Chapter 6

An energy efficient smart metering system using Edge computing in LoRa network

6.1 Introduction

IoT is envisioned to improve the quality and experience of human living [84,85]. As one of the potential applications of IoT [86,87], 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, such as washing machine, air conditioner, grinder, or room heater. It also induces energy-saving behaviour of EUs, load forecasting, electricity price design, and promotions for reducing energy consumption [88–93]. 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 (EUs) [94,95]. 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 deep learning in Edge Computing can help to reduce the size of EMS at the Edge device near to the consumers [96–98]. Such compressed EMS consumes low energy and requires smaller delay while transferring them from consumers to the operators. Figure 6.1 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 consumes low energy [21–24], long-range communication protocols support a wide coverage with high transmission rates but at the cost of high power consumption [3, 25]. Therefore, the selection of suitable communication protocols is crucial to reduce the energy consumption of Edge devices. The LoRa, 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 [5, 99].

In this chapter, we focus on the smart metering application of IoT, where a consumer has different types of appliances generating EMS data. A large number of consumers simultaneously transferring their EMS for energy consumption of appliances to the operators, requires substantial time and energy. The objective of this chapter is to successfully receive the EMS at the operator in a given time period with minimal energy consumption. 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*

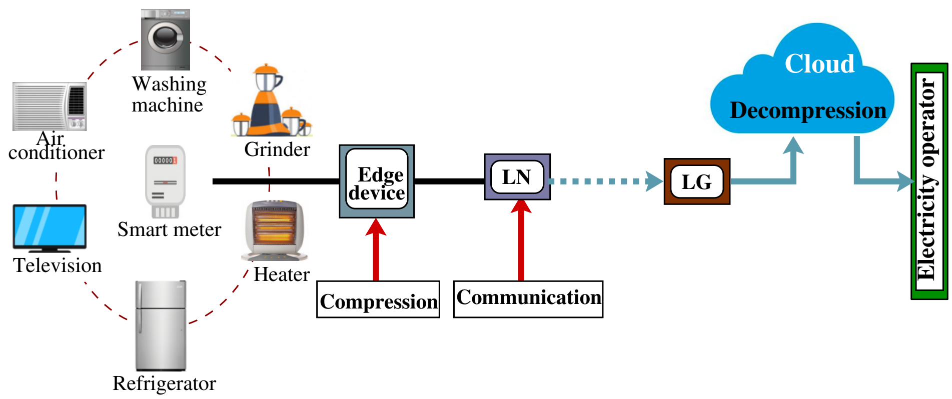


Figure 6.1: Illustration of an example scenario of smart metering.

consumption?

To solve this problem, we present an Energy Efficient Smart Metering (EESM) system that first compresses EMS data at the Edge device, transfers compressed EMS using LoRa network, then decompresses it at the network server, and finally, the decompressed EMS is transferred to the operators. 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 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 LNs in such a way that the energy consumption of all LNs is nearly equal.

6.1.1 Motivation of this work

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

• **Data compression in smart metering:** The authors in [89, 100–102] proposed data compression techniques for smart metering. In [89], authors proposed a nonnegative K-Single Value Decomposition (SVD)-based sparse coding technique for data compression and pattern extraction of electricity consumption data. An adaptive data reduction algorithm is proposed in [100] using compressive sampling technique such that the bandwidth requirement for smart meter data transmission is reduced with minimum loss of information. In [102], smart meter readings are compressed using burrow-wheeler transform and entropy encoding. Lightweight data compression techniques using task offloading for data compression Edge computing are proposed in [97, 98]. The time complexity of the existing data compression techniques [89,100,101] is very high. They require substantial processing power and energy, and hence are not suitable for Edge devices. The work in [97,98] assumed a fixed compression ratio. However, considering a system with a fixed compression ratio hampers efficient utilization of data transmission resources.

• **Data transfer in smart metering:** For transferring EMS data to the operators, the authors in [90–93] proposed techniques to aggregate energy readings of all appliances, transfer the aggregated readings, and finally breakdown the EMS at the operators. In [90] a TreeCNN model is developed for energy breakdown on low-frequency data. Sparse coding based approach [93] has been proposed to breakdown EMS at an interval of one hour. Since the work in [90] considers hourly and days smart meter readings as input and therefore it breakdowns the EMS in days wise, which is not useful for the near real time applications. The basic premise of [92] is that common design and construction patterns for house creating a repeating structure in their energy data. This assumption may not be practically true. Furthermore, the techniques proposed in [90–93] transfer uncompressed EMS to the operators which consumes huge communication energy and delay in large size energy meter readings.

• **Communication protocols in smart metering:** The authors in [21] proposed a ZigBee mesh network 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 [22] which can forecast energy consumption conditions, such as predicting the house energy requirements at different times of the day. In [23], wireless network is analyzed to support services similar to those offered by wired LANs (*e.g.*, Ethernet). Finally, long Term Evolution (LTE) network is considered in [3, 25] for communicating the smart meter reading for high throughput, low latency, and operation in plug and play mode. The long-range communication protocols [3, 25] suffer from colossal power consumption, while short-range communication protocols have limited coverage, and increased hardware and maintenance cost [21–23].

6.1.2 Major contributions

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

• **Processing at Edge Device:** We propose a deep learning 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. The system delay also satisfies the given time period for retrieving of the EMS at the operator. Different from the existing work, we use first and second-order statistics 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 [16] 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 chapter is organized as follows. The next section describes the preliminaries, while Section 6.3 analyzes the delay and energy consumption for compression and communication of EMS data. Section 6.4 presents the EESM system, followed by the experimental and prototype results in Sections 6.5 and 6.6, respectively. The conclusions are offered in Section 6.7.

6.2 Preliminaries

This section defines the system model and terminology used in this chapter.

6.2.1 System model

The proposed energy efficient smart metering (EESM) system assumes each house is equipped with various appliances connected with a smart meter. The smart meter collects ETS of each appliance and then generates an EMS. Next, the Edge device compresses the EMS by using LSTM and sends to the connected LN, as illustrated in Figure 6.2.

The LoRa network consists of multiple LNs connected to a single LG. Let $\mathcal{N} = \{1, 2, \cdots, N\}$ denote the set of LNs, connected to the LG, which use the SF set $\mathcal{F} = \{7, 8, \cdots, 12\}$. The orthogonality of SF facilitates LNs to simultaneously transfer the data on different SFs. The LN further sends the compressed EMS to the LG using the SF $f \in \mathcal{F}$. In this work, we use layer-based Compression-Decompression model where Q is the maximum number of layers of the model for EMS compression. Let Q_n denotes the layer at which LN n compressed data and d_{Q_n} is the compressed data size at layer Q_n. The data size vector $\mathbf{d}_Q$ of N LNs is represented as:

$$\mathbf{d}_Q = [d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}]. \tag{6.1}$$

Let $f_n \in \mathcal{F}$ denote the SF used by LN n to transmit d_{Q_n} data to the LG. The SF vector $\mathbf{f}$ for N LNs is given by

$$\mathbf{f} = [f_1, f_2, \cdots, f_N]. \tag{6.2}$$

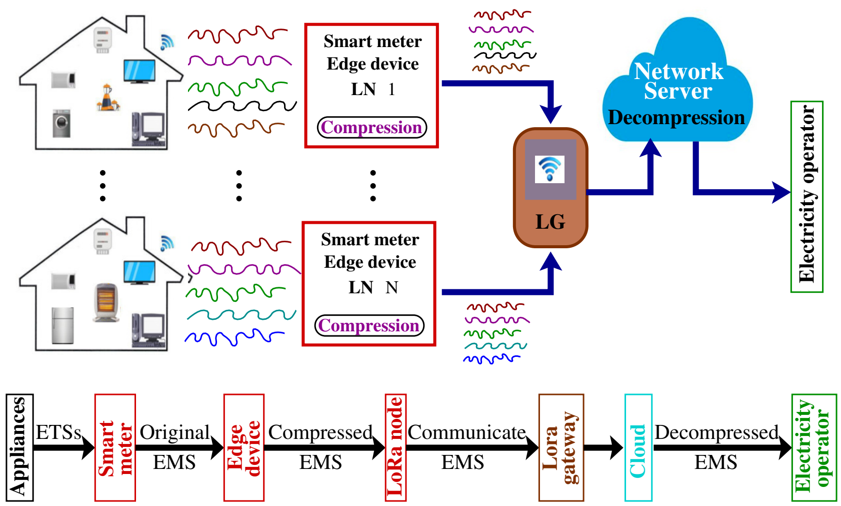


Figure 6.2: An Energy Efficient Smart Metering system (EESM).

