zeta_n - v_n \delta_n}{2 u_n \delta_n}, \tag{33}$$

where $\delta_n = \alpha_n + \beta_n - 1$. The Hessian of the function can be calculated by differentiating the gradient, i.e., $\nabla_{p_{Q_n}} (\gamma_n Acc_n + \Gamma_n - \zeta_n - \delta_n (2 u_n p_{Q_n} + v_n)) = -2 u_n \delta_n$. Let assume the slack variable $\omega_n = 2 \delta_n u_n p_{Q_n}$. Finally, we use Semi-Smooth Newton method for optimal solution of the *EESM Problem*, as shown in Procedure 1.

**Procedure 1.** Semi-Smooth Newton Method for Solving EESM Problem

**Input:** **A**, **B**, u, v;
**Output:** Optimal data size $p_Q$ for fixed SFs;
1: Initialization: Data size vector $p_Q^0 \leq p_1$, slack variable $\omega_n^0 > 0$, iteration $k = 0$, terminating constant $\epsilon$;
2: **do**
3: $(p_Q)^k = diag((p_{Q_1})^k, (p_{Q_2})^k, \ldots, (p_{Q_N})^k)$;
4: $dist = \frac{\nabla^2_{p_{Q_n}} (\mathcal{L}(\mathbf{p}_Q, \alpha, \beta, \gamma, \Gamma, \zeta))}{\nabla_{p_{Q_n}} (\mathcal{L}(\mathbf{p}_Q, \alpha, \beta, \gamma, \Gamma, \zeta))}$;
5: $(p_Q)^{k+1} = (p_Q)^k + dist$;
6: $\omega_n^{k+1} = \gamma_n Acc_n + \Gamma_n - \zeta_n - \delta_n v_n$;
7: $k = k + 1$;
8: **while** $((\omega_n^k)^T (p_Q)^k < \epsilon)$;

#### 5.2.2 Step 2: Find Out the Optimal Set of Data Size and SFs

In this section, we solve the *EESM Problem* to determine the optimal set of SFs i.e., $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ to transmit to the

LG data of optimal size, i.e., $\mathbf{p}_Q = \{p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}\}$. Algorithm 1 illustrates the steps to solve the problem. Lines 2-4 of this algorithm collect data from appliances for all LNs. Lines 5-8 of the algorithm run Procedure 1 for all allocated SFs and LNs in the network. An SF $s_n \in \mathcal{S}$ is allocated to an LN $n \in \mathcal{N}$ for transmitting its data if $n$ lies in the range $[0, d_{s_n}]$ of $s_n$, where $d_{s_n}$ is the maximum distance at which the LN $n$ can transmit on SF $s_n$. Line 8 calculates the optimal data size for fixed SFs. The optimal solution must also satisfies the deadline constraint $C5$. Lines 9-11 of the algorithm ensure that the time taken by LNs for transmitting data to the LG meets deadline. If the calculated total time of any LN exceeds its deadline, then we check for the next solution, otherwise create a *min* Heap tree $H$ based on the minimum energy consumption by inserting the calculated value of $\mathbf{p}_Q$.

**Algorithm 1.** Energy Efficient Smart Metering

**Input:** Time series of smart meters connected with LNs;
**Output:** $p_Q = \{p_{Q_1}, p_{Q_2}, \ldots, p_{Q_N}\}$, $s = \{s_1, s_2, \ldots, s_N\}$;
1: Initialization: $minHeap\ H = \text{NULL}$, $Flag = 0$;
2: **for** $n \leftarrow 1$ *to* $N$ **do**
3: **for** $i \leftarrow 1$ *to* $D_n$ **do**
4: Collect data from appliance $i$ for $\tau$ time interval for all LNs;
5: **for** $s_1 \leftarrow SF7$ **to** $SF12$ **do**
6: **for** $s_2 \leftarrow SF7$ **to** $SF12$ **do**
⋮
7: **for** $s_N \leftarrow SF7$ **to** $SF12$ **do**
8: Call **Procedure 1**;
// Deadline constraint
9: **for** $n \leftarrow 1$ **to** $N$ **do**
10: $\mathbb{T}_n = T_n^{Comp} + T_n^{wait} + T_n^{Comm}$;
11: **if** $\mathbb{T}_n \geq \mathbb{D}_n^{th}$ **then**
12: $Flag \leftarrow 1$;
13: break;
14: **if** $Flag == 0$ **then**
15: Insert $minHeap(H, p_Q, s)$;
16: $Extract\_Min(H, p_Q, s)$;
17: **return** $(p_Q, s)$;

