AI-ERA: Artificial Intelligence-Empowered Resource Allocation for LoRa-Enabled IoT Applications

Arshad Farhad and Jae-Young Pyun

***Abstract*— Adaptive data rate (ADR) is a widely adopted resource assignment approach in long-range wide-area networks (LoRaWANs) for static Internet of Things (IoT) applications such as smart grids and metering. Blind ADR (BADR) has been recommended for mobile IoT applications such as pet and industrial asset tracking. However, ADR and BADR cannot provide appropriate measures to alleviate the massive packet loss problem caused by the unsuitable spreading factors (SFs) assigned to end devices when they are mobile. This paper proposes a novel proactive approach—"artificial intelligence-empowered resource allocation" (AI-ERA)—to address the resource assignment issue in static and mobile IoT applications. The AI-ERA approach consists of two modes, namely offline and online modes. First, a deep neural network (DNN) model is trained with a dataset generated at ns-3 in the offline mode. Second, the proposed AI-ERA approach utilizes the pre-trained DNN model in the online mode to proactively assign an efficient SF for the end device before each uplink packet transmission. The proactive behavior of the AI-ERA improved the packet success ratio by an average of 32% and 28% in static and mobility scenarios compared with the typical LoRaWAN ADR, respectively.**

***Index Terms*— The adaptive data rate, artificial intelligence, deep neural network, Internet of Things, LoRaWAN, resource allocation**

I. Introduction

THE Internet of Things (IoT) paradigm is gaining popularity owing to the support for the connectivity of a large number of end devices (EDs). The US National Intelligence Council has estimated that, by the year 2025, billions of EDs will be connected to the Internet. Therefore, low-power wide-area networks (LPWANs) have been commonly used to provide wide-area connectivity to many static and mobile IoT applications (e.g., smart grids, metering, and asset tracking). By 2023, the number of IoT-enabled utility applications connected via LPWAN technology is expected to surpass 300 million worldwide [1]. A long-range wide-area network (LoRaWAN) is an LPWAN that is anticipated to account for up to 20% of smart meter deployments by 2026 owing to its low deployment cost, support for large-scale and long-range connectivity, scalability, ultra-low energy consumption, and enhanced productivity [1]–[3].

A LoRa-enabled network comprises EDs, a gateway (GW), a cloud network server (NS), and application servers. The EDs transmit uplink (UL) packets to the GW using a series of spreading factors (SFs $\in$ [7, 8, 9, 10, 11, 12]) with a transmit power (TP). The NS-side adaptive data rate (ADR) manages SF and TP based on the maximum signal-to-noise ratio (SNR) value among M UL packets (i.e., $M = 20$), which is recommended for static IoT applications, such as metering and smart grids [4], [5]. However, the deployment of ADR for mobile IoT applications results in an enlarged packet loss, high energy consumption, and an unreasonable convergence period (i.e., the time required to reach a stable SF and packet success) owing to ED movement. Thus, a blind ADR (BADR) adopted on the ED side has been recommended for mobile IoT applications [5]. The ED employing BADR transmits a packet once with SF_{12}, three times with an SF_7, and twice with SF_{10} every hour [5]. However, owing to the sudden changes in signal strength between the ED and the GW, BADR still suffers from packet loss [6].

Artificial intelligence (AI) can be applied for LoRaWAN resource management, allowing EDs to learn the underlying propagation conditions and act intelligently by transmitting UL packets with efficient resource parameters. As an example of resource allocation with AI, [7] showed Model deficit and Algorithm deficit cases. The Model deficit reflects insufficient domain knowledge or non-existent mathematical models. In contrast, Algorithm deficit indicates that a well-established mathematical model is present, but the optimization of the existing algorithms is very complex. In those cases, AI-based solutions with reduced complexity are recommended. Additionally, AI approaches are most suited for incorporating contextual information into decision-making. Therefore, in most IoT applications, we could consider AI to manage issues such as network resource allocation, wireless event detection, channel estimation, and so on [8].

Recently, AI has emerged as a cutting-edge machine learning (ML) approach with the potential to achieve substantial advances in various research fields. However, the application of AI for resource allocation remains immature owing to

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (2018-0-00691, Development of Autonomous Collaborative Swarm Intelligence Technologies for Disposable IoT Devices)

The authors are with the Wireless and Mobile Communication System Laboratory, Department of Information and Communication Engineering, Chosun University, Gwangju 61452, Republic of Korea, e-mails: (arshad@chosun.ac.kr)

Corresponding author: Jae-Young Pyun (jypyun@chosun.ac.kr)

the challenge of uniquely identifying network properties as adequate input features for learning structures [9]. Specifically, resource allocation is anticipated to be even more dynamic and complex in the LoRaWAN network owing to massive EDs and long-range connectivity [10]. Therefore, LoRaWAN network operators must optimize resource parameters efficiently rather than relying on simple conventional ADRs (i.e., ADR and BADR). These ADRs cannot provide appropriate measures to alleviate the massive packet loss problem caused by an inefficient SF when the channel condition drastically changes owing to the ED movement. Hence, predicting a suitable SF based on the channel conditions for LoRa links in a complex dynamic environment under shadowing and interference is still challenging. Therefore, we propose a novel deep neural network (DNN)-based approach, referred to as the "AI-empowered resource allocation" (AI-ERA), to alleviate the packet loss issue caused by the improper use of SFs at the static and mobile ED during communication without prior knowledge of the underlying propagation environments. The proposed AI-ERA approach comprises two modes, namely offline and online modes. By leveraging a pre-trained DNN model in the online mode, the AI-ERA framework yields an appropriate SF for static and mobile IoT applications, resulting in improved network performance. The contributions of the AI-ERA are summarized as follows:

1) To solve the issue of resource management (e.g., the spreading factor) to EDs under the condition of static and mobility in LoRaWAN, the proposed AI-ERA framework leverages the DNN model to achieve this goal. The DNN model is trained in offline mode after the appropriate dataset is prepared (using the ns-3 simulator) and formatted.
2) Upon the successful training of the DNN model, a generalized pre-trained DNN model is utilized during network simulation in the online mode for static and mobile EDs. During simulation, a formatted chunk of input data is fed to the pre-trained DNN model, resulting in a predicted SF for the subsequent uplink transmission.
3) The proposed AI-ERA framework proactively allocates the predicted SF to the ED. The proactive behavior of the AI-ERA increases the chances of successfully delivering a packet to the GW, improving the packet success ratio (PSR). To the best of our knowledge, this is the first study to utilize an AI model for SF management in static and mobile EDs.

The remainder of the article is organized as follows: Section II presents related studies, including ADR enhancements, the traditional ML, deep reinforcement, and AI approaches utilized in LoRaWAN; Section III provides extensive details of the dataset generation and data labeling; Section IV presents the proposed AI-ERA framework; and Section V provides an extensive experimental evaluation of AI-ERA in comparison with conventional ADR approaches. Finally, the concluding remarks are presented in Section VI.

II. Related Studies

To resolve the resource allocation issue in LoRaWAN, researchers have been proposing improved ADRs [11]–[14] or seeking to utilize ML [15]–[18], reinforcement learning [19], [20], and deep learning [21], [22] approaches. These approaches are discussed in the remainder of this section.

A. Enhanced Adaptive Data Rates

An enhanced ADR (EADR) under a static scenario at the NS was proposed in [11] to improve the convergence period of the typical ADR. In the EADR, the value of M was set to 5 instead of 20 UL packets. When the number of received UL packets reaches M, the EADR computes the PSR and is compared to a threshold of 80%. Then, it finds the standard deviation of the SNR values of M packets. If the outcome of the standard deviation is lower than 2.5 dB, then the typical ADR was initiated to find a suitable SF and TP. Their simulation results revealed that EADR enhanced the PSR and convergence period compared to the typical ADR. However, the EADR at the NS side waits for M packets every time after a DR change has occurred, resulting in a large convergence period.

A linear regression-based ADR (LR-ADR) at the NS was proposed to manage the SF and TP of mobile EDs [12]. It utilized basic linear regression to get the latest 10 SNR values for each GW. Then, using the linear regression function, the next expected SNR was predicted, and the average value for all GWs was added to a list for the subsequent stages. Further, the average of the last 20 SNR values was computed, and SF and TP values were identified as similar to the typical ADR.

Another ADR was proposed in [13] to mitigate the packet collision in the LoRaWAN network. Their proposed ADR comprises three steps: (i) recording the arrival time of consecutive transmissions at the NS, (ii) ED grouping based on the same SF, and (iii) mitigating the collision by allocating suitable SF, TP, and UL transmission time to the EDs involved in the collision. Their simulation results improved the PSR and lowered the energy consumption and convergence period by reducing the number of retransmissions.

The authors in [14] proposed a new-dynamic ADR (ND-ADR) to adjust the value of M dynamically (it should be noted that in the typical ADR of LoRaWAN, the value of M is set to 20). In the ND-ADR, the value of M is initially set to 3. Then, ND-ADR combines the RSSI and the average of the SNR (SNR_{avg}) values of M received packets for the demodulation instead of choosing the maximum value of SNR among M. When the SNR_{avg} is greater or lower than a particular threshold defined for each SF, the value of M is increased. Their OPNET simulation results showed improved energy consumption, delay, and throughput performance compared with typical ADR.

