Sequence-To-Sequence Learning for Online Imputation of Sensory Data

Kaitai TONG[1], Teng LI[2,*]

(1. *Faculty of Applied Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada*;
2. *Department of Mechanical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada*)

Abstract: Online sensing can provide useful information in monitoring applications, for example, machine health monitoring, structural condition monitoring, environmental monitoring, and many more. Missing data is generally a significant issue in the sensory data that is collected online by sensing systems, which may affect the goals of monitoring programs. In this paper, a sequence-to-sequence learning model based on a recurrent neural network (RNN) architecture is presented. In the proposed method, multivariate time series of the monitored parameters is embedded into the neural network through layer-by-layer encoders where the hidden features of the inputs are adaptively extracted. Afterwards, predictions of the missing data are generated by network decoders, which are one-step-ahead predictive data sequences of the monitored parameters. The prediction performance of the proposed model is validated based on a real-world sensory dataset. The experimental results demonstrate the performance of the proposed RNN-encoder-decoder model with its capability in sequence-to-sequence learning for online imputation of sensory data.

Key words: Data Imputation; Recurrent Neural Network; Sequence-To-Sequence Learning; Sequence Prediction

1 Introduction

Recent developments of the Internet of Things (IoT) systems facilitate information gathering and online monitoring processes, which also take more and more significant roles in various other fields, such as micro-electromechanical systems (MEMS), intelligent instrumentations, robotics, and so on. Particularly, in monitoring applications, the IoT systems equip different sensing and sensory devices with the access to local or wide area network, providing data collection, aggregation, processing, and analysis [1]. Irrespective of the monitoring objectives, collection and delivery of reliable sensory data are still facing significant challenges in the data acquisition process using IoT sensing systems, especially when involving high-volume online measurements.

Due to limited onboard computational and energy resources, external disturbances, and unexpected failures, the problem of missing data is frequently encountered in IoT-based sensing systems. The missing data leads to unavailable or meaningless sensory data, which can cause instability in the data acquisition process. Such a phenomenon not only will affect system reliability for online applications, but also will influence the quality of the collected data for further analysis and information gathering. For example, missing sensory data may cause a dangerous or even damaging condition for real-time decision-making applications that heavily rely on sensor readings, e.g., in autonomous vehicles. In addition, missing data in a dataset can cause losses of statistical characteristics and introduce statistical bias potentially. In general practice, a dataset with large quantities of missing values without any preprocessing may mislead the implementation of a learning algorithm, which can severely impact the inference outcomes, such as classification or prediction, as a consequence [2].

In order to handle the missing data issue without sacrificing the size of a dataset, an effective scheme is required to predict un-sampled sensor readings by making use of historical observations of the moni-

tored parameters. Such methodology is called data imputation [3]. In the literature, traditional imputation approaches commonly depended on maximum likelihood (ML) methods and multiple imputations (MI) methods [4]. Both of them relied on a strong assumption that the parameters of the datasets were independently and identically distributed (i.i.d) [5, 6]. This assumption may not always be applicable for real-world sensory data.

More importantly, inter-correlations among sensory parameters might exist in reality. The relationships among them are generally nonlinear and complicated. To establish a more complex model for describing the underlying environmental field, the state-of-the-art data-driven methods were actively studied, including artificial neural network (ANN) [7], support vector machine (SVM) [8], support vector regression (SVR) [9], and many of their variations [10].

Although the mentioned methods provide more complicated modelling architectures, they may not be suitable to characterize underlying environmental fields that have complex interdependencies and interactions [11]. Recently, with the fast development of deep learning techniques, deep neural networks (DNN) have been designed to model complex environmental fields. With the strong capabilities in representative learning and nonlinear modeling, the DNNs provide the superior tools for tasks such as data prediction and data imputation. Among them, recurrent neural networks (RNN) are proven to be more suitable for time series prediction. For example, long short-term memory (LSTM) networks have been widely applied as the state-of-the-art RNN architectures [12], which construct a unit with a certain forgetting rate to characterize short and long term time series. Recently, the RNN encoder-decoder model has been introduced for translation, which was considered as sequence-to-sequence learning [13]. The work of [14] further adopted the original RNN-encoder-decoder to time series prediction. Different from these recent work, the proposed model focuses on the one-step-ahead data sequence imputation of the monitored sensory parameters, rather than predicting the upcoming time series of a monitored sensory parameter.

