IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 6, 2019, accepted August 16, 2019, date of publication August 23, 2019, date of current version September 5, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2937098

Multi-Step Data Prediction in Wireless Sensor Networks Based on One-Dimensional CNN and Bidirectional LSTM

HONGJU CHENG[1,2], (Member, IEEE), ZHE XIE[1], YUSHI SHI[1], AND NAIXUE XIONG[3], (Senior Member, IEEE)

[1]College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China
[2]Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou 350116, China
[3]Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA

Corresponding authors: Hongju Cheng (cscheng@fzu.edu.cn) and Naixue Xiong (xiongnaixue@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61370210, and in part by the Science Foundation of Fujian Province, China, under Grant 2019J01245.

ABSTRACT The increase of network size and sensory data leads to many serious problems to the wireless sensor networks due to the limited energy. Data prediction method is helpful to reduce network traffic and increase the network lifetime accordingly, especially by exploring data correlation among the sensory data. Data prediction can also be used to recover abnormal/lost data in case these sensor nodes fail to work. The current prediction methods in wireless sensor networks do not make full usage of the spatial-temporal correlation between wireless sensor nodes, and thus leads to higher prediction error relatively. This paper proposes a novel model for multi-step sensory data prediction in wireless sensor network. Firstly, we introduce the artificial neural networks based on 1-D CNN (One-Dimensional Convolutional Neural Network) and Bi-LSTM (Bidirectional Long and Short-Term Memory) to get the abstract features of different attributes via the pre-processed sensory data. Then, these abstract features are used to obtain one-step prediction. Finally, the multi-step prediction is introduced by using historical data and the prediction results of the previous step iteratively. Experiment results show that after selecting suitable node combinations in which the spatial-temporal correlation is highlighted, the proposed multi-step predictive model can predict multi-step (short and medium term) sensory data, and its performance is better compared with other related methods.

INDEX TERMS Neural networks, predictive models, wireless sensor networks.

I. INTRODUCTION

Wireless sensor networks are widely used to collect environmental data due to its low energy consumption, low cost and large-scale deployment [1]–[3]. With the increase of sensor nodes in wireless sensor networks, many problems are revealed including increased energy consumption, high network transmission delay, bad transmission quality due to data transmission congestion, data transmission is blocked due to partial node failure, etc. The cheap requirement of sensor nodes also results in node/link failure, and thus data lost/abnormal is a common phenomenon in the wireless sensor networks. Data predictive is helpful to solve these problems in wireless sensor networks. Some methods use prediction to reduce data transmission, while some methods use it to correct or recover abnormal data [4]–[8]. The former tries to optimize the data collection process, and the latter focuses on mining relevant information in the collected data to improve data quality. First, in wireless sensor networks, redundant data transmission can be avoided by prediction, which makes wireless sensor networks improve in energy efficiency and data transmission quality [9]–[12]. Second, predicting the state of equipment or area that monitored by wireless sensor networks can increase the lifetime of the equipment or avoid unnecessary accidents [13]–[17]. Eventually, multi-step data prediction is very important to the wireless sensor network due to its ability to predict multi-step (short and medium term) sensory data.

In this paper, various factors are considered for data prediction based on sensory data correlation. These factors can be

The associate editor coordinating the review of this article and approving it for publication was Zhongming Zheng.

described as following: 1) Quality of sensory data: the outliers in sensory data that affect the data quality, some of these outliers are not in the range of normal data, the others are very abnormal compared with their neighborhoods. 2) Correlation between sensory data: Wireless sensor networks often collect redundant sensory data, and each sensory data has a different trend in time and space dimensions. 3) Correlation utilization: After analyzing the correlation between the sensory data, this paper uses the higher correlation to implement multi-step data prediction, but it is necessary to select a model that can learn and utilize the sensory data correlation well.

Some correlations between sensory data are shown in the periodic change of the sensory data gathered by single sensor node, while others are shown in the trend of different sensor nodes. Classical data analysis methods have limited ability to extract such abstract features especially for the data prediction problem. It is difficult for classical methods to extract effective features from original data. However, with the development of deep learning, it is possible to automatically extract relevant features for prediction, which greatly reduces the difficulty of extracting data correlation features for prediction.

Deep learning has developed rapidly in recent years. CNN (Convolutional Neural Networks) and RNN (Recurrent Neural Networks) have been widely used in image recognition, natural language processing, climate and traffic flow prediction [18]–[21]. CNN is a feedforward neural network containing convolution calculations, which is suitable for processing various types of data with translation invariance, such as audio data or image data. RNN is a time recurrent neural network whose internal state can show the dynamic behavior of data in the time dimension and it is suitable for processing time series data. The LSTM (Long and Short-Term Memory) is an RNN with a structure called cell, and the network relies on three gates in cell to choose useful information and discard unwanted information. The cell structure makes the network more accurate than RNN in processing and predicting medium and long-term dependencies in time series. The neural network model based on CNN and LSTM can be used to extract the correlation in data, which is suitable for the prediction problem of sensing data in wireless sensor networks.

After considering the above three factors, this paper builds a multi-step predictive model based on 1-D CNN and LSTM. This model uses CNN to extract translation invariance within data and uses LSTM to deal with medium-term dependence. Thus, the accuracy of short and medium-term predictions are improved.

The remainder of this paper is as follows: Section 2 introduces related works. Section 3 introduces the problem formulation and basic idea behind this paper. Section 4 discusses the data correlation formulation between sensory data. Section 5 introduces the structure of the predictive model proposed in this paper. Section 6 shows the experimental results in this paper. Finally, Section 7 shows the conclusions.

II. RELATED WORK

There are many researches on various prediction problems on wireless sensor networks [13]–[21]. The main problems in the related work can be divided into two different types. One is to improve energy consumption, data transmission, or fault detection in the sensor network through prediction, while another is to predict the state of device or environment monitored by wireless sensor network, including predicting the battery health, energy harvested, or the status of industrial devices. For the first type mentioned above, most of the existing solutions use non-deep learning methods because devices in wireless sensor networks lack sufficient computing resources to support the computation of artificial neural networks. For the second type, there are both non-deep learning methods and deep learning methods to extract data features for prediction. This paper investigates these two types of prediction problems and the according solutions.

Pramod Ganjewar proposed a hierarchical minimum mean square prediction algorithm for reducing data transmission in wireless sensor networks [9]. In this paper, a predictive model based on hierarchical fractional least mean squares (HFLMS) was proposed, which attempts to predict the sensory data by error estimation and only sends the required data to the receiving node by using the proposed adaptive filter to reduce the energy consumption in the wireless sensor networks.

R A Avinash proposed a data prediction method in wireless sensor networks using Kalman filter [10]. This method used predicted data replace the loss data, which makes the sensory data in the wireless sensor network available. The paper predicted the temperature data and find that the temperature data is well predicted and the error range is within an acceptable range. The author concluded that this predictive model can be used not only to predict future data, but also to smooth existing data.

Hamed Nazaktabar used reinforcement learning techniques to establish a dual prediction scheme for reducing the energy consumption of wireless sensor networks [11]. This method learned environmental signals and builds predictive models based on their experience. When the model fails, it only needed to learn and transmit the environmental data at the time of failure. The model uses error bounds to discretize the environmental data, and then used the instantaneous state of the environmental data signal to train the predictive model. It is proved by experiments that the proposed method could significantly reduce the energy consumption of wireless sensor networks through the dual prediction scheme based on enhanced learning.

Adrien Russo proposed a time series predictive model based on self-organizing mapping algorithm in wireless sensor networks [12]. This model can be used for anomaly monitoring or to maintain the integrity of the system. When a node stops working, its value can be replaced by the value obtained by the predictive model. The self-organizing mapping algorithm is an unsupervised deep learning method that maps high-dimensional spaces into two-dimensional spaces

and then forms a grid connected by several adjacent neurons. This neural grid updates the weights by learning algorithms and experiments show that it can make stable prediction results.

Tomoki Kawamura proposed a method for predicting the power consumption of sensor node in a wireless sensor network for train condition monitoring [13]. The paper introduces some problems in this train monitored networks like the frequent changes in the train compartment configuration and the environmental changes that often occur during train travel. These problems often lead to changes in the network configuration and communication environment between wireless sensor, making it difficult to predict their power consumption. To solve the above problems, the paper proposes a Monte Carlo method to predict the power consumption in such wireless sensor networks, and proves the superiority of this method through a series of experiments.

Rafael Lajara proposed a method for predicting battery health in wireless sensor networks [14]. The related parameters like The number of charge and discharge cycles, internal resistance of the node, voltage, output current, and temperature are considered to build an analytical model to predict the battery health. In this paper, the variation trajectory of the above parameters is collected as a training set during a large number of discharge cycles, and a battery health predictive model based on multi-layer perceptron is constructed.