6.2.2 Definitions

Let $\mathscr{A} \in \mathbb{R}^{Y \times Z}$ denote an EMS of size $Y \times Z$, where Y and Z are respectively the number of time series generated by the connected Y appliances to an LN, and the total number of events in each time series. Such events are generated in a fixed time interval τ. The Y time series in $\mathscr{A}$ are denoted by $\mathscr{A} = \mathscr{A}_1, \mathscr{A}_2, \cdots, \mathscr{A}_Y$. The Energy Time Series (ETS) $\mathscr{A}_i$ consists zero and non-zero smart meter readings when the appliance i is OFF and ON, respectively, where $1 \leq i \leq Y$. Figure 6.9 illustrates the ETSs of appliances.

Definition 6.1 ***(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 Energy Time Series (ETS). If the reading is generated through multiple appliances, then it is called Energy Multivariate Time Series (EMS).*

Definition 6.2 ***(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 Y_n denote the number of appliances connected to an LN n generate energy time series. Then the accuracy of LN n is defined as:*

$$Acc_n = \frac{1}{Y_n} \sum_{i=1}^{Y_n} \mathbf{1} \left(x_i == \bar{x}_i \right), \tag{6.3}$$

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

$$\mathbf{1}(\mathscr{F}) = \begin{cases} 1, & \textit{if } \mathscr{F} \textit{ is true,} \\ 0, & \textit{otherwise.} \end{cases} \tag{6.4}$$

6.2.3 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 input vector $\mathbf{x}$ to the LSTM is a sequence of events, $\mathbf{x} = \{x_1, x_2, \cdots, x_t\}$, such as meter readings of an appliance at timestamp t. The principle mechanism of the LSTM model incorporates the gated operations performed on a single LSTM unit consisting of four gates and a cell state. The input gate, forget gate, output gate, and input modulation gate are denoted by i, k, o, g, respectively; and let c denote the cell state. Let $\mathbf{W}_{xj}$ and $\mathbf{W}_{hj}$ $\forall j \in \{i, k, o, g, c\}$

denote the weight matrices corresponding to the input vector $\mathbf{x}$ and previous output state $\mathbf{h}_{t-1}$. The LSTM unit operation at time t is summarized as:

$$i_t = \sigma(\mathbf{W}_{xi}x_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_i), \tag{6.5}$$

$$k_t = \sigma(\mathbf{W}_{xk}x_t + \mathbf{W}_{hk}\mathbf{h}_{t-1} + \mathbf{b}_k), \tag{6.6}$$

$$o_t = \sigma(\mathbf{W}_{xo}x_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{6.7}$$

$$g_t = \tanh(\mathbf{W}_{xg}x_t + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_g), \tag{6.8}$$

$$c_t = k_t \otimes c_{t-1} + i_t \otimes g_t, \tag{6.9}$$

$$\mathbf{h}_t = o_t \otimes \tanh(c_t), \tag{6.10}$$

where $\sigma(\cdot)$ is the logistic sigmoid function defined as $\sigma(a) = \frac{1}{1+e^{-a}}$ and the operator $\otimes$ denotes the element-wise product with the gate value. $\mathbf{b}_j$ denotes the bias vector corresponding to the $\mathbf{j}^{th}$ component, $\forall j \in \{i, k, o, g, c\}$. Let $\mathscr{L}(\cdot, \cdot, \cdot)$ denotes a function that combines all the LSTM operation from Equation 6.5 to Equation 6.10. The update of each LSTM cell with parameter ϕ is given by

$$\mathbf{h}_t = \mathscr{L}(\mathbf{h}_{t-1}, x_t, \phi). \tag{6.11}$$

6.3 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.

6.3.1 Compression-Decompression model

Figure 6.3 illustrates the proposed Compression-Decompression model, where the compression maps a higher-dimensional point in the EMS to a lower-dimensional point.

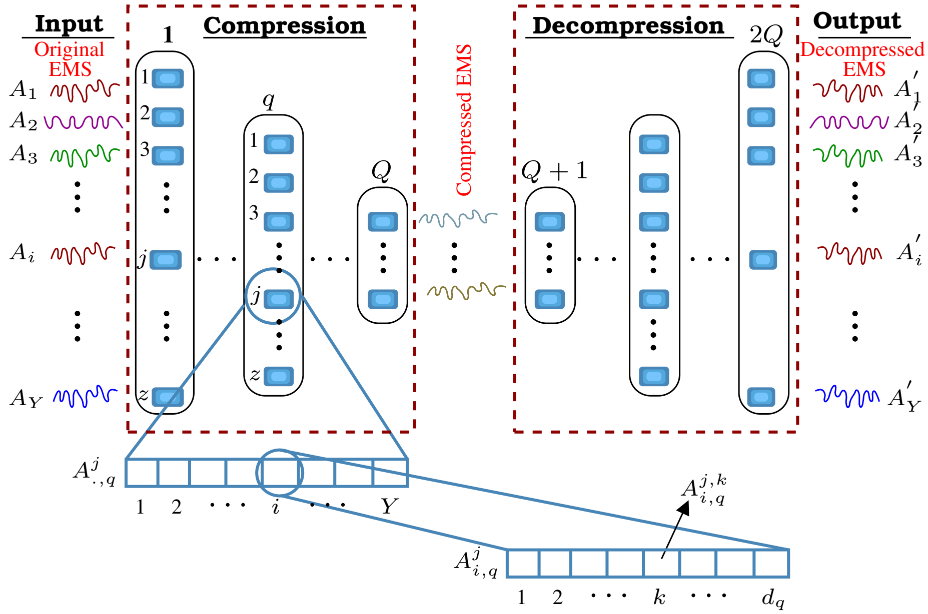


Figure 6.3: Illustration of Compression-Decompression model, where symbol (▣) denotes an LSTM cell.

The model uses LSTM, where at each layer the neuron reduces in a similar pattern. For example, suppose the neurons at layer $q-1$ is 128, then the neurons count reduces to 64 at layer q and 32 at $q+1$. 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.