In this paper, a sequence-to-sequence learning method based on an RNN-encoder-decoder model is introduced. The model simultaneously predicts the missing data of the multiple monitored parameters for multivariate data imputation. Specifically, the encoding layers of the network extract underlying features to represent the driving features within the input time series. The decoding layer of the network predicts the data sequences of the monitored parameters in a one-step-ahead manner. The predictive data sequences are utilized for online imputation of the missing data, afterwards. The proposed model is tested based on a real-world dataset in this paper. Prediction results are compared with the-state-of-the-art methods, which show the superior performance of the proposed method for the imputation of missing sensory data.

The rest of the present paper is organized as follows. Section 2 introduces the formulation of the research objective in data sequence prediction. Section 3 presents the proposed RNN-encoder-decoder for data imputation in detail. The experimental results based on a real-world sensory dataset are demonstrated in Section 4. The final section concludes the present paper.

2 Formulation

The goal of the present paper is to predict the future sensory data of multiple parameters in a one-step-ahead manner by making use of their historical data streams. These historical data sequences are defined by a sliding time window. Given a time window T, the collected data sequences by N different sensors lead to multivariate time series, which is designated by $\boldsymbol{X} = (\boldsymbol{X}^1, \boldsymbol{X}^2, \cdots, \boldsymbol{X}^N)^T$, where $\boldsymbol{X}^n \in \mathbb{R}^T$, $n = 1, 2, \cdots, N$. In a time window, each time step is indexed by t, $t = \{1, 2, \cdots, T\}$.

At each time step t, the measurements over all

sensors can be obtained as $\boldsymbol{X}_t=(\boldsymbol{X}^1{}_t,\boldsymbol{X}^2{}_t,\cdots,\boldsymbol{X}^N{}_t)^T$, $\boldsymbol{X}^n{}_t\in\mathbb{R}$. The one-step-ahead prediction of a data sequence is formulated as:

$$\hat{\boldsymbol{y}}_{T+1}=F(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_T) \tag{1}$$

where $\hat{\boldsymbol{y}}_{T+1}=(\hat{y}^1_{T+1},\hat{y}^2_{T+1},\cdots,\hat{y}^N_{T+1})^T$ denotes the predicted data sequence, F denotes the nonlinear regression function. The proposed RNN-encoder-decoder is introduced to establish the function F in Equation (1). More details are presented in the following section.

3 Proposed RNN Architecture

The proposed RNN model is introduced in this section in detail, which has an encoding-decoding architecture. In the model, encoders are utilized to extract the hidden features within the input time series while the decoders map the hidden features to the predictive outputs of the one-step-ahead data sequences. The detailed architecture of the encoding layers and the decoding layers are presented as follows.

First, the encoding layers exploit the underlying features within the input time series and extract the driving features to make predictions. In the RNN model, the recurrent function is defined to construct the encoding layers. Specifically, at time t, the recurrent function can be represented as:

$$\boldsymbol{h}_{t+1}=f(h_t,\boldsymbol{X}_t), \tag{2}$$

where f denotes the recurrent function, $\boldsymbol{h}$ denotes the hidden state that will be learned via the recurrent function in the encoding layers iteratively.