Yinggao Yue proposed a fault predictive model for strip wireless sensor networks [15]. Based on the theory of kernel function, this paper proposes a fault prediction method, and chooses the radial basis function as its kernel function to predict the fault from two aspects: node hardware fault and network fault. Experiments show that the prediction accuracy in this model is higher than GRNN and PNN.

Alves Maicon Melo uses a wireless sensor network to predict wind turbine damage [16]. The method is based on a time series predictive model, using ARIMA and fuzzy systems to consider the effect of temperature on wind turbine damage prediction. In this paper, the influence of ambient temperature on the blade is considered and the accuracy of the prediction is improved. It is proved by experiments that when the system considers the influence of temperature on the blade, the prediction performance and energy consumption performance of the model are excellent.

Alessandro Cammarano proposed a method for energy harvesting prediction in environmentally powered wireless sensor networks [17]. This approach uses past energy observations to predict future energy availability, enabling environmentally powered wireless sensor networks to adjust their energy management strategies as needed. In the article, the author focuses on solar energy and wind energy, using connected photovoltaic panels, micro wind turbines, and public solar energy curves to get real energy harvesting curves and use them to validate the model presented in this paper. The experiment shows that the proposed method has high prediction accuracy.

III. PROBLEM FORMULATION

The data in the wireless sensor network is collected by sensor nodes. The set of sensor nodes can be defined as $V = (V_n|n = 1, 2, 3, \ldots, N)$, where N is the number of sensor nodes. Assuming in the wireless sensor network, node V_n collects M kinds of sensing data. The set of sensory data of the given m^{th} kind collected by node V_n is defined as $D_{n,m}$. The value of these m^{th} kind of sensory data collected by node V_n at the timestamp t is defined as $d_{n,m}(t)$, where $0 < n \leq N$, $0 < m \leq M, t > 0$.

Sometimes, the sensory data of V_n before T timestamp is known, and the sensory data of V_n after the T timestamp is unknown. In this case, the unknown data is defined as $D'_n = \{d_{n,m}(t)|T < t, 0 < m \leq M\}$. Especially, in case the data of node V_n after timestamp T is unknown, let $V*$ be the set of sensor nodes which still have available data, and the data collected by $V*$ is $D* = \{D_{n,m}|V_n \in V*, 0 < m \leq M\}$. Given V_n, the collected data before timestamp T is known as $(V_n–V'_n)$. The data prediction problem is to predict the unknown data of V_n after timestamp T, i.e., $D'_{n,m}$, by using $(D_n–D'_n)$ and the data set $D*$. Note that the set $D'_{n,m}$ contains different kinds of sensory data collected by node V_n after the timestamp T.

In this paper, we introduce a neural network based predictive model to iteratively predict $d_{n,m}(T)$, $d_{n,m}(T + 1)$, $d_{n,m}(T + 2), \ldots, d_{n,m}(T + E)$, in which E is the times of iterative predictions. NNet$(\cdot)$ represents a well-trained predictive model, and H the number of timestamps used in the predictive model. Let $S(\cdot)$ represent the selected nodes from $V*$ whose correlation with $D_{i,j}$ is not only above a certain threshold but also closest to node V_i. The process of predicting $d_{n,m}(T)$ can be formulated as:

$$d_{n,m}(T) = \text{NNet}(d_{n,m}(T-1), d_{n,m}(T-2), d_{n,m}(T-3), \ldots, \mathrm{d}_{n,m}(T-H), S(V^*)).$$

In this paper, the proposed model NNet$(\cdot)$ predicts the sensory data $d_{n,m}(T)$ by utilizing the sets from $d_{n,m}(T–1)$ to $\mathrm{d}_{n,m}(T–H)$ as well as the subset of V^*. After getting $d_{n,m}(T)$, we can iteratively use the predictive model to get $d_{n,m}(T+1)$, $d_{n,m}(T + 2)$ and so on, and finally we get $d_{n,m}(T + E)$.

In order to comprehensively evaluate the prediction results, several evaluation indicators are used in this paper, including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and R-square. Details of these parameters can be found in Section 6.2.

In order to get accurate prediction results, data correlation between neighbor sensors and target sensor is helpful to improve the multi-step prediction process. A series of methods including quartile method, wavelet denoising, correlation analysis, CNN and LSTM are used to improve the quality of sensory data and extract related features for multi-step prediction. The methods used in this paper can be divided into four processes: data preprocessing, correlation analysis, neural network construction and training, and iterative prediction. Details of these four steps are described as follows.

Data Preprocessing: Most of the noise in the sensory data is originated from different factors such as tough environment, bad sensors, and the network congestion, etc. These factors may result in abnormal sensory data.

Quartile method and wavelet denoising can be used to improve the quality of sensory data. The quartile method removes the abnormal data that deviates from the dataset, in this way the final data does not have elements that deviates from the data distribution. To further improve the data quality, wavelet denoising is used to reduce the noise that quartile method cannot recognize in the sensory data. This abnormal data is in the normal distribution, but it might have different trends compared with their neighborhood.

Correlation analysis: Taking Intel indoor dataset [22] as an example, this paper quantifies the correlation between 54 sensor nodes in wireless sensor network, including the correlation between different kinds of sensory data in a single node and the same kind of sensory data between nodes. We first adopt the Spearman correlation coefficient to quantify the correlation, then build the predictive model by sensory data with the highest correlation coefficient. In order to verify that the correlation coefficient of the sensory data has an impact on the prediction accuracy, multiple sensory data with different correlation coefficients are introduced to predict the sensory data of the same node.

Neural Network Construction and Training: To process time series data, we propose to use the LSTM and CNN to extract features of sensory data, and all these features are merged into merge layer of the neural network. After the structure of the neural network is determined, the MSE is chosen as the loss function, and we select Adam as the optimizer in the model.

Iterative prediction: Once the model training process is finished, the iterative prediction is performed. After getting a new prediction result from the historical data, the historical data and the new prediction result are both used to carry out the next step of data prediction, iteratively. In order to make multi-step prediction stable in short and medium term, two sensor nodes which have the strongest relationship with the target node is chosen in the prediction process.

IV. CORRELATION BETWEEN SENSORY DATA

A. BETWEEN DIFFERENT KINDS OF SENSORY DATA IN A SINGLE NODE

The Intel Indoor Dataset [22] used in this paper contains 4 kinds of data that are collected by 54 sensors, i.e., temperature, humidity, light, and voltage. These data have different correlations with each other, and data with strong correlation can be used to improve the prediction accuracy. In order to quantify the correlation between sensory data, this paper picks up node 4 as an example to calculate the correlation between sensory data. Spearman correlation coefficient is used to perform the calculation process, the Spearman correlation coefficient is calculated as following:

$$\rho = 1 - \frac{\sum 6d_i^2}{n(n^2-1)} \tag{1}$$

In which, d_i is the i^{th} element in the ranking difference set, that is, the difference between the two variables' rank from the correlative two datasets. And n is the number of variables from the correlative two datasets.

The correlation between the sensory data is calculated by (1), including the correlation coefficient between the sensory data such as temperature, humidity, light, and voltage. The Spearman correlation coefficient between each sensory data is shown in Table 1.

TABLE 1. Correlation coefficient of sensory data.

Correlation	Temperature	Humidity	Voltage	Light
Temperature	1.0000	-0.4050	-0.0039	0.3516
Humidity	-0.4050	1.0000	-0.4682	-0.1511
Voltage	-0.0039	-0.4682	1.0000	0.2496
Light	0.3516	-0.1511	0.2496	1.0000

As we can see, Table 1 is symmetrical because the correlation calculations are commutative. It is also seen that correlation is different between different sensory data. For example, the correlation coefficient between temperature and humidity is -0.4050, while the correlation coefficient between temperature and illumination is 0.3516. If the correlation coefficient is smaller than zero, it means there is a negative correlation between these two groups of variables. For multi-step prediction, strong correlation is beneficial to strengthen the learning process of neural network based models, which can make the prediction more accurate. However, the correlation between various types of sensory data in a single sensor is low, and its ability to assist prediction is weak. Therefore, it is necessary to add other data with stronger correlation when building the data predictive models.

B. BETWEEN THE SAME KINDS OF SENSORY DATA AMONG MULTIPLE NODES

The Intel indoor dataset used in this paper contains a variety of sensory data collected by 54 sensor nodes. These sensor nodes are deployed in various corners of the room, and there are obstacles between some nodes. In order to quantify the correlation between different nodes, this paper calculates the Spearman correlation coefficient between node 4 and other sensor nodes, including temperature, humidity, and light.