**Example 2.** Let three LNs, i.e., $N = 3$ connected to 2, 4, and 5 appliances (i.e., $D_1$=2, $D_2$=4, and $D_3$=5). Let assume that $\mathbf{p}_Q^0 = \{p_{Q_1}^0, p_{Q_2}^0, p_{Q_3}^0\}$ is the vector of initial data size of all three LNs. Procedure 1 calls Eq. (23) which leads to call Eqs. (17) and (20) for calculating the energy consumption for compression and communication, respectively. For estimating the compression energy, Eq. (17) needs the number of layers which is calculated based on the initial data size. Next, Eq. (20) calculates the communication energy on a given SF. Procedure 1 is repeated for all three LNs and calculates the optimal data size which consumes minimum energy consumption for the given SFs. This optimal data size is used at Line number 8 in Algorithm 1. Algorithm 1 checks all the combination of SFs among the LNs to calculate the optimal set of SFs $\mathbf{s} = \{s_1, s_2, s_3\}$ and optimal data size $\mathbf{p}_Q = \{p_{Q_1}, p_{Q_2}, p_{Q_3}\}$ which consumes minimum energy and meet the constraints.

**Lemma 2.** *The time complexity of the proposed EESM system is $O(M \log N)$, where $M$ and $N$ are the total number of LNs in the network and the number of LNs connected to the LG, respectively.*

**Proof.** In Algorithm 1, the *for* loops in line 2 and line 3 take $N \times D$ time for collecting the EMS of each appliance for all LNs connected to the LG. Let $\rho$ be the number of iterations for which Procedure 1 calculates the data size for fixed SFs. As line 8 calls Procedure 1 which is inside $N$ " and hence its time complexity is $O(N\rho)$. Similarly, line 10 and line 15 requires $N \times \rho$ time for $N$ times and $N \log N$ times (average time complexity of the $\min$ heap tree), respectively. Therefore, the total time complexity is $O(N\rho + N^2\rho + N^2\rho \log N) = O(N^2\rho \log N)$. Since the total number $M$ of LNs in the network is much higher than $N$, The time complexity of lines 8, 10, and 15 can be approximated as $O(M\rho \log N)$. As Semi-smooth Newton method is used among $N$ LNs, the number of iterations required for the convergence of Procedure 1 is in the order of constant because of the small value of $N$. Therefore the total time complexity, with including $O(N \log N)$ time complexity for line 16, is $O(ND + M \log N + N \log N)$. Since number $D$ of appliances is limited and much smaller than $N$, the time complexity of the proposed EESM system is $O(M \log N)$. □

## 6 Performance Evaluation by Simulation

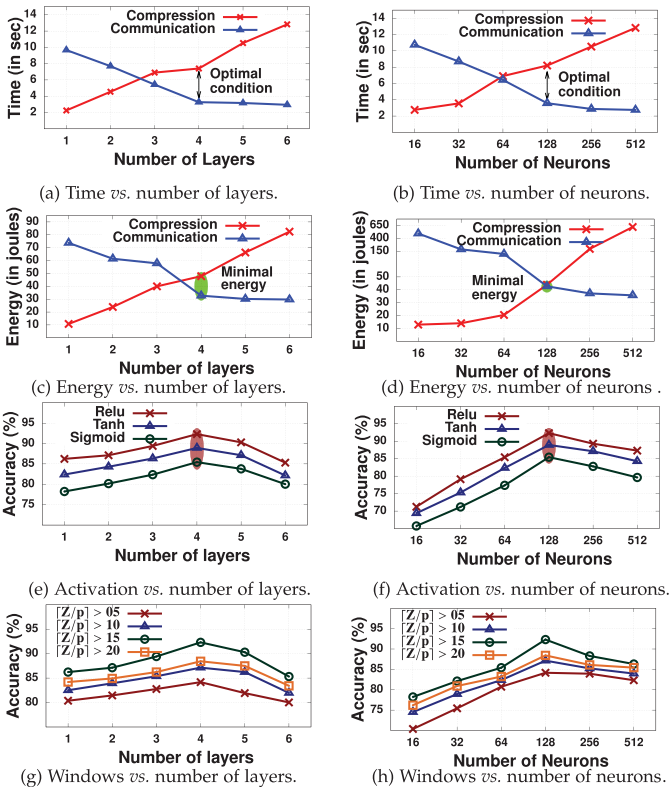
This section validates the performance of the proposed EESM system through simulation experiments and answers the following questions:

- How do the parameters of the compression model affect the compression and communication time, energy consumption, and accuracy of the EESM system? (Section 6.2)
- What is the impact of deployment distribution of LoRa nodes on the time and energy consumption of EESM system? (Section 6.3)
- Does the size of compressed EMS affect the energy consumption and time of EESM system? (Section 6.4)
- What is the impact of the number of appliances and the consumers on the time and energy consumption of EESM system? (Sections 6.5 and 6.6)
- Is the proposed EESM system more effective and energy efficient than the existing works [3], [18]? (Section 6.7)

*Performance Metrics*. In the simulation results, we mainly used time and energy consumption as performance metrics. The time is the sum of the system delay required for compressing the EMS, waiting for accessing the LG, and communicating the compressed EMS to the operators as given in Eq. (21). The energy consumption is the sum of the energy consumed during compression and communication, as given in Eq. (22).

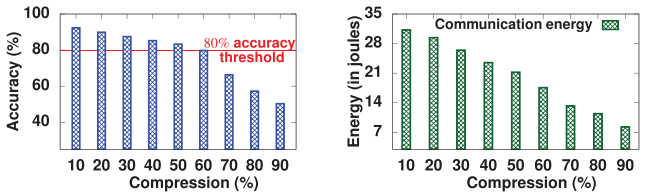
### 6.1 Simulation Setup

The LSTM model in EESM system is implemented in python language using TensorFlow libraries. The protocol for communication of compressed EMS is implemented in Network Simulator-3 which supports multiple channels, SFs, and bi-directional networks with a large number of LNs. Most of the network parameters are obtained from the datasheet of *LoRaWAN Multitech mDot* [24], [29]. The energy consumption of LNs during the idle, transmit, and receive



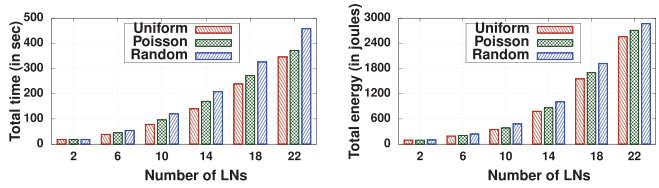

(a) Time *vs.* number of layers. (b) Time *vs.* number of neurons.
(c) Energy *vs.* number of layers. (d) Energy *vs.* number of neurons .
(e) Activation *vs.* number of layers. (f) Activation *vs.* number of neurons.
(g) Windows *vs.* number of layers. (h) Windows *vs.* number of neurons.

Fig. 4. Impact of LSTM parameters on compression and communication time, consumed energy, and accuracy of the system.




(a) Accuracy *vs.* compression. (b) Communication energy *vs.* compression.

Fig. 5. Impact of compression on accuracy and communication energy.




(a) Total time of EESM system. (b) Total energy of EESM system.

Fig. 6. Impact of LN deployment distribution on EESM system.

are $2\mu$J, $32\mu$J, and $11\mu$J, respectively. We assume the battery (energy) capacity of each LN is 2400 mAh. We further consider that all LNs have a duty cycle of 1 percent. We repeat the experiment 100 times by changing the locations of LNs and the average in results. All the results in this work are with 95 percent confidence level though the error bars are not visible in the plots.

### 6.2 Impact of Compression Model Parameters

We study the impact of LSTM model parameters on the time required for compression and communication, energy consumption, and accuracy of the system. The layers of LSTM and the number of neurons are varied from 1 to 6 and 16 to 512, respectively. Figs. 4a and 4b demonstrate that with the increase in the number of LSTM layers, the compression time increases while the communication time decreases. The reduction in communication time is due to the fact that the decreases of EMS size with the inclusion of more layers for compression. The increase in the number of neurons shows a similar pattern, as illustrated in Figs. 4b and 4d. The optimal condition for both communication and compression times is achieved with 4 layers and the neurons are arranged in the decreasing order from 128 to 16 in these four layers. The compression time and communication time at the optimal point are 7.4 secs and 3.9 secs, respectively. Similarly, compression energy and communication energy at the optimal point are estimated as 48 joules and 33 joules, respectively

The number of layers and neurons also affect the accuracy of the system as shown in Figs. 4e and 4f, respectively. Fig. 4e illustrates the impact of different activation functions used at each layer of LSTM. We use Sigmoid, Tanh, and Relu activation function for the comparison. The Relu activation function achieves higher accuracy compared to that of Tanh and sigmoid functions. The best accuracy achieved is around 92 percent with 4 layers of the LSTM model. Figs. 4g and 4h illustrate that the accuracy is the highest when the number of windows $\lceil Z/p \rceil) > 15$.

