IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 12, 2019, accepted January 29, 2019, date of publication February 18, 2019, date of current version March 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899916

An RNN-Based Delay-Guaranteed Monitoring Framework in Underwater Wireless Sensor Networks

XIAOHUI WEI[1,2], YUANYUAN LIU[2], SHANG GAO[2], XINGWANG WANG[2], AND HENGSHAN YUE [2]

[1]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[2]College of Computer Science and Technology, Jilin University, Changchun 130012, China

Corresponding author: Xingwang Wang (xww@jlu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61772228, in part by the National Key Research and Development Program of China under Grant 2017YFC1502306, Grant 2016YFB0201503, and Grant 2016YFB0701101, in part by the Jilin Scientific and Technological Development Program under Grant 20170520066JH and Grant 20190201024JC, and in part by the Graduate Innovation Fund of Jilin University under Grant 101832018C026.

ABSTRACT Real-time underwater monitoring has been widely applied in many applications of underwater wireless sensor networks (UWSNs). Due to the long acoustic communication delays, the real-time data collection in UWSNs is challenging. Moreover, the underwater acoustic transmission faces the problem of high data loss rate, which causes a longer delay time due to the need for packet retransmissions. To address these problems, we propose a recurrent neural network (RNN)-based underwater monitoring framework with the consideration of delay, energy, and data quality. We drop the automatic retransmission mechanism applied in the MAC protocols to reduce the long end-to-end delay and energy cost. Facing high data loss, we propose an efficient RNN learning model, LSTM-Decay, to analyze the raw data with the time-related decay weights features and predict the missing values. The experiments with the real-world underwater sensing datasets show that our learning model can achieve an accurate estimation with different degrees of missing rates and can provide better performance compared with the non-RNN and RNN baselines.

INDEX TERMS Underwater wireless sensor networks, real-time monitoring, missing values, RNN.

I. INTRODUCTION

The demand for real-time monitoring in underwater sensor networks is gradually increasing in various applications, such as monitoring for pollution detection, disaster warning, oil industry, aquaculture [1], [2]. Figure 1 illustrates a typical underwater scenario where sensor nodes are deployed for continuous monitoring. Multiple nodes are deployed underwater to sense data and transmit data by AUVs or relay nodes to sink nodes. Then sink nodes transmit data to the monitoring center by radio links. However, the requirement for real time in UWSNs is challengeable compared with terrestrial wireless networks.

On one hand, the long propagation delay caused by the low sound speed compound the difficulty of real-time data collection [3]. It is known that the propagation speed for an acoustic link is about 1,500 meters/sec, which is five orders of magnitude lower than that of a radio link. There have been plenty of works about dedicated transfer protocols for underwater acoustic networks [4], [5]. In these protocols, retransmissions are used to address the packet error but data retransmissions cause longer delay and extra energy cost. Especially in the multi-hop network, the increased number of hops further aggravates the propagation delay [6]. On the other hand, underwater data transmission is prone to high-frequency data loss due to harsh underwater environments. Figure 1 shows that data transmission may be influenced by several reasons, such as complex ambient noise, packet collision, hardware failure [7]–[9]. Missing values may imply rich information that affects the quality of collected data and cause failure prediction or analysis [10]. Hence, it's essential to handle missing values to assure the desired data quality.

To reduce the end-to-end delay and improve the monitoring quality, we focus on the design of low-latency

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

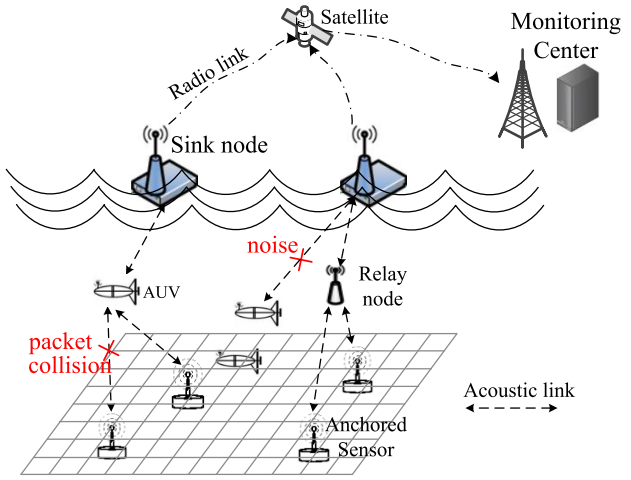


FIGURE 1. Illustration of data transmission in UWSN.

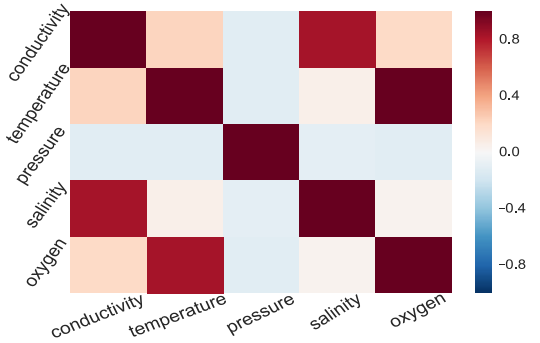


FIGURE 2. Pearson correlation coefficients between parameters of water quality.

monitoring framework while ensuring the quality of collected data. In the framework, we first drop the automatic retransmission mechanism applied in transmission protocols for real-time underwater monitoring. Data loss caused by packet errors is traded off to reduce the long delay and energy consumption. Since higher rates of packet loss will degrade the quality of collected data, we move the compensation for larger data loss to the data center. When faced with varied and high-frequency missing rates in UWSNs, we propose an RNN-based data learning model to efficiently process data loss.

Commonly used imputation methods include smoothing, regression, interpolation, and K-nearest neighbor (KNN) methods. The main idea of these methods is to compute the missing value according to a discrete set of known data points. These methods are efficient for single static datasets but are not suitable for complex data sets. In some cases, the consecutive observation on sensor nodes brings time series data. Moreover, most environmental data are multidimensional and time series. For instance, in environmental monitoring applications, several environmental parameters such as temperature, salinity, conductivity, all need to be collected for water quality measurement. To handle missing data, we need to consider the effects of variable correlations and time series. As shown in Figure 2, the Pearson correlation coefficients between environmental parameters illustrate variable correlations. In feeding monitoring applications, the water quality affects the fish growth and the combinational analysis of environmental parameters can be used to predict the feeding process [2].

As the prevalence of the incorporation of WSNs into the IoT, extensive research explored efficient data processing methods by leveraging the computation capability of the cloud computing services [11], [12]. The feature of multivariate time series inspires us to import Machine Learning (ML) model, Recurrent Neural Networks (RNNs), to capture complex patterns for data imputation and prediction. Instead of two-step imputation-prediction process, we build learning model to make prediction taking missing values into account. With the cloud resource, we use an RNN-based model to train features of missing values with the input. RNNs provide superior ability to exploit the long-term temporal dependencies and variable correlations [13]. The prediction result can further reduce the size of data and the number of data packet transmission so as to achieve energy saving.

RNNs, such as Long Short-Term Memory (LSTM) [14] and Gated Recurrent Unit (GRU) [15], have been widely applied in many applications with time series or sequential data. In our paper, we design a modified LSTM model, named LSTM-Decay to train multivariate environment parameters. To assure the prediction performance, we import a time-related decay weight for missing values. The modified model concatenates time-related missing features with variable correlations to train sensing data. Hence, the learning model cannot only capture correlations between time series data but also exploit the impact of missing values. Considering specific applications in UWSNs, the RNN-based learning model is able to analyze spatial- and variable-related data where data can be unsampled or sampled.

Our **contributions** of this paper are as follows:

(1) We propose a delay-guaranteed underwater monitoring framework. In the framework, the processing of data loss caused by packet collisions is migrated to the data center. Data retransmissions in transfer protocols are dropped to satisfy the constrained delay for real-time monitoring.

(2) Then an RNN-based model, LSTM-Decay, is presented to estimate multivariate time series data with different types of missing values. When considering missing values, we model input variables and import a trainable time-related decay weight for each correlated variable. The proposed RNN-based network both considers the training of spatial and multivariate sensor data collected in UWSNs.

(3) Simulations are conducted to assess the performance of dropping data retransmission. Then experiment results on real-world ocean datasets show that the proposed RNN-based model can provide better performance with different degrees of missing rates compared with Non-RNN and RNN baselines.