B. Machine Learning

For an ML-based LoRaWAN system, [16] proposed load balancing (for DL traffic) with unsupervised, supervised, and Markov decision process ML techniques. Their proposed methodology comprises four primary stages: data preprocessing, pattern analysis, classification, and decision-making. The simulation results with 5000 EDs in a multicell environment showed a PSR improvement of 50% and an energy consumption enhancement of 20%.

Another approach for resolving the collision issue of UL packets with ML in an unconfirmed mode of LoRaWAN was presented in [18]. First, the dataset was generated using a random SF assignment approach, where the packets were classified as interfered, unsuccessful, or successful. When the trained model classified a packet as interfering, the SF was increased. For a packet classified as unsuccessful at the GW by the pre-trained model, the SF was decreased. In contrast, the SF was not changed when a packet was classified as successful.

Prediction of LoRaWAN behavior using ML was reported in [23]. First, unsupervised learning was utilized for profiling EDs (i.e., clustering) through a well-known K-means method. The profiling method aimed to group the EDs with similar transmission characteristics (e.g., the same SF and packet size). Then, the decision tree classifier and long short-term memory (LSTM) models were used to predict traffic patterns.

C. Deep Reinforcement Learning

A method based on deep reinforcement learning to assign physical layer parameters (e.g., frequency, SF, and TP) to overcome the massive collision problem was proposed in [20]. Furthermore, a deep Q-network was used at the GW, wherein an agent provided combinations of SF and TP to mitigate collisions. Consequently, a 500% improvement in the PSR was reportedly achieved. However, only a low-density network comprising 100 EDs in a 4.5-km area was considered for the performance evaluation.

D. Deep Learning

A regression approach to predict collisions with an extended Kalman filter (EKF)-based LSTM was proposed in [21]. The authors used the LSTM-EKF model as the backbone based on a pre-trained conventional LSTM model. The primary aim was to predict future collisions, and the dataset was generated using the LoRaSim simulator. The simulation results indicated that the LSTM-EKF model afforded better results in terms of future collision prediction as compared with the conventional LSTM. However, no proactive solution has been proposed to alleviate collisions by allocating a suitable SF to EDs.

Compared with existing research, which employs traditional ML, [15]–[18] deep reinforcement [20], and deep learning [21] approaches for mainly predicting the collision by considering static LoRaWAN network, in this paper, we train the AI model and utilize the trained model in ns-3 for efficient resource allocation (e.g., SF) for static and mobility environments.

III. Dataset Generation and Data Labeling

This section presents the simulation environment, dataset generation, and data labeling procedure of the proposed AI resource allocation approach.

A. Simulation Environment Setup

We have utilized the LoRaWAN module[1] in the ns-3 simulator. During the simulation, 500 EDs are randomly deployed in a radius of 5 km with one GW placed in the middle (similar with [6], [10], [11]), as shown in Fig. 1. The antenna height of ED and GW is set to 1.5m and 15m, respectively. In addition, the simulation environment utilizes log-distance propagation, shadowing, and interference (i.e., inter-SF and intra-SF) models[2]. The interference model and the interference threshold matrix utilized in the simulation is based on [10], [24]. The rest of the detailed experimental conditions are listed in Table I.

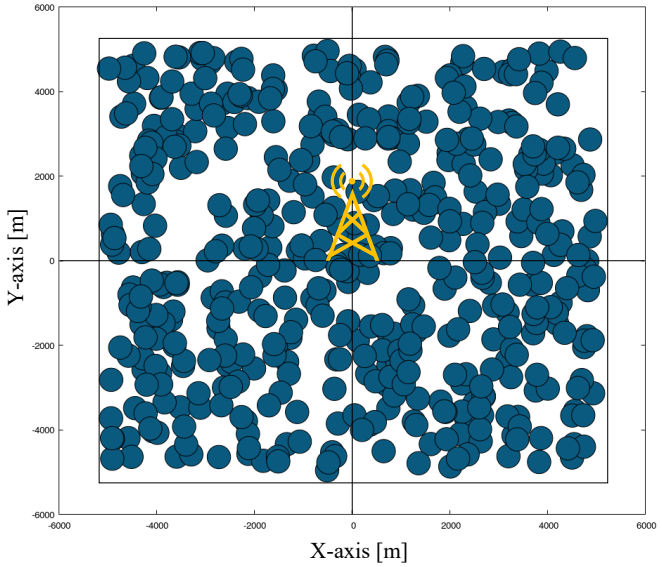


Fig. 1: Simulation environment with 500 EDs deployed in a radius of 5-km.

TABLE I: Simulation parameters utilized in ns-3 for dataset collection.

Parameter	Value
Simulation time [days]	10
UL interval [min]	10 (similar to blind ADR [5])
UL transmission limit	1
UL packet size [bytes]	21 [5]
Frequency region	EU-868
UL & DL channels [MHz]	868.1, 868.3, 868.5
UL channel bandwidth [kHz]	125
Dedicated DL Channel [MHz]	869.525 for DR0 (i.e., SF_{12}) only
Propagation model	log-distance
Shadowing model	de-correlation distance = 110 m [25]
	variance = 6 dB [26]
Type of interference	intra- and inter-SF [10]
Interference threshold matrix	adopted from [24], [27]

B. Dataset Generation

The preliminary step in the proposed framework is to generate the dataset for the training of the AI model. We performed a simulation for 10 days, wherein every ED transmits 6 UL packets assigned with SF_7 to SF_{12} (i.e., UL_1, UL_2,..., UL_6) with every 10-min interval under the condition of a confirmed mode without a retransmission mechanism, as highlighted in Fig. 2. When the NS receives a UL packet from the ED, an

[1]https://github.com/signetlabdei/lorawan

[2]inter-SF interference means that the collision occurs with different SF over the same channel, and intra-SF interference is referred to as the collision caused by the same SF over the same channel.

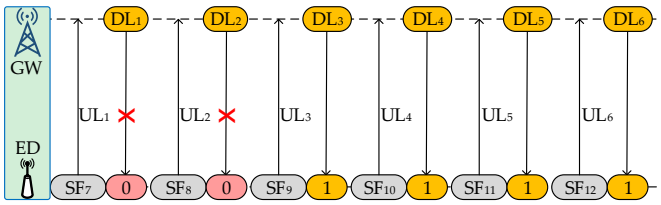


Fig. 2: An example showing a single *group* (g_1) with 6 UL packet transmissions (using SF_7 to SF_{12}) for a single device from the same locations in a one-hour UL interval.

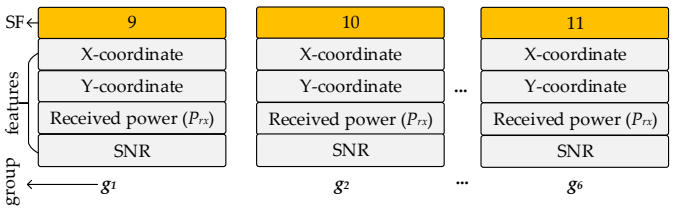


Fig. 3: Example of SF labeling of each group with corresponding four features including X- and Y-coordinates, P_{rx}, and SNR.

ACK is transmitted via the GW in a first receive window with the same SF and channel utilized at the transmission time of the UL packet. If the ACK fails in the first receive window, the NS transmits ACK again in the second receive window using a dedicated channel of 869.525 MHz with SF_{12}. Fig. 2 shows an example of UL packet transmission and its ACK reception. Here, an ACK failure is represented with "0," and successful ACK reception for each corresponding UL packet is marked with "1"—in a confirmed mode, while a UL packet is considered successful if the ED receives ACK. To this end, the X- and Y-coordinates of the ED, received signal power (P_{rx}), SNR, and ACK status (i.e., either 0 or 1) are stored in the dataset.

During the dataset generation, we disabled the retransmission mechanism since it can cause a bottleneck at the GW during UL packet reception and downlink ACK transmission. Since the GW in LoRaWAN is not full-duplex, ACK transmission is prioritized over the incoming UL packet reception [10].

C. SF Labeling

The SF labeling is broadly categorized into: SF labeling of individual group and generating input sequence with a labeled SF.

1) SF labeling of individual group: The 6 UL packets, represented with UL_1, UL_2,..., UL_6 transmitted with SF_7 to SF_{12} are organized to one group, as shown in Fig. 2. From each group, the lowest SF is chosen based on successful ACK reception. However, if no ACK in any group for UL_1, UL_2,..., UL_6 was received, SF_{12} is assumed. For instance, SF_9 is the lowest among successful ACK receptions for SF_{10}, SF_{11}, and SF_{12}, as highlighted in Fig. 2. Eventually, the chosen SF_9 is labeled for the corresponding ED channel features, X- and Y-coordinates, P_{rx} (which is determined by using [24]), and SNR, as shown in Fig. 3.

Even though the UL packets were generated from the same ED, the ACK reception for each SF was completely different at each measured time owing to the underlying dynamic propagation environment, shadowing model, and inter-SF and intra-SF interferences.

2) Generating input sequence with labeled SF: The labeling of each input sequence is further divided into two cases: $Case-1$ and $Case-2$.