In the present paper, the LSTM network is selected as the recurrent function. The LSTM network has the capabilities in modeling multivariate time series, especially by considering both the long term and short term interdependencies inside time series [12]. A LSTM unit can be formulated as:

$$\begin{bmatrix}\boldsymbol{i}_t\\ \boldsymbol{f}_t\\ \boldsymbol{o}_t\end{bmatrix}=\sigma\left(\begin{bmatrix}\boldsymbol{W}_i\\ \boldsymbol{W}_f\\ \boldsymbol{W}_o\end{bmatrix}[\boldsymbol{h}_{t-1}\oplus\boldsymbol{X}_t]+\begin{bmatrix}\boldsymbol{b}_i\\ \boldsymbol{b}_f\\ \boldsymbol{b}_o\end{bmatrix}\right),$$

$$\boldsymbol{j}_t=\tanh(\boldsymbol{W}_j[\boldsymbol{h}_{t-1}\oplus\boldsymbol{X}_t]+\boldsymbol{b}_j)$$

$$\boldsymbol{s}_t=\boldsymbol{f}_t\otimes s_{t-1}+\boldsymbol{i}_t\otimes\boldsymbol{j}_t$$

$$\boldsymbol{h}_t=\boldsymbol{o}_t\otimes\tanh(\boldsymbol{s}_t) \tag{3}$$

where $\oplus$ denotes the concatenation operator, $\otimes$ denotes the element-wise multiplication operator, σ denotes the logistic sigmoid function, and *tanh* denotes the hyperbolic tangent function. The basic unit of the LSTM structure is shown in Fig. 1.

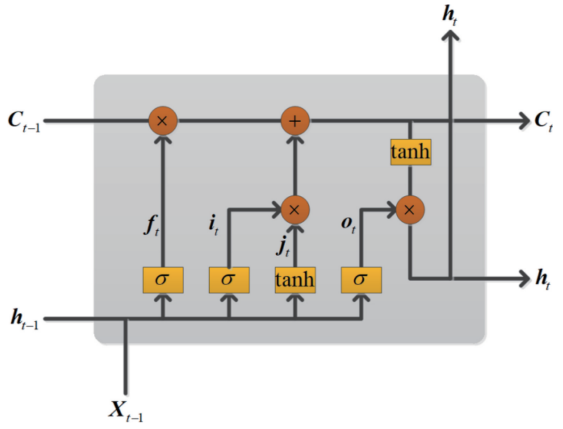

Fig. 1 Basic unit of the LSTM structure.

When encoding the multivariate inputs, instead of inputting the time series to the recurrent function directly, an attention layer is integrated to weight the input vectors before embedding them into the LSTM layers. The attention mechanism was studied in the work of [13, 14] to align between the input vectors and the hidden states, in order to pay more attention on the more important input vectors. In the attention layer, the attention value is defined through a multilayer perceptron in a probability form α, which is defined as:

$$\alpha m_t=\frac{\exp(a^m_t)}{\sum_{m=1}^{M}\exp(a^m_t)}, \tag{4}$$

where $a^m_t=\boldsymbol{v}^T{}_a\tanh(\boldsymbol{W}_a[\boldsymbol{h}_{t-1};\boldsymbol{C}_{t-1}]+\boldsymbol{U}_a\boldsymbol{X}m_{1:T}+\boldsymbol{b}_a)$ represents the alignment between the hidden states $\boldsymbol{h}_{t-1}$, cell states $\boldsymbol{C}_{t-1}$, and the input vectors $\boldsymbol{X}^m_{1:T}$. The probability $\alpha^m{}_t$ indicates the importance of the input variable m for the predictive output at time step t.

As a result, the recurrent function is updated as:

$$\boldsymbol{h}_{t+1}=LSTM(\boldsymbol{h}_t,\boldsymbol{\alpha}_t\otimes\boldsymbol{X}_t), \tag{5}$$

where $\boldsymbol{\alpha}_t = (\alpha_t^1, \alpha_t^2, \cdots, \alpha_t^m)^T$ and $\boldsymbol{\alpha}_t \otimes \boldsymbol{X}_t$ denotes the weighted input vectors.