The structure of the Intel lab is shown in Fig. 1, in which furniture in the lab, i.e., chairs and tables, is drawn as circles and rectangles. The sensor node is marked by a black hexagon in which the node number is shown. There are two decimals nearby the sensor node, and the above one is the correlation coefficient with node 4 calculated by the denoised temperature data, and the one below is the correlation coefficient calculated by the sensory data without Wavelet denoising method. It can be seen from Fig. 1 that the correlation between nodes' temperature data has a relationship with the

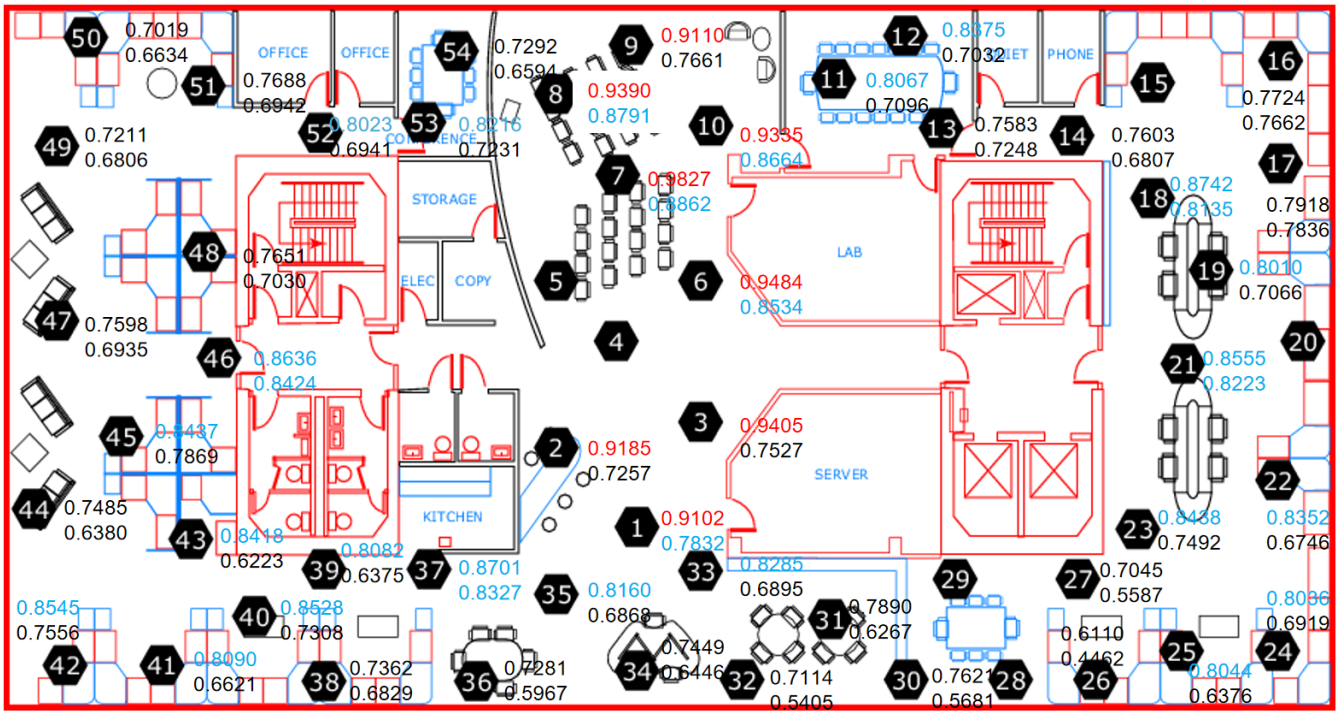


FIGURE 1. Correlation in temperature between node 4 and other nodes.

distance between nodes. But, by observing the correlation between node 4 and 42, 41, this kind of relationship is not so obvious in this dataset. There may be several reasons for this kind of phenomena.

1) Data noise in some nodes, which may affect the accuracy of correlation coefficient. For example, the correlation coefficients between nodes 45, 43 calculated from original data in the above figure are 0.7869 and 0.6223, while the correlation coefficients calculated from denoised data are 0.8437 and 0.8418. It can be seen that the denoising method makes the correlation coefficients between these nodes at the same level. Because data denoising can only remove part of the noise, there is still a lot of noise in the denoised data, which will have effect on the correlation coefficient.

2) Possible air conditioning system such as a central air conditioner in the indoor environment. Air conditioning systems are often deployed in indoor environments. These air conditioners can significantly adjust the temperature and humidity in the room, this may lead to a weak correlation between nodes close with each other, or a strong correlation between nodes that far from each other.

3) The indoor environment is divided into several rooms, and the different usage of the room may cause a large difference in correlation between nodes that deployed in different rooms. The correlation coefficients between node 8, 53 and 4 in the above figure are very different, which are 0.9380 and 0.8216 respectively.

These above reasons may cause the correlation coefficient to be higher compared with that with far distance. But in general, the correlation coefficient is gradually reduced as the distance increases. Therefore, when selecting nodes to improve data predicting, we should choose those nodes in the same room and close with the target.

Fig. 2 shows the Spearman correlation coefficient between node 4 and other nodes in light data. Similar to Fig. 1, there are two decimals next to the sensor node in Fig. 2, which shows the correlation between the current node and node 4 via light data. It is known from the figure that the correlation between light data is not directly related to the distance, and the nodes in the edge of the building have a higher correlation with node 4, such as node 42, 38, 54, 16, 24 and so on. The correlation coefficient of node 2 is 0.8923, but the correlation of node 35 is 0.9250, higher than node 2, which have a longer distance with node 4. In Fig. 2, node 2 is deployed around some obstacles, so the collected light data may be affected by obstacles. This scenario might be helpful to explain why the correlation coefficient with node 4 is lower. Light data is highly susceptible to indoor or outdoor lighting and obscuration, which leads to lower correlation compared to temperature data. In this way, we can see that the light data of neighborhood nodes is not suitable for prediction.

Fig. 3 shows the Spearman correlation coefficient between node 4 node and other nodes in humidity data, which is similar to Fig. 1 and 2. It decreases with the distance increasing, but the decreasing speed of humidity is different in different directions. For example, the speed is slow in the upper right direction, but it is very fast in the lower right and upper

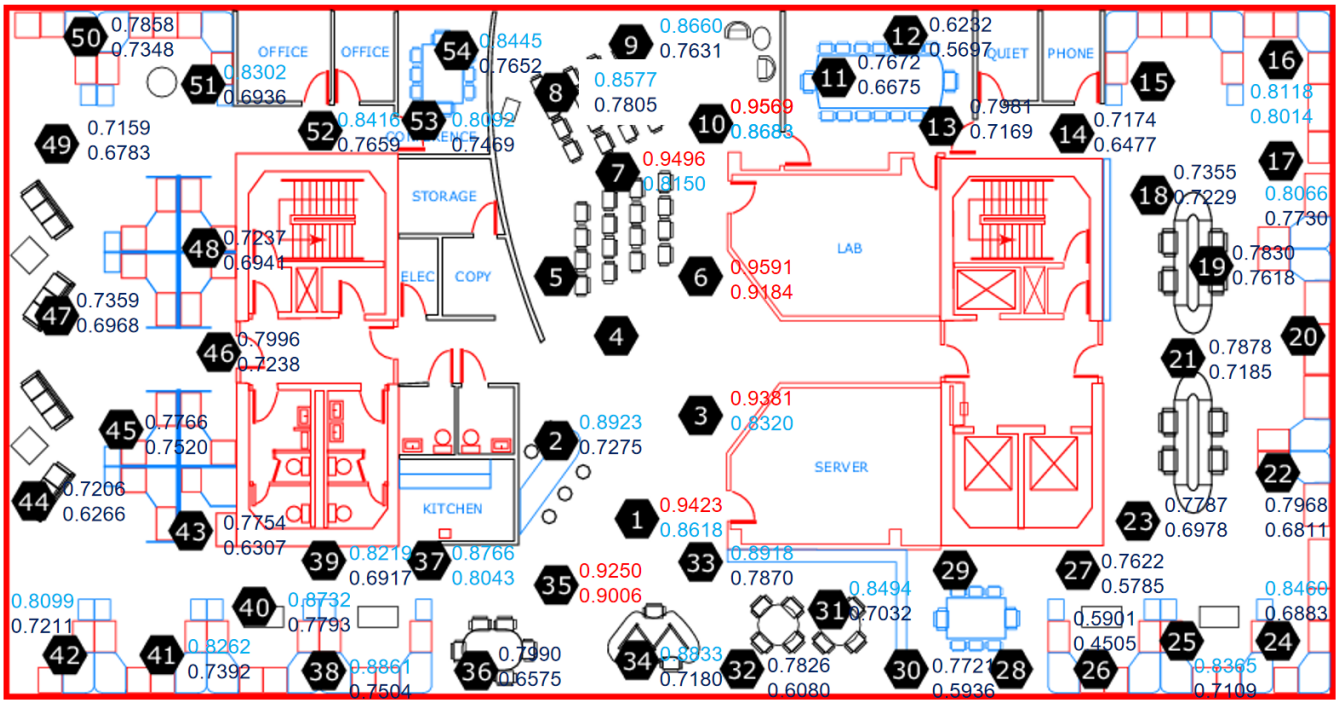


FIGURE 2. Correlation in light between node 4 and other nodes.