The remaining of the paper is organized as follows. We give an overview of our proposed framework and present a delay guaranteed transmission method in Section II. In Section III, We introduce the LSTM structure and present a modified LSTM and discuss data imputation for spatial and multivariate sensor data in UWSNs. The RNN-based data estimation framework is proposed in Section IV. Experimental results are shown in Section V and Section VI discusses related work. We conclude the paper in Section VII.

II. OVERVIEW FOR THE DELAY-GUARANTEED FRAMEWORK

We consider the application scenarios where the delay of data transmission is constrained. The aim of the framework is to make low-latency data collection while ensuring the quality of data. Hence, we consider both the delay time and data quality in the framework. To achieve the low delay time, the designed monitoring network first makes a change based on existing transmission protocols, such as Slotted FAMA [4]. Then we focus on quality-guaranteed data processing with missing values in the data collection terminal.

The efficiency of underwater acoustic communications is constrained by high latency and low transmission quality. In our framework, we first design a delay guaranteed transmission method to reduce the delay time and energy consumption of packet transmission. Based on our observation and investigation, massive data transmissions cause the longer propagation layer. We mainly aim to reduce the extra latency caused by data retransmission applied in networking protocols.

To illustrate the method, we use a commonly used MAC protocol, Slotted FAMA as an example. The protocol requires each packet (RTS, CTS, DATA or ACK) has to be transmitted at the beginning of one slot. An ARQ technique was added to acknowledge the data reception by ACK or NACK packets. When a terminal receives the NACK packet, it must wait long enough for data retransmission and a new ACK or NACK has to be sent. In our method, we drop the automatic retransmission mechanism. The data packet will be sent after receiving a CTS packet and we don't confirm whether the packet was transmitted successfully. In this case, the delay time will be reduced and the extra energy cost of retransmission is avoided.

For a network scenario with multiple hops, the source node transmits packets to the sink node by one or several relay nodes as shown in Figure 1. The increased communication nodes bring more serious packet loss. The simulation results in Section V-B validate the significant improvement on the average end-to-end delay and energy consumption when dropping retransmission in a multi-hop network. When setting a 3-hop network shown in Figure 8(a), the average end-to-end delay is 3.95, 53% less than that with retransmissions. The proposed method can be seen as an improvement on existing communication protocols and it can be used on other protocols including multi-channel protocols [6].

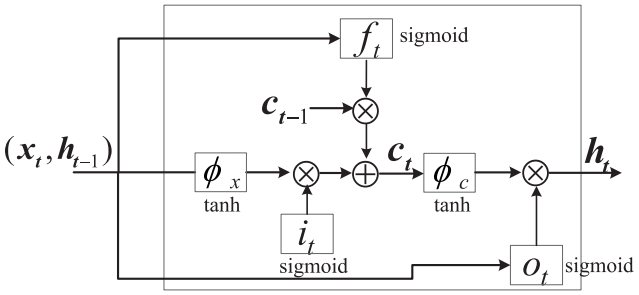


FIGURE 3. The illustration of the general LSTM model.

However, it's obvious that dropping retransmissions may cause high packet loss rates. Then we adopt an RNN-based model that is applicable for missing value estimation even with high missing rates. Owing to the limited power of sensor nodes, more applications move time/resource-consuming tasks to the cloud computing platform that can provide significant computation resources (e.g., CPU, memory). The use of cloud resources motivates us to conduct in-depth data analysis to extract valuable knowledge and provide cost-efficient suggestions. Inspired by clinical data analysis [13], we use the LSTM model to explore how to predict missing underwater data with the quality guarantee.

III. LSTM-BASED DATA PREDICTION WITH MISSING VALUES

In this section, we first introduce the basic learning model, LSTM and modify the model to train the input with missing features.

A. LSTM MODEL

Figure 3 shows a LSTM structure that is equipped with several memory cells. These memory cells are used to store previous experiences to capture long-term time dependencies. A LSTM is a sequence of units that share the same parameters across all time steps [16]. Formally, given a series of sensing data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_T$, $\boldsymbol{x}_t \in \mathbb{R}^D$ represents the t-th vector of all environmental parameters. Here, T denotes the length of variables and D denotes the input dimension. x_t^d denotes the t-th observation for the parameter d. Let $s_t \in \mathbb{R}$ denote the time stamp when $\boldsymbol{x}_t$ is observed. The LSTM model contains three memory cells: input, forget and output gates, denoted as $\boldsymbol{i}, \boldsymbol{f}, \boldsymbol{o}$, respectively. The learning process updates these gates to train datasets. We first introduce how to update functions of LSTM:

$$\boldsymbol{i}_t = \sigma(W_i \boldsymbol{x}_t + U_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \quad (1)$$

$$\tilde{\boldsymbol{c}}_t = \phi(W_c \boldsymbol{x}_t + U_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \quad (2)$$

$$\boldsymbol{f}_t = \sigma(W_f \boldsymbol{x}_t + U_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \quad (3)$$

$$\boldsymbol{o}_t = \sigma(W_o \boldsymbol{x}_t + U_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \quad (4)$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \quad (5)$$

$$\boldsymbol{h}_t = \phi(\boldsymbol{c}_t) \odot \boldsymbol{o}_t \quad (6)$$

where σ represents an element-wise *sigmoid* function that has the value in (0,1), ϕ represents an element-wise

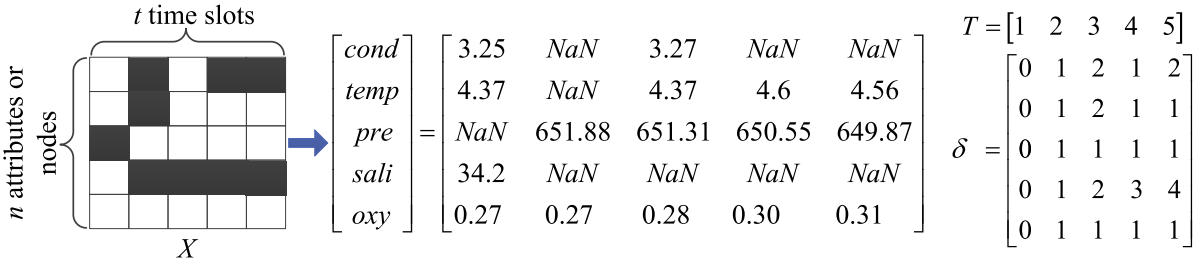


FIGURE 4. An example of data vector (5-dimensional input variables: conductivity, temperature, pressure, salinity and oxygen concentration) and the data instance has different types of loss patterns; time stamp T, Time interval δ.

tanh function and $\odot$ is the Hadamard (element-wise) product. $W_{i,f,o,c} \in \mathbb{R}^{H\times D}$, $U_{i,f,o,c} \in \mathbb{R}^{H\times H}$, are the coefficient matrix for input nodes and hidden states. Vectors $\boldsymbol{b} \in \mathbb{R}^H$ are constant parameters where H refers to the number of hidden units. For each hidden layer, we use the function to train these model parameters for all gates. The input vector $\boldsymbol{x}$ corresponds to environmental parameters and $\boldsymbol{h}_t$ is the output of the hidden layer. In the model, c_{t-1} reflects the effect of time series factor, which denotes the result of the previous moment stored in the memory. Eq.(3) is used to measure the extent of memorizing previous experiences, which is a weight of c_{t-1} in Eq.(5). Eq.(5) estimates the new memory state that is updated based on a weighted previous memory c_{t-1} and the new input $\boldsymbol{x}_t$ with a *tanh* activation.

If to predict environment parameters at each time step, we can stack Softmax layer on top of the last LSTM [17]. The stacked layer uses a soft-max activation function to compute the next value where the hidden layer output $\boldsymbol{h}_i$ is the input: $\hat{\boldsymbol{y}}_{i+1} = softmax(W_p^\top \boldsymbol{h}_i + \boldsymbol{b}_p)$.

B. MODELING INPUT DATA

To address the issue of missing values, we import time-related decay weights for the input variables [13]. Then the weights are trained along with the input.

