Received January 17, 2021, accepted February 4, 2021, date of publication February 9, 2021, date of current version March 25, 2021.
Digital Object Identifier 10.1109/ACCESS.2021.3058191

A Seq2Seq Learning Approach for Link Quality Estimation Based on System Metrics in WSNs

WEIWEI MIAO, ZHENGYANG DING, HAO TANG, ZENG ZENG, MINGXUAN ZHANG, AND SHAQIAN ZHANG
Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company, Nanjing 210024, China
Corresponding author: Zeng Zeng (zeng_sgcc@163.com)

ABSTRACT Link quality estimation (LQE) is a fundamental problem in Wireless Sensor Networks (WSNs). LQE is not only a prerequisite for efficient routing but also significantly impacts the energy consumption of sensor nodes. Despite its importance, LQE remains an open problem due to the time-varying nature of WSNs. Existing approaches mainly rely on physical layer measurements to estimate link quality. However, considering the hardware and environment variations, modeling the correlation between physical layer measurements and link quality is a nontrivial task, rendering it difficult to obtain accurate link quality estimations. For example, our study reveals that various packet delivery rates may correspond to the same RSSI value. In this paper, we propose a novel method SeqLQE to predict link quality using system metrics (e.g., radio-on time, number of packets received) rather than physical layer measurements. We systematically design and collect runtime metrics during network operation. We then adopt a Seq2Seq learning-based model to capture the structure of correlation between link quality and system metrics. Through extensive experiments, we show that SeqLQE achieves an MSE error of 0.0226, which is 6 times better than widely used linear models.

INDEX TERMS Wireless sensor networks, link quality estimation, Seq2Seq.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of tens or hundreds of sensor nodes spatially distributed over a wide geographical area. Because of its real-time sensing, low-power wireless transmission, and self-organized networking characteristics, WSNs are widely used in various applications such as environmental monitoring, indoor positioning, target tracking, and health caring [1]–[5].

Despite the successful applications, the long-term and stable operation of WSNs is still challenging. First, sensor nodes are usually deployed in outdoor and complex environments such as forests and factories [1], [6]. The wireless transmission tends to be unstable due to interference, shadowing, and multipath fading. Second, sensor nodes are restricted to stringent power supplies. Since wireless communication consumes an unneglectable amount of energy, retransmissions of data packets will drain the battery rapidly and reduce networks' lifetime.

In recent years, online link quality estimation (LQE) has attracted considerable attention since it has great potential in overcoming these challenges [7]. The benefits of LQE are two-fold. On the one hand, LQE can assist routing protocols to establish routing paths and improve network data yield [8]–[10]. On the other hand, with accurate LQE results, a sensor node can choose to send data packets on a more reliable link, reducing energy consumption caused by retransmissions.

Existing LQE methods mainly rely on physical layer measurements such as Received Signal Strength Indicator (RSSI) and Signal to Noise Ratio (SNR) to predict link quality. In general, they can be classified into three categories: *rule-based*, *statistics-based* and *learning-based* approaches. Rules-based approaches [11]–[13] identify link states with pre-defined rules on input metrics. Statistics-based approaches [14]–[18] aim to model link quality using statistical models such as Kalman filter and Support Vector Machine (SVM). Learning-based approaches [19]–[21] adopt deep learning techniques such as Reinforcement Learning (RL) to model the relationship between link quality and physical layer measurements.

This study is motivated by the deployment of a sensor network which consists of 20 sensor nodes and operates for two months. Through careful study of collected traces,

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

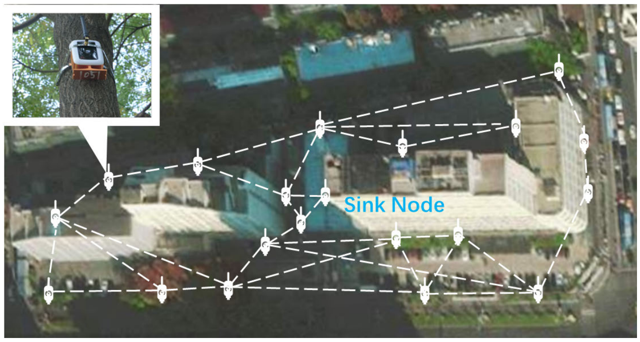


FIGURE 1. System deployment.

we find it is hard to obtain a well-defined correlation between link quality and physical layer measurements over different links. The rationale is, the correlation pattern between link quality and physical layer parameters may be diverse due to hardware-specific variations [22], [23]. To tackle this issue, we propose SeqLQE, a novel approach to predict link quality using system metrics (e.g., radio-on time, number of packets received). SeqLQE investigates the internal structure among system metrics of each sensor node. It uses a Seq2Seq learning model to represent the relationship between link quality and system metrics and predict link quality.

Our main contributions are summarized as follows:

- Unlike previous approaches, SeqLQE does not solely rely on physical layer measurements to predict link quality, considering that the correlation pattern between link quality and physical layer parameters may diverse due to hardware-specific variations [22], [23].
- To the best of our knowledge, SeqLQE is the first to exploit system metrics to identify link quality in WSNs. SeqLQE employs Seq2Seq learning to capture the correlation structure.
- We validate SeqLQE with traces from a sensor network consisting of 20 sensor nodes. The results show that with SeqLQE, the average prediction error is 0.0226 only.

Our proposed approach can be utilized in many protocols such as density-based routing [24], [25], which prefers to select a routing path towards a dense group of sensor nodes in routing decisions. As a result, even if the targeted sensor node becomes unreachable due to battery drain or intermediate link failures, packets can still be re-routed to neighboring sensor nodes. Our method can be used to enhance the robustness of density-based routing. The probability of successfully transmitting a packet can be modeled as a joint function of the target node's density and future link quality. Thus, we can avoid selecting next-hop nodes purely based on density. With our method, the routing strategy can be designed as an optimal trade-off between density and robustness.

The remainder of this paper is organized as follows. We introduce the related work of LQE in Section II. Then in Section III, we present the motivation of this study. The detailed system design is presented in Section IV. Furthermore, in Section V, we conduct comprehensive experiments to evaluate SeqLQE. Finally, we conclude this paper and discuss future work in Section VI.

II. RELATED WORK

In recent years, many efforts have been made on LQE. Based on the prediction methods, existing studies can be classified into the following categories.