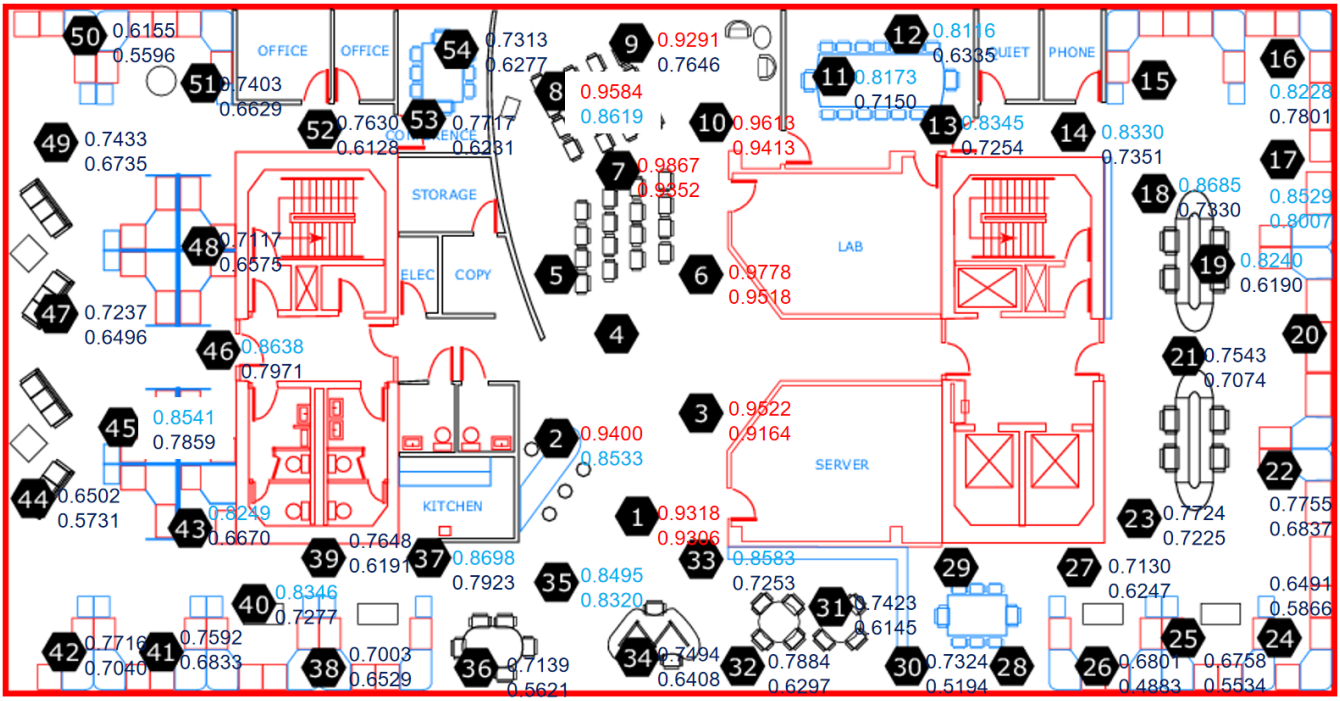


FIGURE 3. Correlation in humidity between node 4 and other nodes.

left direction. In general, as we can see from Fig. 3, humidity data and temperature data have a strong correlation and they are suitable for multi-step predictive models based on spatial correlation.

V. DATA PREDICTION MODEL

A. 1-D CONVOLUTIONAL NETWORK

With the help of data preprocessing, the quality of the sensory data can be greatly improved. In order to extract data

features quickly and accurately for data prediction, in this paper, we introduce to use the one-dimensional convolutional layer and the one-dimensional pooling layer as the first two hidden layers of the neural network model. The convolutional neural network mainly includes three characteristics, local perception, spatial arrangement, parameter sharing. Details of these three characteristics can be seen as below.

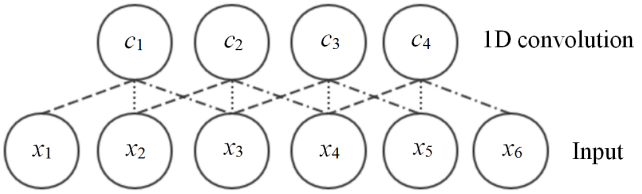


FIGURE 4. Example for the process of 1-D convolution.

Local perception: The convolutional layer is different from the fully connected layer, since its hidden unit only connects with part of the input layer. Fig. 4 shows an example for the process of 1-D convolution, in which c_1 to c_4 are feature maps and x_1 to x_6 are inputs. The single hidden unit of the one-dimensional convolution is only connected to three inputs of the input layer. This kind of connection greatly reduces the number of parameters and accelerates the training process of the neural network. The size of the input area that connected to the hidden unit is called receptive field.

In the above figure, c_1 to c_4 are calculated by convolution, where c_1 is calculated by convolution with x_1, x_2, and x_3, and the connections between c_1 and x_1, x_2, x_3 all have different weights. When calculating c_2, these three weights are corresponding to x_2, x_3, and x_4, respectively. The connections with the same line style have the same weight.

Spatial Arrangement: The parameter of convolutional layer consists of three variables, including size of convolution kernel, stride, and padding. They determine the size of output feature map. The size of convolution kernel is the number of variables used for convolution calculation. In Fig. 4, c_1 is calculated by convolution by x_1, x_2 and x_3, so the size of convolution kernel in the above one-dimensional convolution is 3. The stride is the distance that the convolution kernel needs to move when the convolution calculation goes on. c_1 is calculated by x_1, x_2 and x_3. When the convolution kernel moves by one step, c_2 is to be calculated by x_2, x_3, and x_4. The padding is used to offset the reduction of feature map size caused by the convolution calculation. As is shown in Fig. 4, there are 6 variables in the input layer, so the size of calculated feature map is 4. If an input is added to each side of the input layer, the feature map size will be 6. The size of the output feature map is calculated by (2):

$$w_{out} = \frac{w_{in} + 2 * padding - F}{stride} + 1 \tag{2}$$

In which, w_{out} is the size of input feature map, w_{in} is the size of input feature map, *padding* is the number of elements filled at the both ends of the input, F is the size of convolution kernel, and *stride* is the stride mentioned above. According to (2), the size of output feature map is 4.

Parameter sharing: Parameter sharing can greatly reduce the number of parameters. In the same filter, the calculation of all feature maps shares the same set of weights, n filters have n sets of weights. Parameter sharing and sparse connections greatly reduce the number of free variables, enabling convolutional neural networks to extract features with fewer computational resources.

The feature map calculated by the 1-D convolution layer is down-sampled in the 1-D pooling layer. The pooling layer selects and filters the feature map output from the convolution layer. The widely used pooling methods are Max Pooling and Mean Pooling, and the parameters of pooling methods are the pool size, the stride and the padding. The pool size is the number of data used for the pooling calculation, and the stride and the padding in pooling layer are the same as convolutional layers. Fig. 5 shows an example for the process of 1-D pooling. In this example, the feature map is composed of 6 elements including c_1 to c_4 and two *pd*, and the elements of 1-D pooling p_1, p_2, and p_3 are calculated by these 6 elements.

In the above figure, c_1, c_2, c_3 and c_4 are the feature maps calculated by the upper layer, *pd* is the elements of padding, p_1, p_2, and p_3 are the outputs of pooling layer, wherein the stride is 2 and the pooling size is 2.

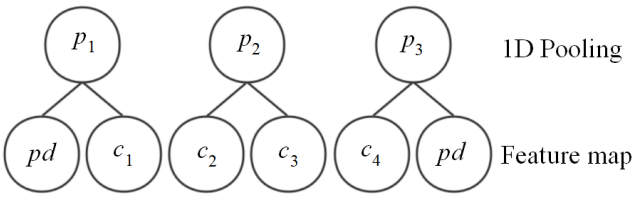


FIGURE 5. Example for the process of 1-D pooling.

In Fig. 5, p_1 is calculated from pooling *pd* and c_1. Max Pooling or Mean Pooling can be used as the pooling method. The Max Pooling method is to select the biggest one from *pd* and c_1 as the value of p_1. The Mean Pooling method takes the average of pd and c_1 as the value of p_1, and the values of p_2 and p_3 is calculated the same way. Through these two kinds of pooling method, the pooling layer gets its output p_1, p_2, an p_3.

After passing through the 1-D convolutional layer and the 1-D pooling layer, features are input into the bidirectional LSTM neural network for further extraction.