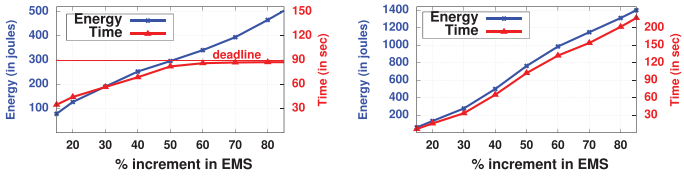
Figs. 5a and 5b illustrate the effect of compression on accuracy and energy consumption during communication. The results demonstrate that as the compression of EMS increases, the accuracy of EMS regeneration at the electricity operator decreases. It is because, during decompression, we have to regenerate original EMS using less number of features. Next, if we assume an accuracy threshold of 80 percent then compression beyond 60 percent is not acceptable even if communication energy decreases, as shown Fig. 5b.

### 6.3 Impact of LN Deployment Distribution

Next, we illustrate the impact of the deployment distribution of LNs in the network on the total time and energy consumption of the EESM system. We have performed experiments on the following three distributions of LN deployment: (i) *uniform* distribution, where $N$ LNs are deployed in the range of 6 SFs such that $\left(\frac{N}{6}\right)$ LNs come in the range of each SF; (ii) *Poisson* distribution, in which the LNs are deployed by using $\frac{e^{-\Gamma}\Gamma^{\kappa}}{\kappa!}$, where $\kappa$ is the number of times a SF gets allocated to an LN and $\Gamma$ is the mean of SF allocation, and (iii) *random* distribution where the LNs are randomly deployed in the network. Figs. 6a and 6b respectively show the impact of the number of LNs in three distributions on the required time and energy in the network. An interesting observation from these results is that uniform distribution of LNs gives better result than the other two distributions.

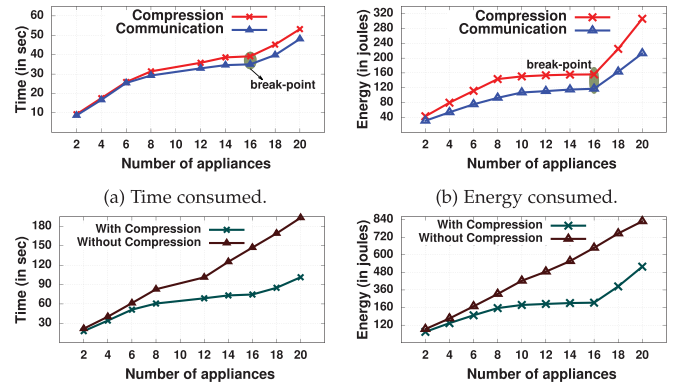
### 6.4 Impact of EMS Size

Next, we study the impact of the size of EMS on the required time and energy. For experimental purpose, we considered 6 appliances EMS as the base size and increment in percentages of the base size. We also considered 3 consumers (LNs) and 90secs as the average deadline of these




(a) Using EESM system (b) Approach without Edge computing

Fig. 7. Impact of EMS size on time and energy of EESM system.




(a) Time consumed. (b) Energy consumed.

(c) Total time of EESM system. (d) Total energy of EESM system.

Fig. 8. Impact of number of appliances attached to an Edge device.

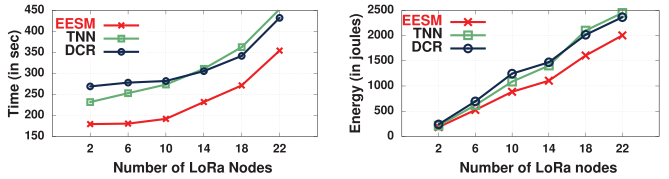
LNs. As the EMS size increases, the compression of the acquired data also increases for completing the task within the specified deadline. Fig. 7 shows that the proposed approach transfers the EMS to the operator within the deadline, whereas without Edge computing scheme the required time and energy consumption is huge. We observe that EESM system takes less time and energy than the approach without Edge computing when the size of EMS is large because other scheme does not compress data and hence requires large communication time.

### 6.5 Impact of the Number of Appliances

Figs. 8a and 8b respectively illustrate the time and energy increase with growing number of appliances. This is because a larger number of appliances generate more EMS, thus increasing the data size for compression and communication. A sharp increase in the compression and communication time is observed when the number of appliances reaches 16 (breakpoint). The energy consumption has also the same breakpoint. Figs. 8c and 8d illustrate the time and energy with and without using the compression model, respectively. As expected, the proposed EESM system compresses the EMS, consuming less energy and time.