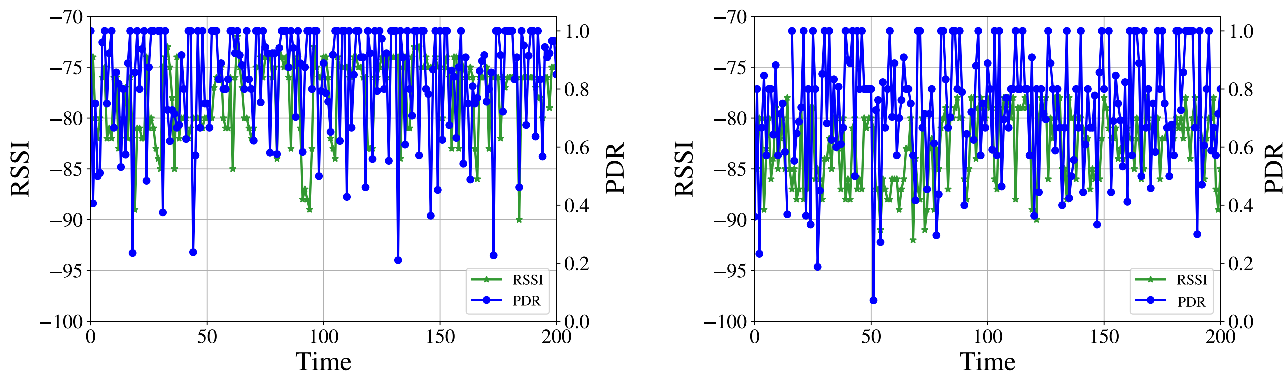
A. RULE-BASED APPROACHES

Four-Bit [11] proposed by Fonseca *et al.* was a rule-based hybrid estimator combining information from the physical, link, and network layers. It provided four bits: the white bit from the physical layer, the ack bit from the link layer, the pin bit, and the compare bit from the network layer. And since TinyOS 2.1, Four-Bit estimator has been adopted in the standard Collection Tree Protocol (CTP) [26]. The Triangle Metric [12] combined the physical measurements geometrically into a robust estimator. In specific, the link quality is estimated by computing the distance of point $(\overline{SNR}_w, \overline{LQI}_w)$ to the origin, where $\overline{SNR}_w$ and $\overline{LQI}_w$ is the sum of SNR and LQI values of received packets divided by the number of transmitted packets respectively. Spuhler *et al.* [27] proposed a preamble-based estimator that exploits information from chip errors in the preamble symbols. The experiments showed that it was at least three times faster than Four-Bit.

In this paper, the proposed method differs from rule-based approaches in that it is a data-driven approach to predict link quality in the near future and requires no pre-defined rules to perform prediction.

B. STATISTICS-BASED APPROACHES

One of the earliest statistics-based approaches was proposed by Lai *et al.* [28]. It presented an experimental study of link quality in WSNs and showed that the relationship between Packet Success Rate (PSR) and SNR could be modeled with a sigmoid function. Then, Son *et al.* [29] found that in the presence of concurrent transmissions, there was a gray region (about 6db) with mixed PSR at the same SNR value. Senel *et al.* [18] proposed a Kalman filter-based estimator to track the change of SNR and predicted PSR with a pre-calibrated SNR-PSR map function. 4C [30] combined RSSI, SNR, LQI, and Packet Reception Ratio (PRR) as input and predicted the probability of successfully delivering the next packet. 4C utilized naive Bayes classifier, logistic regression, and artificial neural networks as prediction models independently. The results showed that logistic regression worked well with small training data. Then, TALENT [14] aimed to predict future link quality instead of current value, since routing protocols can harness predictions of the future to establish low-cost delivery paths. The authors trained a logistic regression classifier using a stochastic gradient descent learning algorithm. TALENT's output is the probability of link quality higher than a pre-defined threshold in the near future.



(a) Node 1003

(b) Node 1026

FIGURE 2. RSSI and PDR values of two randomly selected sensor nodes in 200 time windows, where each time window is 10 minutes.

Similar to statistics-based approaches, SeqLQE is also data-driven. But we employ the deep learning technique, which is more capable of modeling the dynamics of input metrics.

C. LEARNING-BASED APPROACHES

Recently, the success of deep learning in time series analysis has fostered research toward their adoption in solving LQE problems. Sun *et al.* [20] proposed a wavelet-neural-network-based LQE algorithm that decomposed SNR into a time-varying nonlinear part and a non-stationary random part. Link quality predictions are then obtained through the mapping function between SNR and PRR. Luo *et al.* [21] designed a stacked autoencoder-based link quality estimator to extract link features and evaluate the features to divide links into five different quality grades. RL-Probe [19] studied the probing mechanism to estimate link quality by leveraging reinforcement learning.

In this study, we also adopt the deep learning technique to model input metrics and PDR's relationship. But instead of physical layer measurements, we propose to utilize system metrics to predict link quality.

Besides LQE, clustering techniques [31]–[33] also play an important role in achieving energy efficiency in WSNs. For example, the ECH approach [33] minimizes data redundancy and maximizes network lifetime by using a sleeping-waking mechanism for overlapping and neighboring nodes.

III. MOTIVATION

A. DATA COLLECTION

Inspired by GreenOrbs project [34], [35], we deployed 20 telosB sensor nodes in a $215m \times 96m$ area for two months, as shown in Figure 1. Each node is equipped with temperature, humidity, and light sensors. Sensor nodes are either placed on buildings or hang on trees. And the sink node is placed on top of a building. Every 10 minutes, a sensor node reports its sensor readings and run-time information to the sink node.

B. LINK QUALITY METRICS

1) PDR (PACKET DELIVERY RATIO)

PDR is the target we aim to predict in this study. It is obtained from the total number of packets received by the receiver divided by the total number of packets transmitted from the sender. The definition of PDR is as follows:

$$PDR = \frac{\#\ of\ delivered\ packets}{\#\ of\ sent\ packets}.$$

2) RSSI (RECEIVED SIGNAL STRENGTH INDICATOR)

RSSI is a widely used metric to predict PDR. According to IEEE 802.11 standard [36], RSSI is defined as the relative received signal strength of wireless communication links. It indicates the signal power received by the radio and thus reflects the link status. In specific, the wireless communication module in $CC2420$ chip provides the analog value of the received signal strength, which is stored in the chip register, and the digital signal can be obtained through ADC conversion. The value in the register is $RSSI\_VAL$, the empirical deviation value is $RSSI\_OFFSET$, and the power representation of RF is expressed as follows:

$$Power = RSSI + RSSI\_OFFSET(dbm).$$

Through a comprehensive analysis using the collected traces, we find that physical layer measurements such as RSSI and SNR are insufficient to predict link quality in real-world systems. For example, Figure 2 depicts the PDR and RSSI values of two typical sensor nodes in our system. It is easy to find out that the correlation between them is relatively small. We use the Pearson correlation coefficient [37] to measure the linear correlation between PDR and RSSI, and the results are 0.05 and 0.03, respectively. The reason is that the ideal wireless communication model may not fit in real deployed systems. And Figure 3 plots the relationship between RSSI and PDR. We can find that various PDR values may correspond to the same RSSI value. Similar results about SNR can also be found in [29].