6.3.1.1 Compression

We use LSTM model with Q layers as Compression (see Figure 6.3) for mapping input sequences $\mathscr{A} \in \mathbb{R}^{Y \times Z}$ to a reduced-dimensional sequences $\mathscr{A}' \in \mathbb{R}^{Y \times Z'}$, where $Z' \leq Z$. The compression divides each time series $A_i \in \mathscr{A}$ into fixed d length windows, where the number of windows $z = \lceil Z/d \rceil$ and $1 \leq i \leq Y$. 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 $\{\{A_1^1, \cdots, A_1^z\}, \{A_2^1, \cdots, A_2^z\}, \cdots, \{A_Y^1, \cdots, A_Y^z\}\}$. The k^{th} event of A_i^j window is denoted by $A_i^{(j,k)}$, where $1 \leq k \leq d$. 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 d_q, where $1 \leq q \leq Q$, the input layer $A^j_{(i,1)}$=A^j_i, and $d_1 = d$. 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 $A^{(j,k)}_{(i,q)}$ as shown in Figure 6.3. To compress $\mathscr{A}$ to $\mathscr{A}'$, we reduce the length of windows with layers of the Compression model, *i.e.*, $d_{q'} < d_q$ and $1 \leq \{q < q'\} \leq Q$. The compression uses the following mathematical expression to generate the events of window $A^j_{(i,q)}$ at LSTM layer q.

$$A^j_{(i,q)} = \mathscr{F}_e(\mathbf{W}_{iq-1} \odot \mathbf{A}_{(i,q-1)} + U^{j-1}_{iq} h^{j-1}_{(i,q)} + b_{iq-1}), \tag{6.12}$$

where $\mathscr{F}_e(.)$ denotes the activation function and $\odot$ the dot product. $\mathbf{W}_{iq-1} \in \mathbb{R}^{d_q \times p_{q-1}}$ 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 $b_{iq-1} \in \mathbb{R}^{d_q}$ denotes the bias. The output of the last layer (i.e., Q^{th} layer) of compression is the compressed events $\mathscr{A}$ given by,

$$\mathscr{A}' = \{\{A^1_{(1,Q)}, \cdots, A^z_{(1,Q)}\}, \cdots, \{A^1_{(Y,Q)}, \cdots, A^z_{(Y,Q)}\}\},$$

where,

$$\begin{aligned} A^j_{(i,Q)} &= \mathscr{F}_e(\mathbf{W}_{iQ-1} \odot \mathbf{A}_{(i,Q-1)} + U^{j-1}_{iQ} h^{j-1}_{(i,Q)} + b_{iQ-1}), \\ &= \mathscr{F}_e(\mathbf{W}_{iQ-1} \odot \cdots \odot f_e(\mathbf{W}_{i1} \odot A^j_{(i,1)} + U^{j-1}_{i2} h^{j-1}_{(i,2)} + b_{i1}) + U^{j-1}_{iQ} h^{j-1}_{(i,Q)} + b_{iQ-1}), \\ &= \{A^{(j,1)}_{(i,Q)}, A^{(j,2)}_{(i,Q)}, \cdots, A^{(j,d_Q)}_{(i,Q)}\}. \end{aligned} \tag{6.13}$$

6.3.1.2 Decompression

The Decompression gradually transfers $\mathscr{A}'$ to produce an estimate $\bar{\mathscr{A}} \in \mathbb{R}^{Y \times Z}$. Figure 6.3 illustrates the block diagram of the proposed Decompression model. It uses LSTM on each compressed window $A^j_{(i,Q)}$ which captures a wide range of temporal dependencies among the events of the window. To initialize the decompression, we assume its initial state $h_1^1 = h_2^1 = \cdots = h_Y^1 = 0$. Next, the decompression uses the fully-connected Q layers to map to each window from $\mathbb{R}^{d_Q}$ back to $\mathbb{R}^d$. Similar to compression, the events window $A^j_{(i,q)}$ is produced at layer q by the following expression.

$$A^j_{(i,q)} = \mathscr{F}_d(\mathbf{W'}_{iq-1} \odot \mathbf{A}_{(i,q-1)} + U'^{j-1}_{iq} h^{j-1}_{(i,q)} + b'_{iq-1}), \tag{6.14}$$

where $\mathscr{F}_d(.)$ denotes the activation function, $\odot$ the dot product. $\mathbf{W'}_{iq-1} \in \mathbb{R}^{d_q \times d_{q-1}}$ 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 $b'_{iq-1} \in \mathbb{R}^{d_q}$, where $Q+1 \leq q \leq 2Q$. The output of the last layer of decompression ($2Q^{th}$ layer) is the events $\mathscr{A}'$, *i.e.,*

$$\bar{\mathscr{A}} = \{\{A^1_{(1,2Q)}, \cdots, A^z_{(1,2Q)}\}, \cdots, \{A^1_{(Y,2Q)}, \cdots, A^z_{(Y,2Q)}\}\},$$

where,

$$A^j_{(i,2Q)} = \{A^{(j,1)}_{(i,2Q)}, A^{(j,2)}_{(i,2Q)}, \cdots, A^{(j,d_{2Q})}_{(i,2Q)}\}. \tag{6.15}$$

6.3.1.3 Compression-Decompression parameters learning

The compression and decompression are non-linear mapping of $\mathscr{A} \to \mathscr{A}'$ and $\mathscr{A}' \to \bar{\mathscr{A}}$, respectively. By using Equations 6.11, 6.13 and 6.15, 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

$$A^j_{(i,Q)} = \mathscr{L}_e(A^j_{(i,1)}, h^{j-1}_{i,1}, \chi), \tag{6.16}$$

$$A^j_{(i,2Q)} = \mathscr{L}_d(A^j_{(i,Q)}, h^{j-1}_{i,Q}, \chi'), \tag{6.17}$$

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 discrepancy error of the Compression-Decompression model of $A_i \in \mathscr{A}$ time series is given by

$$Err_{\bar{x}}(A_i) = \frac{1}{zd} \sum_{j=1}^{z} \sum_{k=1}^{d} \left\| A_{(i,2Q)}^{(j,k)} - A_{(i,1)}^{(j,k)} \right\|_2^2. \tag{6.18}$$

The mean error of the model is first-order statistics, a simplistic way to compare two time series [103]. Next, we consider second-order statistics, *i.e.*, variance information of $\mathscr{A}$ and $\bar{\mathscr{A}}$ for computing the variance discrepancy error. Similar to Equation 6.18, the variance discrepancy error of the Compression-Decompression model of $A_i \in \mathscr{A}$ time series is given by

$$V_j(A_{(i,1)}) = \frac{1}{d-1} \sum_{k=1}^{d} \left\| A_{(i,1)}^{(j,k)} - \frac{1}{d} \sum_{k=1}^{d} A_{(i,1)}^{(j,k)} \right\|_2^2,$$
$$V_j(A_{(i,2Q)}) = \frac{1}{d-1} \sum_{k=1}^{d} \left\| A_{(i,2Q)}^{(j,k)} - \frac{1}{d} \sum_{k=1}^{d} A_{(i,2Q)}^{(j,k)} \right\|_2^2,$$
$$Err_v(A_i) = \frac{1}{z} \sum_{j=1}^{z} \left\| V_j(A_{(i,1)}) - V_j(A_{(i,2Q)}) \right\|_2^2.$$

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

$$\mathcal{L} = \frac{1}{Y} \sum_{i=1}^{Y} (Err_{\bar{x}}(A_i) + \Upsilon Err_v(A_i)),$$
$$\mathcal{I}^{\star} = \arg \min_{I=\{\mathbf{W},\mathbf{W}',U,U'\}} \mathcal{L}. \tag{6.19}$$

By solving Equation 6.19, the Compression-Decompression model learns the weight matrices $(\mathbf{W}, \mathbf{W}', U, U')$ and the corresponding bias vectors $(\mathbf{b}, \mathbf{b}')$. The learned weight matrices and bias vectors minimize the loss or discrepancy between the actual data and decompressed data. In other words, the minimization of loss reciprocates to the maximization of accuracy. Thus, we can infer that the reduction in the discrepancy between the actual value and decompressed value results in accuracy improvement. The accuracy of the Compression-Decompression model can be written as

$$Acc = 1 - \mathcal{L} = 1 - \frac{1}{Y}\sum_{i=1}^{Y}(Err_{\bar{x}}(A_i) + \Upsilon Err_v(A_i)). \tag{6.20}$$

Using Equation 6.20, we can compute the accuracy Acc_n of generated Y_n time series of LN n as defined in Definition 2.