After encoding the multivariate time series inputs to the encoding layers, the predictive outputs are generated through the RNN decoding layer. With the similar structure, an attention layer is embedded into the model before the LSTM stacks of the decoders. The attention probability β is defined as:

$$\beta_m^i = \frac{\exp(b_m^i)}{\sum_{i=1}^{T} \exp(b_m^i)}, \tag{6}$$

where $\boldsymbol{b}_t^i = \boldsymbol{v}_a^T \tanh(\boldsymbol{W}_a[\boldsymbol{h}_{t-1}; \boldsymbol{C}_{t-1}] + \boldsymbol{U}_a h_{1:T}^i + \boldsymbol{b}_a)$. The probability β_m^i indicates the importance of the hidden state i when making prediction of the m th parameter of the predicted data sequence output. The recurrent function of the decoding layer can be formulated as:

$$\boldsymbol{d}_{m+1} = LSTM(\boldsymbol{d}_m, [\hat{\boldsymbol{y}}_{T+1}^m \oplus \boldsymbol{c}_m]), \tag{7}$$

where $c_m = \sum_{i=1}^{T} \beta_m^i \boldsymbol{h}_i$, $m = 1, 2, \cdots, M$.

Given the learned hidden states and the temporally distribution weights of them, the predictive outputs of the monitored variable m within the predicted data sequence at time step $t = T+1$ can be obtained through a linear map, which is formulated as:

$$\begin{aligned} \hat{y}_{T+1}^m &= F(\boldsymbol{X}_{1:T}^1, \boldsymbol{X}_{1:T}^2, \cdots, \boldsymbol{X}_{1:T}^M), \\ &= \boldsymbol{v}^T(\boldsymbol{W}_y[\boldsymbol{c}_m; \boldsymbol{d}_m] + \boldsymbol{b}_y) + b. \end{aligned} \tag{8}$$

The overall architecture of the proposed RNN model is shown in Fig. 2. As shown there, first the input vectors are handled by the attention layers before inputting into the LSTM layers in the encoding structure. Afterwards, the hidden states are also handled by the attention layers before embedding into the LSTM layers in the decoding structure. In the training process, the data sequence of the next time step is used as the a sequence label.

For training the proposed RNN encoder-decoder model, since the overall architecture is differentiable, back-propagation algorithm with an Adam optimizer [15] is implemented. Mean square error (MSE) is selected as the metric of the loss function in the training procedure, which is defined as:

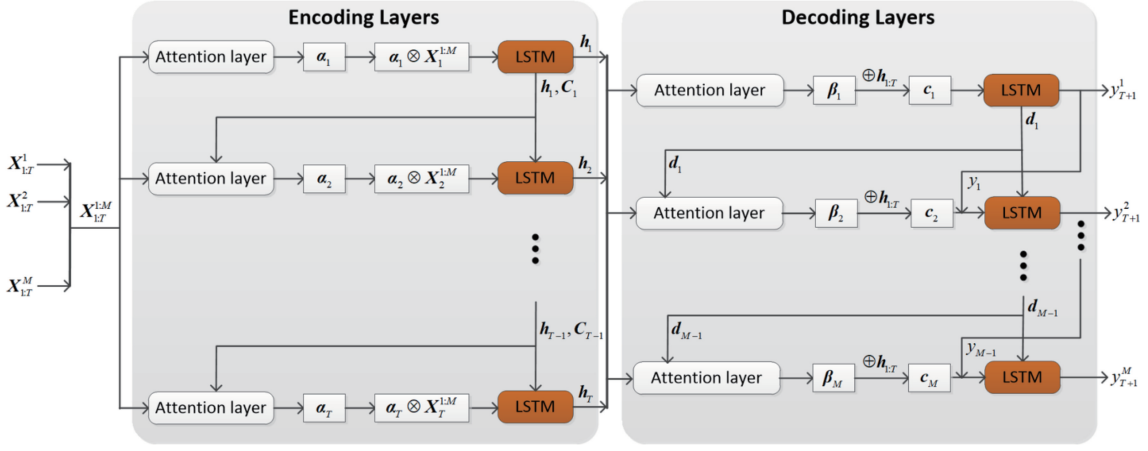

Fig. 2 The overall architecture of the proposed RNN model.