TABLE 1. System metrics.

Metric	Description
THL	the time-has-lived counter, incremented by one at each packet forwarding
ETX	the expected transmission count
RadioOnTime	the total radio-on time in milliseconds
ReceivedPacketsCounter	the number of received packets
OverflowDropsCounter	the number of packet dropped because of buffer overflow
SelfTransmitCounter	the number of self-transmitted packets
RetransmitCounter	the number of retransmissions
LoopCounter	the number of routing loops detected
ParentChangeCounter	the number of parent change events
TaskPostCounter	the number of posted task in TinyOS
TaskExecCounter	the number of executed tasks in TinyOS
DuplicateCounter	the number of duplicated packets

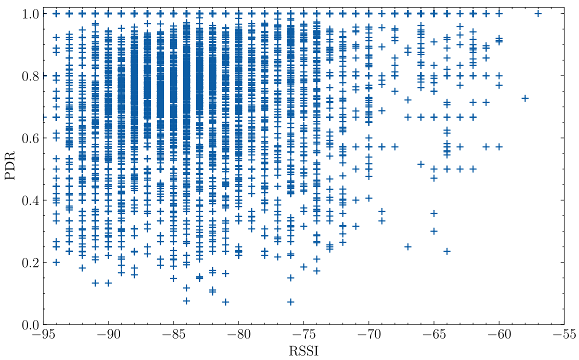


FIGURE 3. RSSI and PDR relationship.

C. SYSTEM METRICS

In this study, we propose to take advantage of run-time information to predict PDR rather than RSSI values. Table 1 summarizes the run-time information we collected. Basically, we investigate the following categories of system metrics:

- Time-related metrics. For example, RadioOnTime measures the cumulative radio-on time during an interval of 10 minutes.
- Traffic-based metrics. For example, ReceivedPacketsCounter records the number of packets received in an interval of 10 minutes, and OverflowDropsCounter denotes the number of packets dropped due to buffer overflow.
- Link-related metrics. For example, THL is the time-has-lived counter, incremented by one at each packet forwarding.
- Task-related metrics. For example, TaskPostCounter records the number of tasks posted by the sensor node.

IV. SYSTEM DESIGN

In this section, we first formally define the link quality monitoring problem in WSNs. Then, we introduce the framework of our solution step by step, as illustrated in Figure 4.

A. PROBLEM FORMULATION

Problem Definition 1: Suppose a wireless sensor network consists of N sensor nodes. In every time window t, a sensor node s_i reports its uplink's packet delivery ratio PDR_i^t, as well as its system metrics $X_i^t = (x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,p}^t)$. The sink node collects all the data packets and obtains dataset $\{(X_i^t, PDR_i^t)|1 \leq i \leq N\}$. The objective is to predict $\{PDR_i^{t+1}|1 \leq i \leq N\}$.

TABLE 2. Definition of symbols.

Symbol	Definition
N	the number of sensor nodes
T	the number of time windows
p	the number of system metrics
s_i	the $i-th$ sensor node, $1 \leq i \leq N$
PDR_i^t	the packet delivery ratio of sensor node s_i in time window t
X_i^t	the vector system metrics of sensor node s_i in time window t
$x_{i,j}^t$	the $j-th$ system metric of sensor node s_i in time window t

B. DATA PREPARATION

We build a dataset $\{(X_i^t, PDR_i^t)|1 \leq i \leq N, 1 \leq t \leq T\}$, where $N = 20$ is the number of sensor nodes and $T = 8640$ is the number of time windows in two months. A data point X_i^t is a 12-dimensional integer-valued vector, each entry of which records a system metric using two bytes. However, due to packet loss and data corruption, we only collected 160,195 valid data packets from the network, which is less than $N * T = 172, 800$. We have to preprocess the data to aid further analysis.

1) MISSING VALUES

There are two types of missing values, *i.e.*, single missing value and continuous missing values. For single missing values, we use the linear interpolation method to fill missing data. For continuous missing values, we choose to delete these incomplete data from the dataset. In our dataset, we delete 174 data points in total. After imputation of missing values, the size of the dataset becomes 172, 626.

2) NORMALIZATION

For prediction tasks, data without normalization could cause training failures. We use min-max normalization to rescale

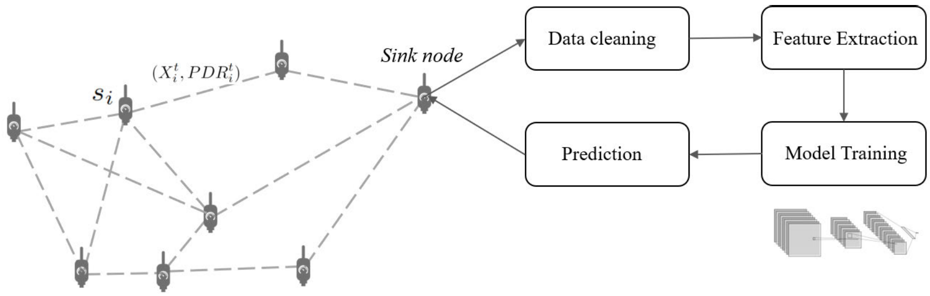


FIGURE 4. System workflow.

each metric into range [0, 1] as follows,

$$X_{i,j}^t \leftarrow \frac{X_{i,j}^t - min_{i,t}X_{i,j}^t}{max_{i,t}X_{i,j}^t - min_{i,t}X_{i,j}^t}.$$

C. FEATURE EXTRACTION

Using system metrics to predict PDR is a non-trivial task. We first study the correlation between these metrics and PDR and find that none of these metrics has a high correlation score with PDR. As shown in Figure 5, the most correlated metric with PDR is RetransmitCounter, with a correlation score of -0.26 only. For other metrics, most of the correlations scores are within $[-0.05, 0.05]$.