B. BIDIRECTIONAL LSTM

LSTM has great advantages in processing and predicting time series data. It is a special form of RNN. Both LSTM and RNN have a network module with chain structure. In RNN, the module consists of a single neuron structure, but in LSTM, this module consists of cells with three gates. The cell relies on three gates for feature selection, including the input gate, output gate, and forget gate. The loop body of LSTM is shown

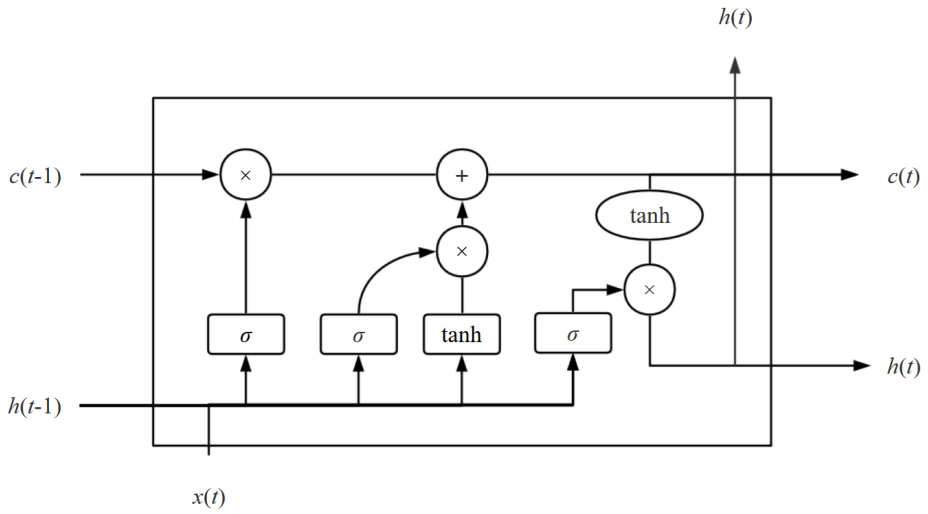


FIGURE 6. Structure of cell.

in Fig. 6. These symbols in the figure are to be introduced in the following (3)-(8).

Fig. 6 shows the structure of the cell, which mainly includes the input gate, the output gate and the forget gate. The calculation methods for these three types of gates are illustrated by the following:

$$i(t) = \sigma(W_i x(t) + U_i h(t-1) + b_i) \tag{3}$$

Equation (3) describes the calculation process of the input gate in the cell, where $h(t-1)$ is the output of the previous cell, $x(t)$ is the current cell input, σ is the sigmoid function, and W_i and U_i are the weights of the input gate.

$$f(t) = \sigma(W_f x(t) + U_f h(t-1) + b_f) \tag{4}$$

Equation (4) describes the calculation process of forget gate in the cell. This gate determines which information in the cell needs to be discarded, and W_f and U_f in the equation are forgot gate weights.

$$\tilde{C}(t) = \tanh(W_c x(t) + U_c h(t-1) + b_c) \tag{5}$$

$$C(t) = f(t) * C(t-1) + i(t) * \tilde{C}(t) \tag{6}$$

Equation (5) and (6) describe the update processes, where (5) is the candidate memory unit which generates alternative update information, and (6) is the process of updating the state of the cell. The information from forgot gate is combined with the update information to calculate a new state, where W_c and U_c are the weights of the alternative new state, and $*$ is the Hadamard product.

$$o(t) = \sigma(W_o x(t) + U_o h(t-1) + b_o) \tag{7}$$

$$h(t) = o(t) * \tanh(C(t)) \tag{8}$$

Equation (7) and (8) describe the calculation process of output gate. First, the sigmoid layer is used to get the state of the cell to be output, then the updated cell state is processed by tanh function, and updated cell state is multiplied by $o(t)$ to get $h(t)$. U_o is the output gate weight.

The cell mentioned above is the core of LSTM neural network. Based on this structure, a bidirectional LSTM network is created to extract features of sensory data.

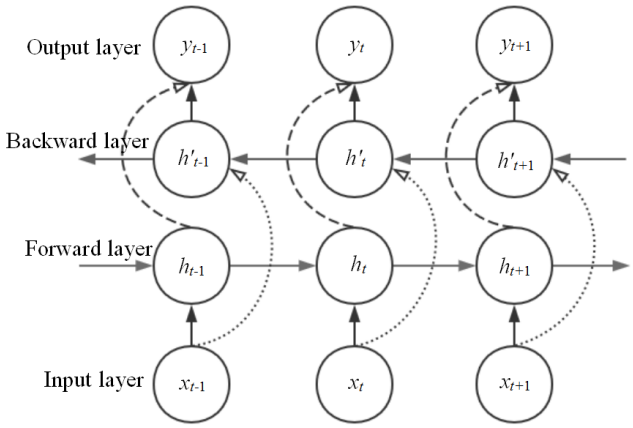


FIGURE 7. Example for the structure of bidirectional LSTM.

Fig. 7 shows an example for the bidirectional LSTM. The input data x_{t-1}, x_t, x_{t+1} is processed by both the forward and backward layer and finally the hidden state h_{t-1}, h_t, h_{t+1} and h'_{t-1}, h'_t, h'_{t+1} are obtained respectively. Then these hidden states are fused to obtain the output, i.e., h_t, h'_t are fused to get y_t in the output layer. The case is similar to y_{t+1} and y_{t-1}.

The bidirectional LSTM can extract more context information than normal LSTM. The forward and backward time series are used to get the information of the current timestamp in the past and the future, so that the network can make better time series prediction [23]. There is no direct connection between the backward and forward layer, which ensures that the structure is acyclic. In case the input layer data x_t, the results of the forward and backward layers are

combined at the output layer to make output y_t. After each sensory feature is processed by the bidirectional LSTM and passes through the fully connected layer, all sensory features are merged through merge layer.

C. PROPOSED NEURAL NETWORK FOR PREDICTION

The same type of sensory data from several sensor nodes is used for multi-step prediction in the proposed model. In order to extract the abstract features of sensory data from multi-node, parallel multiple networks are used to learn the features of each input data. The proposed model uses sensory data from three nodes for multi-step prediction. Regarding node 4 as an example, the temperature data of node 4 and its two neighbors are used for prediction. The selection of neighbor nodes mainly considers the Spearman correlation coefficient between the neighboring node and the target node. In Section 5.2, the correlation coefficient of each type of sensory data is studied, and the correlation between temperature and humidity is relatively stable, which is suitable for the proposed prediction method. Fig. 8 shows the network structure used in this paper, where node 2 and 3 are selected to assist the prediction for temperature of node 4:

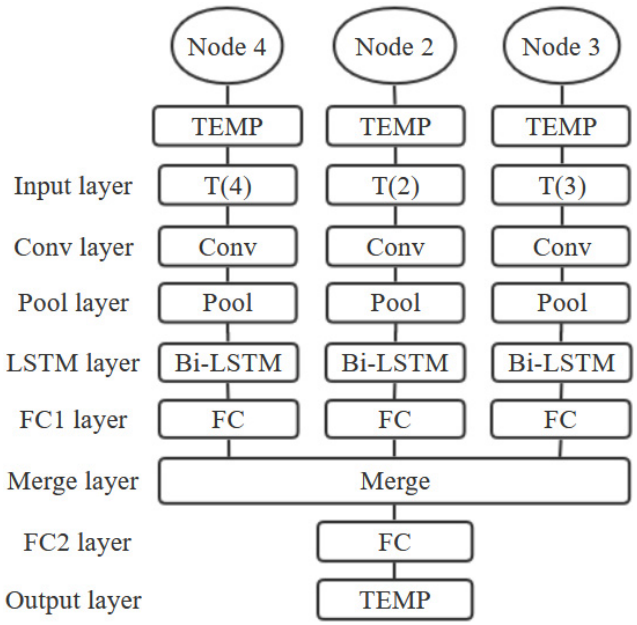


FIGURE 8. Structure of multi-step predictive model.

In Fig. 8, T(4), T(2), and T(3) are the temperature sequences of nodes 4, 2, and 3, respectively. Each sequence has an independent length, and the length is the time step of each sequence. The Conv layer in the above figure is a one-dimensional convolution layer. The Pool layer is a one-dimensional pooling layer that down-samples the abstract features to prevent overfitting. The LSTM layer is a bidirectional LSTM that uses cells to extract long-term dependencies in abstract features, which helps extract temporally related features in the sensory data. FC1 layer is a fully connected layer for unifying abstract features of different shapes into one shape. Merge layer incorporates multiple sensory features for data prediction. In this paper, three kinds of prediction features are chosen by using three parallel network structures. The purpose of this parallel structure is to adjust the time step of each sensory data separately.

Experiments in Section 6.1 have shown that a reasonable combination of time steps can achieve better prediction results. The output dimensions of each layer in the predictive model are shown in Table 2:

TABLE 2. Output dimensions of layers in the proposed model.