Figure 4 shows a data instance that records values of five environmental parameters measured by Ocean Networks Canada [18]. The distance depicts several typical data loss patterns in WSNs. The first rows show element frequent loss in row. The second and third rows show element random loss. The fourth row shows successive elements loss in row. These patterns have been detailed in [19]. In real world, data missing always happens as a combination of these loss patterns. When considering the effect of missing data, we import a binary indicator variable $m_t^d \in \{0, 1\}$, where $m_t^d = 1$ if x_t^d is observed and $m_t^d = 0$ denotes x_t^d is missing or noisy. With the indicator variable, we have

$$x_t^d = m_t^d x_t^d + (1 - m_t^d)\hat{x}_t^d \tag{7}$$

where $\hat{x}$ denotes the substituted value. If replaced with the mean value $\bar{x}^d, \hat{x}_t^d = \sum_{t=1}^T m_t^d x_t^d / \sum_{t=1}^T m_t^d$. If replaced with the last measurement $\hat{x}_t$, that is $\hat{x}_t^d = x_{t'}^d$.

Besides, a time-related vector δ_t^d for each variable b stores the time interval from the last observation to t-th record.

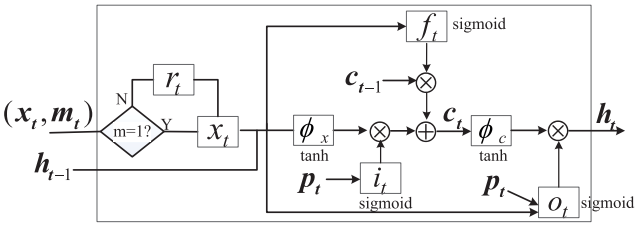


FIGURE 5. The modified LSTM model that considers the preprocessing of input variables with time decaying parameters. Sometimes, to model the influence of external intervention, the modified LSTM can also add the replenishment vector p_t in the gate functions, for instance, adding in the input and output gate.

The continuous data missing will increase the time interval. With the stored time stamps, the time interval can be computed as follow:

$$\delta_t^d = s_t - s_{t-1} + (1 - m_t^d)\delta_{t-1}^d, \quad \delta_1 = 0 \tag{8}$$

where δ in Eq.(8) is to label the effect of time series on missing data. Figure 4 also shows the corresponding time interval matrix of the data instance. The frequent or continuous data loss causes a larger value for δ that reflects the time effect on missing features. For simplify, the straightforward method is to fill in the input with common imputation methods described in Section IV-A. To explore the missing pattern during the training process, we import the decay parameters and modify the LSTM unit as shown in Figure 5. First, we define a vector of decay rate for each variable, denoted as $r_t \in \mathbb{R}^D$. The setting of r_t depends on the value of the time interval. According to Newton's Law of cooling [20], r_t can be expressed as

$$r_t = e^{-(W_r\delta_t + b_r)} \tag{9}$$

Eq.(9) specifies a negative exponential rate to monotonically decrease r_t with the increase of time interval since we consider the influence of input variables is time-decayed. Model parameters a_r, W_r, b_r will be obtained through training. Then, we use the time decaying parameter to concatenate the forward and mean imputation methods.

$$\begin{aligned} \hat{x}_t^d &= r_t^d x_{t'}^d + (1 - r_t^d)\bar{x}^d \\ \Longrightarrow x_t^d &= m_t^d x_t^d + (1 - m_t^d)(r_t^d x_{t'}^d + (1 - r_t^d)\bar{x}^d) \end{aligned} \tag{10}$$

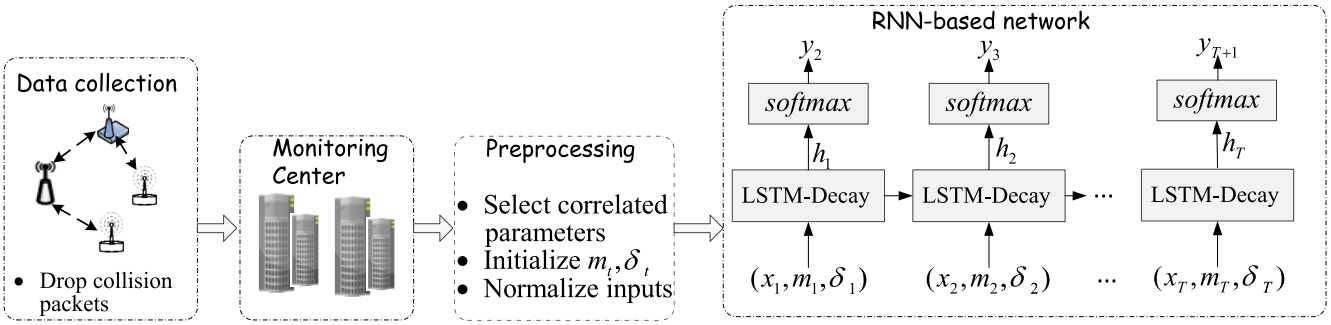


FIGURE 6. The delay-guaranteed monitoring framework with LSTM-Decay.

With the modified input setting, we need to add the indicator variable $\boldsymbol{m}$, the decay rate $\boldsymbol{r_x}$ for the input node. Hence, we can modify the structure of the LSTM unit as shown in Figure 5. The modified LSTM adds time decaying weights into the training model.

In practical scenarios, data may be influenced by some external factors, such as human intervention. During training data, it is required to consider the influence to assure an accurate prediction model. To achieve this, we can quantize the factor and add it to the gate function, which can be implemented by modifying the memory cell. The choice of the modified gates depends on specific scenarios. For instance, in feeding process monitoring, the initial design of LSTM is to model the correlation between environmental parameters and fish feeds. When training the LSTM model, we can add the intervention of feed replenishment as shown in Figure 5. Let us denote a replenishment vector $\boldsymbol{p_t} \in R^H$ that records the information of feed replenishment. When mapped to the model, the vector $\boldsymbol{p_t} \in R^H$ affects the result of input gates and cell states.

IV. RNN-BASED DATA ESTIMATION FOR UWSNs

Combined with the modified LSTM model, we next detail the RNN-based data estimation framework designed for underwater sensor data.

A. RNN-BASED DATA PREDICTION WITH MISSING VALUES

We first introduce several simple imputation methods that can be applied to fill in the missing values for those input sets: zero-, mean- and forward-filling strategies.

- The zero-filling approach is simply to set the missing value $x_t^d = 0$ if x_t^d is missing, but it has not good performance especially when the missing rate is high.
- The mean-filling approach imputes the missing data with the mean estimated by all observations in the training data, denoted as $\bar{x}^d$.
- The forward-filling approach assumes the value of missing data is the same as its previously observed measurement. In this case, we set the missing value $x_t^d = x_{t'}^d$.

We can use LSTM or GRU to train datasets where missing values are filled with these methods. In this case, we do not need to modify the LSTM network architecture. However, simple imputation cannot be well combined with next-step prediction model. The insufficient exploration of missingness greatly affects the performance of the training model, especially for large-scale data loss [10]. In WSN, it's common that collected data from different nodes are spatial or variable correlated [21]. So we can utilize this supplementary information to improve the accuracy of data estimation.

B. RNN-BASED NETWORK ARCHITECTURE

Figure 6 illustrates the process of the delay-guaranteed underwater monitoring framework that includes the modification of data communication and the design of RNN-based learning model. Considering the difference of specific application scenarios, we consider the following two types of data collected in UWSNs:

1) SPATIAL SENSOR DATA

Spatial data corresponds to a scenario where several sensor nodes are deployed in different locations to measure one parameter. For instance, underwater environmental monitoring will deploy multiple temperature probes or turbidity meters at different locations of a region to measure temperature or turbidity. In this case, multiple sensor nodes comprise a sensor network. When to impute missing values, we can leverage the spatial correlation of nodes to build the learning model. Before training, we first select the information of k-nearest neighbors (KNN) as inputs. Assume there are n nodes deployed to measure n sets of temperature values, denoted as $T_1, \cdots, T_n$. Owing to the uncertain of underwater environment mentioned above, it's possible that each node has different degrees of data loss. If we consider to fill T_i, we first select k neighbor nodes with temperature values $T_{i+1}, \cdots, T_{i+k}$. With the proposed model, we can construct $\{T_i, T_{i+1}, \cdots, T_{i+k}\}$ as input variables. The input variables are modeled by Eq.(10). Based on training values, we can predict values for the $k+1$ groups of data. When extending a single node from the entire network, the network can be divided into several clusters using the k-means algorithms. Sensor data in a cluster can be seen as a set of input variables in a training process where we exploit the spatial and temporal relationship among nodes in the same cluster.