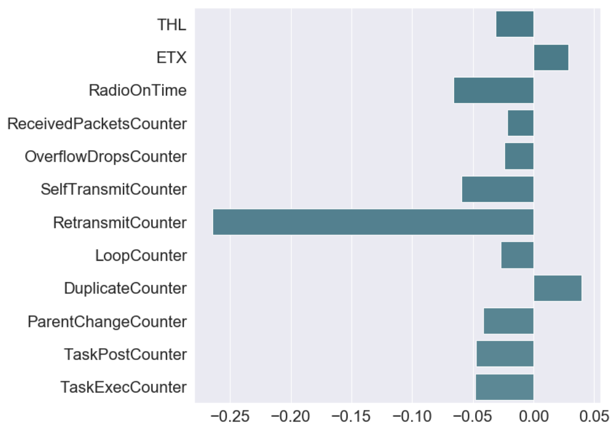


FIGURE 5. Correlations with PDR.

We build the input dataset for prediction as follows:

- First, we split the original dataset into $T - m + 1$ sequences, where $T = 8640$ is the number of time windows, and m is the size of each sequence.
- Then, for each sequence $D_i = [X^i, X^{i+1}, \ldots, X^{i+m-1}]$, we use it to predict the PDRs in the $(i + m)$-th time window, *i.e.*, $y_i = [PDR_1^{i+m}, PDR_2^{i+m} \ldots, PDR_N^{i+m}]$.
- Thus, we obtain a dataset of sequences, denoted as (D, y). We split the dataset into training and testing datasets. In specific, the first 80% is used for training and the rest 20% is used for testing, denoted as (D_{train}, y_{train}) and (D_{test}, y_{test}) respectively.

D. PREDICTION MODEL

In recent years, sequence to sequence learning [38] has been proven successful on a variety of tasks such as machine translation, image captioning, video summarization, and *etc.* [39]–[41]. Since our target is to predict PDR sequences with sequence inputs, we adopt the Seq2Seq model [38] to build the PDR prediction framework in this study.

As illustrated in Figure 6, the model consists of both an encoder and a decoder. Our solution's core idea is first to use the encoder to capture the internal structure of system metrics and then make the decoder generate PDR predictions based on these hidden representations. Different from the original seq2seq model where the encoder and decoder are implemented by LSTM [42], we use Gated Recurrent Unit (GRU) instead. The benefit is, compared to the LSTM cell, a GRU has fewer parameters and thus trains faster. As illustrated in Figure 7, in the GRU block, let h_t and $\hat{h}_t$ denote the activation and candidate activation at time step t respectively, and h_{t-1} is the activation at last time step. Given x_t as the input vector, the key formulas of GRU can be written as:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{1}$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{2}$$
$$\hat{h}_t = \phi(W_h x_t + U_h(r_t \odot h_{t-1} + b_h)) \tag{3}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \tag{4}$$

where z_t is the update gate vector which represents how much the activation is updated, r_t is the reset gate vector which determines how to reset a key gate. And operator $\odot$ is Hadamard product, *a.k.a.*, element-wise multiplication. The activations of update and reset gates are logistic sigmoid function $\sigma(\cdot) = sigmoid(\cdot)$, and $\hat{h}_t$ is computed with a hyperbolic tangent function, *i.e.*, $\phi(\cdot) = tanh(\cdot)$. Matrix Ws, Us and vector bs are trainable parameters of the model. Specifically, W_z, W_r, W_h are the weights of feedforward connections, and U_z, U_r, U_h are recurrent weights.

To learn the prediction model, we first feed training dataset (D_{train}, y_{train}) into the model. The encoder network then extracts the hidden representations h from the observations:

$$h = f_e(D; \theta_e),$$

where h captures the internal structures of system metrics and θ_e are model parameters. Then, after fed with h, the decoder

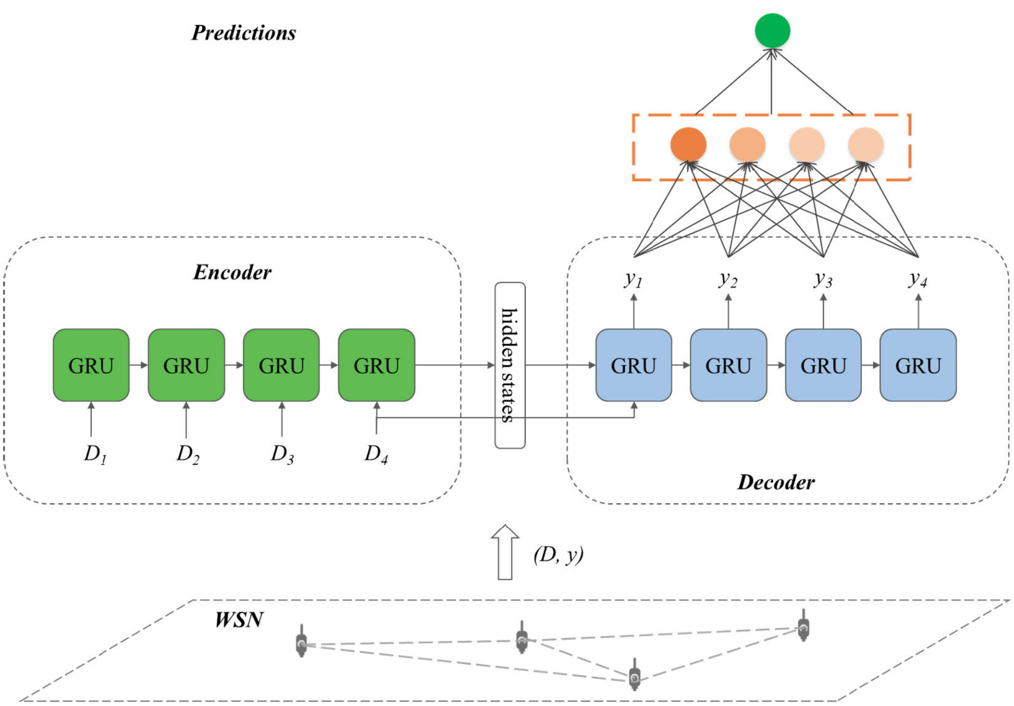


FIGURE 6. The prediction model.

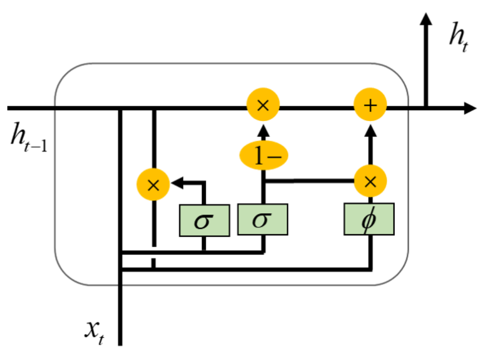


FIGURE 7. A GRU cell.

generates sequential estimation of PDRs:

$$\hat{y} = f_d(h; \theta_d).$$

Consequently, we have:

$$\hat{y} = f_d(f_e(D; \theta_e); \theta_d),$$

where θ_e, θ_d are model parameters. The loss function of the model is defined as the sum of L2 loss of outputs and L2 regularization for model parameters:

$$loss = (y - \hat{y})^2 + \lambda * (\theta_e^2 + \theta_d^2)$$

The learning algorithm is illustrated in 1.