Dimension	T(4)	T(2)	T(3)
Input layer	(Ts(4), 1)	(Ts(2), 1)	(Ts(3), 1)
Conv layer	(C_1, F_1)	(C_2, F_1)	(C_3, F_1)
Pool layer	(P_1, F_1)	(P_2, F_1)	(P_2, F_1)
LSTM	$2N_1$	$2N_1$	$2N_1$
FC1	FC_1	FC_1	FC_1
Merge layer		M_1	
FC2		FC_2	
Output layer		1	

In Table 2, the output dimensions of each sensory data in Input layer are (Ts(4), 1), (Ts(2), 1), (Ts(3), 1), respectively, where Ts(4), Ts(2), Ts(3) is the timestamp used for each sensory data, and the second dimension is 1, indicating that only one type of sensory data is used. In the Conv layer, C_1, C_2, and C_3 are the feature map shapes of sensory data, and the size of output feature map can be calculated by (2) from the size of convolution kernel, padding, stride and the input feature shape. F_1 is the number of filters. P_1, P_2, and P_3 in the Pool layer are the shapes of the feature maps after the pooling operation. N_1 is the number of neurons in the LSTM layer. Since the bidirectional LSTM neural network is used in this paper, it has both forward and backward directions, so the output shape is twice the N_1. The output shapes of remaining layer FC_1, FC_2, and M_1 is the number of neurons.

Since the loss function used in this paper is the error of single-step prediction, it is necessary to adjust the parameters to reduce the average error of multiple iterative prediction. In order to get better multi-step prediction results, several parameters and node selection experiments are carried out.

VI. SIMULATION RESULT

A. PARAMETERS SELECTION

The parameters that need to be adjusted in this paper include the time step in the Input layer, the size of convolution kernel, the number of filters in the Conv layer, the pooling size in the Pool layer, and the number of neurons in the LSTM layer. Taking the examples of using historical temperature data of node 4, 7 and 10 to predict unknown temperature

TABLE 3. Output dimensions of layers in the proposed model.

Input	Conv		Pool		LSTM	Error
Time step	Kernel	Filter	Pool size	stride	Neurons	RMSE
50, 10, 10	5	1	2	1	32	0.727
10, 10, 10	5	1	2	1	32	0.788
30, 10, 10	5	1	2	1	32	0.721
40, 10, 10	5	1	2	1	32	0.716
40, 10, 10	5	1	**3**	1	32	0.700
40, 10, 10	5	1	**4**	1	32	0.690
40, 10, 10	5	1	**5**	1	32	0.682
40, 10, 10	**6**	1	5	1	32	0.943
40, 10, 10	**4**	1	5	1	32	1.118
40, 10, 10	**5**	**2**	5	1	32	0.689
40, 10, 10	5	**4**	5	1	32	0.627
40, 10, 10	5	4	5	**2**	32	0.631
40, 10, 10	5	4	5	**3**	32	0.645
40, 10, 10	5	4	5	1	**16**	0.647
40, 10, 10	5	4	5	1	**64**	0.680

data of node 4, a variety of parameter combinations can be used to verify the prediction performance. Table 3 shows the experiment results under various parameter combinations. The error in the table is the average error of 1000 prediction processes, in which the batch size is set to 200, and epoch is 15.

The first and second rows in the table are the layers and their adjustable parameters. When time step = (40, 10, 10), convolution kernel = 5, filter = 4, pooling size = 5, stride = 1 and the number of neurons in LSTM layer is 32, the most accurate prediction is with RMSE = 0.627. The size of convolution kernel in convolutional layer has a great influence on the prediction, as can be seen from Table 3.

B. NODE SELECTION

The proposed multi-step predictive model needs to use the sensory data of neighboring nodes. Multiple groups of nodes are selected to iteratively predict the trend of sensory data of node 4. The experiment results show that the multi-step predictive model based on the correlation between nodes can make a good prediction when good neighboring nodes are selected.

Fig. 8 describes the data predict result for the temperature of node 4, which uses the temperature data of node 3 and 6 as assistance. The position of node 4, 3 and 6 is shown in Fig. 1, and it is clear that the distance between node 4 and 3, 6 is very close. The Spearman correlation coefficient of temperature data between node 4 and 3 is 0.9405, the correlation coefficient between node 4 and 6 is 0.9484.

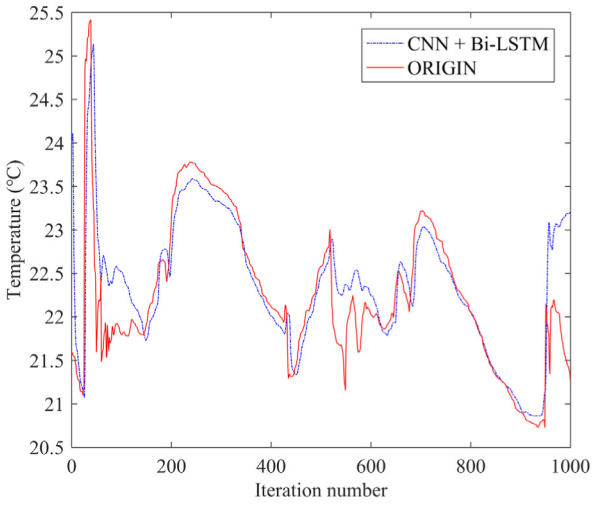


FIGURE 9. Using node 3 and 6 to predict temperature of node 4.

In Fig. 9, the blue dotted line is the prediction result, and the red solid line is the original data. With the help of temperature data of nodes 3 and 6, the temperature data of node 4 is used to iteratively predict 1000 temperature data.

Fig. 10 shows the prediction for humidity of node 4, which chooses the humidity data of node 3, 6 as assistance. In Fig. 3, there is a strong correlation between the humidity data of node 4 and 3, 6, which are 0.9522 and 0.9778, respectively. With the strong correlation of humidity data between node 3, 6 and 4, the proposed model has made a good prediction. In order to further study the relationship between error of

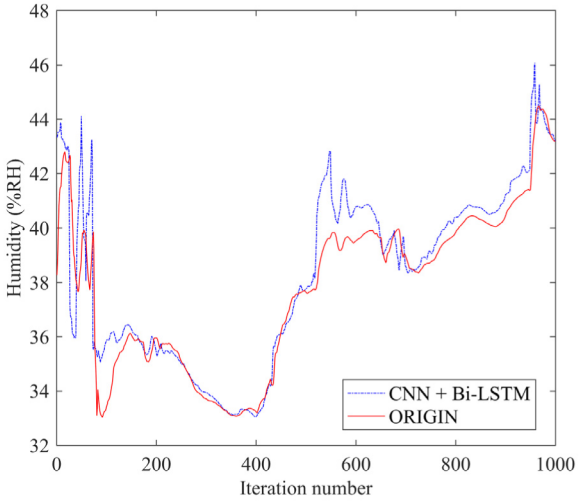


FIGURE 10. Using node 3 and 6 to predict humidity of node 4.

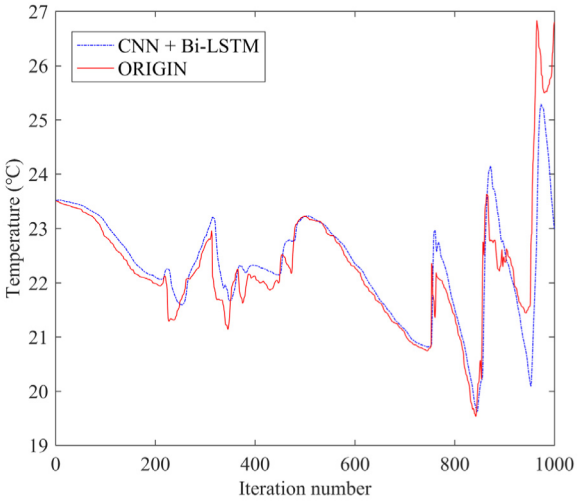


FIGURE 11. Using node 7 and 10 to predict temperature of node 4.

prediction and correlation coefficient, node 7 and 10 are used for prediction. The results of predicting the temperature of node 4 with node 7 and 10 are shown in Fig. 11.

In Fig. 11, the red solid line is the original data, the blue dotted line is the predicted data, abscissa of Fig. 11 is the number of iterations, and the ordinate is the temperature in degrees Celsius. the prediction error is small at the beginning of the iteration, but the error increases significantly in the last 200 iterations. There are two main reasons for this phenomenon. First, there is continuous accumulation of errors in the iteration. Second, some nodes may meet unusual situations. Fig. 12 below shows the prediction for humidity data in node 4, which uses node 7 and 10 to assist the predicting process.

In this paper, several node combinations are used to test the predictive model. The prediction error is quantified by 4 evaluation indicators including RMSE (Root Mean Square Error), MAPE (Mean Absolute Percentage Error),

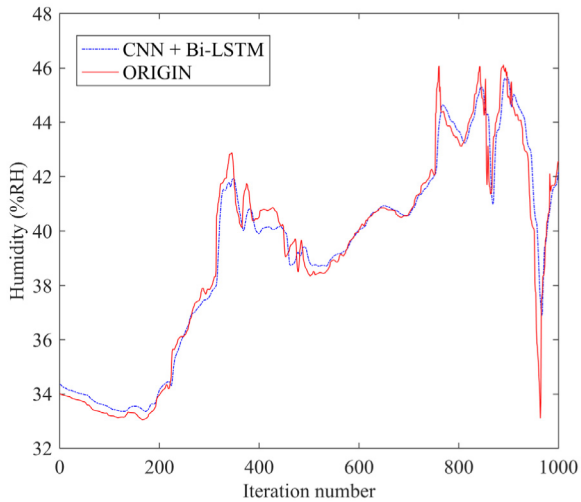


FIGURE 12. Using node 7 and 10 to predict humidity of node 4.