2) MULTIVARIATE SENSOR DATA

As we have exemplified above, there are lots of applications that need to collect multivariate data to make synthesis analysis. In this case, we consider attribute-related parameters. Training data with the modified LSTM can use the correlations to exploit the missing patterns for time series prediction. To further reduce the training cost, we can select the most relevant parameters as input variables, which can be implemented by Pearson correlation coefficients. For instance, Figure 2 plots different Pearson correlation coefficients between several representative parameters for water quality. The result shows that pressure values have no significant effect on other parameters. We can train the RNN model with principal correlated parameters: conductivity, temperature, salinity, and oxygen. Before training, the preprocessing is to find the most relevant parameters. Besides, we can also combine the neighbor information with other correlated parameters to provide more comprehensive information for model training.

It's also common that sensing data are collected in a combination of the two discussed data types. Based on the discussion, we propose an RNN-based data estimation network as shown in Figure 6. A preprocessing operation is added before training the LSTM model. For different applications, we can select neighbors or attribute-related parameters as the input. Then we need to model missing values by setting time-related decay weights. Before using LSTM, we also need to preprocess time series data where we need to convert the time series prediction to a supervised learning scheme. The conversion makes preparation for data training. On the top of LSTM models, a softmax function is stacked to predict the next value based on h_i. The instance of water quality monitoring explains how to make the basic RNN model applicable to real scenarios and the same network can be used for other applications with application-specific modification.

Our proposed RNN-based network structure is also suitable for training data that may be compressed on the sender. Since energy saving in UWSNs is significant for reducing resource consumption and extending their lifetime, it's common that the collected data have been preprocessed using some data reduction methods (e.g., sampling) [11], [22]. In this case, the collected data is compressed and we can use the LSTM model to train these sampled datasets.

V. EXPERIMENT

In this section, simulations are conducted to show the efficiency of dropping retransmission. Then we use real datasets to train the proposed RNN-based model.

A. SETTINGS

1) SIMULATION SETTINGS OF UWSN

To assess the performance of dropping the automatic retransmission mechanism, we operate the numerical simulation of the data packet transmission using Java. The underwater network can be seen a grid topology (a $5Km * 5Km$ area) referring to [6]. The transmission range is $1Km$ and the speed of sound in water is $1500m/s$. The data packet size is set to 1024 bits. The control packet size is 50 bits and the bit rate is set to 1000 bits per second. The power consumption in transmitting, receiving state are 10W and 80 mW. Simulations are conducted to compare the change of delay time and energy cost when adding and dropping the retransmission mechanism.

2) LSTM SETTINGS

Then we apply the modified LSTM model to train datasets collected for water quality monitoring, which is a common and important application in underwater sensor networks. With real-world datasets, the performance of LSTM-Decay is evaluated and compared with several common data imputation or prediction methods. We also vary missing rates for different variables to explore the impact of different variables on learning results. The structure of the RNN model refers to Figure 6. Our LSTM training models have 2 hidden layers with 50 LSTM cells, recurrent dropout of 0.5. We train on 60% of data, 20% each for validation and testing. The prediction performance is evaluated using two metrics, RMSE and R-square.

B. ANALYSIS OF DELAY AND ENERGY COST

Figure 7(a) shows the average end-to-end delay while varying the number of sensor nodes. Here we compare the average time of each sensor node spent on sending data packets to its sink node in a 2-hop scenario. The comparison curves validate that abandoning retransmission can effectively reduce the delay time. In our settings, the number of sink nodes increases in proportion to the total node number, which can help to relieve the increase of average delay time. Hence, the average delay will be reduced as the increase of the node number where nodes are denser. Figure 7(b) shows the comparison of the total energy consumption. In a network with 450 nodes, the energy saving is up to 30% when dropping retransmission. The cost difference between retransmission and no retransmission represents the extra communication cost of retransmission caused by packet collisions.

To evaluate the effect of the multi-hop network scenario, we simulate the process of data transmission with increasing hops from single-hop to 4-hop. In the multi-hop network, the source node needs to forward packets to its relay node. Figure 8(a) shows the end-to-end delay time becomes longer as the increase of hops because of the growth of the transfer path. The efficiency of dropping retransmission is more prominent in the multi-hop transmission due to the increase of the data collision probability. From Figure 8(b) we can also see how data retransmission affects the total energy consumption. The simulation results show that we can accelerate data collection by appropriately dropping packet retransmission for those applications with constrained delay requirements.

We also compute the packet delivery ratio (PDR) as shown in Figure 8(a). The lower the packet delivery ratio is, the higher the degree of missing data is. In the setting

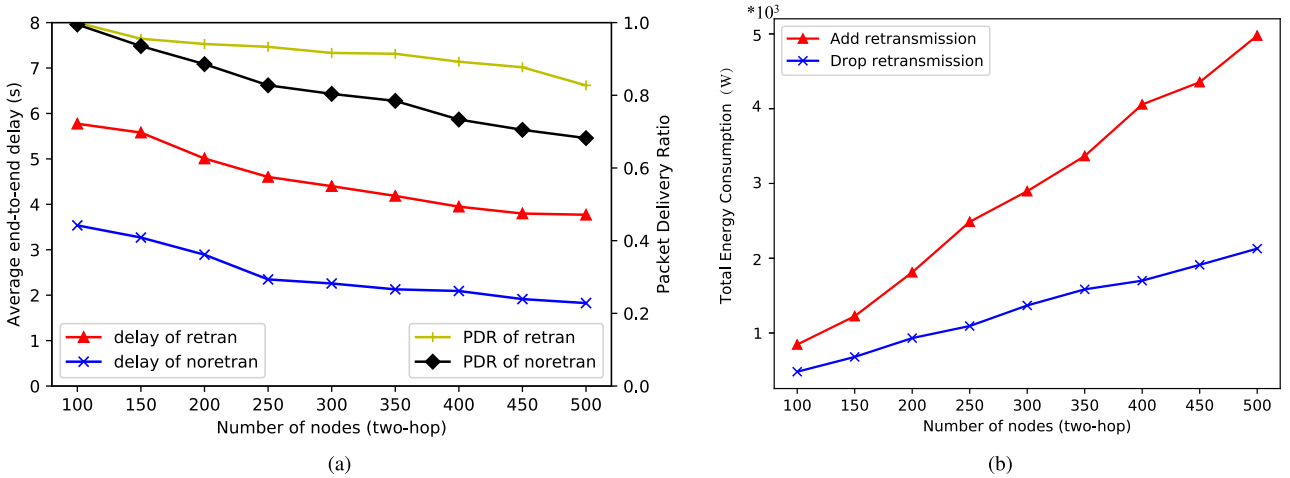


FIGURE 7. Comparison of delay time and energy consumption. (a) Average end-to-end delay. (b) Total energy consumption.

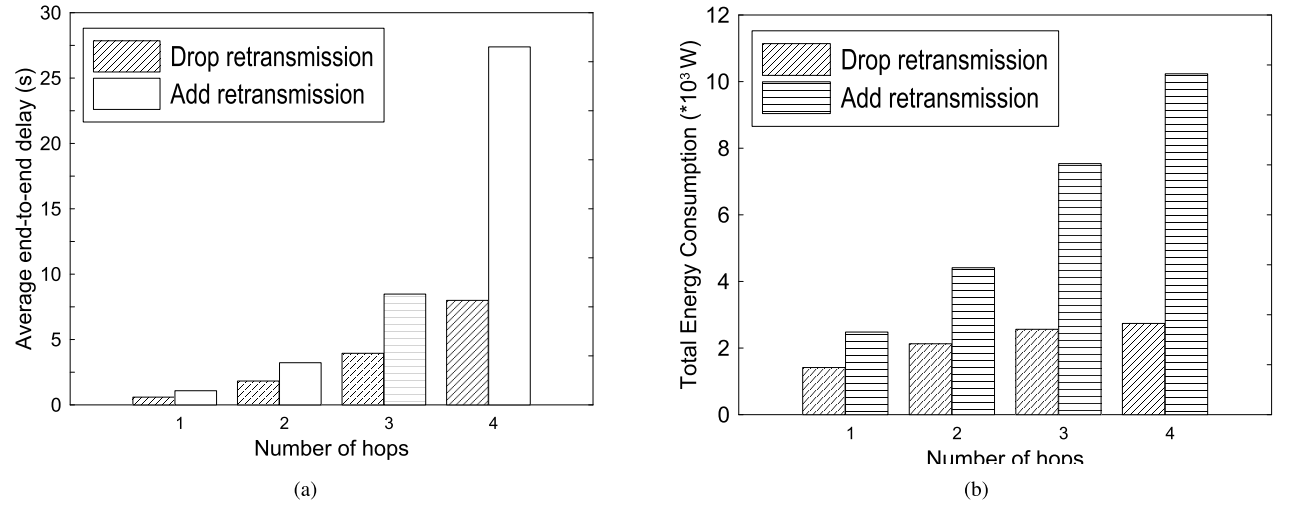


FIGURE 8. Comparison of delay time and energy consumption. (a) Average end-to-end delay. (b) Total energy consumption.

with 250 nodes, the packet delivery ratio with and without retransmission are 0.933 and 0.826, respectively. For these missing data, our proposed RNN-based learning model can assure efficient estimation.