Algorithm 1 Learning Algorithm

Input: $(D_{train}, y_{train}), (D_{test}, y_{test})$
Output: the prediction model

build *model* with a computation graph
initialize all model parameters θ_e, θ_d
repeat
  randomly draw a batch $\mathcal{B}$ from (D_{train}, y_{train})
  feed *model* with $\mathcal{B}$ and obtain predictions $\hat{y}$
  $loss = (y - \hat{y})^2 + \lambda * (\theta_e^2 + \theta_d^2)$
  compute the gradient of *loss*
  back propagate and update θ_e, θ_d
  feed *model* with testing data and compute the loss
  save *model*
until *model* converges
return *model*

V. EXPERIMENT AND ANALYSIS

A. EXPERIMENTAL SETUP

In this paper, we built a sensor network system to perform environmental monitoring, as shown in Figure 1. The base station is a GPU server running Ubuntu 17.10 with an i7 CPU, 64 GB RAM, and NVIDIA GTX 1080Ti. We use Tensorflow [43] to train the prediction model and perform online inference. TinyOS [44] is adopted as the embedded operating system for sensor nodes, and we modified the nesC program to report sensors nodes' internal states and sensing data. In order to prolong the lifetime of the network, the sensor nodes operate at low-duty-cycle mode and wake up periodically to sense the environment and transfer data packets. After finishing these tasks, the sensor nodes switch to a sleeping state. In particular, we set the working period to be 10 minutes for each sensor.

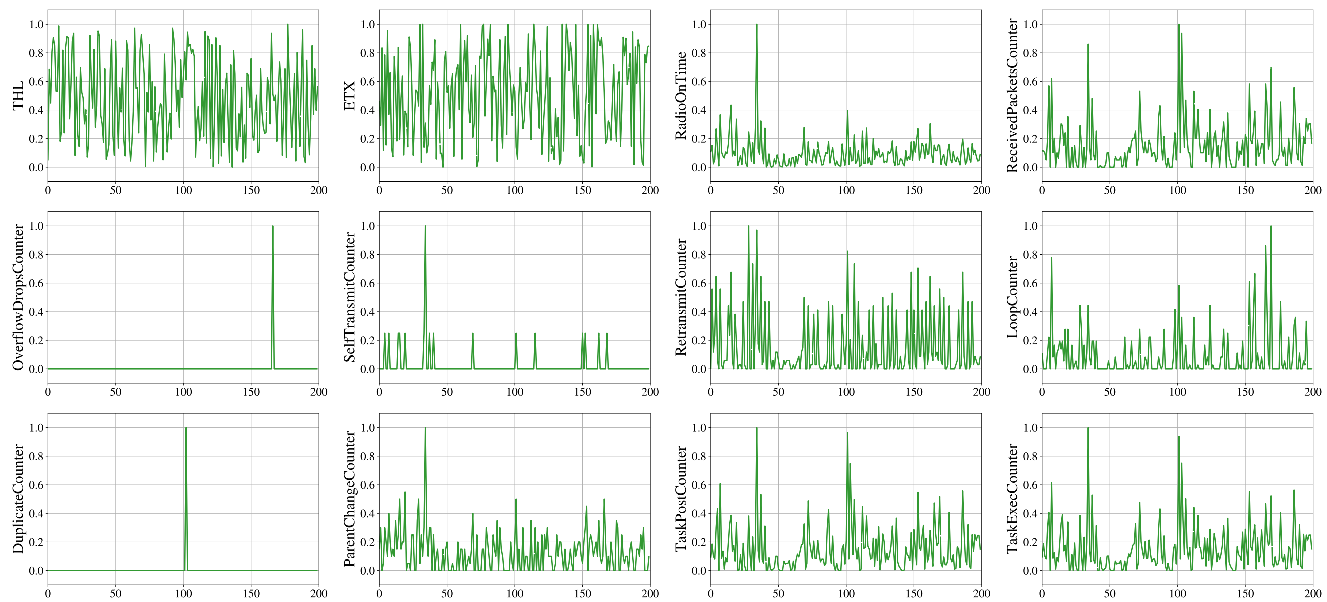


FIGURE 8. System metrics after preprocessing.

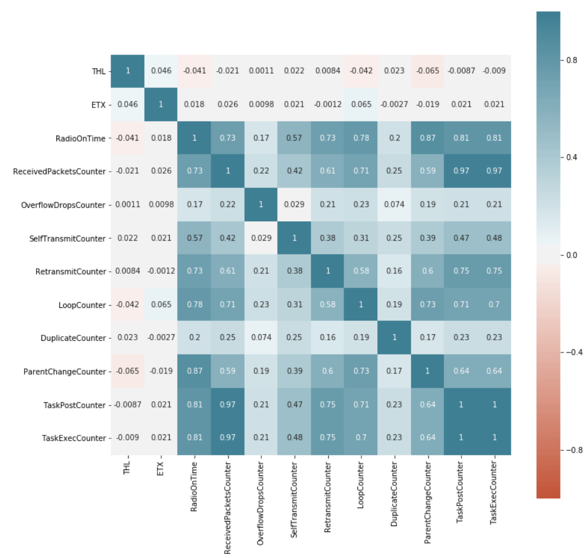


FIGURE 9. Pairwise correlations among system metrics.

It is worth noticing that SeqLQE does not rely on any specific network structure or working mode, such as duty-cycling. It only requires each sensor node to periodically report its system metrics to the base station. Thus, it also applies to cluster-based or hop-by-hop WSNs. In this study, we choose to adopt the duty-cycling mode because it helps to extend the deployed network's lifetime to two months without replacing any sensor node.

B. DATA ANALYSIS

We then preprocess data according to the method described in Section IV-B. A piece of preprocessed data from sensor node 1023 is displayed in Figure 8.

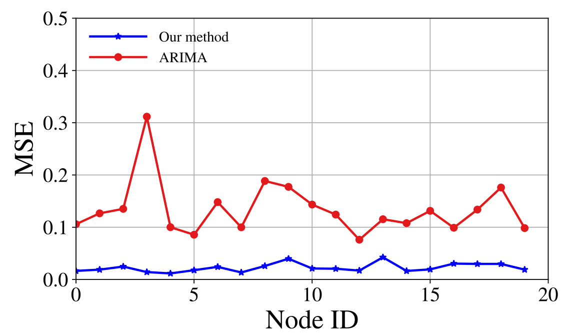


FIGURE 10. Comparison of prediction errors.