Example 1: Let an EESM system consist of an LN connected to two appliances. The example EMS of the appliances is $\mathscr{A} \in \mathbb{R}^{2\times 10}$={{0,0,0,0,4,2,2,4,0,0,0,0},{0,0,2,3,7,8,3,9,7,3}}, where zero and non-zero values show that the appliances are OFF and ON, respectively. Using the compression model, the compressed EMS is $\mathscr{A}' \in \mathbb{R}^{2\times 4}$ and the decompressed EMS is $\bar{\mathscr{A}} \in \mathbb{R}^{2\times 10}$={{0,0,0,0,4,3,2,4,0,0,0,0},{{0,0,2,3,7,8,3,9,7,3}}. This example shows that the EESM system compressed 20 smart meter readings to 8 and therefore it needs 60% less communication energy and delay. It also shows that $\bar{\mathscr{A}}$ has one smart meter reading error.

6.3.2 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. The EESM system incorporates the compression and communication delay for the EMS.

The energy consumption of the EESM system is computed as the sum of the energy consumed due to the compression and communication of EMS.

6.3.2.1 Compression

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

Lemma 6.1 *Given a LoRa network with Q_n layers of an LN n, where d_q is the length of the window at layer q. Let $\mathscr{A} \in \mathbb{R}^{Y_n \times Z}$ be an Energy Multivariate Time Series of length Z with Y_n dimensions, generated by appliances with a sampling rate of κ_n. Then the predicted delay for compressing the data is given by*

$$T_n^{comp} = \sum_{i=1}^{Y_n} \sum_{q=1}^{Q_n - 1} \frac{\kappa_n^i \tau l_{q+1}(2d_q + \eta)}{l}. \tag{6.21}$$

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

$$E_n^{comp} = \sum_{i=1}^{Y_n} \sum_{q=1}^{Q_n - 1} \frac{\kappa_n^i \tau l_{q+1}(2d_q + \eta)}{l} E_c. \tag{6.22}$$

Proof: To estimate the runtime, we count the required FLOating Point operations (FLOPs). Equation 6.21 shows that an event window of length l requires $d_q \times (2d_{q-1})$ 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 $\sum_{q=1}^{Q_n-1} 2d_q d_{q+1}$. Such compression also uses $\sum_{q=1}^{Q_n-1} d_{q+1}$ activation functions per window, requiring $\sum_{q=1}^{Q_n-1} d_{q+1}\eta$ FLOPs, where η is the distinct operation in the activation function. The total runtime of compression per window is $\sum_{q=1}^{Q_n-1} 2d_q l_{q+1} + \sum_{q=1}^{Q_n-1} l_{q+1}\eta = \sum_{q=1}^{Q_n-1} l_{q+1}(2d_q + \eta)$. The i^{th} ETS of an LN n with κ_n^i sampling rate consists of $\kappa_n^i \tau / l$ windows during τ time duration, where $1 \leq i \leq Y_n$. The total FLOPs for $\mathscr{A}$ is the sum of the required

FLOPs for all Y_n time series and given in Equation 6.21. 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 Equation 6.22. □

6.3.2.2 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 f_n and bandwidth W for transmitting data to the LG with coding rate c. The transmission rate from LN n to LG can be obtained from [45], which is given as

$$r_n^f = W \times \frac{f_n}{2^{f_n}} \times \frac{4}{4+c}. \tag{6.23}$$

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 d_{Q_n} data of LN n are given by

$$T_n^{comm} = \frac{d_{Q_n}}{r_n^f}, \tag{6.24}$$

$$E_n^{comm} = \frac{d_{Q_n}}{r_n^f} E_o. \tag{6.25}$$

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 f_n for transmitting data at the LG. Then in the worst case, LN n has to wait $T_n^{wait} = \sum_{\substack{i=1, \\ f_n==f_i}}^{N} T_i^{comm}$ time for getting its turn. $T_n^{wait} = 0$ for LN n, if selected SF f_n is free. Therefore the total delay of LN n is the sum of compression, waiting, and communication delay, given as

$$\mathbb{T}_n = T_n^{comp} + T_n^{wait} + T_n^{comm}. \tag{6.26}$$

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 d_{Q_n} and SF f_n. It is calculated from Equations 6.22 and 6.25. That is,

$$\xi_n = E_n^{comp} + E_n^{comm}. \tag{6.27}$$

6.4 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.

6.4.1 Problem formulation

Let LN $n \in \mathcal{N}$ uses SF f_n for transmitting the compressed d_{Q_n} data to the LG, where $1 \leq n \leq N$, $1 \leq Q_n \leq Q$, and $f_n \in \mathcal{F}$. Let us define

EESM Problem:

$$\min_{\mathbf{d}_Q,\mathbf{f}} \quad \sum_{n=1}^{N} \xi_n, \tag{6.28a}$$

$$\text{s.t.} \quad C1:\ \xi_n \leq E_1^{th}, \tag{6.28b}$$

$$C2:\ ||\xi_n - \xi_j|| \leq E_2^{th}, \tag{6.28c}$$

$$C3:\ Acc_n \geq \mathbb{A}^{th}, \tag{6.28d}$$

$$C4:\ d_n^{min} \leq d_{Q_n} \leq d_1, \tag{6.28e}$$

$$C5:\ \mathbb{T}_n \leq \mathbb{D}_n^{th}, \tag{6.28f}$$

where ξ_n can be obtained from Equation 6.27, $\forall n \in \mathcal{N}, j \in \mathcal{N}/n$, and $\{f_n, f_j\} \in \mathcal{F}$.

Constants: The input to the **EESM Problem** includes the set of LNs ($\mathcal{N}$), their connected appliances sampling rate (κ_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 ξ_n of an LN n comprises E_n^{comp} and E_n^{comm} which depends on the datasize d_{Q_n} and SF f_n. Therefore, the total energy in the LoRa network depends on vector $\mathbf{d}_Q = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$ and SF vector $\mathbf{f} = \{f_1, f_2, \cdots, f_N\}$ 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 $\mathbf{d}_Q$ on SF $\mathbf{f}$ 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 d_n^{min} and d_1. Constraint $C5$ ensures that each LN scheduled on an SF transmits its data within a specified deadline.

6.4.2 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.,* $\mathbf{f} = \{f_1, f_2, \cdots, f_N\}$ to transmit data to the LG of optimal size, *i.e.,* $\mathbf{d}_Q = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$.

6.4.2.1 Step 1: Solve the EESM problem for fixed SFs

We convert the problem into a Quadratic Programming Problem (QPP) [104], define the Karush Kuhn Tucker (KKT) conditions, and use Semi-Smooth Newton method for searching the optimal value of $\mathbf{d}_Q = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$ for fixed SFs.