MAE (Mean Absolute Error), and R-square. RMSE is sensitive to few outliers and it is suitable for measuring the stability of prediction. MAPE calculates the ratio of the prediction error to the true value and it is suitable for measuring the relative error of prediction. MAE calculates the average prediction errors, which is insensitive to few outliers compared to RMSE. R-square measures the fitness of the predicted value to the original value.

TABLE 4. Prediction error with temperature.

Nodes	RMSE	MAPE	MAE	R^2
4.2.3	0.271	0.833	0.195	0.981
4.3.6	0.505	1.375	0.306	0.637
4.7.10	0.627	1.446	0.331	0.681
4.8.9	0.755	2.005	0.457	0.534

TABLE 5. Prediction error with humidity.

Nodes	RMSE	MAPE	MAE	R^2
4.2.3	0.492	0.81	0.333	0.972
4.3.6	1.154	1.811	0.714	0.872
4.7.10	0.714	1.108	0.437	0.964
4.8.9	1.844	3.86	1.213	0.899

Table 4 shows the prediction error for the temperature data, and Table 5 shows the errors of predicting humidity, the above four evaluation indicators are used to quantify the error in different aspects.

According to the prediction errors in the above two tables and the correlation between these nodes, it is known that when the correlation of chosen nodes are high, the prediction

TABLE 6. Prediction error with low correlation.

Nodes	RMSE	MAPE	MAE	R^2
4.33.35	0.731	2.487	0.545	0.167
4.1.33	0.787	2.868	0.637	0.676
4.32.34	1.157	3.215	0.766	0.645
4.7.32	1.128	3.159	0.736	0.321
4.30.31	1.174	3.324	0.778	0.608
4.11.12	1.333	3.447	0.809	0.162

errors becomes low. In order to further study the relationship between prediction error and correlation coefficient, we choose temperature as the example to carry out the prediction experiment under low correlation. The experimental results are shown in Table 6.

We study two factors that affect the prediction result, i.e., distance and correlation. From the above prediction experiment of various node combinations, Fig. 13 is shown to explain the relationship between prediction error and the correlation. The two connected points are the nodes of a combination, the node with low correlation is drawn left with red, and the node with high correlation is drawn right with blue, as we can see in Fig. 13.

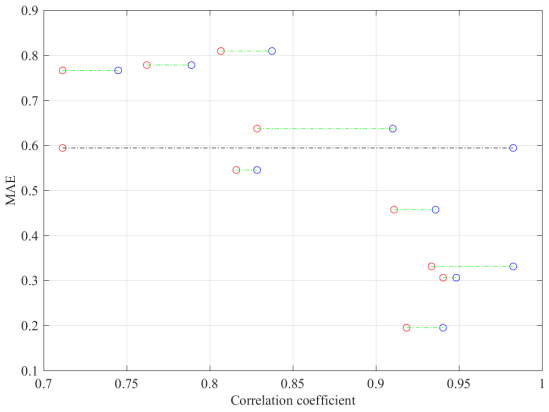


FIGURE 13. Relationship between correlation coefficient and error.

The above experiments show that there is a relationship between node correlation and prediction error, and the prediction is more sensitive to the node with lower correlation. The distance between nodes is another factor that may affect the prediction. Based on the combinations in Table 4. The relation between the distance and the prediction error is shown in Fig. 14.

As we can see, the two nodes connected by dashed line have the highest and lowest correlation coefficients among all the selected nodes separately. The correlation coefficients between node correlations and prediction error are calculated to quantify their relationship. The correlation coefficient between node with lower correlation and the prediction error

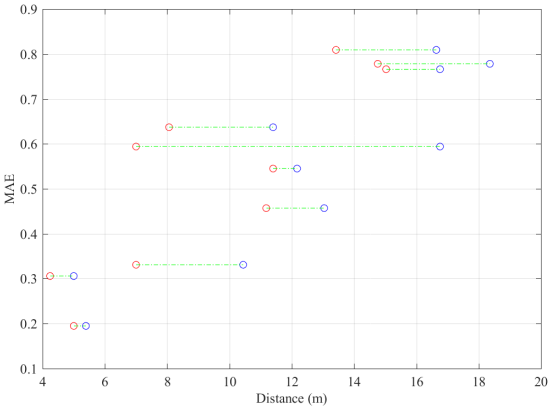


FIGURE 14. Relationship between correlation distance and error.

is −0.8198, and the correlation coefficient between node with higher correlation and the prediction error is −0.7291, so the node with lower correlation has a stronger impact on the prediction error. When selecting nodes to predict the lost data, we shall try to avoid the cask effect.

C. PERFORMANCE COMPARISION

In this section, Convolutional Neural Networks, Bidirectional LSTM, and Gated Recurrent Unit Network are used to compare the performance.

1) Convolutional neural network (CNN): A feedforward neural network that uses convolution calculations. Its one-dimensional form has a lot of applications in natural language processing or regression of time series data.

2) Bidirectional LSTM (Bi-LSTM): A neural network that has many applications in natural language processing and temporal data prediction, including sentiment analysis, speech recognition, traffic flow prediction, and so on.

3) Gated Recurrent Unit Network (GRU): A neural network based on RNN, which is similar to the LSTM and has some great applications in processing time series data.

In order to make a comprehensive evaluation on the prediction, multiple evaluation indicators are used, which are RMSE, MAPE, MAE, R-square. The calculated evaluation indicators of comparative experiment are shown in Table 7:

TABLE 7. Prediction evaluation of compared models.

	RMSE	MAPE	MAE	R^2
Proposed model	0.627	1.446	0.331	0.681
CNN	0.739	1.743	0.398	0.557
Bi-LSTM	0.714	1.669	0.385	0.587
GRU	0.802	2.011	0.464	0.478

Table 7 shows the prediction of the proposed model, CNN, Bi-LSTM, and GRU under the same data set. Taking an iterative prediction of 1000 time steps as an example, the