1) $Case-1$**:** Once the SF labeling of the individual group is completed, we generate an input sequence of 6 groups

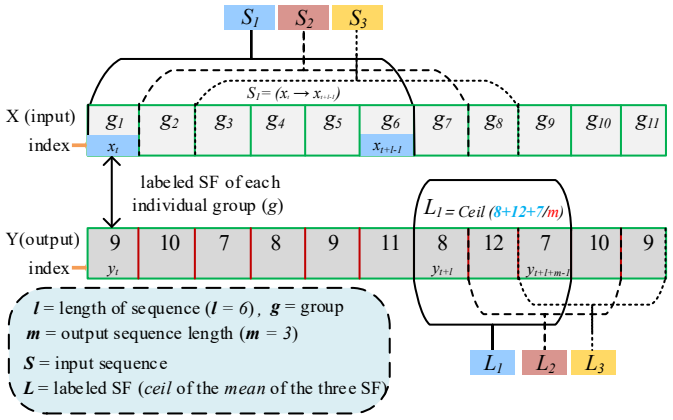


Fig. 4: $Case-1$: Example of input sequence (S) with the corresponding labeled SF (L). The S_1 shows one input sequence (i.e., from g_1 to g_6) with corresponding one labeled SF (i.e., L_1).

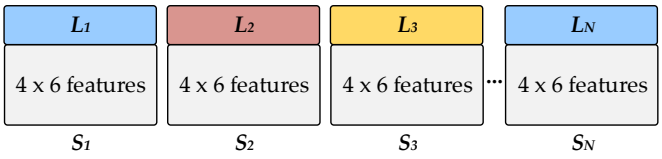


Fig. 5: $Case-1$: Final input sequence representations (S_1 to S_N) with labeled SF (L_1 to L_N). Each S corresponds to 24 flattened inputs (e.g., 4 features and 6 groups) for the DNN model.

with a corresponding labeled SF, as shown in Fig. 4. The input sequence selection and a corresponding labeled SF can be modeled mathematically as (based on Fig. 4)

$$S_t = \{x_t, x_{t+1}, ..., x_{t+l-1}\}, \tag{1}$$

$$Y_t = \{y_{t+l}, ..., y_{t+l+m-1}\}, \tag{2}$$

$$L_t = \left\lceil \frac{1}{m} \times \sum_{n=t+l}^{T} y_n \right\rceil, \text{ where } y_n \in Y_t, \tag{3}$$

$$T = t + l + m - 1, \tag{4}$$

where in (1), S_t represents the input sequence, x_t is the starting index of the input data and t iterates from 1 to $N - l - m$, l is the length of the input sequence ($l = 6$) and N is the total length of the dataset. Eq. (2) shows the output sequence, where m is the length

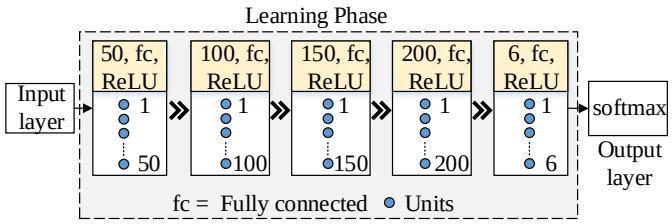


Fig. 6: Training phase of the DNN model with the vectorized input layer of size 24 (i.e., 4×6 features) and five fully connected hidden layers.

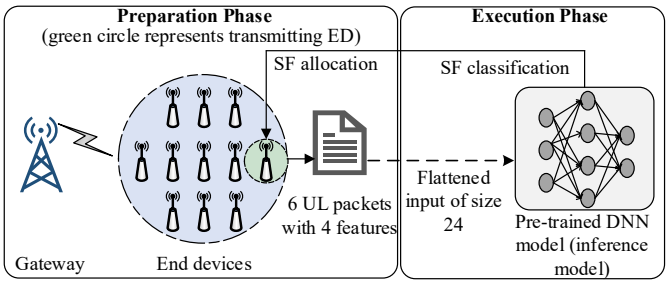


Fig. 7: The online mode working procedure of the proposed AI-ERA framework for resource allocation to EDs.

of the output sequence ($m = 3$). Finally, (3) is used to calculate the final labeled SF by taking a ceil after averaging the m sequence length (i.e., 3-SFs). The final input sequence (S_1 to S_N) with labeled SF (L_1 to L_N), as indicated in Fig. 5, is used as input for the DNN model.

Furthermore, in this work, we found 6 groups for training the DNN model in a heuristic way, generating the best training performance. Generally, packets with the same SF delivered at the same channel environment do not always provide the same results. For the training target, the lowest SF among 6 input samples was chosen based on successful ACK to lower the time-on-air (ToA). Another factor for selecting the number of 6 groups was a reduction of the convergence period caused by the ADR in the network. A larger than 6 in the number of groups generates a larger convergence period, which could not be acceptable to some IoT applications.

2) $Case-2$**:** For the $Case-2$, the input sequence is labeled according to (1), (2), and (3), where the value of m is set to 1 [28]. As an example, for input sequence, S_1, the corresponding SF_8 will be labeled, as illustrated in Fig. 4.

IV. The Proposed AI-ERA Framework

In this paper, the proposed AI-ERA uses a one-time generated dataset to train the DNN model, and then the pre-trained DNN model is used in ns-3 to estimate the best SF for the static or mobile EDs before each UL packet transmission. The proposed AI-ERA framework consists of two modes, namely offline and online. In the remaining section, these two modes are discussed in detail.

A. AI-ERA: Offline Mode

The offline mode of the proposed AI-ERA framework is mainly responsible for training the DNN model, as illustrated in Fig. 6. Before the DNN model training, our input data consisting of 4×6 features are flattened, i.e., we adopted the vectorized input layer of size 24. It is because the DNN model cannot handle two- or three-dimensional data. The labeled data is divided into training (80%), validation (10%), and testing (10%). To train the DNN model correctly, we utilized fully connected layers via the $ReLu$ function to update the weights and biases. Finally, $softmax$ (in $pytorch$, cross-entropy[3]. is taking care of it), which classifies the SFs, is utilized at the output layer. In multiclass classification issues, the $softmax$ activation function is typically used in the last layer, mathematically represented as

$$I_k = \sigma(z)_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } k = 1, 2, ..., K, \qquad (5)$$

where K is the number of classes (6 classes, in this paper, indicating from SF_7 to SF_{12}), I_k is the index of the k-th SF class, and z = (z_1, z_2,..., z_K) represents the output of the layer before the softmax activation function [29].

Furthermore, this trained DNN structure (inference model) with the best hyperparameters is utilized as a backbone to yield the best SF in the online mode.

B. AI-ERA: Online Mode

The online mode refers to classifying an appropriate SF in ns-3 with a new input of 6 UL packets and 4 features generated from an ED during UL communication. The online mode is divided into two phases, preparation and execution, as shown in Fig. 7.

1) Preparation Phase: This phase manages the new data for determining the best SF using the pre-trained DNN model. The new input data is prepared with a UL history of size 6 and 4 features for each ED during communication (i.e., a vectorized input layer of size 24). In our online test mode, EDs (static or mobile) transmit UL packets to the GW. This phase collects the input UL data in the same form as the training dataset without prior knowledge of the ACK status. Once the UL history reaches a size of 6, the execution phase begins.

2) Execution Phase: The pre-trained model in the proposed AI-ERA framework waits only for the first six UL packets and uses a sliding window after that, as shown in Fig. 8. In this phase, a vectorized input layer of size 24 is fed to the pre-trained DNN model to obtain one suitable SF. Finally, the GW forwards the received packet to the NS for further processing, such as monitoring and tracking, as illustrated in Fig. 7.

As an example of the execution phase and resource management in a dynamic environment, Fig. 9 illustrates the ED movement within a radius of 5 km (gateway is placed at positions X-axis: 0.0, Y-axis: 0.0), where the ED randomly chooses

[3]https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

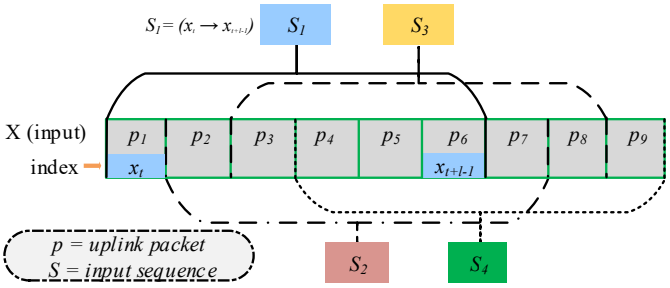


Fig. 8: Formatted input sequence of 6 uplink packets with 4 features for the pre-trained model in the simulation environment. The input sequence selection criteria are based on a moving window for accommodating mobile end devices.