$$Loss = ||\hat{\boldsymbol{y}}(\Theta) - \boldsymbol{y}||^2, \tag{9}$$

where Θ denotes the set of the learnable parameters that will be trained in the training procedure. For a sliding time window, the data sequences of all the monitored parameters at next time steps are prepared as the labels and learned by the network via the training procedure.

4 Experiment

This section demonstrates the prediction performance of the proposed model in forecasting multivariate data sequences, which is based on the experiments of a real-world dataset from sensory data in an indoor sensing network. The description of the dataset and the experiments are given in the following

subsections.

4.1 Real-World Dataset

To examine the prediction performance of the proposed model, the sensory data collected at Intel Berkeley Research Lab was selected as the experimental dataset [16], which ranges from February 28th, 2004 to April 5th, 2004. The dataset measured and recorded four physical parameters of the study indoor environment, i.e., temperature, humidity, light intensity, and battery voltage, from 54 Mica2Dot sensors with a sampling rate of approximately every half minute. The sensors were located over the monitored area as shown in Fig. 3(a).

In the dataset, all four monitored parameters at each station were sensed and recorded as the time series with approximately 30 seconds in between each two consecutive time stamps. If a specific time stamp does not exist in the dataset, it indicates a loss of data at that corresponding time. The rates of the missing data at all the 54 sensor stations are summarized and visualized in Fig. 3(b) by plotting at their corresponding locations. In the figure, lighter color represents lower missing rate while darker color indicates higher missing rate. As it is shown, the loss of data commonly existed in the dataset at all the sensors over the spatial scale.

The time series of the four monitored parameters taken from No. 1 sensor is displayed in Fig. 4(a). A value of -1 is assigned to annotate the missing data of a sample. The subfigures display the variation of each monitored parameter and the missing data over time variation. To visualize the distribution of the missing data more clearly, Fig. 4(b) shows the detailed conditions of the missing data by zooming at a subinterval of the acquisition time period that is highlighted by the grey boxes A in Fig. 4(a). From this zoomed display, it can be perceived that the amount of missing data is large. The missing rate at this sensor station is about 44%.

Besides the overall missing rate found in this dataset, it can also be observed that a complete consecutive missing period over a long time steps

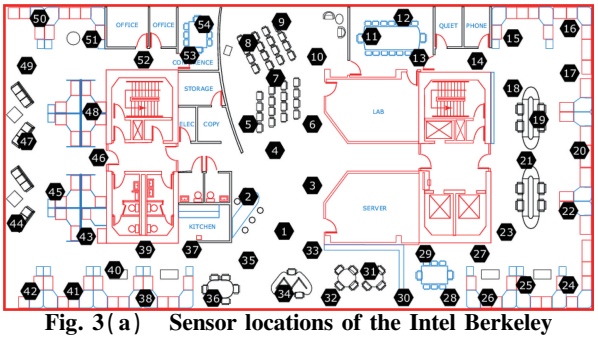

Fig. 3(a) Sensor locations of the Intel Berkeley Research Lab dataset [16].

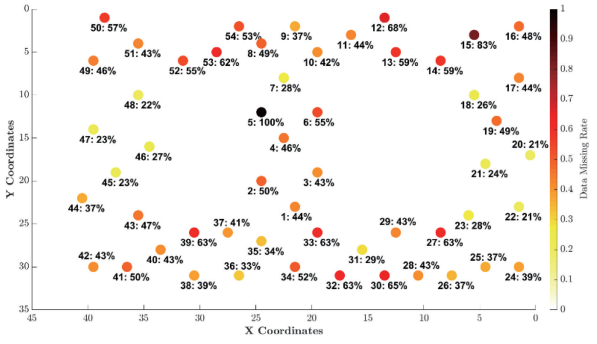

Fig. 3(b) Missing rates of the sensors in the Intel lab dataset.