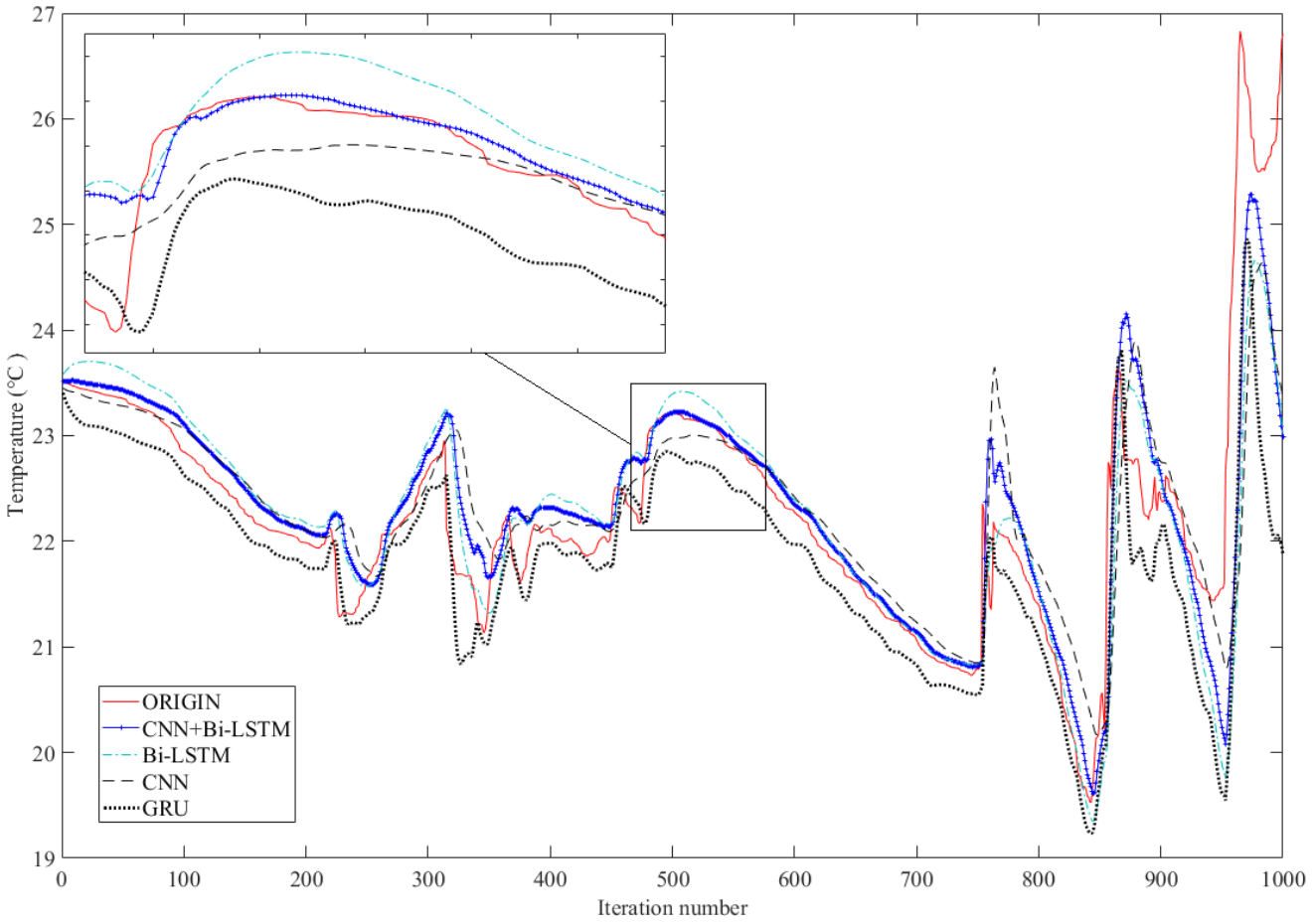


FIGURE 15. **Prediction results of compared models.**

prediction of proposed model, CNN, Bi-LSTM, and GRU are shown in Fig. 15.

VII. CONCLUSION

In wireless sensor networks, sensor nodes collect a large number of sensory data, some of these data have similar variations that can be used to build a multi-step predictive model. Three parallel structures base on 1-D CNN and Bi-LSTM are used to extract abstract features in sensory data, and they are combined into the merge layer for data prediction. Once the model is well trained, we use the predictive model iteratively to get the result of multi-step prediction. The experiments show that the relationships between prediction error and correlation coefficient, distance are exist, and the proposed model can make a stable and accurate prediction on temperature and humidity data in short and medium-term.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.

[2] I. Khan, F. Belqasmi, R. Glitho, N. Crespi, M. Morrow, and P. Polakos, "Wireless sensor network virtualization: A survey," *IEEE Commun. Surveys Tut.*, vol. 18, no. 1, pp. 553–576, 1st Quart., 2015.

[3] B. Prabhu, R. Mahalakshmi, S. Nithya, P. D. Manivannan, and S. Sophia, "A review of energy efficient clustering algorithm for connecting wireless sensor network fields," *Int. J. Eng. Res. Technol.*, vol. 2, no. 4, pp. 477–481, Apr. 2013.

[4] O. Ojuroye, R. Torah, S. Beeby, and A. Wilde, "Smart textiles for smart home control and enriching future wireless sensor network data," *Sensors Everyday Life*, vol. 22, pp. 159–183, Oct. 2016.

[5] S. Z. Erdogan and T. T. Bilgin, "A data mining approach for fall detection by using k-nearest neighbour algorithm on wireless sensor network data," *IET Commun.*, vol. 6, no. 18, pp. 3281–3287, Dec. 2012.

[6] S. Boubiche, D. E. Boubiche, A. Bilami, and H. Toral-Cruz, "Big data challenges and data aggregation strategies in wireless sensor networks," *IEEE Access*, vol. 6, pp. 20558–20571, 2018.

[7] X. Li, X. Tao, and Z. Chen, "Spatio-temporal compressive sensing-based data gathering in wireless sensor networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 2, pp. 198–201, Apr. 2018.

[8] K. Li, W. Ni, L. Duan, M. Abolhasan, and J. Niu, "Wireless power transfer and data collection in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2686–2697, Mar. 2018.

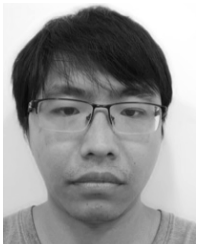
[9] P. Ganjewar, S. Barani, and S. J. Wagh, "A hierarchical fractional LMS prediction method for data reduction in a wireless sensor network," *Ad Hoc Netw.*, vol. 87, no. 1, pp. 113–127, 2019.

[10] R. A. Avinash, H. R. Janardhan, S. Adiga, B. Vijeth, S. Manjunath, S. Jayashree, and N. Shivashankarappa, "Data prediction in wireless sensor networks using Kalman filter," in *Proc. Int. Conf. Smart Sensors Syst. (IC-SSS)*, Bengaluru, India, Dec. 2015, pp. 1–4.

[11] H. Nazaktabar, K. Badie, and M. N. Ahmadabadi, "RLSP: A signal prediction algorithm for energy conservation in wireless sensor networks," *Wireless Netw.*, vol. 23, no. 3, pp. 919–933, 2017.

[12] A. Russo, F. Verdier, and B. Miramond, "Energy saving in a wireless sensor network by data prediction by using self-organized maps," *Proc. Comput. Sci.*, vol. 130, pp. 1090–1095, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918305271
[13] T. Kawamura, S. Ryuo, and N. Iwasawa, "Power consumption prediction method for train-health monitoring wireless sensor networks," *Electron. Commun. Jpn.*, vol. 101, no. 6, pp. 24–32, 2018.
[14] R. Lajara, J. J. Pérez-Solano, and J. Pelegrí-Sebastiá, "Predicting the batteries' state of health in wireless sensor networks applications," *IEEE Trans. Ind. Electron.*, vol. 65, no. 11, pp. 8936–8945, Nov. 2018.
[15] Y. Yue, J. Li, H. Fan, Q. Qin, L. Gu, and L. Du, "Fault prediction based on the kernel function for ribbon wireless sensor networks," *Wireless Pers. Commun.*, vol. 97, no. 3, pp. 3277–3292, 2017.
[16] M. M. Alves, L. Pirmez, S. Rossetto, F. C. Delicato, C. M. de Farias, P. F. Pires, I. L. dos Santos, and A. Y. Zomaya, "Damage prediction for wind turbines using wireless sensor and actuator networks," *J. Netw. Comput. Appl.*, vol. 80, pp. 123–140, Mar. 2017.
[17] A. Cammarano, C. Petrioli, and D. Spenza, "Online energy harvesting prediction in environmentally powered wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 17, pp. 6793–6804, Sep. 2016.
[18] X. Zhang, F. Chen, and R. Huang, "A combination of RNN and CNN for attention-based relation classification," *Procedia Comput. Sci.*, vol. 131, pp. 911–917, 2018.
[19] C. W. Tian, Y. Xu, and L. K. Fei, "Enhanced CNN for image denoising," *CAAI Trans. Intell. Technol.*, vol. 4, no. 1, pp. 17–23, 2019.
[20] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 11, pp. 999–1016, 2018.
[21] X. Qing and Y. Niu, "Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM," *Energy*, vol. 148, pp. 461–468, Apr. 2018.
[22] Intel. (2004). *Intel Lab Data*. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html
[23] Y. Yao and Z. Huang, "Bi-directional LSTM recurrent neural network for Chinese word segmentation," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2016, pp. 345–353.

HONGJU CHENG (M'11) received the B.E. and M.E. degrees in EE from the Wuhan University of Hydraulic and Electric Engineering, in 1997 and 2000, respectively, and the Ph.D. degree in computer science from Wuhan University, in 2007. Since 2007, he has been with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He has published almost 60 papers in international journals and conferences. His research interests include mobile ad hoc networks, wireless sensor networks, and wireless mesh networks. He is serving as an Editor or a Guest Editor for several international journals.



ZHE XIE was born in Jiuquang, Gansu, China, in 1994. He received the B.S. degree in software engineering from Fuzhou University, China, in 2017, where he is currently pursuing the M.S. degree. His research interests include mobile ad hoc networks, wireless sensor networks, and wireless mesh networks.

YUSHI SHI was born in Quanzhou, Fujian, China, in 1996. She received the B.S. degree in software engineering from Fuzhou University, China, in 2018, where she is currently pursuing the M.S. degree. Her research interests include mobile ad hoc networks, wireless sensor networks, and wireless mesh networks.

NAIXUE XIONG received the Ph.D. degrees in software engineering from Wuhan University and in dependable networks from the Japan Advanced Institute of Science and Technology, respectively.

Before he attends Colorado Technical University, he was with the Wentworth Technology Institution, Georgia State University, for many years. He is currently a Professor with the Department of Computer Science, Southwestern Oklahoma State University, USA. Dr. Xiong published more than 100 international journal papers and international conference papers. His research interests include cloud computing, security and dependability, parallel and distributed computing, networks, and optimization theory.

He is a Senior Member of the IEEE Computer Society. He has been a General Chair, a Program Chair, a Publicity Chair, a PC Member, and an OC Member of more than 100 international conferences. He is the Chair of Trusted Cloud Computing Task Force, the IEEE Computational Intelligence Society, and the Industry System Applications Technical Committee. He is serving as an Editor-in-Chief, an Associate Editor, or an Editor Member for more than ten international journals.