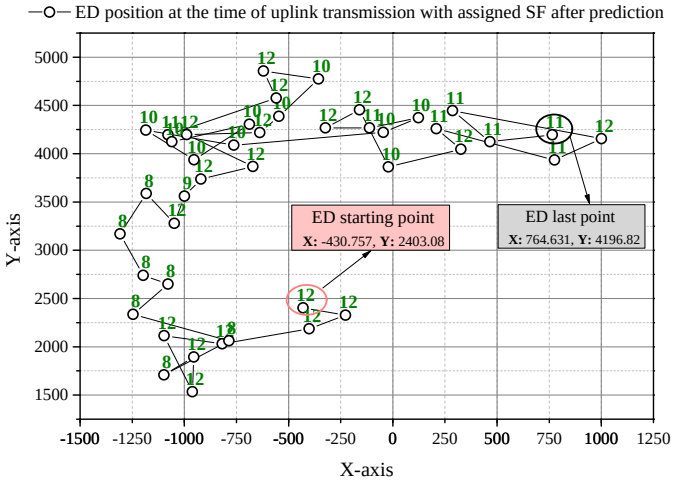


Fig. 9: Example of ED movement and SF assignment by the AI-ERA framework in a dynamic network.

its speed between 1.0 and 2.0 m/s. During the experiment, only one ED was deployed with an initial SF of 12, transmitting 6 uplink packets per hour. Fig. 9 shows the 42 positions marked with the allocated SF predicted by the DNN model. Initially, the ED transmits the first 6 uplink packets with SF_{12}. Once the input sequence reaches 6 (i.e., 6 uplink packets), the chunk of formatted data is fed into the pre-trained model. As a result, a predicted SF is assigned to the concerned ED at the time of every uplink.

C. Computational Complexity of AI-ERA

The proposed AI-ERA framework is implemented on the ED side. It can also be deployed at the NS side; however, the convergence period would be significantly increased because the AI-ERA framework should wait for M UL packets received at the NS to be activated [11], [30]. Therefore, we deployed the AI-ERA framework on the ED side to classify a suitable SF proactively before each UL transmission by starting the prediction of SF as soon as the UL packet transmission reaches 6, thereby increasing the probability of successful transmission with the selected SF.

However, employing the AI-ERA on the ED side can lead to excessive computational resources. Therefore, the computational complexity of the proposed AI-ERA is computed

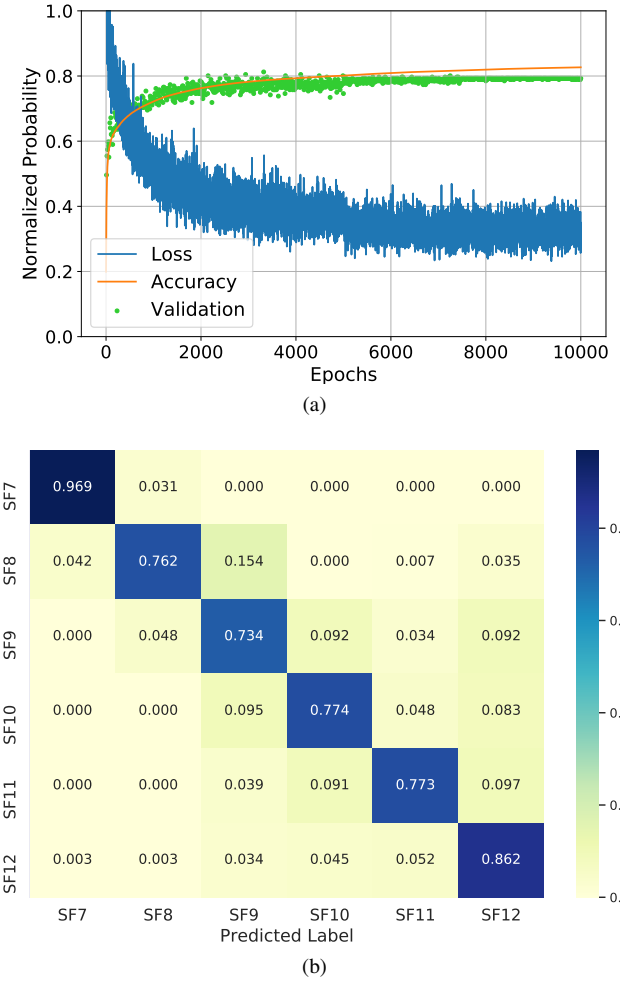


Fig. 10: DNN model performance during the offline mode for the $Case-1$ with the vectorized input layer of size 24 (i.e., 4×6 features) and five fully connected hidden layers: (a) model accuracy and (b) confusion matrix.

as $\mathcal{O}(n^2)$, excluding the inference model. In addition, the computational complexity of the inference model (during the online mode) in terms of floating point operations (FLOPs) per one input sample (i.e., vectorized input layer of size 24 features) is estimated as 13.54 MMac (Mega FLOPs), and the space complexity is determined as 52,906 parameters. Also, the time complexity of AI-ERA exceeds that of the typical ADR and BADR at the ED side (which is computed as $\mathcal{O}(c)$ in [10]). Therefore, there is a trade-off between the attained performance (e.g., enhanced PSR, ultra-low energy consumption, and short convergence period) and the computational cost.

V. Experimental Results and Analysis

A. AI-ERA Analysis in the Offline Mode

Here, we present the training parameters, the DNN structure, and the DNN model performance in terms of accuracy and confusion matrix obtained using the AI-ERA approach during the offline mode.

1) Training Parameters: The offline results with the DNN model consisting of five hidden layers were obtained in a GPU

TABLE II: Validation accuracy analysis of the AI models with 4 features (i.e., X- and Y- coordinates, P_{rx}, and SNR) and 6 groups in the offline mode.

No. of layers	Neurons	Input sequence length	Parameters	FLOPs	Case-1 [%]	Case-2 [%]
LSTM						
1	200	Input sequence size 6 with 4 features	350.71 k	85.42 MMac	79.30	78.12
2	200		1.31 M	333.03 MMac	80.08	77.73
3	200		2.28 M	580.63 MMac	78.91	77.34
4	200		3.24 M	828.23 MMac	79.30	76.17
5	200		4.2 M	1.08 GMac	78.52	77.34
GRU						
1	200	Input sequence size 6 with 4 features	268.31 k	64.02 MMac	79.30	76.56
2	200		990.71 k	249.67 MMac	79.30	76.95
3	200		1.71 M	435.32 MMac	79.69	76.95
4	200		2.44 M	620.97 MMac	76.95	78.12
5	200		3.16 M	806.63 MMac	79.30	76.17
DNN Model						
1	24,6	$4 \times 6 = 24$ (flattened)	150	36.87 KMac	53.52	53.12
2	24, 50, 6		1.56 k	396.86 KMac	59.38	54.69
3	24, 50, 100, 6		6.96 k	1.78 MMac	71.09	69.53
4	24, 50, 100, 150, 6		22.41 k	5.73 MMac	77.73	76.95
5	24, 50, 100, 150, 200, 6		52.91 k	13.54 MMac	80.47	78.12

NVIDIA GeForce GTX 1050 Ti environment using PyTorch 1.9.0, with a batch size of 256, a learning rate of 0.0001, a dropout ratio of 0.5, and an Adam optimizer. Furthermore, during the training process, the DNN model was validated after every 10^{th} epoch and tested with a 10% unknown UL history to evaluate the model accuracy. When the training and validation losses reached their maximum epochs, they achieved a plateau. At this point, the DNN model could be regarded as completely trained. This pre-trained model was further utilized in the online mode to assign the best SF to the EDs before each UL packet transmission.

2) Analysis of the AI models: Table II lists different AI models for the $Case-1$ and $Case-2$ (Section: III-C.2) SF classification cases during the offline mode.

All AI models with different layers have been trained with 80% data, validated with 10% data, and tested with 10% unknown labeled data in order to evaluate the model accuracies. In general, the accuracy of the $Case-1$ is higher than the $Case-2$. The complexity of all AI models in terms of trainable parameters (space complexity) and FLOPs (time complexity) is increasing with the increasing number of layers. However, compared to LSTM and GRU, the complexity of the DNN model is much lower, with similar performance (in the case of layer 5). Therefore, based on the highest accuracy and lower complexity, we chose DNN with 5 layers of $Case-1$ and utilized its inference model in online mode.

Furthermore, we chose a support vector machine (SVM) algorithm for comparison with the proposed AI-ERA framework. To develop the SVM model, we have utilized the $scikit-learn$ library. Before training the SVM model, we normalized the labeled data between the interval [0, 1], which is a fundamental step for the SVM. In addition, the SVM model was trained with the same dataset as AI-ERA. Eventually, it produced validation accuracies of 48.29% and 46.13% in $Case-1$ and $Case-2$, respectively.

3) DNN Model Performance: During the offline mode, the performance of the DNN model and the confusion matrix is shown in Fig. 10. First, the model accuracy was recorded as 82%, as shown in Fig. 10 (a). The accuracy and validation

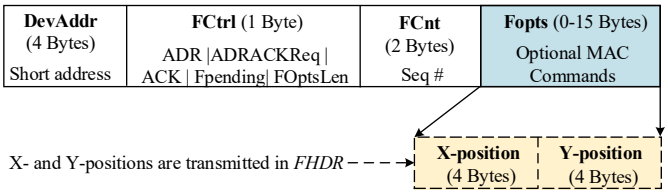


Fig. 11: The modified LoRa frame header containing X- and Y-positions of the ED.

increase and the loss decrease with increasing epochs. Second, the confusion matrix is shown in Fig. 10 (b), which lists the accuracies of individual SF classification. The DNN model learns the relationship between the UL transmission conditions and the corresponding best-labeled SF during the offline mode. As a result, it achieves a good performance of 82% because the pre-trained DNN model provides dynamic SF prediction and assigns the predicted SF to static and mobile EDs.