C. PERFORMANCE OF LEARNING MODEL

1) DATA

The real-world datasets used to evaluate the RNN-based learning model come from the Ocean Networks Canada Data Archive [18]. Ocean Networks Canada deploys different types of sensors (e.g., Temperature Sensor, pH sensor) to monitor ocean properties and assess marine environmental conditions. We select datasets collected in Barkley Canyon, Northeast Pacific Ocean. Two types of datasets are used in our experiments:

(1) *Spatial sensor data.* We collect a temperature array measured by temperature probes located at depth 870m (126.1^o W, 48.3^o N). The datasets record temperatures of a region deployed with 8 sensor modules.

(2) *Multivariate sensor data.* We collect datasets measured by CTD devices and oxygen sensors located at depth 643m. The datasets store measurement of several environmental parameters collected from June 2018 to Nov 2018. Environmental parameters include conductivity, temperature, pressure, salinity and oxygen concentration.

Figure 2 plots Pearson correlation coefficients between parameters in the multivariate dataset. We can use the proposed model to learn missing patterns for efficient prediction. Figure 9 shows the results of data statistics for these parameters. In the test, we set different missing rates for different parameters to validate LSTM-Decay can mine the variable relationship with missing data.

2) EVALUATION METRICS

The performance of RNN models for predicting parameter values can be evaluated by two commonly used metrics, RMSE and R^2:

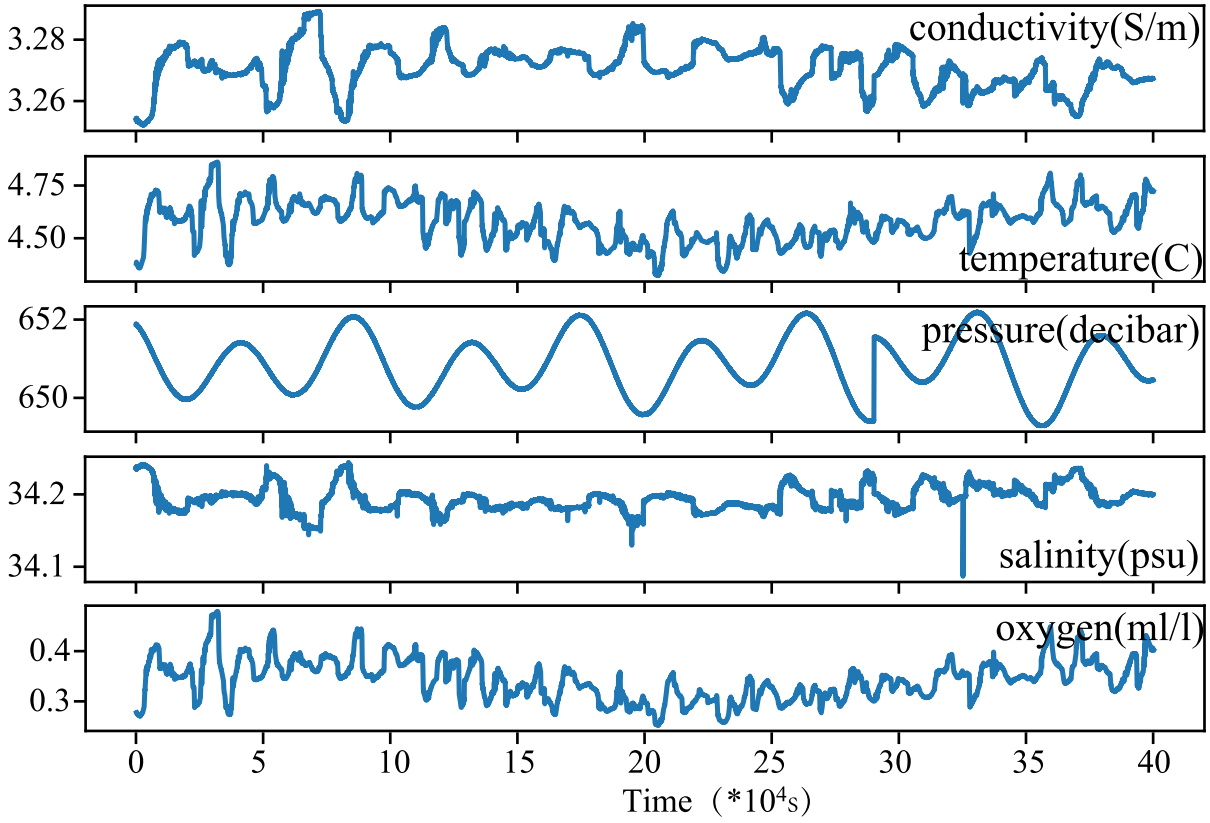


FIGURE 9. Data statistics of 5-dimensional environmental parameters measured for water quality monitoring. Different parameters are labeled as a form of parameter(unit) in each subfigure.

a: ROOT MEAN SQUARED ERROR (RMSE)

RMSE can reflect the magnitude of error. We use it to measure the difference between values predicted by RNNs and observed values. The computation of RMSE is defined as:

$$Loss_{RMSE} = \sqrt{\frac{1}{T}\sum_{i=1}^{i=T}(y_i - \hat{y}_i)^2} \tag{11}$$

b: COEFFICIENT OF DETERMINATION (R^2/R-SQUARE)

R^2 is used to measure how well observed values are replicated by the prediction model.

$$Loss_{R^2} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2} \tag{12}$$

where the closer the value R^2 is to 1, the more accurate the prediction is.

3) EVALUATION RESULTS

To test the performance of LSTM-Decay model, we compare the prediction result with Support Vector Regression (SVR) and several baseline methods introduced in Section IV-A:

- **SVR.** SVR is a popular non-linear regression method employed for time series prediction [23]. In our test, we use SVR with the RBF kernel function.
- **RNN baselines.** We respectively implement LSTM models with zero, mean, forward imputation methods. These methods are named as LSTM-Zero, LSTM-Mean, LSTM-Forward.

Besides, we also compare the performance of SVR and LTSM by combining the KNN imputation method, named SVR-KNN, LSTM-KNN, respectively. The two methods are suitable for spatial sensor data. We evaluate these methods on CTD and temperature datasets. Since these datasets are nearly complete, we need to manually generate datasets with missing values when randomly given a missing rate. To construct missing values, we first set the missing rate of each input parameters as 0.2 to compare the accuracy with baseline methods. The missing rate can also be set other values in the range [0,1] and we will test the impact of different rates on prediction performance.

a: PERFORMANCE COMPARISON WITH BASELINE METHODS

With LSTM-Decay, Figure 10 plots the training and test results for spatial and multivariate sensor data where we show the result of conductivity prediction as a representative of environmental parameters. The result intuitively interprets the proposed model can efficiently predict the change of time series data when modeling inputs with missing values.