### 6.6 Impact of the Number of Consumers

This section illustrates the impact of the number of consumers in the system on required time and energy consumption. In Fig. 9, EESM result illustrates the system time and energy for the different number of consumers in the system. The results show that the increment in the consumers increases the time and energy consumption of the system. It is because more LNs send more data to the LG and requires more time during communication. The energy consumption of the system also increases because the increment in the number of LNs reduces the chance of getting the energy-efficient SFs for transmitting the data.




(a) Time consumption. (b) Energy consumption.

Fig. 9. Comparison of time and energy consumption of the EESM system with existing works.

### 6.7 Comparison With Existing Approaches

This section compares the EESM system with Tree-structure Neural Network model (TNN) [3] and Data Characterization and Reduction scheme (DCR) [18]. We compare the proposed work with [3] because it considers the appliances that are constantly ON, such as fridge and ON/OFF appliances, such as washing machine. The proposed EESM system also works on both types of appliances. We have also considered [18] for comparison as they use Edge computing technique for reducing data at the edge devices.

Some appliances are ON for a very short time period and not frequently such as toaster or water purifier. For successfully recognize such appliances, we collect the smart meter reading of the appliances with high sampling rate. The results in Fig. 9 illustrate the total energy consumption and time of the system. To make the comparison fair, we incorporate similar computation and communication mechanisms as in TNN and DCR. The results show that the approach in [3] consumes more energy than the proposed one. This is because it does not reduce the length of the energy time series. However, the length of the energy time series in the practical scenario is very long. The authors in [3] also claimed that the existing work is suitable for low sampling rate. Fig. 9 also illustrates that the approach in [18] takes more energy and time. This is because they use redundancy algorithm for reducing data which requires more time than our LSTM model for compression. Another reason for outperforming of the proposed work with existing work is that EESM system can compress the EMS based on the available LoRa network parameter. The EEMS system uses full length of the LoRa packets with minimum packets. The existing approaches compress the EMS on a fixed size and each packet of the LoRa network is not used its full length.

## 7 Prototype Experiments

We built a prototype of EESM system using Edge devices and LoRa network for compression and transmitting the EMS from the consumers to the operator. The prototype, as illustrated in Fig. 10, was deployed in six houses in an apartment at IIT (BHU), Varanasi. Each house in EESM system considers as consumer which consists different set of appliances. The houses and appliances are represented by

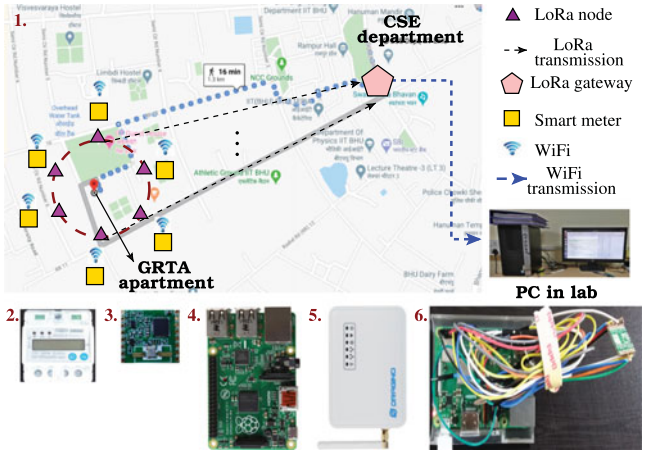



Fig. 10. Smart metering prototype deployed at IIT (BHU), Varanasi. Prototype components: (1) Deployment area, (2) Smart meter, (3) LoRa IC, (4) Raspberry Pi-3, (5) LoRa Gateway, and (6) LoRa node (Pi + LoRa IC).

$\{h_1, \ldots, h_6\}$ and $\{a_1, \ldots, a_{12}\}$, respectively, as shown in Table 2.

### 7.1 Prototype Specification and Overview

Each house has a smart meter and all the appliances of a house are connected with its smart meter for estimating the energy consumption. The smart meter transmits the energy consumption data to the connected Edge device attached with the Lora node (LN). The hardware specification of the prototype is shown in Table 3. The Raspberry Pi 3 board works as an Edge device for compressing the EMS and the LN works as transceivers for communicating the compressed EMS to the LG. The LG, present in the department, forwards the compressed data to the Network Server using the internet connection. Upon receiving data from LG, the NS decompresses the data and forwards to the electricity operator. Here, a Dell Inspiron desktop in the lab is acting as both the NS and electricity operator. A python script is developed for performing the entire operations, including compression, communication, and decompression. All the components used in the prototype are shown in Fig. 10. All the results in this work are with 95 percent confidence level though the error bars are not visible in the plots.