• *Convert **EESM problem** into QPP:* For this purpose, we convert the problem in Equation 6.28 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 d_{q+1}, *i.e.,* $l_{q+1} = d_{q+1}$. The energy consumption of LN n for every time series data is given as follows.

$$\xi_n = \sum_{q=1}^{Q_n-1} d_{q+1}(a_1 d_q + a_2) + a_3 d_{Q_n}, \tag{6.29}$$

where $a_1 = \frac{2Y_n \kappa_n \tau E_c}{l}$, $a_2 = \frac{\eta Y_n \kappa_n \tau E_c}{l}$, and $a_3 = \frac{E_o}{r_n^f}$. Q_n is the number of layers needed for compressing d_{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.,* $d_q = d_{Q_n}(x')^{Q_n - q}$, where $x' = (1 + \frac{x}{100})$ and $1 \leq q \leq Q_n$. So, Equation 6.29 can be rewritten as

$$\xi_n = a_1 d_{Q_n}^2 (x' + x'^3 + \cdots + x'^{(2Q-3)}) + a_2 d_{Q_n} (1 + x' + \cdots + x'^{(Q-2)}) + a_3 d_{Q_n},$$
$$= \underbrace{a_1 \frac{1-(x'^2)^{2Q-3}}{1-x'^2}}_{u_n} d_{Q_n}^2 + \underbrace{\left(a_2 \frac{1-(x')^{Q-2}}{1-x'} + a_3 \right)}_{v_n} d_{Q_n}. \tag{6.30}$$

• *KKT Conditions for the* ***EESM Problem****:* The objective function of the problem (Equation 6.28) is a convex optimization problem with inequality and equality constraints. 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.*,

$$\xi_n = E_1^{th} - \gamma_{1n}, \qquad \forall n \in \mathcal{N}, f_n \in \mathcal{F}, \tag{6.31}$$

$$||\xi_n - \xi_j|| = E_2^{th} - \gamma_{2n}, \qquad \forall \{n, j\} \in \mathcal{N}, \{f_n, f_j\} \in \mathcal{F}, n \neq j, \tag{6.32}$$

$$Acc_n = \mathbb{A}^{th} + \gamma_{3n}, \qquad \forall n \in \mathcal{N}, \tag{6.33}$$

$$d_{Q_n} = d_n^{min} + \gamma_{4n}, \qquad \forall n \in \mathcal{N}, \tag{6.34}$$

$$d_{Q_n} = d_1 - \gamma_{5n}, \qquad \forall n \in \mathcal{N}, \tag{6.35}$$

where $\{\gamma_{in}\}_{i=1}^5$ 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

$$\min \; \mathbf{u}^T \mathbf{d}_Q^2 + \mathbf{v}^T \mathbf{d}_Q$$
$$s.t. \; \mathbf{A}\mathbf{d}_Q = \mathbf{B}, \tag{6.36}$$

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

$$\mathbf{A} = \begin{Bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \mathbf{A}_3 \\ \mathbf{A}_4 \\ \mathbf{A}_5 \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}^T\mathbf{d}_Q + \mathbf{v}^T \\ \mathbf{u}^T\mathbf{d}_Q + \mathbf{v}^T - \xi_j/\mathbf{d}_Q \\ \mathbf{Acc}/\mathbf{d}_Q \\ 1 \\ 1 \end{Bmatrix}, \text{and} \quad \boldsymbol{B} = \begin{Bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \\ \mathbf{b}_4 \\ \mathbf{b}_5 \end{Bmatrix} = \begin{Bmatrix} E_1^{th} - \gamma_{1n} \\ E_2^{th} - \gamma_{2n} \\ \mathbb{A}^{th} + \gamma_{3n} \\ d^{min} + \gamma_{4n} \\ d_1 - \gamma_{5n} \end{Bmatrix}.$$

Next, we introduce Lagrangian multipliers $\boldsymbol{\alpha} = \{\alpha_n\}_{n=1}^N$, $\boldsymbol{\beta} = \{\beta_n\}_{n=1}^N$, $\boldsymbol{\lambda} = \{\lambda_n\}_{n=1}^N$, $\boldsymbol{\Gamma} = \{\Gamma_n\}_{n=1}^N$, and $\boldsymbol{\zeta} = \{\zeta_n\}_{n=1}^N$ one for each constraint. The Lagrangian form of Equation 6.36 is as follows.

$$\begin{aligned}\mathcal{L}(\mathbf{d}_Q, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\Gamma}, \boldsymbol{\zeta}) =& \mathbf{u}^T\mathbf{d}_Q^2 + \mathbf{v}^T\mathbf{d}_Q - \sum_{n=1}^{N} \alpha_n(\mathbf{A}_1\mathbf{d}_Q - \boldsymbol{b}_1) - \sum_{n=1}^{N} \sum_{j=1, j\neq n}^{N} \beta_{nj}(\mathbf{A}_2\mathbf{d}_Q - \boldsymbol{b}_2) \\ &+ \sum_{n=1}^{N} \lambda_n(\mathbf{A}_3\mathbf{d}_Q - \boldsymbol{b}_3) + \sum_{n=1}^{N} \Gamma_n(\mathbf{A}_4\mathbf{d}_Q - \boldsymbol{b}_4) - \sum_{n=1}^{N} \zeta_n(\mathbf{A}_5\mathbf{d}_Q - \boldsymbol{b}_5).\end{aligned} \tag{6.37}$$

By using the gradient method, we can find the optimal value of d_{Q_n} for fixed SF. Differentiating Equation 6.37 *w.r.t.*, d_{Q_n}, we obtain

$$d_{Q_n} = \frac{\lambda_n Acc_n + \Gamma_n - \zeta_n - v_n\delta_n}{2u_n\delta_n}, \tag{6.38}$$

where $\delta_n = \alpha_n + \beta_n - 1$. The Hessian of the function can be calculated by differentiating the gradient, *i.e.*, $\nabla_{d_{Q_n}}(\lambda_n Acc_n + \Gamma_n - \zeta_n - \delta_n(2u_n d_{Q_n} + v_n)) = -2u_n\delta_n$. Let assume the slack variable $\omega_n = 2\delta_n u_n d_{Q_n}$. Finally, we use Semi-Smooth Newton method [105] for optimal solution of the **EESM Problem**, as shown in Procedure 6.1.

Procedure 6.1: Semi-Smooth Newton method for solving EESM Problem.

Input: $\mathbf{A}$, $\mathbf{B}$, $\mathbf{u}$, $\mathbf{v}$;
Output: Optimal data size $\boldsymbol{d}_Q = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$ for fixed SFs ;

Initialization: Data size vector $\boldsymbol{d}_Q^0 \leq d_1$, slack variable $\omega_n^0 > 0$, iteration $k = 0$, terminating constant ϵ;

do

```
// Compressed data size
```
$(\boldsymbol{d}_Q)^k = diag((d_{Q_1})^k, (d_{Q_2})^k, \cdots, (d_{Q_N})^k)$;
```
// Using gradient and Hessian and Equation 6.37
```
$dist = \frac{\nabla^2_{d_{Q_n}}(\mathcal{L}(\mathbf{d}_Q, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\Gamma}, \boldsymbol{\zeta}))}{\nabla_{d_{Q_n}}(\mathcal{L}(\mathbf{d}_Q, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\Gamma}, \boldsymbol{\zeta}))}$;
```
// Update compressed data size
```
$(\boldsymbol{d}_Q)^{k+1} = (\boldsymbol{d}_Q)^k + dist$;
```
// Update slack variable
```
$\omega_n^{k+1} = \lambda_n Acc_n + \Gamma_n - \zeta_n - \delta_n v_n$;
```
// Next iteration
```
$k = k + 1$;
```
// Optimality test
```

while $((\omega_n^k)^T (\boldsymbol{d}_Q)^k < \epsilon)$;

6.4.2.2 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.,* $\mathbf{f} = \{f_1, f_2, \cdots, f_N\}$ to transmit to the LG data of optimal size, *i.e.,* $\mathbf{d}_Q = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$. Algorithm 6.1 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 6.1 for all allocated SFs and LNs in the network. The SF allocation can be done based on the distance from the LG. An SF $f_n \in \mathcal{F}$ is allocated to an LN $n \in \mathcal{N}$ for transmitting its data if n lies in the range $[0, ran_{f_n}]$ of f_n, where ran_{f_n} is the maximum distance at which the LN n can transmit on SF f_n. For example, if an LN lies in the range of SF_8 with respect to the covered distance from the LG then $SF_8, SF_9, SF_{10}, SF_{11}$, and SF_{12} are the allocated SFs of the LN. 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 $\mathbf{d}_Q$.

Algorithm 6.1: Energy Efficient Smart Metering.