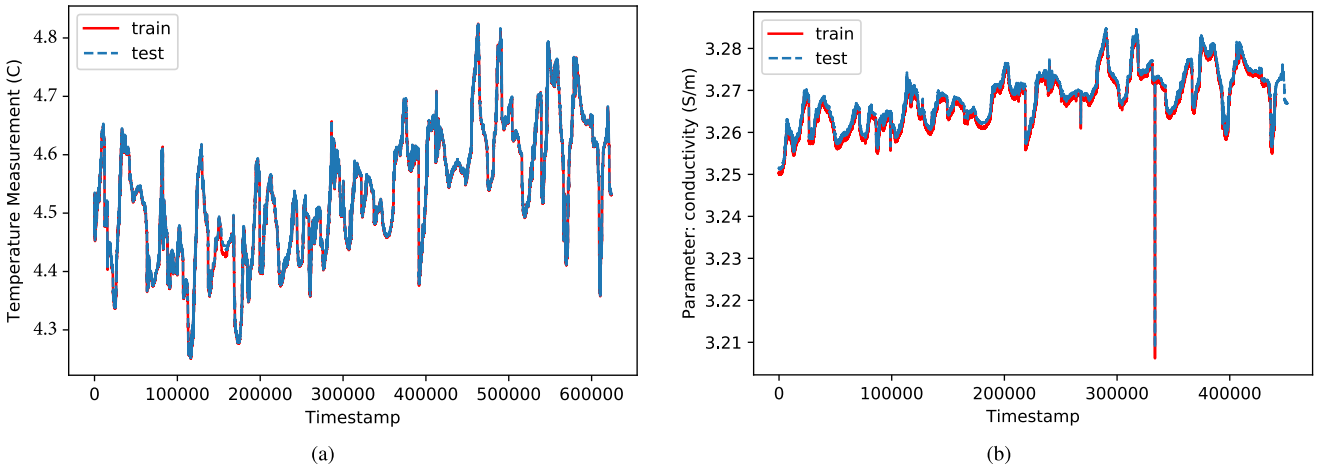


FIGURE 10. Comparison of train and test result with the proposed LSTM-Decay model. (a) Spatial sensor data. (b) Multivariate sensor data.

TABLE 1. Performance comparison of different models by RMSE and R-square.

	Model	RMSE	Model	RMSE
Multivariate sensor data: CTD datasets	LSTM with complete datasets	0.01237	SVR with complete datasets	0.02657
	LSTM-Zero	0.02217	SVR-Zero	0.04692
	LSTM-Mean	0.03147	SVR-Mean	0.04894
	LSTM-Forward	0.02842	SVR-Forward	0.05451
	LSTM-Decay	**0.0197**		
	Model	RMSE	Model	RMSE
Spatial sensor data: Temperature Datasets	LSTM with complete datasets	0.02884	SVR with complete datasets	0.05026
	LSTM-Zero	0.05160	SVR-Zero	0.08323
	LSTM-Mean	0.06819	SVR-Mean	0.08640
	LSTM-Forward	0.04775	SVR-Forward	0.09057
	LSTM-KNN	0.05194	SVR-KNN	0.1223
	LSTM-Decay	**0.03772**		

Table 1 shows the estimation performance using SVR and LSTM models with simple imputation methods (Zero, Mean, Forward), KNN-based imputation, respectively. We also make the prediction on original complete datasets as a comparison. The accuracy estimation with RMSE and R-square indicates that RNN-based models outperform the SVR method. When existing missing values, the advantage of RNN-based methods is more prominent. Moreover, our proposed model, LSTM-Decay, obtains higher accuracy than those LSTM models with simple imputation methods since we consider the time effect on missing values during data training. The training process not only learns correlations between parameters but also exploits the influence of missing values with time steps.

b: PERFORMANCE WHEN VARYING MISSING RATES

We respectively vary the missing rate of each parameter to evaluate the impact of different missing rates on prediction results. The missing rate varies from 0.2 to 0.8 each for an environment parameter. Figure 11 shows the performance comparison of conductivity prediction measured by R^2. The results on different parameters further indicate the superiority of the LSTM-Decay model. When increasing the missing rate from 0.2 to 0.8, the prediction performance of models with simple imputation methods is significantly reduced, especially with high missing rate ($\geq$ 0.6). This is because these imputation methods cannot efficiently fill datasets with high missing rates. The filled data have a large deviation with no-missing datasets, which causes performance degradation of predicting environmental parameters. The prediction performance with LSTM-Decay has a little reduction but is almost stable since it can recognize missing values with our input model. With LSTM-Decay, we can find the prediction accuracy when setting missing rates for salinity is lower than other parameters. When setting the missing rate to 0.6, LSTM-Decay with missing data for salinity has an R^2 of 0.8515 while for other parameters the values of R^2 are 0.8847, 0.8744, 0.8664. The reason is that the correlation between salinity and conductivity is highest, which help the training process to mine variable correlations.

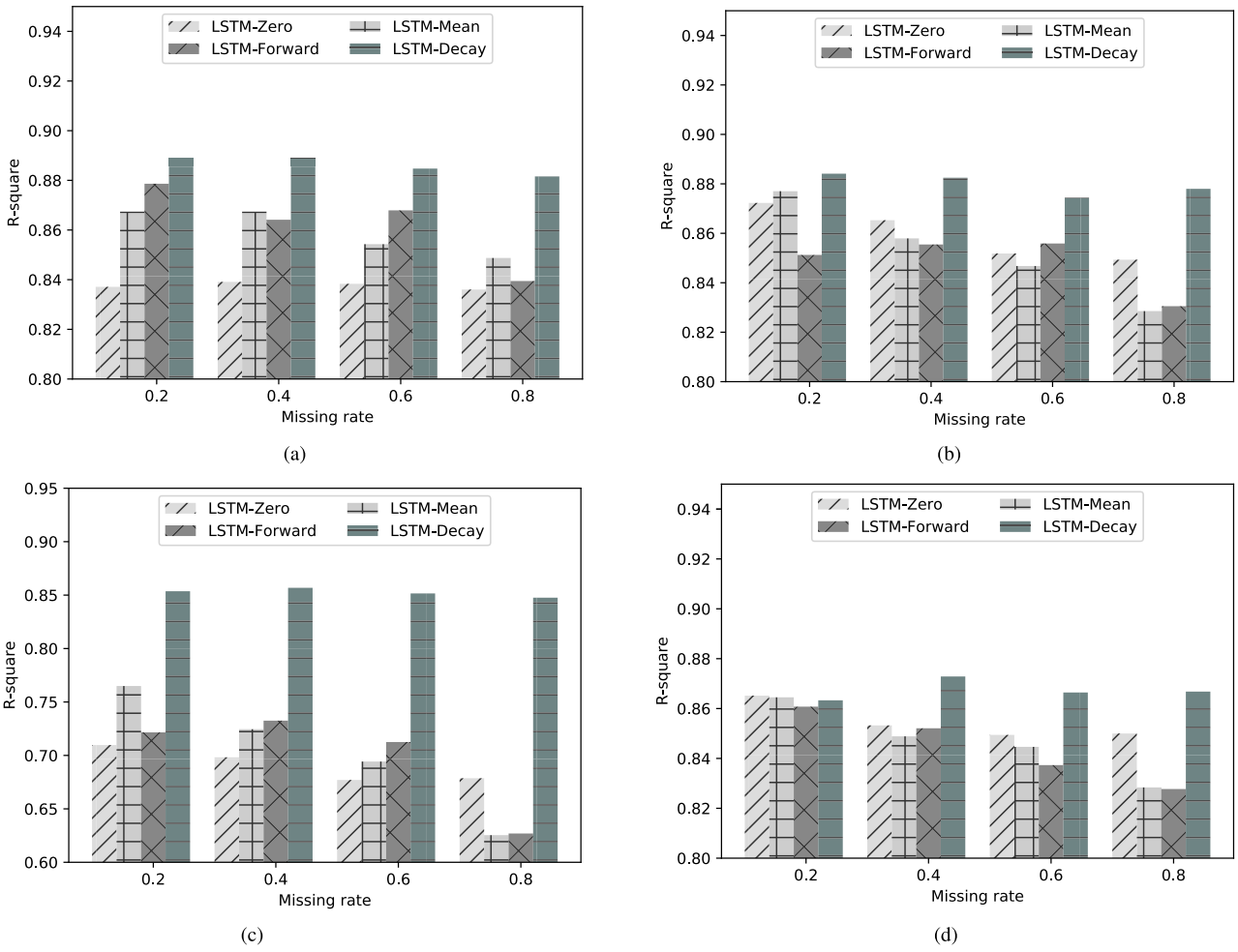


FIGURE 11. (a) Temperature. (b) Pressure. (c) Salinity. (d) Oxygen. The impact of setting different missing rates for each parameter on prediction performance.