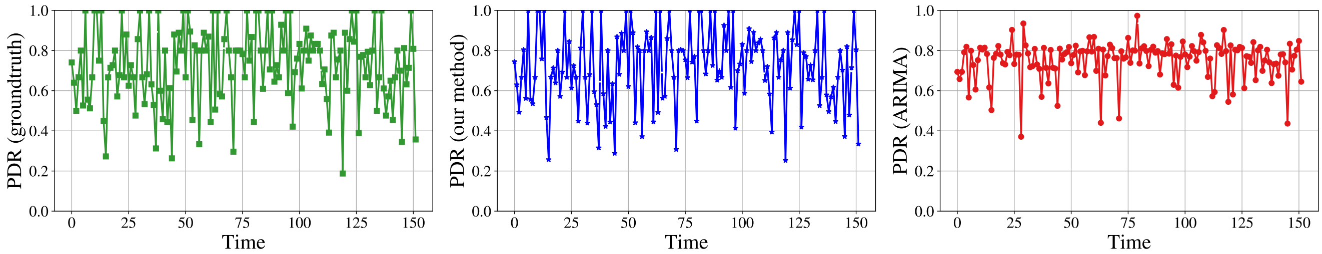
We also investigate the pairwise correlations among system metrics and plot the results in Figure 9. We can find that some metrics are closely correlated (with a correlation score higher than 0.75), such as ReceiveCounter and TaskPostCounter. The rationale is that every time a sensor node receives a task, it has to perform certain tasks in TinyOS, *e.g.*, turning on the radio, validating packets and *etc.*. As a result, these two metrics are positively related. Besides, we can also find that metrics like THL are not correlated with others.

C. EXPERIMENT RESULTS

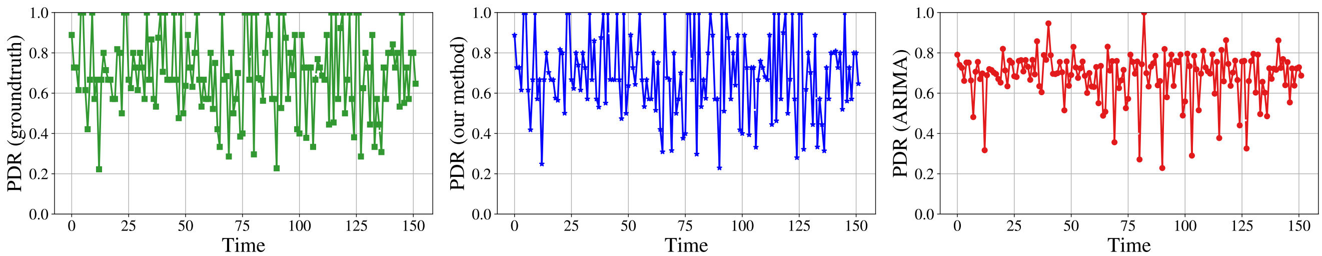
In this paper, we use mean squared error (MSE) as the major evaluation metric to measure the prediction error.

$$MSE = \frac{1}{N}(y_t - \hat{y}_t)^2,$$

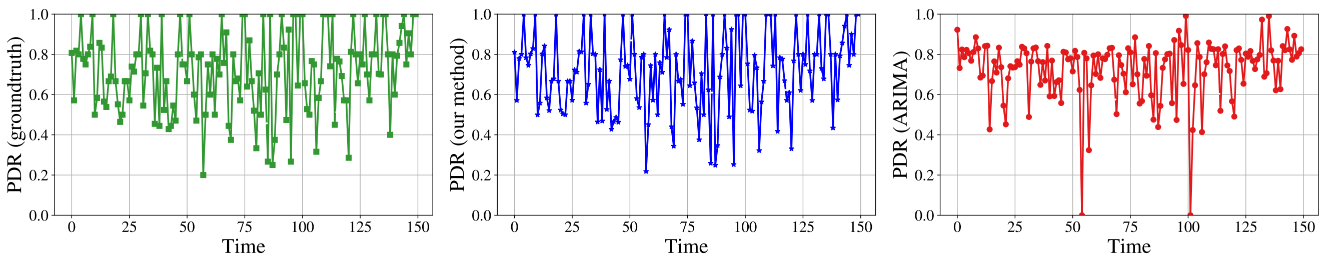
where N is the number of sensor nodes, y_t is the actual vector of PDRs of size $N \times 1$ in time window t, $\hat{y}_t$ is the prediction vector.



(a) Node 1006



(b) Node 1018



(c) Node 1022

FIGURE 11. Comparison of groundtruth, SeqLQE and ARIMA.

We compare our proposed solution with the widely adopted time series prediction model ARIMA [45]. As shown in Figure 10, we can see that for all the 20 sensor nodes, the MSE error of our method is less than ARIMA. And the average MSE of our method is 0.0226, while the average MSE of ARIMA is 0.1342.

In Figure 11, we plot the PDR time series of ground-truth, PDR prediction results by our method, and PDR prediction results by ARIMA. The three sensor nodes are randomly selected from the dataset. Compared with ARIMA, our method's prediction results are more similar to the ground truth time series and have fewer outliers.

D. TIME COMPLEXITY ANALYSIS

The proposed approach consists of three phases: (a) data processing; (b) model training; (c) PDR prediction. In phase (a), all packets received at the base station are preprocessed, and features are extracted. Thus, the running time is linear with the number of data packets, *i.e.*, $O(N \cdot T)$, where N is the number of sensor nodes in the network, and T is the number of time windows. In phase (b), since the computational complexity of learning per weight per time step with stochastic gradient descent (SGD) is $O(1)$ [46], the computation cost of each step is $O(T \cdot |W|)$, where $|W|$ is the number of weights (parameters) in the learning model. Considering that the size of training dataset $\mathcal{D}_{train}$ is $O(T)$, the running time of this phase is thus $O(T \cdot |W|)$. Similarly, the time complexity of prediction in phase (c) is $O(|W|)$. Therefore, the overall time complexity of SeqLQE is $O(N \cdot T + T \cdot |W|)$.