### 7.2 Experimental Results

This section discusses the dataset of smart meter readings and the experimental results.

TABLE 2
List of Appliances in Six Houses (✓ = Yes; ×= No)

| Houses | Appliances | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $a_1$ | $a_2$ | $a_3$ | $a_4$ | $a_5$ | $a_6$ | $a_7$ | $a_8$ | $a_9$ | $a_{10}$ | $a_{11}$ | $a_{12}$ |
| $h_1$ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | ✓ | ✓ |
| $h_2$ | ✓ | × | ✓ | ✓ | ✓ | ✓ | ✓ | × | × | × | ✓ | × |
| $h_3$ | × | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | × | ✓ | × | ✓ |
| $h_4$ | × | ✓ | × | ✓ | × | × | ✓ | ✓ | ✓ | × | ✓ | ✓ |
| $h_5$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $h_6$ | × | × | × | × | ✓ | ✓ | ✓ | × | ✓ | ✓ | ✓ | ✓ |

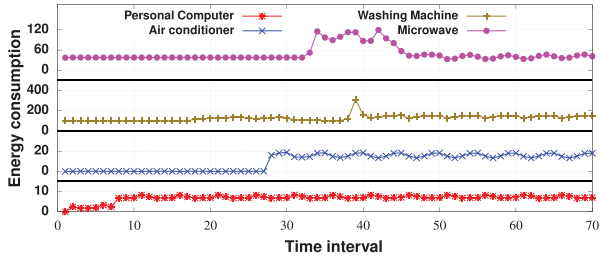



Fig. 11. Energy consumption of appliances in EESM system.

#### 7.2.1 Dataset Creation

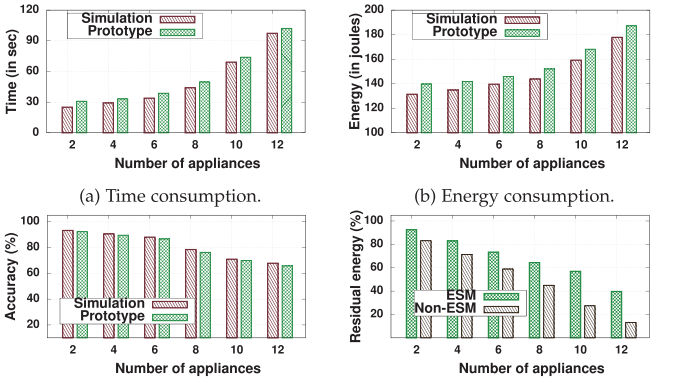
We conduct an experiment to create a dataset using EESM system. Energy consumption of the appliances captured at the interval of 20 seconds and the experiment for data collection performed for a total 240 hours. Fig. 11 illustrates the energy consumption of four appliances. Appliances whose energy consumption is 0, which means that respective appliances are OFF. The energy consumption of the Washing machine is consistently 0 from initial to 30 time instances because this time the Washing machine is OFF. It can also be seen that there is a high peak in the energy consumption of Washing machine from a time interval 33 to 46 and afterwards consumes low energy comparatively. This is because of the washer and dryer ON at the same time for the interval 33 to 46 and then washer is OFF.

#### 7.2.2 Result 1: Number of Appliances

We evaluate the impact of the appliances on the accuracy, time, and energy consumption of the system. The system periodically collects the smart meter readings of $d$ appliances individually whether it is ON or OFF, where $d = \{2, 4, 6, \ldots\}$. The simulation results achieve better performance than the prototype results. This is because the signal strength of the LoRa network varies with the distance and the new obstacles coming in between transmitter and receiver. The performance of the LoRa network also depends on the different network load on the LNs and LG. Network Simulator-3 does not fully consider these challenges and hence performs better than the prototype results. Figs. 12a and 12b demonstrate that the average time and energy consumption of the EESM system increases with the increase in the number of appliances. This is because when we increase the number of appliances, size of EMS also increases as we have considered energy consumption reading of appliances are captured at a defined interval irrespective of their energy consumption and the ON or OFF state. The number of appliances also affect the accuracy of the system and battery life of the LoRa node (operatilithium-ion battery) as illustrated in Figs. 12c and 12d. It shows that the