B. Experimental Setup in the Online Mode

1) Application Scenario: We considered the following two IoT applications: smart grids (metering) [1], [3], [31] and industrial asset tracking [32], [33], as proposed by Semtech and GSMA-3GPP, respectively. First, we considered a packet size of 30 bytes corresponding to metering and industrial asset tracking applications for static and mobile EDs. In the case of metering, the ED location information in the LoRa frame header (FHDR) is not required. However, location information needs to be sent inside the FHDR for tracking and monitoring purposes for industrial asset tracking. The industrial asset tracking packet size includes 8 bytes of X- and Y-coordinates. Therefore, we have modified the LoRa FHDR to transmit the location information of the EDs, as shown in Fig. 11.

2) Simulation Scenario: This study considered EDs working in the confirmed mode and deployed in a single GW scenario within a radius of 5 km. For the mobile scenario, a random-walk two-dimensional mobility model was adopted in the case of industrial asset tracking applications [10], [33]. The EDs

TABLE III: Simulation parameters utilized in the online phase.

Parameter	Description and Value
Maximum UL transmissions	8 (including 7 retransmissions)
ED movement speed [m/s]	1.0∼2.0 ([6])
Change in direction	after 200 m
Frequency region	EU-868 MHz
UL channels	[868.1, 868.3, 868.5] in MHz
Initial SF and TP	SF = 12 and TP = 14 dBm

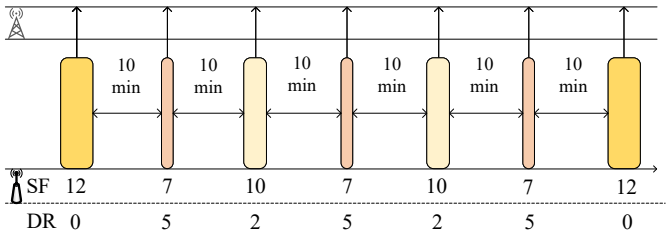


Fig. 12: SF selection of the typical BADR used for LoRaWAN.

transmit six UL packets per hour during the simulation for 24 h. In this paper, each simulation is run 10 times and the results are presented with a 95% confidence interval, similar to [34]–[36]. The remaining simulation parameters are listed in Table III.

In addition, the simulator used in this study was the LoRaWAN module[4] with the addition of (1) the mobility model, (2) the BADR protocol, (3) modified LoRa FHDR for tracking applications [5], (4) Gaussian ADR (GADR) [6], and (5) the proposed AI-ERA framework.

C. Description of the existing ADRs utilized in the Analysis

Here, we present the existing ADRs, considered for comparison with the proposed AI-ERA.

1) Typical ADR of LoRaWAN for Static Applications: The ADR at the NS side (recommended for static application [5]) decreases the value of SF by 1 and increases/decreases the value of TP by 2 after M (i.e., M = 20) UL packets. The values of SF and TP are decided based on the highest SNR value (SNR_m) among the M packets. Once the typical ADR identifies these parameters, the NS transmits them in a MAC command $LinkADRReq$ as unconfirmed DL (i.e., no ACK notification is required from the corresponding ED).

2) Typical BADR of LoRaWAN for Mobile Applications: BADR was designed for mobile applications (e.g., pet-tracking) on the ED side. In BADR, each ED transmits a data packet with GPS status to the NS once with an SF of 12, three times with an SF of 7, and twice with an SF of 10 with an interval of 10 minutes, as shown in Figure 12 [5]. To replicate the BADR behavior, we have modified the FHDR, as shown in Fig. 11

3) Gaussian ADR (GADR): The GADR was proposed for static and mobile applications, such as smart grids and pet-tracking [6]. It works at the NS side for SF and TP management, based on $M = 20$ UL packets. First, the GADR computes the mean (μ) and variance (σ) of the SNR of the M packets. Then, the GADR utilizes the high pass and low pass filters using the thresholds of $\mu + \sigma$ and $\mu - \sigma$ to find the centralized SNR values in the effective range. Finally, it averages the centralized values of the SNR, which is used to decide the SF and TP of the ED.

[4]https://github.com/signetlabdei/lorawan

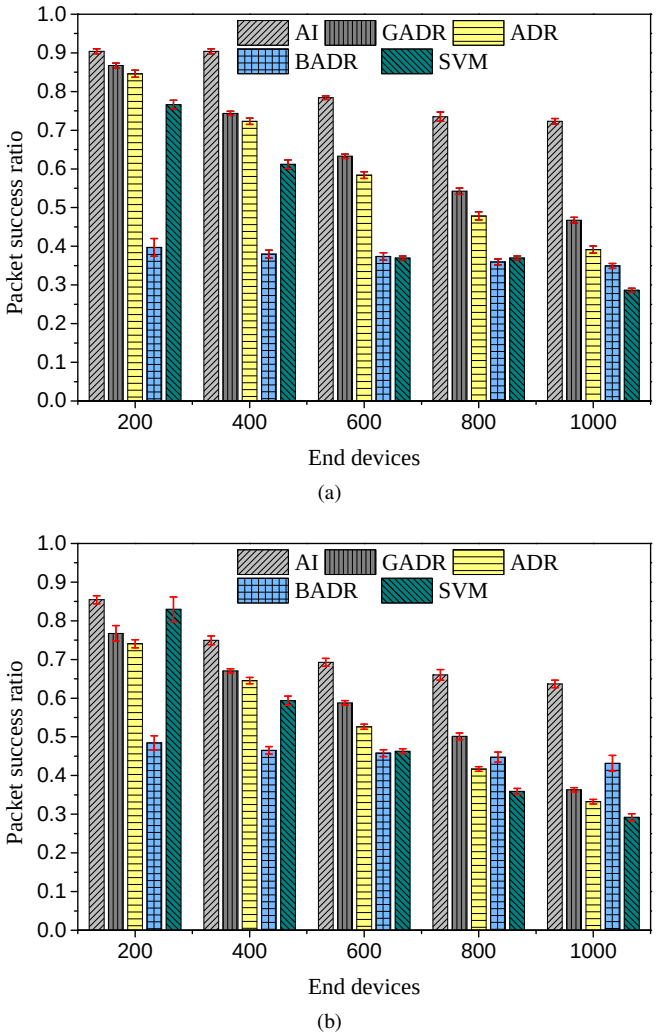


Fig. 13: Packet success ratio with a 95% confidence interval: (a) static scenario and (b) mobile scenario.

D. AI-ERA Analysis in the Online Mode

The proposed AI-ERA framework was evaluated in terms of the PSR, packet loss ratio (PLR), energy consumption, and convergence period.

1) Packet Success Ratio (PSR): Fig. 13 shows a decrease in the PSR for all approaches when the number of EDs increases in both static and mobile scenarios. The PSR decrease in the static scenario is because of intra- and inter-SF interferences. However, the PSR of BADR in both scenarios is static owing to the use of only SF_7, SF_{10}, and SF_{12}. In these experiments, the main cause of high packet loss is the arrival of packets at the GW under the required sensitivity thresholds when they are transmitted with SF_7. The required sensitivity thresholds for each SF at the GW and ED are listed in [6].

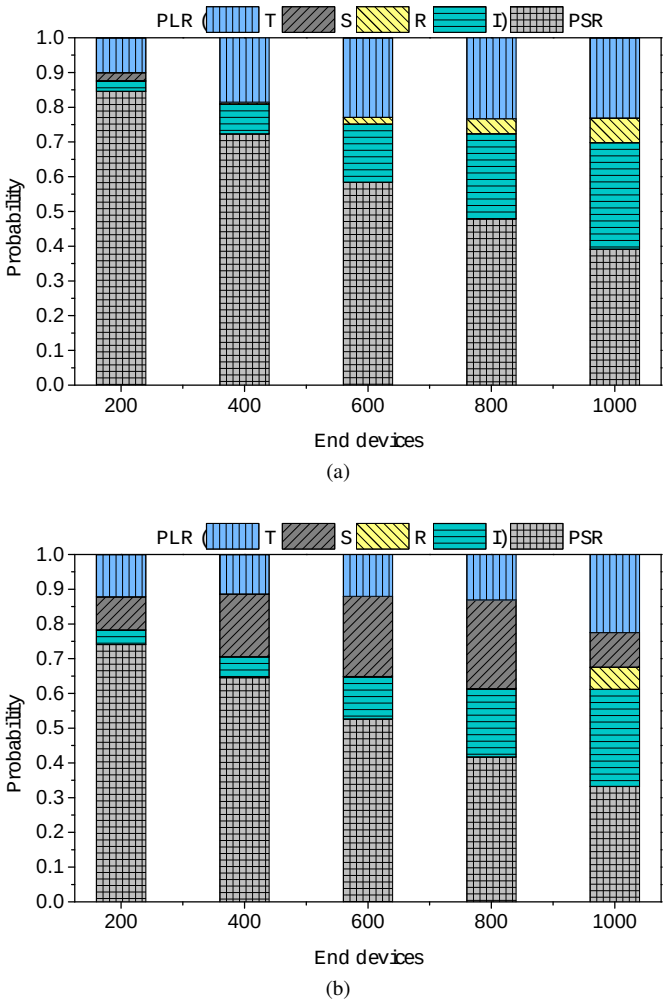


Fig. 14: Average PSR and PLR for ADR: (a) static scenario and (b) mobile scenario.