VI. CONCLUSION AND FUTURE WORK

The LQE problem in WSNs [5] has attracted much attention in recent years. Precise LQE results can assist routing protocols in establishing low-cost delivery paths and reducing sensor nodes' energy consumption. Despite its importance, LQE is still an open problem. The reason is, due to hardware and environment variations, the correlation between physical layer measurements and link quality is difficult to model accurately. In this paper, we aim to predict link quality using system metrics rather than physical layer measurements.

To this end, we propose to adopt the Seq2Seq model to capture the internal structure of system metrics. Through extensive experiments, we show that SeqLQE achieves an MSE error of 0.0226. This result is 6 times better than widely used linear models.

Despite the accuracy of SeqLQE, its time complexity is higher than linear models as analyzed in Section V-D. To accelerate training, we currently perform all the computation tasks in the base station with a modern GPU. A possible improvement is first training an offline model with historical data and then online updating the model. As analyzed previously, the time complexity of updating is $O(|W|)$. In the future, we will also explore lightweight methods that can perform prediction in the network.

REFERENCES

[1] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, "Does wireless sensor network scale? A measurement study on GreenOrbs," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 1983–1993, Oct. 2013, doi: 10.1109/TPDS.2012.216.

[2] X. Mao, X. Miao, Y. He, X.-Y. Li, and Y. Liu, "CitySee: Urban CO_2 monitoring with sensors," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 1611–1619, doi: 10.1109/INFCOM.2012.6195530.

[3] A. Maddumabandara, H. Leung, and M. Liu, "Experimental evaluation of indoor localization using wireless sensor networks," *IEEE Sensors J.*, vol. 15, no. 9, pp. 5228–5237, Sep. 2015.

[4] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous WSN for healthcare: Recent advances and future prospects," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 311–318, Aug. 2014, doi: 10.1109/JIOT.2014.2329462.

[5] A. Saifullah, M. Rahman, D. Ismail, C. Lu, R. Chandra, and J. Liu, "SNOW: Sensor network over white spaces," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. CD-ROM*, Nov. 2016, pp. 272–285.

[6] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi, "SWATS: Wireless sensor networks for steamflood and waterflood pipeline monitoring," *IEEE Netw.*, vol. 25, no. 1, pp. 50–56, Jan. 2011, doi: 10.1109/MNET.2011.5687953.

[7] G. Cerar, H. Yetgin, M. Mohorcic, and C. Fortuna, "Machine learning for wireless link quality estimation: A survey," 2018, *arXiv:1812.08856*. [Online]. Available: http://arxiv.org/abs/1812.08856

[8] A. Yadav, Y. N. Singh, and R. R. Singh, "Improving routing performance in AODV with link prediction in mobile adhoc networks," *Wireless Pers. Commun.*, vol. 83, no. 1, pp. 603–618, Jul. 2015, doi: 10.1007/s11277-015-2411-5.

[9] L. Qin and T. Kunz, "Increasing packet delivery ratio in DSR by link prediction," in *Proc. 36th Annu. Hawaii Int. Conf. Syst. Sci.* Big Island, HI, USA: IEEE Computer Society, Jan. 2003, p. 300, doi: 10.1109/HICSS.2003.1174846.

[10] R. Huang, L. Ma, G. Zhai, J. He, X. Chu, and H. Yan, "Resilient routing mechanism for wireless sensor networks with deep learning link reliability prediction," *IEEE Access*, vol. 8, pp. 64857–64872, 2020, doi: 10.1109/ACCESS.2020.2984593.

[11] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "Four-bit wireless link estimation," in *Proc. HotNets*, 2007, pp. 1–7.

[12] C. A. Boano, M. A. Zuniga, T. Voigt, A. Willig, and K. Romer, "The triangle metric: Fast link quality estimation for mobile wireless sensor networks," in *Proc. 19th Int. Conf. Comput. Commun. Netw.*, Aug. 2010, pp. 1–7.

[13] H.-J. Audeoud and M. Heusse, "Quick and efficient link quality estimation in wireless sensors networks," in *Proc. 14th Annu. Conf. Wireless Demand Netw. Syst. Services (WONS)*, Isola, France, Feb. 2018, pp. 87–90, doi: 10.23919/WONS.2018.8311667.

[14] T. Liu and A. Cerpa, "TALENT: Temporal adaptive link estimator with no training," in *Proc. 10th ACM Conf. Embedded Netw. Sensor Syst.*, Toronto, ON, Canada, 2012, pp. 253–266, doi: 10.1145/2426656.2426682.

[15] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Los Angeles, CA, USA, 2003, pp. 14–27, doi: 10.1145/958491.958494.

[16] J. Shu, S. Liu, L. Liu, L. Zhan, and G. Hu, "Research on link quality estimation mechanism for wireless sensor networks based on support vector machine," *Chin. J. Electron.*, vol. 26, no. 2, pp. 377–384, Mar. 2017.

[17] K. Srinivasan, M. A. Kazandjieva, M. Jain, and P. Levis. (2008). *Prr is Not Enough*. [Online]. Available: http://sing.stanford.edu/pubs/sing-08-01.pdf

[18] M. Senel, K. Chintalapudi, D. Lal, A. Keshavarzian, and E. J. Coyle, "A Kalman filter based link quality estimation scheme for wireless sensor networks," in *Proc. IEEE GLOBECOM-IEEE Global Telecommun. Conf.*, Washington, DC, USA, Nov. 2007, pp. 875–880, doi: 10.1109/GLOCOM.2007.169.

[19] E. Ancillotti, C. Vallati, R. Bruno, and E. Mingozzi, "A reinforcement learning-based link quality estimation strategy for RPL and its impact on topology management," *Comput. Commun.*, vol. 112, pp. 1–13, Nov. 2017, doi: 10.1016/j.comcom.2017.08.005.

[20] W. Sun, W. Lu, Q. Li, L. Chen, D. Mu, and X. Yuan, "WNN-LQE: Wavelet-neural-network-based link quality estimation for smart grid WSNs," *IEEE Access*, vol. 5, pp. 12788–12797, 2017, doi: 10.1109/ACCESS.2017.2723360.

[21] X. Luo, L. Liu, J. Shu, and M. Al-Kali, "Link quality estimation method for wireless sensor networks based on stacked autoencoder," *IEEE Access*, vol. 7, pp. 21572–21583, 2019, doi: 10.1109/ACCESS.2019.2898690.

[22] A. Cerpa, N. Busek, and D. Estrin, "SCALE: A tool for simple connectivity assessment in lossy environments," Center Embedded Netw. Syst. (CENS), Univ. California, Los Angeles, CA, USA, CENS Tech. Rep. 0021, Sep. 2003, pp. 1–13.