Input: Time series of smart meters connected with the LNs;
Output: $\boldsymbol{d_Q} = \{d_{Q_1}, d_{Q_2}, \cdots, d_{Q_N}\}$, $\boldsymbol{f} = \{f_1, f_2, \cdots, f_N\}$;

Initialization: $minHeap\ H = \text{NULL}$, $Flag = 0$;
for $n \leftarrow 1$ *to* N **do**
    for $i \leftarrow 1$ *to* Y_n **do**
        Collect data from appliance i for τ time interval for all LNs;
for $f_1 \leftarrow SF_7$ **to** SF_{12} **do**
    for $f_2 \leftarrow SF_7$ **to** SF_{12} **do**
        ⋮
        for $f_N \leftarrow SF_7$ **to** SF_{12} **do**
            Call **Procedure 6.1**;
            `// Deadline constraint`
            for $n \leftarrow 1$ **to** N **do**
                $\mathbb{T}_n = T_n^{Comp} + T_n^{wait} + T_n^{Comm}$;
                if $\mathbb{T}_n \geq \mathbb{D}_n^{th}$ **then**
                    $Flag \leftarrow 1$;
                    break;
            if $Flag == 0$ **then**
                Insert $minHeap(H, \boldsymbol{p_Q}, \boldsymbol{s})$;
$Extract\_Min(H, \boldsymbol{d_Q}, \boldsymbol{f})$;
return $(\boldsymbol{d_Q}, \boldsymbol{f})$;

Example 2: Continue Example 1 with three LNs, *i.e.,* $N = 3$ connected to 2, 4, and 5 appliances (*i.e.,* Y_1=2, Y_2=4, and Y_3=5). Let assume that $\mathbf{d}_Q^0 = \{d_{Q_1}^0, d_{Q_2}^0, d_{Q_3}^0\}$ is the vector of initial data size of all three LNs. $d_{Q_1}^0$ vector is shown in Example 1. Procedure 6.1 calls Equation 6.28 which leads to call Equations 6.22 and 6.25 for calculating the energy consumption for compression and communication, respectively. For estimating the compression energy, Equation 6.22 needs the number of layers which is calculated based on the initial data size. Next, Equation 6.25 calculates the communication energy on a given SF. Procedure 6.1 is repeated for all three LNs and calculates the optimal data size which consumes minimum energy consumption for the given SFs.

Optimal data of first LN is shown in Example 1. This optimal data size is used at Line number 8 in Algorithm 6.1. Algorithm 6.1 checks all the combination of SFs among the LNs to calculate the optimal set of SFs $\mathbf{f} = \{f_1, f_2, f_3\}$ and optimal data size $\mathbf{d}_Q = \{d_{Q_1}, d_{Q_2}, d_{Q_3}\}$ which consumes minimum energy and meet the constraints.

Lemma 6.2 *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 6.1, the **for** loops in line 2 and line 3 take $N \times Y$ time for collecting the EMS of each appliance for all LNs connected to the LG. Let ρ be the number of iterations for which Procedure 6.1 calculates the data size for fixed SFs. As line 8 calls Procedure 6.1 which is inside N "for loops" which run at most 6 times for each. This is because the number of possible SFs for each LN is at most 6, 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 $O(M\rho \log N)$, where $M \approx N^2$. As Semi-smooth Newton method is used among N LNs, the number of iterations required for the convergence of Procedure 6.1 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 (average time complexity of the min heap tree), is $O(ND + M \log N + N \log N)$. Since number Y of appliances is limited and much smaller than N, the time complexity of the proposed EESM system is $O(M \log N)$.

□

6.5 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?

• Does the size of compressed EMS affect the energy consumption and time of EESM system?

• What is the impact of the number of appliances and the consumers on the time and energy consumption of EESM system?

• Is the proposed EESM system more effective and energy efficient than the existing works [90, 100].

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 Equation 6.26. The energy consumption is the sum of the energy consumed during compression and communication as given in Equation 6.27.

6.5.1 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. We have modified the traffic control model (lorawan-tracing-example.cc). The network is set up by randomly deploying the LNs in a disc-shaped field of radius 10Km. Most of the network parameters are obtained from the datasheet of *LoRaWAN Multitech mDot* [48, 49]. The LNs and LG are configured to use 125 KHz bandwidth with 868.1 MHz channel frequency. The energy consumption of LNs during the idle, transmit, and receive are 2μJ, 32μJ, and 11μJ, 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% and follow perfect orthogonality in SFs. 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% confidence level though the error bars are not visible in the plots.

6.5.2 Impact of compression model parameters

We study the impact of LSTM model parameters, *i.e.*, number of layers and neurons, 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. Part (a) and part (b) of Figure 6.4 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. However, the addition of layers leads to increase processing time and energy consumption. The increase in the number of neurons shows a similar pattern, as illustrated in Part (c) and part (d) of Figure 6.4. 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

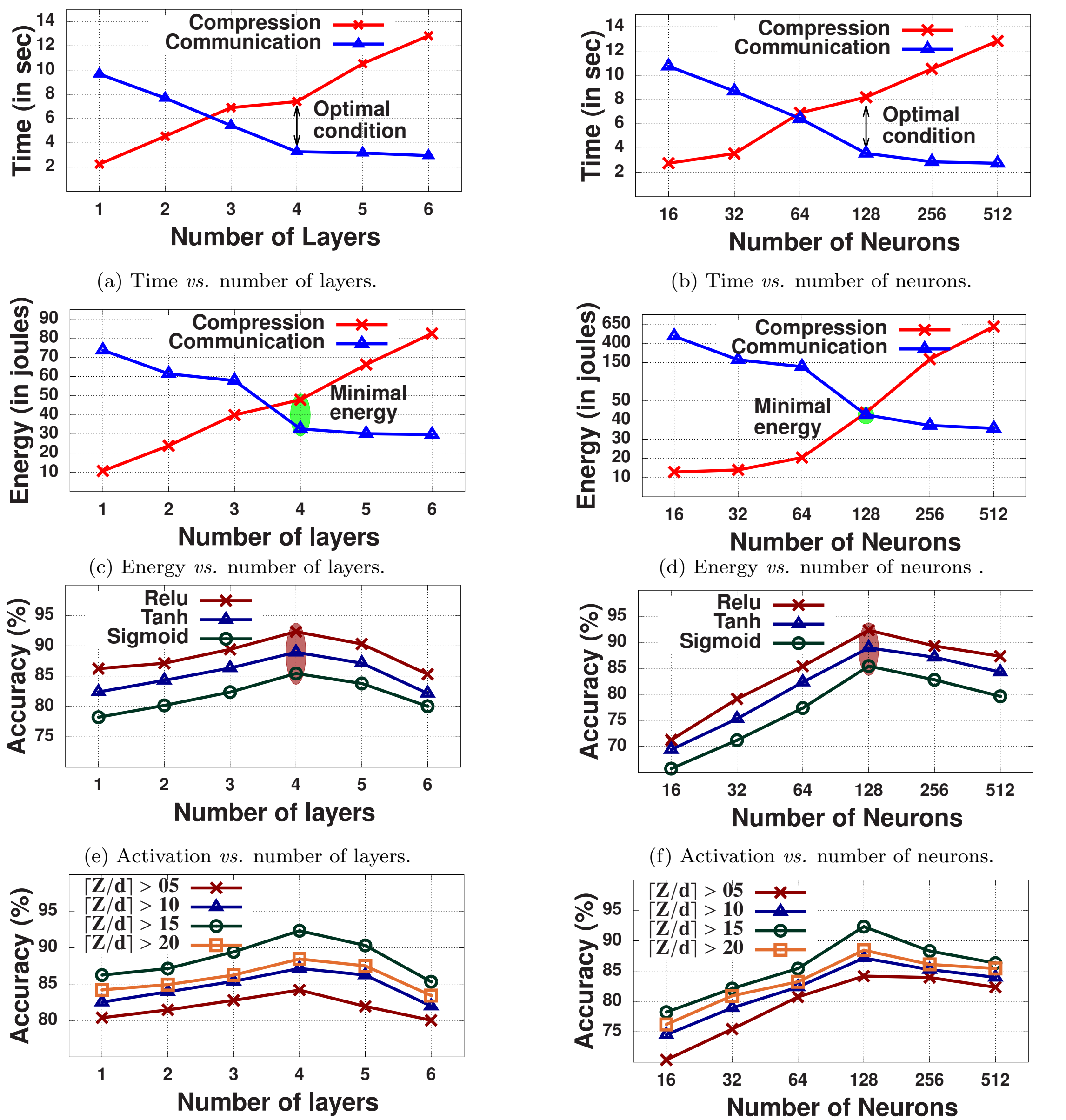
The number of layers and neurons also affect the accuracy of the system. Part (e) of Figure 6.4 illustrates the impact of different activation functions used at each layer of LSTM. We use Sigmoid, Tanh, and Relu activation function for the comparison [106]. The Relu activation function achieves higher accuracy compared to that of Tanh and sigmoid functions. The best accuracy achieved is around 92% with 4 layers of the LSTM model. The number of neurons influences the accuracy of the LoRa network when different activation functions are used, as illustrated in part (f) of Figure 6.4. The windows for dividing a time-series also play a vital role in determining the accuracy of the LoRa network for smart metering. Part (g) and part (h) of Figure 6.4 illustrate that the accuracy is the highest when the number of windows $\lceil Z/d \rceil) > 15$.