Then we evaluate the performance of LSTM-Decay when simultaneously varying missing rates for 0.1 to 0.4 for all parameters. Figure 12 shows the performance of prediction models with simple imputation methods decreases significantly as increasing missing rates. Our proposed shows a better prediction performance and the advantage is more prominent when the missing rate is high.

c: PERFORMANCE ON SAMPLED DATASETS

Due to the limited energy and long transmission delay, the most common method for communication cost reduction is to sample/compress data in the data sending terminal. In this case, the collected data in the monitoring center is sampled and we also need to consider predict parameters on sample data. Next, the prediction performance is evaluated under different sampling intervals. As shown in Figure 13, we test the model performance when setting sampling intervals from 10s to 60s in a dataset with an average missing rate of 20 %. We find that the larger sampling interval causes the degradation of prediction accuracy. LSTM-Decay performs the best performance among these methods. The result of the prediction model also provides a guideline that we can make efficient data reduction when given an error bound. The proposed model can be used to train sampled data with good performance.

In the experiments, we first simulated the scenario of data packet transmission to validate the efficiency of dropping retransmission mechanism. When mapping to real datasets, we then preprocessed environmental data with different data missing rates and then use the processed data to train the performance of our proposed LSTM model. We use the combined evaluation results to simulate the execution process of our proposed delay-guaranteed framework.

VI. RELATED WORK

We have observed that UWSNs have been attracting increasing attention for its potential for (nearly) real-time data collection [1]. There have been a variety of communication protocols designed for underwater sensor networks [4], [5]. Generally, underwater acoustic communication has two characteristics: lossy in space and inconsistent in time [6]. Unstable data transmission brings high-frequency data loss

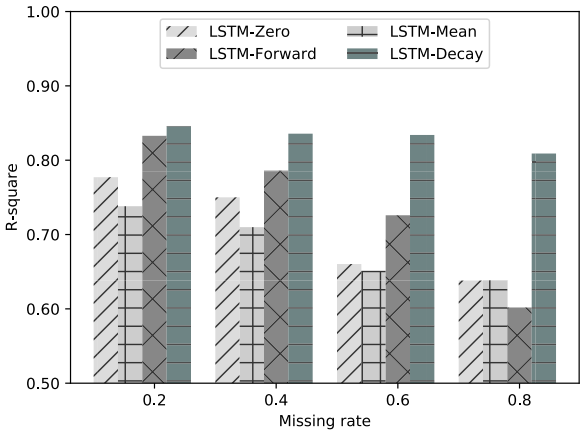


FIGURE 12. Varying the missing rate for ***all*** parameters from 0.2 to 0.8.

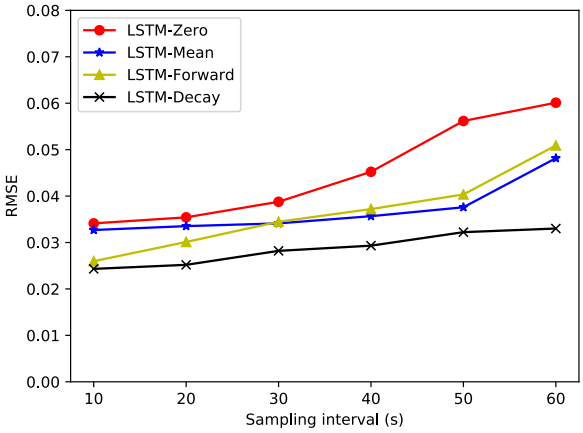


FIGURE 13. Performance comparison when setting different sampling intervals.

in network space. Multiple factors may cause the missing of data values, such as the malfunction of sensor nodes, packet collision, drop readings for saving cost [24], [25]. The received data may be inconsistent in time since the long delay of the update messages causes the received data may be the retransmission of legacy data instead of fresh data. Hence, data itself collected from acoustic communication networks is incomplete. The quality of collected data directly affects the results of subsequent data processing, analysis.

Data imputation for lossy data is important for making accurate decisions, especially in real-time underwater monitoring. Existing data mining algorithms classify common imputation methods into two categories. One is to directly fill missing values with substituted values that may be mean or most frequently observed values, or we can build learning models to estimate missing data [26].

There have been a number of imputation methods employed to fill in missing values in WSNs. Simple imputation methods include interpolation, expectation maximization, linear regression, etc [27], [28]. However, these methods are not suitable if data loss is not random (there is continuous data loss), which is frequently in underwater networks. In WSNs, sensor nodes are deployed to measure a region. By leveraging spatial information of neighbors, K-nearest neighbors (KNN) are used to estimate missing data [21], [29]. However, KNN is efficient in a dense sensor network where we can find available non-missing neighbor data to estimate missing values. In UWSNs, the deployment of sensor nodes is much sparser than that in terrestrial WSNs. It's possible that the reference values obtained from neighbor nodes cannot represent missing values or the data collected from neighbors are missing.

By exploiting the time and spatial correlation, more complex methods have been developed for better estimation. Reference [30] utilizes association rules to mine temporal and spatial correlations. Reference [31] estimates missing values based on the temporal and spatial correlation by assigning different weights for these two dimensions. However, the correlation information may be unknown if data are incomplete, which constrains the use of these methods. Reference [19] models WSN-specific data loss patterns for massive data loss. With the spatial, temporal and low-rank features, they reconstruct massive missing values based on compressive sensing.

Another type of data imputation methods is to build learning models to estimate missing data. Existing researches include Principal Component Analysis (PCA), random forests, etc [32]–[34]. But these methods need a lot of extra time to train hyperparameters to fill missing values. As the prevalence of machine learning in IoTs, deep learning methods have been applied to exploit implicit information from unaware environments [35], [36]. RNNs, such as LSTM, have shown strong prediction performance for multivariate time series data. Reference [37] proposed an improved LSTM-based network to predict future sea surface temperature values. The network includes one fully connected LSTM layer and one convolution layer. RNNs can not only store previous experiences by memory cells to exploit the long-term temporal dependencies, but also explore correlations between variables. Hence, we consider the usage of RNN-based models to mine more information with incomplete datasets. Instead of a two-step imputation-prediction procedure, the learning model directly combines the exploration of missing features and sensing data prediction where the missing feature is modeled and trained along with inputs.

VII. CONCLUSION

Real-time monitoring in UWSNs is difficult limited by the long delay time and frequent data loss. Inspired by these problems, we proposed a delay-guaranteed monitoring framework while ensuring the quality of collected data. To achieve the constrained delay, we drop the automatic retransmission mechanism applied in transfer protocols and migrate the processing of data loss to the data center. To assure the quality of collected data, we utilized an RNN-based model, LSTM, to exploit long-term temporal dependencies to perform efficient missing data estimation. We proposed a modified model, LSTM-Decay, to train the multivariate time

series sensing data with the exploration of missing features. The missing feature with a time-related weight is trained along with inputs in the LSTM architecture. The proposed RNN-based learning model both considers the training of spatial and multivariate sensor data collected in UWSNs. Experiment results with real-world ocean datasets show that the estimation performance with LSTM-Decay shows good performance both on unsampled or sampled datasets when setting different degrees of missing rates.

REFERENCES

[1] R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, "Underwater wireless sensor networks: A new challenge for topology control–based systems," *ACM Comput. Surv.*, vol. 51, no. 1, p. 19, 2018.

[2] L. Parra, J. Rocher, J. Escrivá, and J. Lloret, "Design and development of low cost smart turbidity sensor for water quality monitoring in fish farms," *Aquacultural Eng.*, vol. 81, pp. 10–18, May 2018.

[3] G. Han, S. Shen, H. Song, T. Yang, and W. Zhang, "A stratification-based data collection scheme in underwater acoustic sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10671–10682, Nov. 2018.

[4] M. Molins and M. Stojanovic, "Slotted FAMA: A MAC protocol for underwater acoustic networks," in *Proc. IEEE OCEANS Asia–Pacific*, May 2007, pp. 1–7.

[5] H. Yan, Z. J. Shi, and J.-H. Cui, "DBR: Depth-based routing for underwater sensor networks," in *Proc. Int. Conf. Res. Netw.* Berlin, Germany: Springer, 2008, pp. 72–86.