[23] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proc. 2nd Int. Conf. Mobile Syst., Appl., Services (MobiSYS)*, 2004, pp. 125–138, doi: 10.1145/990064.990081.

[24] V. Lenders, M. May, and B. Plattner, "Density-based anycast: A robust routing strategy for wireless ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 852–863, Aug. 2008.

[25] R.-S. Ko and C.-C. Lai, "Density-based routing mechanism for sensor networks," in *Proc. 10th IEEE Symp. Comput. Commun. (ISCC)*, Dec. 2005, pp. 341–346.

[26] O. Gnawali, R. Fonseca, K. Jamieson, M. A. Kazandjieva, D. Moss, and P. Levis, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM Trans. Sens. Netw.*, vol. 10, no. 1, pp. 16:1–16:49, 2013, doi: 10.1145/2529988.

[27] M. Spuhler, V. Lenders, and D. Giustiniano, "BLITZ: Wireless link quality estimation in the dark," in *Wireless Sensor Networks* (Lecture Notes in Computer Science), vol. 7772. Ghent, Belgium: Springer, 2013, pp. 99–114, doi: 10.1007/978-3-642-36672-7_7.

[28] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian, "Measurement and characterization of link quality metrics in energy constrained wireless sensor networks," in *Proc. GLOBECOM IEEE Global Telecommun. Conf.*, San Francisco, CA, USA, Dec. 2003, pp. 446–452, doi: 10.1109/GLOCOM.2003.1258278.

[29] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst. (SenSys)*, Boulder, CO, USA, 2006, pp. 237–250, doi: 10.1145/1182807.1182831.

[30] T. Liu and A. Cerpa, "Foresee (4C): Wireless link prediction using link features," in *Proc. 10th Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, Chicago, IL, USA, 2011, pp. 294–305. [Online]. Available: http://ieeexplore.ieee.org/document/5779046/

[31] H. El Alami and A. Najid, "Energy-efficient fuzzy logic cluster head selection in wireless sensor networks," in *Proc. Int. Conf. Inf. Technol. Org. Develop. (IT4OD)*, Mar. 2016, pp. 1–7.

[32] J.-S. Lee and C.-L. Teng, "An enhanced hierarchical clustering approach for mobile sensor networks using fuzzy inference systems," *IEEE Internet Things J.*, vol. 4, no. 4, pp. 1095–1103, Aug. 2017, doi: 10.1109/JIOT.2017.2711248.

[33] H. E. Alami and A. Najid, "ECH: An enhanced clustering hierarchy approach to maximize lifetime of wireless sensor networks," *IEEE Access*, vol. 7, pp. 107142–107153, 2019, doi: 10.1109/ACCESS.2019.2933052.

[34] W. Dong, Y. Liu, Y. He, T. Zhu, and C. Chen, "Measurement and analysis on the packet delivery performance in a large-scale sensor network," *IEEE/ACM Trans. Netw.*, vol. 22, no. 6, pp. 1952–1963, Dec. 2014, doi: 10.1109/TNET.2013.2288646.

[35] X. Miao, K. Liu, Y. He, Y. Liu, and D. Papadias, "Agnostic diagnosis: Discovering silent failures in wireless sensor networks," in *Proc. 30th IEEE Int. Conf. Comput. Commun., Joint Conf. IEEE Comput. Commun. Societies*, Shanghai, China, 2011, pp. 1548–1556, doi: 10.1109/INFCOM.2011.5934945.

[36] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ANSI/IEEE Std. 802.11, 1999 edition, ANSI/IEEE, Standard, 1999.
[37] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processing*. Berlin, Germany: Springer, 2009, pp. 1–4.
[38] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, pp. 3104–3112. [Online]. Available: http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks
[39] K. Zhang, K. Grauman, and F. Sha, "Retrospective encoders for video summarization," in *Computer Vision—ECCV 2018* (Lecture Notes in Computer Science), vol. 11212. Munich, Germany: Springer, 2018, pp. 391–408, doi: 10.1007/978-3-030-01237-3_24.
[40] C. Liu, F. Sun, C. Wang, F. Wang, and A. Yuille, "MAT: A multimodal attentive translator for image captioning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Melbourne, VIC, Australia, Aug. 2017, pp. 4033–4039, doi: 10.24963/ijcai.2017/563.
[41] J. Lin, X. Sun, X. Ren, S. Ma, J. Su, and Q. Su, "Deconvolution-based global decoding for neural machine translation," in *Proc. 27th Int. Conf. Comput. Linguistics (COLING)*, Santa Fe, NM, USA: Association Computational Linguistics, 2018, pp. 3260–3271. [Online]. Available: https://www.aclweb.org/anthology/C18-1276/
[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, and M. Kudlur, "Tensorflow: A system for large-scale machine learning," in *Proc. 12th USENIX Symp. Operating Syst. Design Implement.* Savannah, GA, USA: USENIX Association, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi
[44] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. L. Hill, M. Welsh, E. A. Brewer, and D. E. Culler, "Tinyos: An operating system for sensor networks," in *Ambient Intelligence*. Berlin, Germany: Springer, 2005, pp. 115–148, doi: 10.1007/3-540-27139-2_7.
[45] J. D. Hamilton, *Time Series Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 2020.
[46] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *CoRR*, vol. abs/1402.1128, Feb. 2014. [Online]. Available: http://arxiv.org/abs/1402.1128

WEIWEI MIAO received the M.S. degree in automation theory and application from Southeast University, Nanjing, China, in 1992.

He is currently a Principal Researcher with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company. His research interests include power communication systems, wireless access networks, and communication network management.

ZHENGYANG DING received the B.S. degree in wireless communication and the M.S. degree in communication and electronics from Southeast University, Nanjing, China, in 1992.

He is currently a Senior Engineer with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company.

HAO TANG received the M.S. degree in computer science and technology from Nanjing University, Nanjing, China, in 2015.

He is currently an Assistant Researcher with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company. His research interests include software process improvement and practical software engineering.

ZENG ZENG received the M.S. degree in computer science and technology from Nanjing University, Nanjing, China, in 2016.

He is currently a Senior Researcher with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company. His research interests include the Internet of Things and smart grids.

MINGXUAN ZHANG received the Ph.D. degree in wireless network and mobile communication from Southeast University, Nanjing, China, in 2016.

He is currently a Senior Researcher with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company. His research interests include wireless communication and wireless power networks.

SHAQIAN ZHANG received the M.S. degree in information science and engineering from Southeast University, Nanjing, China, in 2007.

He is currently a Senior Engineer with the Information and Telecommunication Branch, State Grid Jiangsu Electric Power Company. His research interests include communication networks and network security.