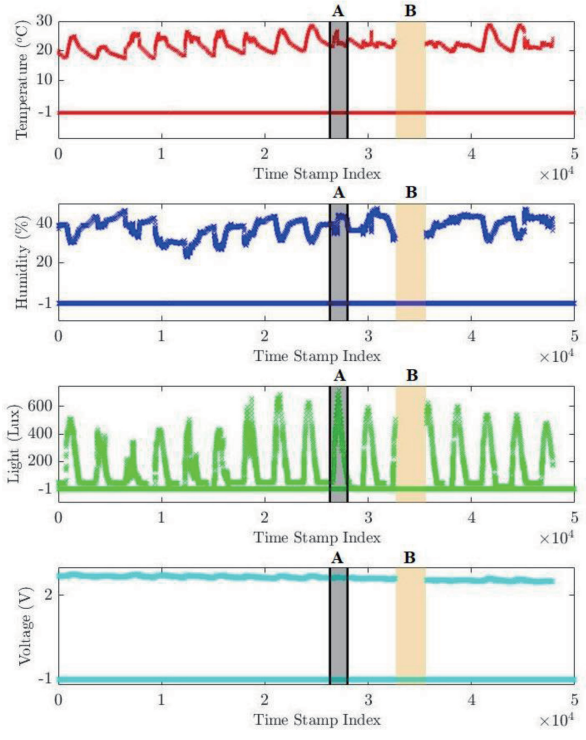

Fig. 4(a) Time series of the monitored parameters at No. 1 sensor.

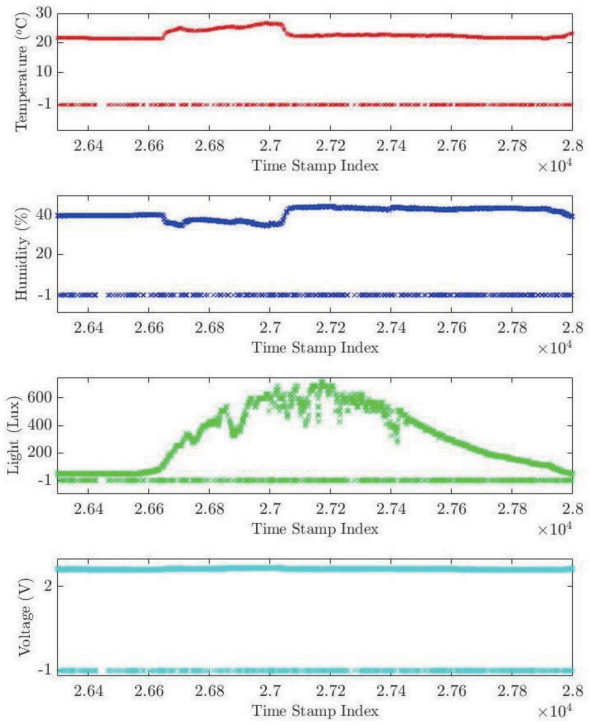

Fig. 4(b) Zoomed displays of the time series of the monitored parameters.

occurred, for example, over 3000 among time steps in $[3(10^4), 4(10^4)]$ in Fig. 4(a), highlighted by the orange boxes B. Such dense missing rate over a large time period is commonly found throughout all 54 sensors. If missing data occurred, it is highly likely that the samples of four parameters were missed simultaneously. This phenomenon further reveals the significant issue of missing data that is addressed in this paper.

4.2 Experimental Results

The proposed RNN model is validated based on the Intel lab dataset that is introduced in the previous subsection. The proposed model is also compared with the state-of-the-art models for data imputation, including autoregressive integrated moving average model (ARIMA), nonlinear autoregressive network with exogenous inputs model (NARX), and LSTM-RNN model. To compare with these models fairly, the corresponding model parameters are tuned to provide their best performance in the experiments.