TABLE 3
Hardware Specification of Prototype

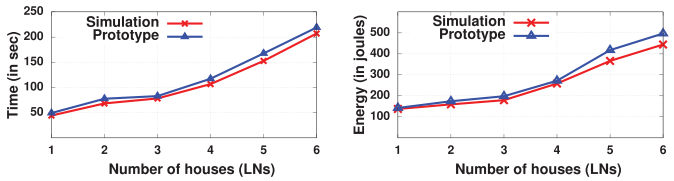
| Device | Specification |
|---|---|
| Appliances | $a_1$ to $a_{12}$ (Connected with smart meter) |
| Smart meter | DDS238-4W single-phase (Attached with Edge device) |
| Edge device | Raspberry Pi-3 (Attached with LN) |
| LoRa Node | RFM95W-868S2 (Communicate to LG) |
| LoRa Gateway | Drigano LG01-SIOT |




(a) Time consumption. (b) Energy consumption.

(c) Accuracy of the system. (d) Residual energy of the LoRa nodes.

Fig. 12. Impact of appliances on time, energy, accuracy, and residual energy.




(a) Time consumption. (b) Energy consumption.

Fig. 13. Impact of number of consumers on time and energy consumption of EESM system.

residual energy of LNs in the EESM system is higher than the non-EESM system.

### 7.2.3 Result 2: Number of Consumers

Similar to the impact of consumers (i.e., houses) as discussed in the simulation experiments (Section 6.6), here we illustrate the impact of the number of consumers on time and energy consumption. The results in Fig. 13 illustrate that both time and energy consumption increase with an increase in the number of consumers and the connected appliances to the consumer. An interesting observation from the result is that energy consumption difference between the simulation and prototype results increases as the number of consumers increases. The reason behind this is the increment in the practical challenges as we increase the number of consumers.

## 8 Conclusion

In this paper, we proposed an Energy Efficient Smart Metering system using edge computing in LoRa network. The system built a deep learning based compression-decompression model and estimated the required energy and time for compression and communication of smart meter readings. The work incorporates Semi-smooth Newton method to find the appropriate compression size and presented an algorithm for selecting the suitable SFs in LoRa network. The experimental and prototype results show that EESM system has achieved high energy efficiency and successfully transfer the EMS within the given time.

The analysis of this work considers the SF parameter of the LoRa network. However, LoRa network also consists of other parameters, such as bandwidth and coding rate. Future directions of research include the extension of the analysis by considering these parameters for improving the energy efficiency. Along with energy efficient smart metering, secure communication of EMS is an essential future direction, which is not covered in this paper.

## Acknowledgments

The authors would sincerely like to thank the anonymous reviewers for their valuable comments and suggestions. The work of Hari Prabhat Gupta and Tanima Dutta was supported in part by SERB grants under award numbers ECR/2016/000406 and ECR/2017/002419, respectively. The work of Sajal K. Das was supported in part by the NSF grants under award numbers CNS-1818942, CCF-1725755, CNS-1545037, and SaTC-2030624.

## References

[1] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019.

[2] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2142–2151, May 2017.

[3] Y. Jia, N. Batra, H. Wang, and K. Whitehouse, "A tree-structured neural network model for household energy breakdown," in *Proc. World Wide Web Conf.*, 2019, pp. 2872–2878.

[4] J. Kolter and T. Jaakkola, "Approximate inference in additive factorial HMMs with application to energy disaggregation," in *Proc. 15th Int. Conf. Artif. Intell. Statist.*, 2012, pp. 1472–1482.

[5] N. Batra, H. Wang, A. Singh, and K. Whitehouse, "Matrix factorisation for scalable energy breakdown," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4467–4473.

[6] J. Kolter, S. Batra, and A. Ng, "Energy disaggregation via discriminative sparse coding," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1153–1161.

[7] T. Sirojan, S. Lu, B. T. Phung, D. Zhang, and E. Ambikairajah, "Sustainable deep learning at grid edge for real-time high impedance fault detection," *IEEE Trans. Sustain. Comput.*, early access, Nov. 07, 2018, doi: 10.1109/TSUSC.2018.2879960.

[8] K. Li, "Computation offloading strategy optimization with multiple heterogeneous servers in mobile edge computing," *IEEE Trans. Sustain. Comput.*, early access, Mar. 12, 2019, doi: 10.1109/TSUSC.2019.2904680.

[9] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, Secondquarter 2020.

[10] W. Zhang, Y. Wen, Y. J. Zhang, F. Liu, and R. Fan, "Mobile cloud computing with voltage scaling and data compression," in *Proc. IEEE 18th Int. Workshop Signal Process. Advances Wireless Commun.*, 2017, pp. 1–5.