6.5.3 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 LNs. As the EMS size increases, the compression of the acquired data also increases for completing the task within the specified deadline. Figure 6.5 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. EESM system compressed EMS to the optimal compressed data size so that EMS can reach to the operator within the specified deadline.

6.5.4 Impact of the number of appliances

As discussed earlier, each of the appliances generates energy time series that are attached to an Edge device used for compressing the EMS. Part (a) and part (b) of Figure 6.6 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



(a) Time *vs.* number of layers.

(b) Time *vs.* number of neurons.

(c) Energy *vs.* number of layers.

(d) Energy *vs.* number of neurons .

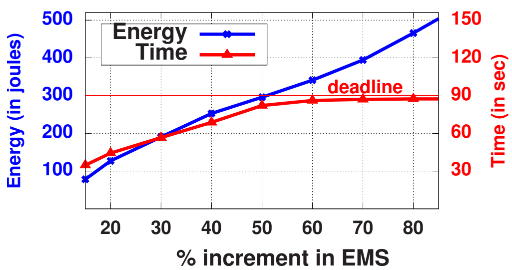
(e) Activation *vs.* number of layers.

(f) Activation *vs.* number of neurons.

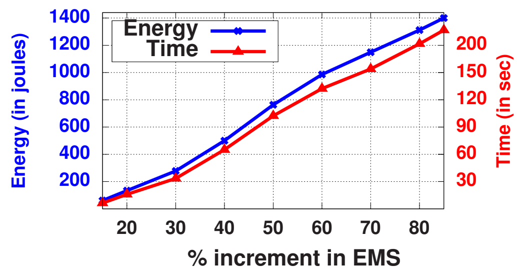
(g) Windows *vs.* number of layers.

(h) Windows *vs.* number of neurons .

Figure 6.4: Impact of LSTM parameters on compression and communication time, consumed energy, and accuracy of the system.



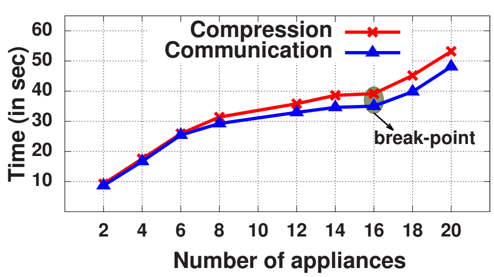
(a) Using EESM system



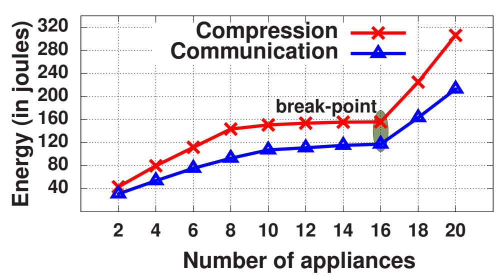
(b) Approach without Edge computing

Figure 6.5: Impact of EMS size on time and energy of EESM system.

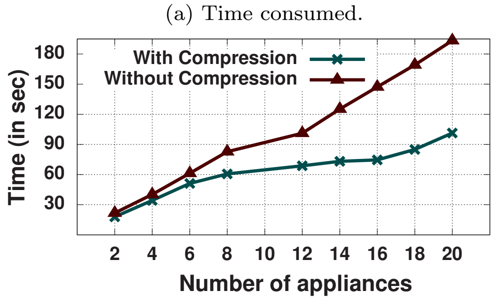
the compression and communication time is observed when the number of appliances reaches 16 (breakpoint). At the breakpoint, the EESM system takes more time than the given deadline. Therefore, for transferring the EMS to the operator within the deadline, more reduction of data is needed which requires high energy consumption for compression. Hence, energy consumption has also the same breakpoint. Part (c) and part (d) of Figure 6.6 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. These results also show similar breakpoint.



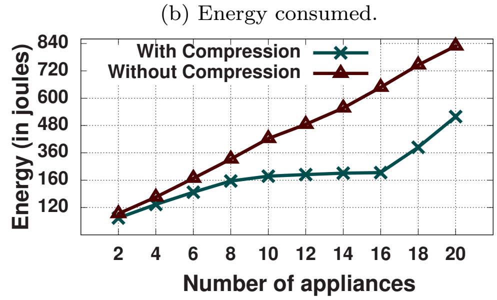
(a) Time consumed.



(b) Energy consumed.



(c) Total time of EESM system.



(d) Total energy of EESM system.

Figure 6.6: Impact of number of appliances attached to an Edge device.

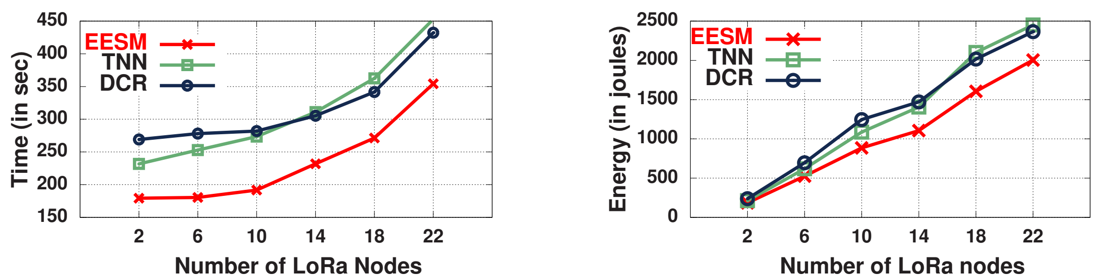
6.5.5 Impact of the number of consumers

Each consumer in EESM system has an Edge device which is embedded with a smart meter and a LN. This section illustrates the impact of the number of consumers in the system on required time and energy consumption. In Figure 6.7, 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.

6.5.6 Comparison with existing approaches

This section compares the EESM system with Tree-structure Neural Network model (TNN) [90] and Data Characterization and Reduction scheme (DCR) [100]. We compare the proposed work with [90] 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 [100] 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 recognise such appliances, we collect the smart meter reading of the appliances with high sampling rate. However, high sampling rate creates long energy time series. The results in Figure 6.7 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 [90] consumes more energy than the proposed one. This is because it doesn't reduces the length of the energy time series. However, the length of the energy time series in the practical scenario is very long. The long time series requires more fixed size communication messages on LoRa network and therefore increases the energy consumption and delay. The authors in [90] also claimed that the existing work is suitable for low sampling rate. Figure 6.7 also illustrates that the approach in [100] 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. The 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. Such partially used packets incense the energy consumption and delay of the system.