To have more reliable data for training and validation procedures in the experiments, sensor stations with relatively low overall missing rate are preferable. Data from sensor station No. 8 is therefore selected as the target data source to compare the performance of the models. In the experiments, linear interpolation is utilized to fill in the local missing data with a missing period lower than 10-unit time stamps for training and validation dataset. Missing records beyond this period are discarded when feeding the training and validation data into the network model in the learning procedure. Table 1 provides the overview of the statistical characteristics of the data information at the target sensor station after preprocessing.

By considering the spatial correlation between sensors, collected data at the stations that are surrounded within approximately 5 meters is also included in the input vector, namely, sensor No. 7, 9, and 10. The data preprocessing is carried out for their sensory data. The sensor stations No. 53 and 54 are not considered to have spatial correlation with the target sensor station due to the isolation caused by the existence of the wall in between (see Fig. 3(a)). Given the prepared data, the whole dataset is divided into 80% for training, 10% for validation, and the remaining 10% as test set.

Adam optimizer is used in the training procedure. The learning parameters are set as: batch size = {64, 128, 256, 512}, training epoch = 50, dimension of the encoder hidden feature = {32, 64, 128, 256}, dimension of the decoder hidden feature = {32, 64, 128, 256}. The root mean square error (RMSE) and the mean absolute error (MAE) are utilized as the evaluation metrics to measure the prediction performance of these methods. The two performance metrics are defined as follows:

$$RMSE=\frac{1}{I}\sum_{i=1}^{I}(\hat{y}_i-y_i)^2,$$

$$MAE=\frac{1}{I}\sum_{i=1}^{I}|\hat{y}_i-y_i|,$$

where I denotes the total number of the predicted variables.

Let the time window size n represent the amount of data acquired from a sensor for the time window $[p+1, p+n]$ with no missing values, where p is set to the index in the test set. Prediction is based on a time window shifting mechanism where for every shift the predicted value for the corresponding parameter of interest will replace the nth element in the time window, while eliminating the first element. The total number of shifting steps in the test procedure is determined by the period of missing data and equals to I, which is set to 500 in the experiments. This procedure is repeated for all four monitored parameters, and the obtained RMSE and MAE values are shown in Table 2.

The complexity of the proposed model characterizes the relations of the historical collections and the surrounding measurements when making predictions of the missing data. As given in Table 2, the proposed model can provide superior prediction results on the Intel lab dataset for all four monitored parameters. The values of the RMSE and MAE results for light intensity are higher for all four models than those of other parameters. This is due to its high standard deviation value demonstrated in Table 1. In contrast, the prediction results of voltage are comparably low due to its low standard deviation.

The proposed model considers the nonlinearity and weighted impact from both temporal and spatial aspects that may potentially have influence on the monitored parameters in prediction. When large volumes of data are missing in a sensing system for online monitoring, the proposed RNN-encoder-decoder model is able to generate a series of reliable imputed data of multiple parameters over a long-term time scale.

Table 1 Statistical characteristics of the dataset at sensor station No. 8.

Monitored Parameters	Min	Max	Mean	Standard Deviation
Temperature(℃)	17.1660	26.5050	21.1364	2.3150
Humidity (%)	23.5990	46.3610	38.0215	4.6228
Light (Lux)	0.4600	1847.40	661.2896	730.4307
Voltage (V)	2.5822	2.7496	2.6554	0.0346

Table 2 Prediction performance of the compared models.