[11] X. Xing, J. Song, L. Lin, M. Tian, and Z. Lei, "Development of intelligent information monitoring system in greenhouse based on wireless sensor network," in *Proc. 4th Int. Conf. Inf. Sci. Control Eng.*, 2017, pp. 970–974.

[12] M. Collotta and G. Pau, "An innovative approach for forecasting of energy requirements to improve a smart home management system based on BLE," *IEEE Trans. Green Commun. Netw.*, vol. 1, no. 1, pp. 112–120, Mar. 2017.

[13] M. S. A. Muthanna, M. M. A. Muthanna, A. Khakimov, and A. Muthanna, "Development of intelligent street lighting services model based on LoRa technology," in *Proc. IEEE Conf. Russian Young Researchers Electr. Electron. Eng.*, 2018, pp. 90–93.

[14] S. Kulkarni *et al.*, "Enabling a decentralized smart grid using autonomous edge control devices," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7406–7419, Oct. 2019.

[15] C. Karupongsiri, K. S. Munasinghe, and A. Jamalipour, "A novel random access mechanism for timely reliable communications for smart meters," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3256–3264, Dec. 2017.

[16] J. P. Shanmuga Sundaram, W. Du, and Z. Zhao, "A survey on LoRa networking: Research problems, current solutions, and open issues," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 371–388, Firstquarter 2020.
[17] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in NS-3," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017.
[18] S. Tripathi and S. De, "An efficient data characterization and reduction scheme for smart metering infrastructure," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4300–4308, Oct. 2018.
[19] A. Abuadbba, I. Khalil, and X. Yu, "Gaussian approximation-based lossless compression of smart meter readings," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5047–5056, Sep. 2018.
[20] A. B. Said *et al.*, "A deep learning approach for vital signs compression and energy efficient delivery in mhealth systems," *IEEE Access*, vol. 6, pp. 33 727–33 739, 2018.
[21] N. Sriraam and C. Eswaran, "Performance evaluation of neural network and linear predictors for near-lossless compression of EEG signals," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 87–93, Jan. 2008.
[22] X. Tong, C. Kang, and Q. Xia, "Smart metering load data compression based on load feature identification," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2414–2422, Sep. 2016.
[23] X. Huang and S. Zhou, "Dynamic compression ratio selection for edge inference systems with hard deadlines," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8800–8810, Sep. 2020.
[24] Lorawan MultiTech mDot, 2020. [Online]. Available: https://www.semtech.com/uploads/documents/SX1272_DS_V4.pdf
[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[26] C. Chatfield, *Introduction to Multivariate Analysis*. Abingdon, U.K.: Routledge, 2018.
[27] H. Pan, H. Han, S. Shan, and X. Chen, "Mean-variance loss for deep age estimation from a face," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5285–5294.
[28] J. Kolter and M. Johnson, "REDD: A public data set for energy disaggregation research," *Artif. Intell.*, vol. 25, pp. 1–6, 2011.
[29] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, pp. 1–30, 2017.

**Preti Kumari** received the MTech degree from IIT Patna, India, in 2016. She is currently working toward the PhD degree in computer science and engineering, Indian Institute of Technology (BHU) Varanasi, India. Her research interests include sensor networks, IoT, and mobile computing.

**Rahul Mishra** is currently working toward the PhD degree in computer science and engineering, Indian Institute of Technology (BHU) Varanasi, India. His research interests include sensor networks, wireless ad hoc networks, and algorithms.

**Hari Prabhat Gupta** (Senior Member, IEEE) received the PhD degree in computer science and engineering from IIT Guwahati, India, in 2014. Currently, he is an assistant professor at the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, India. His research interests include IoT, ubiquitous computing, and WSN.

**Tanima Dutta** (Member, IEEE) received the PhD degree in computer science and engineering from IIT Guwahati, India, in 2014. Currently, she is an assistant professor of computer science and engineering, Indian Institute of Technology (BHU) Varanasi, India. Her research interests include multimedia, ubiquitous computing, and machine learning algorithms.

**Sajal K. Das** (Fellow, IEEE) is currently a professor of computer science and Daniel St. Clair Endowed chair at the Missouri University of Science and Technology, Rolla, Missouri. His research interests include wireless sensor networks, mobile and pervasive computing, cyber-physical systems and IoTs, smart environments, cloud and fog computing, cyber security, and social networks. He serves as a founding editor-in-chief of the *Elsevier's Pervasive and Mobile Computing Journal*, and as associate editor of several journals including the *IEEE Transactions of Mobile Computing*, the *IEEE Transactions on Dependable and Secure Computing*, and the *ACM Transactions on Sensor Networks*.