[6] F. Bouabdallah, C. Zidi, R. Boutaba, and A. Mehaoua, "Collision avoidance energy efficient multi-channel MAC protocol for underwater acoustic sensor networks," *IEEE Trans. Mobile Comput.*, to be published. doi: 10.1109/TMC.2018.2871686.

[7] Y. Luo, L. Pu, Z. Peng, Z. Zhou, J.-H. Cui, and Z. Zhang, "Effective relay selection for underwater cooperative acoustic networks," in *Proc. IEEE 10th Int. Conf. Mobile Ad-Hoc Sensor Syst. (MASS)*, Oct. 2013, pp. 104–112.

[8] G. Han, J. Jiang, L. Shu, and M. Guizani, "An attack-resistant trust model based on multidimensional trust metrics in underwater acoustic sensor network," *IEEE Trans. Mobile Comput.*, vol. 14, no. 12, pp. 2447–2459, Dec. 2015.

[9] X. Wang, D. Wei, X. Wei, J.-H. Cui, and M. Pan, "HAS^4: A heuristic adaptive sink sensor set selection for underwater AUV-aid data gathering algorithm," *Sensors*, vol. 18, no. 12, p. 4110, 2018.

[10] H. Wu, J. Xian, J. Wang, S. Khandge, and P. Mohapatra, "Missing data recovery using reconstruction in ocean wireless sensor networks," *Comput. Commun.*, vol. 132, pp. 1–9, Nov. 2018.

[11] G. M. Dias, B. Bellalta, and S. Oechsner, "A survey about prediction-based data reduction in wireless sensor networks," *ACM Comput. Surv.*, vol. 49, no. 3, p. 58, 2016.

[12] J. Jiang, G. Han, C. Zhu, S. Chan, and J. J. P. C. Rodrigues, "A trust cloud model for underwater wireless sensor networks," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 110–116, Mar. 2017.

[13] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, 2018 Art. no. 6085.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] K. Cho *et al.* (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation." [Online]. Available: https://arxiv.org/abs/1406.1078

[16] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "DeepCare: A deep dynamic memory model for predictive medicine," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Cham, Switzerland: Springer, 2016, pp. 30–41.

[17] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proc. Mach. Learn. Healthcare Conf.*, 2016, pp. 301–318.

[18] Oceans Networks Canada, University of Victoria, Victoria, BC, Canada. (Nov. 1, 2018). *Ocean Networks Canada Data Archive*. [Online]. Available: http://www.oceannetworks.ca

[19] L. Kong, M. Xia, X.-Y. Liu, M. Wu, and X. Liu, "Data loss and reconstruction in sensor networks," in *Proc. 32nd IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2013, pp. 1654–1662.

[20] R. H. S. Winterton, "Newton's law of cooling," *Contemp. Phys.*, vol. 40, no. 3, pp. 205–212, 1999.

[21] L. Pan and J. Li, "K-nearest neighbor based missing data estimation algorithm in wireless sensor networks," *Wireless Sensor Netw.*, vol. 2, no. 02, p. 115, 2010.

[22] M. Marjani *et al.*, "Big IoT Data analytics: Architecture, opportunities, and open research challenges," *IEEE Access*, vol. 5, pp. 5247–5261, 2017.

[23] D. Basak, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Inf. Process. Lett. Rev.*, vol. 11, no. 10, pp. 203–224, 2007.

[24] J. Jiang, G. Han, L. Shu, S. Chan, and K. Wang, "A trust model based on cloud theory in underwater acoustic sensor networks," *IEEE Trans. Ind. Informat.*, vol. 13, no. 1, pp. 342–350, Feb. 2017.

[25] J. Liu, M. Yu, X. Wang, Y. Liu, X. Wei, and J. H. Cui, "RECRP: An underwater reliable energy-efficient cross-layer routing protocol," *Sensors*, vol. 18, no. 12, p. 4148, 2018.

[26] B. Efron, "Missing data, imputation, and the bootstrap," *J. Amer. Stat. Assoc.*, vol. 89, no. 426, pp. 463–475, 1994.

[27] C. G. N. de Carvalho, D. G. Gomes, J. N. De Souza, and N. Agoulmine, "Multiple linear regression to improve prediction accuracy in WSN data reduction," in *Proc. IEEE 7th Latin Amer. Netw. Oper. Manage. Symp. (LANOMS)*, Oct. 2011, pp. 1–8.

[28] B. Zhang, Y. Liu, J. He, and Z. Zou, "An energy efficient sampling method through joint linear regression and compressive sensing," in *Proc. IEEE 4th Int. Conf. Intell. Control Inf. Process. (ICICIP)*, Jun. 2013, pp. 447–450.

[29] L. Z. Wong, H. Chen, S. Lin, and D. C. Chen, "Imputing missing values in sensor networks using sparse data representations," in *Proc. 17th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, 2014, pp. 227–230.

[30] H. Chok and L. Gruenwald, "Spatio-temporal association rule mining framework for real-time sensor network applications," in *Proc. 18th ACM Conf. Inf. Knowl. Manage.*, 2009, pp. 1761–1764.

[31] Z. Gao, W. Cheng, X. Qiu, and L. Meng, "A missing sensor data estimation algorithm based on temporal and spatial correlation," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 10, 2015, Art. no. 435391.

[32] R. Magán-Carrión, J. Camacho, and P. García-Teodoro, "Multivariate statistical approach for anomaly detection and lost data recovery in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 6, 2015, Art. no. 672124.

[33] G. Pachauri and S. Sharma, "Anomaly detection in medical wireless sensor networks using machine learning algorithms," *Procedia Comput. Sci.*, vol. 70, pp. 325–333, Dec. 2015.

[34] M. T. Asif *et al.*, "Spatiotemporal patterns in large-scale traffic speed prediction," in *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 794–804, Apr. 2014.

[35] G. M. Dias, M. Nurchis, and B. Bellalta, "Adapting sampling interval of sensor networks using on-line reinforcement learning," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 460–465.

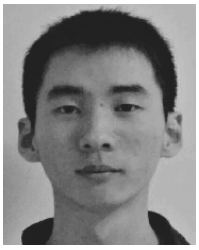
[36] Z. Sun, L. Zhou, and W. Wang, "Learning time-frequency analysis in wireless sensor networks," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3388–3396, Oct. 2018.

[37] Y. Yang, J. Dong, X. Sun, E. Lima, Q. Mu, and X. Wang, "A CFCC-LSTM model for sea surface temperature prediction," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 2, pp. 207–211, Feb. 2018.

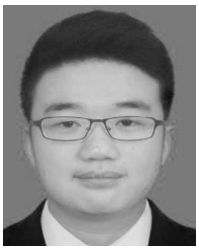
XIAOHUI WEI is currently a Professor and the Dean of the College of Computer Science and Technology, Jilin University. He is also the Director of the High Performance Computing Center, Jilin University. His current major research interests include resource scheduling for large distributed systems, infrastructure-level virtualization, large-scale data processing systems, and fault-tolerant computing.

YUANYUAN LIU received the B.S. and M.A. degrees from the College of Computer Science and Technology, Jilin University, Changchun, Jilin, China, in 2013 and 2016, respectively, where she is currently pursuing the Ph.D. degree. Her research interest includes approximate computing and big data analysis.

SHANG GAO received the B.Math. degree (Hons.) in computer science from the University of Waterloo, Canada, in 2006, and the M.S. and Ph.D. degrees in computer science from the University of Calgary, Canada, in 2009 and 2014, respectively. From 2006 to 2007, he was an Engineer with SAP Labs China, Shanghai. Since 2014, he has been an Associate Professor with the College of Computer Science and Technology, Jilin University, China. His research interests include algorithmic game theory, cloud computing, and data systems.



XINGWANG WANG received the Ph.D. degree from the College of Computer Science and Technology, Jilin University, Changchun, Jilin, China, in 2018, where he is currently a postdoctor. His research interest includes approximate computing and mobile computing.



HENGSHAN YUE received the B.S degree from the College of Computer Science and Technology, Jilin University, Changchun, Jilin, China, in 2016, where he is currently pursuing the Ph.D. degree. His research interests include approximate computing, GPGPU architecture, and HPC computing.