	ARIMA		NARX		LSTM		Proposed model	
Monitored Parameters	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
Temperature(℃)	1.7523	1.4119	1.4672	1.3189	1.17912	1.0625	0.9769	0.9446
Humidity (%)	3.2473	2.6115	2.1393	1.7875	1.4675	1.3177	1.0498	1.0038
Light (Lux)	360.7864	318.0221	319.3094	261.9225	276.8231	255.3528	258.4632	237.7896
Voltage (V)	0.0272	0.0230	0.0247	0.0234	0.0098	0.0092	0.0075	0.0071

5 Conclusion

This paper introduced an RNN-encoder-decoder architecture with the capability of sequence-to-sequence learning, focusing on the data imputation problem in sensory data from online monitoring systems. The proposed RNN model could effectively handle historical time series and make predictions of the upcoming data sequences of multiple target parameters simultaneously. The experimental results demonstrated the superiority of the proposed RNN model on an indoor sensing systems with multiple monitored parameters. When generating the training dataset in the experiments, the data records with the missing items of any monitored parameters were discarded. In the future, data records that are partially missing will be investigated and integrated into the training dataset. In addition, the proposed method can be extended to handle more complex environmental fields with dynamic changes.

References

[1] Botta, A., De Donato, W., Persico, V., and Pescapé, A., 2016. Integration of cloud computing and internet of things: a survey. Future generation computer systems, 56, pp.684-700.

[2] de León, A.D.L.V., Chen, B., and Gillet, V.J., 2018. Effect of missing data on multitask prediction methods. *Journal of cheminformatics*, 10(1), p.26.

[3] Cheema, J.R., 2014. A review of missing data handling methods in education research. *Review of Educational Research*, 84(4), pp.487-508.

[4] Shin, T., Davison, M.L., and Long, J.D., 2017. Maximum likelihood versus multiple imputation for missing data in small longitudinal samples with nonnormality. *Psychological methods*, 22(3), p.426.

[5] Little, R.J. and Rubin, D.B., 2019. *Statistical analysis with missing data* (Vol. 793). Wiley.

[6] Bleninger, S., 2018. *KriMI: a Multiple Imputation Approach for Preserving Spatial Dependencies: Imputation of Regional Price Indices Using the Example of Bavaria* (Vol. 33). University of Bamberg Press.

[7] Gholami, V., Booij, M.J., Tehrani, E.N., and Hadian, M.A., 2018. Spatial soil erosion estimation using an artificial neural network (ANN) and field plot data. *Catena*, 163, pp.210-218.

[8] Li, T., Ji, Y., Zhang, M., and Li, M., 2017. Determining optimal CO2 concentration of greenhouse tomato based on PSO-SVM. *Applied engineering in agriculture*, 33(2), pp.157-166.

[9] Kaneda, Y., Ibayashi, H., Oishi, N., and Mineno, H., 2015. Greenhouse environmental control system based on SW-SVR. *Procedia Computer Science*, 60, pp.860-869.

[10] Ali, I., Greifeneder, F., Stamenkovic, J., Neumann, M., and Notarnicola, C., 2015. Review of machine learning approaches for biomass and soil moisture retrievals from remote sensing data. *Remote Sensing*, 7(12), pp.16398-16421.

[11] Nutkiewicz, A., Yang, Z., and Jain, R.K., 2018. Data-driven Urban Energy Simulation (DUE-S): A framework for integrating engineering simulation and machine learning methods in a multi-scale urban energy modeling workflow. *Applied energy*, 225, pp. 1176-1189.

[12] Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., and Schmidhuber, J., 2017. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), pp. 2222-2232.

[13] Bahdanau, D., Cho, K., and Bengio, Y., 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv*:1409.0473.

[14] Qin, Y., Song, D., Chen, H., Cheng, W., Jiang, G., and Cottrell, G., 2017. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv*:1704.02971.

[15] Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv*: 1412. 6980.

[16] "Intel Lab Data." [Online] http://db.csail.mit.edu/labdata/labdata.html.

Copyright: © 2019 by the authors. This article is licensed under a Creative Commons Attribution 4.0 International License (CC BY) license (https://creativecommons.org/licenses/by/4.0/).