Specifically, the PLR was higher in the mobile scenario than in the static scenario. This is because the propagation environment changes significantly when the ED moves around the GW. Therefore, the ED employing ADR, GADR, and SVM when receives a DL $LinkADRReq$ MAC command with the newly configured SF and TP when a packet transmitted with these parameters can no longer be useful because it may not be delivered to the GW, resulting in packet loss.

Compared with typical ADR approaches, the proposed AI-ERA framework performs significantly better in static and mobile scenarios. Such outstanding performance is because the DNN model learns the channel and ED transmission conditions for the best SF selection in the offline mode, thereby providing a better SF assignment for the transmitting ED before each UL transmission. This proactive behavior of the AI-ERA reduces the chances of interference, resulting in PSR enhancements of 32% and 28% in the static and mobility scenarios, respectively, as compared with the ADR. The improvement in each scenario (i.e., static or mobility) is calculated by averaging the improvement in PSRs of all cases (i.e., from N=200 to 1000).

2) Packet Loss Ratio (PLR): Figs. 14 to 17 show the PSR and PLRs measured for all the ADR approaches in the static and mobile scenarios. For the observation of PSR and PLR, the amount of sent or retransmitted packets is exclusively applied to PSR or PLR.

(i) **ADR:** Fig. 14 shows that the PLR increases in a typical ADR with an increasing number of EDs. Typical ADRs employ a high SF during the initial deployment, resulting in interference (PLR-I) owing to the high ToA. Therefore, a high SF can negatively affect the capacity of the communication channel [37]. Packet losses occur owing to several reasons in these LoRa experiments [10]. When two or more packets collide (called intra- or inter-SF interference), massive packet loss occurs [10]. After a packet loss occurs, the packet is retransmitted, increasing the network capacity and resulting in increased PLR owing to ACK transmission at the GW (PLR-T). In contrast, the packet loss owing to the arrival of packets under the required sensitivity thresholds at the GW (PLR-S) and the reception path unavailability at the GW (PLR-R) is negligible.

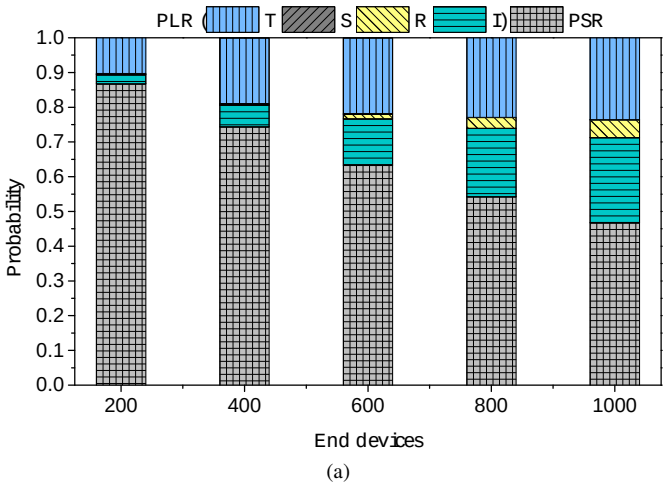
However, in the mobile scenario, the impact of PLR-S is significant owing to the ED movement, as shown in Fig. 14 (b). The ADR cannot assign an efficient SF because of the abrupt SNR changes; thus, the packets arrive under the required sensitivity thresholds at the GW. This packet loss increases retransmission, and the GW becomes a bottleneck, resulting in an increased PLR-T. Moreover, in the case of $N = 1000$, significant interference can be observed in Fig. 14 (b) caused by inter- and intra-SF interferences, and therefore, the PSR is reduced significantly.

(ii) **GADR:** The GADR mainly identifies the SF and TP using Gaussian filter thresholds [6]. In the static scenario shown in Fig. 15 (a), the GADR improves the PSR. However, the PLR-I and PLR-T of the GADR in the static scenario were similar to those of the ADR. Therefore, a high SF (e.g., SF_{12}) that generates the highest ToA can be frequently used but remains vulnerable to interference.

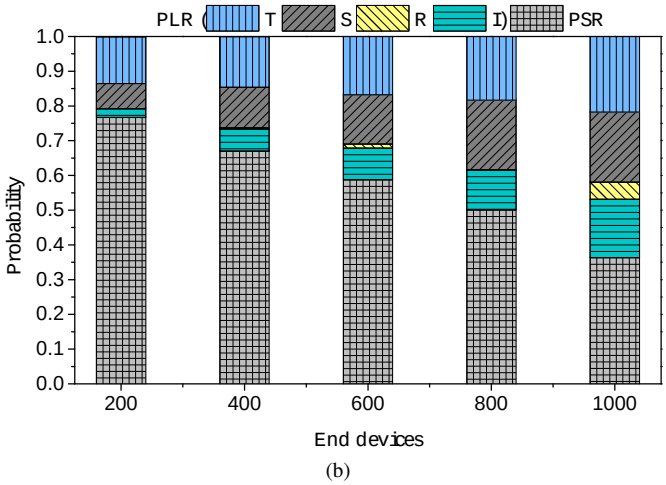
Nevertheless, in the mobile scenario shown in Fig. 15 (b), the impact of PLR-S is lower than that of the ADR. This is because the GADR can efficiently determine the SF and TP configurations concerning Gaussian filters. However, some packets still arrive under the required sensitivity thresholds, causing interference and increasing the chances of retransmission owing to changes in the propagation environment. Furthermore, the retransmission increases bidirectional communication; therefore, PLR-T increases when the impact of PLR-R is negligible.

(iii) **BADR:** The impact of the PLR-S and PLR-T of the BADR follows a trend similar to that of the cases of ADR and GADR in both static and mobile scenarios, as shown in Fig. 16. This is due to the fixed and blind strategies of SF selection among SF_7, SF_{10}, and SF_{12}. The fixed strategy of SF selection in the case of BADR causes PLR-S when packets arrive with inadequate power at the GW, resulting in a lower PSR.

(iv) **AI-ERA:** As shown in Fig. 17 (a), the AI-ERA framework performed exceptionally well in terms of PLR-S in the static scenario owing to its proactive behavior in the online

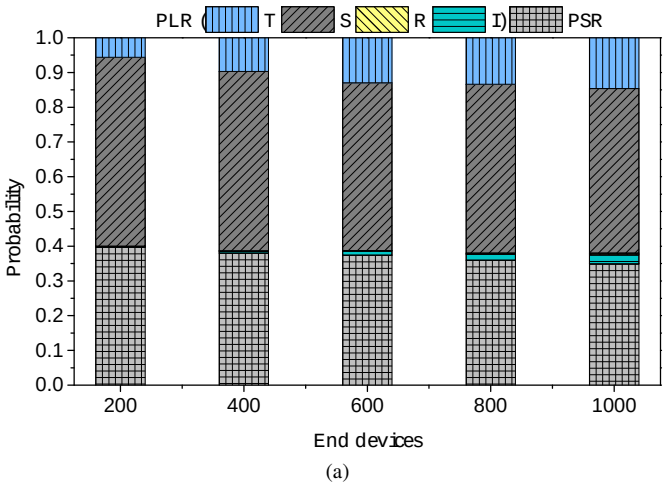


(a)

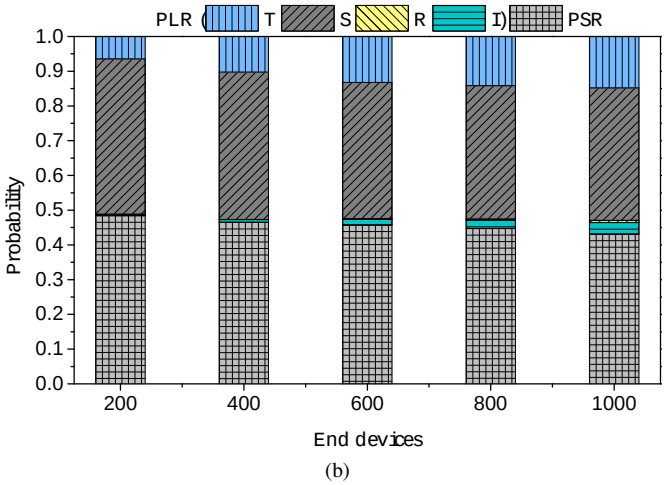


(b)

Fig. 15: Average PSR and PLR for GADR: (a) static scenario and (b) mobile scenario.



(a)



(b)

Fig. 16: Average PSR and PLR for BADR: (a) static scenario and (b) mobile scenario.

mode while assigning an appropriate SF before UL packet transmission. Therefore, the static ED successfully delivered a packet with an adequately higher sensitivity than required. However, a small fraction of PLR-S was observed for the proposed AI-ERA approach owing to the misclassification of the suitable SF.