(a) Time consumption.

(b) Energy consumption.

Figure 6.7: Comparison of time and energy consumption of the EESM system with existing works.

6.6 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 Figure 6.8, 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 $\{h_1, \cdots, h_6\}$ and

$\{a_1, \cdots, a_{12}\}$, respectively, as shown in Table 6.1.

Table 6.1: List of appliances in six houses (✓ = Yes; ✗= No), where personal computer (a_1), air conditioner (a_2), washing machine (a_3), microwave (a_4), refrigerator (a_5), heater (a_6), television (a_7), mixer grinder (a_8), induction (a_9), toaster (a_{10}), geyser (a_{11}), and water pump (a_{12}).

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	✗	✗	✗	✗	✓	✓	✓	✗	✓	✓	✓	✓

6.6.1 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 LN. The hardware specification of the prototype is shown in Table 6.2. 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 (NS) 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. To successfully run the python script, we install mini-conda on Raspberry Pi and install all necessary Keras packages. All the components used in the prototype are shown in Figure 6.8. All the results in this work are with 95% confidence level though the error bars are not visible in the plots.

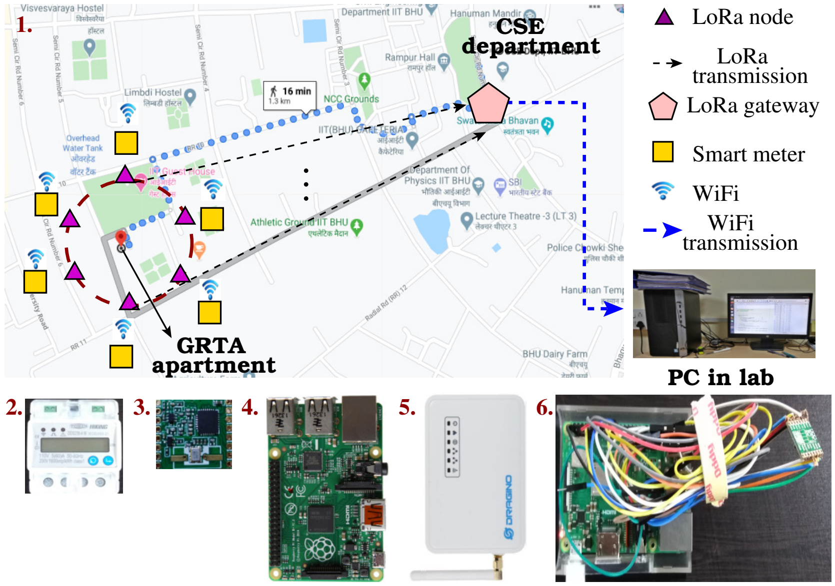


Figure 6.8: Smart metering prototype deployed at IIT (BHU), Varanasi. Prototype components: (1) Deployment area, (2) Smart meter, (3) LoRa IC, (4) Raspberry Pi-3, (5) LoRa Gateway, and (6) LoRa node (Pi + LoRa IC).

Table 6.2: Hardware specification of prototype.

Device	Specification
Appliances	a_1 to a_{12} as Shown in Table 6.1 (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

6.6.2 Experimental results

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

6.6.2.1 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. Figure 6.9 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.

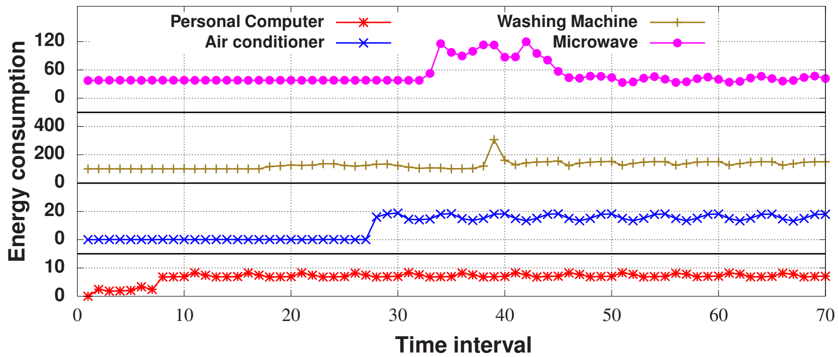


Figure 6.9: Energy consumption of appliances in EESM system.

6.6.2.2 Result 1: Number of appliances

We evaluate the impact of the appliances on the accuracy, time, and energy consumption of the system. We compare the simulation and prototype results for varying number of appliances. The system periodically collects the smart meter readings of y appliances individually whether it is ON or OFF, where $y = \{2, 4, 6, \cdots\}$. 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. Part (a) and part (b) of Figure 6.10 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 LN (operatilithium-ion battery) as illustrated in part (c) and part (d) of Figure 6.10. The results illustrate that the accuracy and residual energy (remaining energy of LNs) decrease with an increase in the number of appliances. It shows that the residual energy of LNs in the EESM system is higher than the non-EESM system.

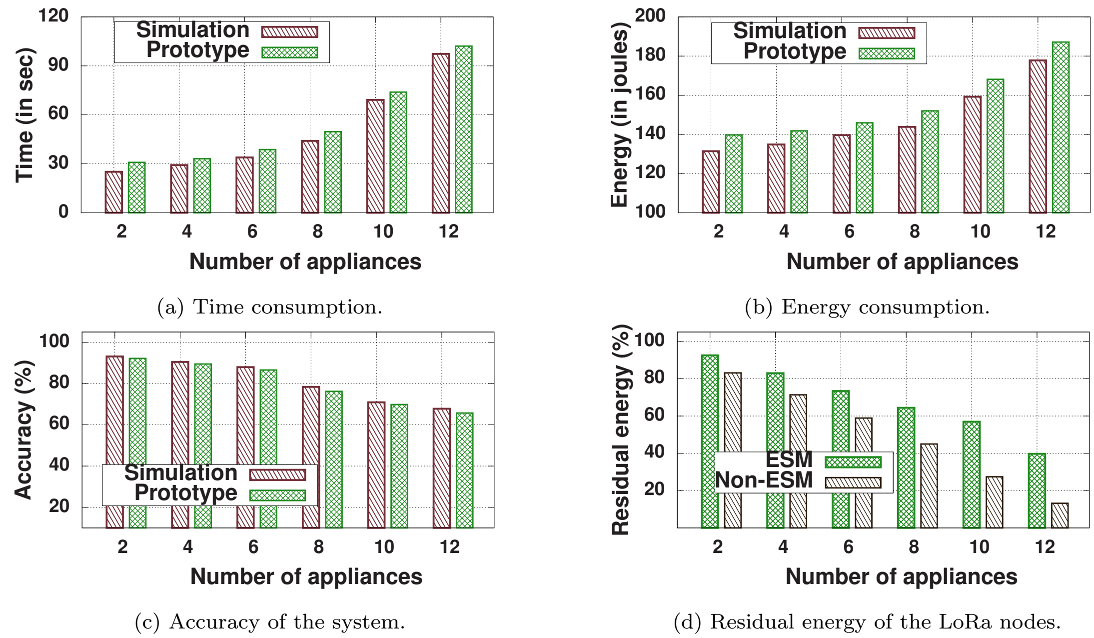


Figure 6.10: Impact of appliances on time, energy, accuracy, and residual energy.

6.7 Conclusion

In this chapter, we proposed an Energy Efficient Smart Metering (EESM) 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 other parameters, such as bandwidth and coding rate. Future directions of research include the extension 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 chapter.