However, as shown in Fig. 17 (b), the impact of PLR-S in the mobile scenario is slightly higher than that in the static scenario owing to the ED movement. The ED movement causes drastic changes in location, P_{rx}, and SNR, resulting in an inaccurate SF classification owing to the overfitting problem. However, the proposed AI-ERA framework significantly outperforms the conventional ADR approaches in terms of PSR and PLR in both static and mobile scenarios.

(iv) **SVM:** The overall performance of the SVM model for the static and mobility scenarios is shown in Fig. 18. The PSR performance is decreasing with the increase in the number of EDs owing to the high impact of the interference caused by the higher SF in the network during the online phase. Since the accuracy of the SF prediction was relatively low during the offline mode, it resulted in low performance in the online mode while predicting an accurate SF for the EDs. It is evident in Fig. 18 that both static and mobile EDs suffer from high interference owing to high SF selection during the UL transmission. However, due to the prediction of high SF (e.g., SF_{11} and SF_{12}), the packet loss occurring due to arriving under the required sensitivity threshold is negligible.

3) Energy Consumption: In this paper, the energy consumption is computed as the energy consumed by the EDs divided by the total number of successful receptions (in the confirmed mode, a packet is termed successful when it receives a corresponding ACK from the NS). The parameters utilized in the energy consumption simulation are taken from [10], [11].

Fig. 19 shows the energy consumption measured at the ED employing ADR approaches. The energy consumption trend increases for all the ADR approaches when the number of EDs increases. However, The energy consumption of the proposed AI-ERA framework is much lower than that of the conventional ADR approaches in static and mobile scenarios, as shown in Fig. 19. This is because the SF selection in AI-ERA is based on prior knowledge, resulting in a suitable SF assignment to EDs at the time of UL transmission. This proactive assignment of efficient SFs increases the chances of successful packet delivery, thereby mitigating retransmissions

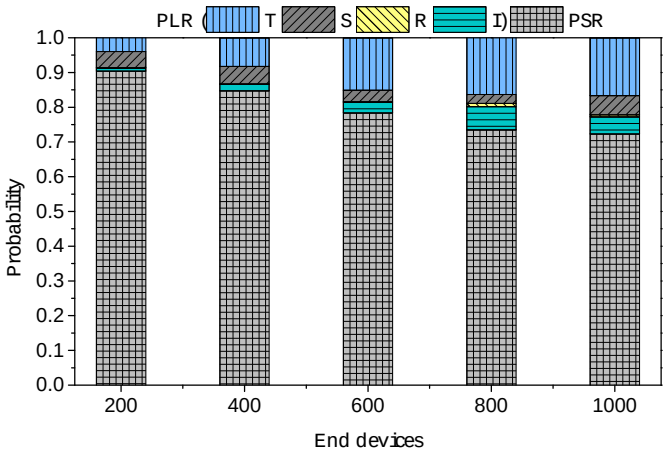


Fig. 17: Average PSR and PLR for the proposed AI-ERA framework: (a) static scenario and (b) mobile scenario.

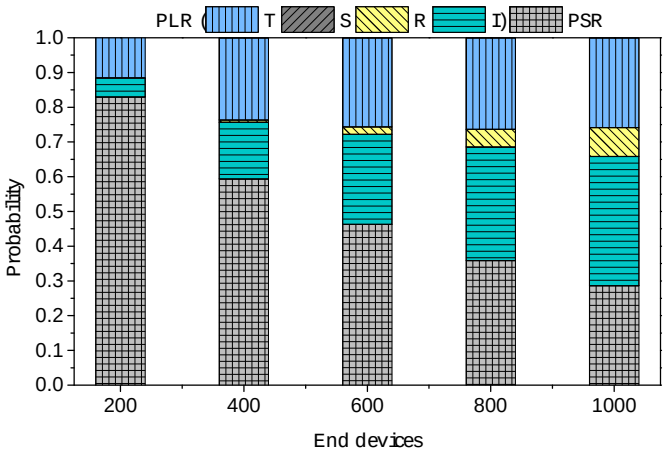


Fig. 18: Average PSR and PLR for the SVM: (a) static scenario and (b) mobile scenario.

from the ED.

However, EDs employing ADR, GADR, and SVM transmit packets with relatively high energy-consuming communication parameters (i.e., SF_{12} and TP=14 dBm), resulting in a massive PLR owing to high interference. Furthermore, these lost packets are retransmitted with the same communication parameters, increasing energy consumption. This is because the energy consumption of packet transmission is mainly based on the SF, TP, ToA, and retransmission parameters [38]. In contrast, the primary aim of the BADR is to lower energy consumption, which is realized as shown in Fig. 19. Thus, the energy consumption of the BADR reveals that it is more energy-efficient than the typical ADR and GADR.

4) Convergence Period: Fig. 20 shows the average convergence period in terms of per-hour PSR for a simulation time of 24 h with $N = 1000$ in the static and mobile scenarios.

The ADR and GADR approaches require initial waiting for M UL packets to be received at the NS before altering the SF and TP. Therefore, convergence periods of 20 and 18 h are observed in a static ED scenario for the ADR and GADR, respectively, as shown in Fig. 20 (a). Similarly, in the mobile scenario, the ADR and GADR exhibit convergence periods of 17 and 19 h, respectively. When a UL packet is lost, both the ADR and GADR must wait longer, which is a tedious operation that yields a long convergence period.

Conversely, the BADR suffers from a convergence period of 6 h in static and mobile scenarios, respectively, then maintains a stable lower PSR than the other ADR approaches, as shown in Fig. 20. Furthermore, the BADR approach does not require waiting time to alter the SF or TP. Instead, it follows a predetermined blind pattern using SF_7, SF_{10}, and SF_{12}. The fixed and blind patterns can lead to packets arriving under the required sensitivity thresholds at the GW with a lower SF. However, when a packet with SF_7 is lost at the first attempt, the next UL packet can be transmitted with SF_{10}. Similarly, SVM is adopted at the ED side for suitable SF allocation, where no convergence is observed. However, the per-hour PSR is much lower compared to other ADR approaches and the proposed AI-ERA framework.

In contrast, the proposed AI-ERA framework exhibits convergence periods of 6 h in static and mobile scenarios, respectively, as shown in Fig. 20. Thus, the AI-ERA approach significantly reduces the convergence period by efficiently classifying the SF of every ED before UL packet transmission.

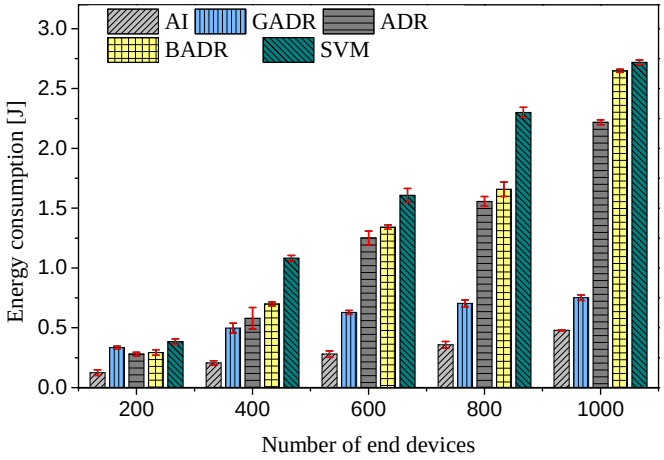


Fig. 19: Energy consumption in Joules with a confidence interval of 95%: (a) static scenario and (b) mobile scenario.

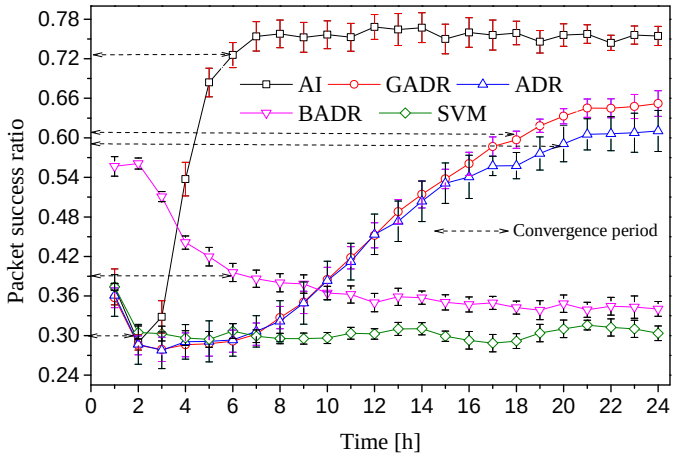


Fig. 20: Convergence period: per hour PSR with a 95% of the confidence interval: (a) static scenario and (b) mobile scenario.

VI. Conclusions

This paper proposed a novel, proactive artificial intelligence-empowered resource allocation (AI-ERA) approach to address the resource assignment issue in static and mobile IoT applications. First, we trained the DNN model that forms the basis of the AI-ERA framework in the offline mode. Then, the pre-trained DNN model was utilized in the online mode to obtain a suitable SF for every UL transmission accurately. Consequently, compared with conventional ADR approaches and SVM, the proposed AI-ERA framework demonstrated a substantially higher PSR, a short convergence period, and ultra-low energy consumption by mitigating the impact of interference and lowering the effect of packets arriving under the sensitivity threshold. Therefore, we believe that the proposed lightweight AI-ERA framework can be utilized for static (e.g., smart metering) and mobile (e.g., asset/livestock tracking) IoT applications requiring reliability, high PSR, and ultra-low energy consumption.

References

[1] Semtech Corporation, "Lora® devices: Smart utilities real world solutions," 2018. [Online]. Available: https://info.semtech.com/smart-utilities-e-book

[2] ABI Research, "Water utilities propelling the next wave of growth for smart meter market," 2019. [Online]. Available: https://www.abiresearch.com/press/global-smart-meter-revenue-reach-us14-billion-2026/

[3] LoRa Alliance, "Why lorawan® is the logical choice for asset-tracking connectivity?" 2020. [Online]. Available: https://lora-alliance.org/wp-content/uploads/2020/11/WhitePaper_AssetTracking.pdf

[4] LoRa Alliance, "Lorawan™ 1.1 specification," 2020. [Online]. Available: https://lora-alliance.org/sites/default/files/2020-06/rp_2-1.0.1.pdf

[5] Semtech Corporation, "Lorawan mobile applications: Blind adr," 2019. [Online]. Available: https://lora-developers.semtech.com/uploads/documents/files/LoRaWAN_Mobile_Apps-Blin_ADR_Downloadable.pdf

[6] A. Farhad, D.-H. Kim, S. Subedi, and J.-Y. Pyun, "Enhanced lorawan adaptive data rate for mobile internet of things devices," *Sensors*, vol. 20, no. 22, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/22/6466

[7] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[8] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and iot networks: Potentials, current solutions, and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 1251–1275, 2020.

[9] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[10] A. Farhad, D.-H. Kim, and J.-Y. Pyun, "R-ARM: Retransmission-Assisted Resource Management in LoRaWAN for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7347–7361, 2022.

[11] J. Finnegan, R. Farrell, and S. Brown, "Analysis and enhancement of the lorawan adaptive data rate scheme," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7171–7180, 2020.

[12] V. Moysiadis, T. Lagkas, V. Argyriou, A. Sarigiannidis, I. D. Moscholios, and P. Sarigiannidis, "Extending adr mechanism for lora enabled mobile end-devices," *Simulation Modelling Practice and Theory*, vol. 113, p. 102388, 2021.

[13] M. A. Lodhi, L. Wang, and A. Farhad, "ND-ADR: Nondestructive adaptive data rate for lorawan internet of things," *International Journal of Communication Systems*, p. e5136, 2022.

[14] C. Jiang, Y. Yang, X. Chen, J. Liao, W. Song, and X. Zhang, "A new-dynamic adaptive data rate algorithm of lorawan in harsh environment," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8989–9001, 2022.

[15] T. Hossain, Y. Doi, T. Tazin, M. A. R. Ahad, and S. Inoue, "Study of lorawan technology for activity recognition," in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, 2018, pp. 1449–1453.

[16] C. A. Gomez, A. Shami, and X. Wang, "Machine learning aided scheme for load balancing in dense iot networks," *Sensors*, vol. 18, no. 11, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3779

[17] J. Cho, D. Hwang, and K.-H. Kim, "Improving tdoa based positioning accuracy using machine learning in a lorawan environment," in *2019 International Conference on Information Networking (ICOIN)*. IEEE, 2019, pp. 469–472.

[18] T. Yatagan and S. Oktug, "Smart spreading factor assignment for lorawans," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1–7.

[19] N. Aihara, K. Adachi, O. Takyu, M. Ohta, and T. Fujii, "Q-learning aided resource allocation and environment recognition in lorawan with csma/ca," *IEEE Access*, vol. 7, pp. 152 126–152 137, 2019.

[20] I. Ilahi, M. Usama, M. O. Farooq, M. U. Janjua, and J. Qadir, "Loradrl: Deep reinforcement learning based adaptive phy layer transmission parameters selection for lorawan," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*. IEEE, 2020, pp. 457–460.

[21] S. Cui and I. Joe, "Collision prediction for a low power wide area network using deep learning methods," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 205–214, 2020.

[22] M. Guerrero, C. Cano, X. Vilajosana, and P. Thubert, "Towards dependable iot via interface selection: Predicting packet delivery at the end node in lorawan networks," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2707

[23] F. Cuomo, D. Garlisi, A. Martino, and A. Martino, "Predicting lorawan behavior: How machine learning can help," *Computers*, vol. 9, no. 3, 2020. [Online]. Available: https://www.mdpi.com/2073-431X/9/3/60

[24] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of lora networks in a smart city scenario," in *IEEE International Conference on communications (ICC), May 2017*. IEEE, 2017, pp. 1–7.

[25] 3GPP, "Cellular system support for ulta-low complexity and low throughput internet of things (ciot)," 2016. [Online]. Available: https://itectec.com/archive/3gpp-specification-tr-45-820/

[26] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in lorawan: Pitfalls and countermeasures," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), Capri, Italy, 20-22 June*. IEEE, 2018, pp. 1–7.

[27] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of lorawan performance under different parameter settings," *IEEE Internet of Things Journal*, 2019.

[28] A. Farhad, D.-H. Kim, J.-S. Yoon, and J.-Y. Pyun, "Deep learning-based channel adaptive resource allocation in lorawan," in *2022 International Conference on Electronics, Information, and Communication (ICEIC)*, 2022, pp. 1–5.

[29] G. Shen, J. Zhang, A. Marshall, L. Peng, and X. Wang, "Radio frequency fingerprint identification for lora using deep learning," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2604–2616, 2021.

[30] K. Anwar, T. Rahman, A. Zeb, Y. Saeed, M. A. Khan, I. Khan, S. Ahmad, A. E. Abdelgawad, and M. Abdollahian, "Improving the convergence period of adaptive data rate in a long range wide area network for the internet of things devices," *Energies*, vol. 14, no. 18, 2021. [Online]. Available: https://www.mdpi.com/1996-1073/14/18/5614

[31] Semtech Corporation, "Real-world lorawan® network capacity for electrical metering applications," 2017. [Online]. Available: https://info.semtech.com/network_capacity_white_paper_download

[32] ——, "Smart supply chain and logistics," 2017. [Online]. Available: https://www.semtech.com/uploads/technology/LoRa/app-briefs/Semtech_SupChain_AssetTracking-Airport_AppBrief-FINAL.pdf

[33] GSMA-3GPP, "3gpp low power wide area technologies," 2016. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2016/10/3GPP-Low-Power-Wide-Area-Technologies-GSMA-White-Paper.pdf

[34] A. R. Bessa, M. S. Santos, J. E. Linhares, M. D. Valadão, D. A. Amoedo, E. V. Mattos, A. M. Pereira, L. M. Torres, M. Valney, A. M. Santos *et al.*, "Design. implementation and evaluation of a private lorawan network for industrial internet of things (iiot) applications," in *2022 IEEE International Conference on Consumer Electronics-Taiwan*. IEEE, 2022, pp. 513–514.

[35] T. Attia, M. Heusse, and A. Duda, "Message in message for improved lorawan capacity," in *2021 International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2021, pp. 1–9.

[36] V. Ribeiro, R. Holanda, A. Ramos, and J. J. Rodrigues, "Enhancing key management in lorawan with permissioned blockchain," *Sensors*, vol. 20, no. 11, p. 3068, 2020.

[37] C. Caillouet, M. Heusse, and F. Rousseau, "Optimal sf allocation in lorawan considering physical capture and imperfect orthogonality," in *GLOBECOM 2019 - IEEE Global Communications Conference, Dec 2019*, 2019, pp. 1–8.

[38] J. Finnegan, K. Niotaki, and S. Brown, "Exploring the boundaries of ambient RF energy harvesting with lorawan," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5736–5743, 2021.

Arshad Farhad received his B.S. degree in Information Technology from the University of Peshawar, Pakistan, an M.S. degree in Telecommunication and Networks from Bahria University, Pakistan, and a Ph.D. degree in Information and Communication Engineering from Chosun University, Gwangju, Korea, in 2012, 2015, and 2022, respectively. He is currently an Assistant Professor and affiliated with the Wireless and Mobile Communication Systems Laboratory at the Department of Information and Communication Engineering. His current research endeavors revolve around implementing machine and deep learning techniques to manage resources at the Medium Access Control (MAC) layer in the Internet of Things (IoT) protocols.

Jae-Young Pyun (Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Chosun University, Gwangju, South Korea, in 1997 and 1999, respectively, and the Ph.D. degree in electronics engineering from Korea University, Seoul, South Korea, in 2003. From 2003 to 2004, he was with Samsung Electronics, Suwon, South Korea, where he was involved in research and development of mobile phone communication systems. In 2004, he joined the Department of Information and Communication Engineering, Chosun University, where he is currently a Professor. He has conducted numerous research projects in the fields of IoT protocol and IoT applications. He has contributed 210 articles, including more than 70 international journals. He also holds more than 25 patents on wireless communication, IoT protocol, positioning system, IR-UWB security, and multimedia communication. His current research interests include machine learning, deep learning, IoT protocol design, and IoT applications with indoor positioning system, real-time location system, and UWB radar. Prof. Pyun has been a member of IEICE, IEEK, and KICS since 2004.