

RESEARCH

Open Access

# Data prediction model in wireless sensor networks based on bidirectional LSTM



Hongju Cheng<sup>1,2</sup>, Zhe Xie<sup>1</sup>, Leihuo Wu<sup>1</sup>, Zhiyong Yu<sup>1\*</sup>  and Ruixing Li<sup>3</sup>

## Abstract

The data collected by the wireless sensor nodes often has some spatial or temporal redundancy, and the redundant data impose unnecessary burdens on both the nodes and networks. Data prediction is helpful to improve data quality and reduce the unnecessary data transmission. However, the current data prediction methods of wireless sensor networks seldom consider how to utilize the spatial-temporal correlation among the sensory data. This paper has proposed a new data prediction method multi-node multi-feature (MNMF) based on bidirectional long short-term memory (LSTM) network. Firstly, the data quality is improved by quartile method and wavelet threshold denoising. Then, the bidirectional LSTM network is used to extract and learn the abstract features of sensory data. Finally, the abstract features are used in the data prediction by adopting the merge layer of the neural network. The experimental results show that the proposed MNMF model has better performance compared with the other methods in many evaluation indicators.

**Keywords:** Wireless sensor networks, Data prediction, Spatial-temporal correlation, LSTM

## 1 Introduction

The Internet of Thing (IoT) has developed rapidly in recent years, in which the wireless sensor network is becoming popular with low energy consumption, multi-function and large-scale deployment by sensing, collecting, processing, and transmitting the sensory data through cooperation between nodes [1, 2]. However, the number of data transmission between common nodes and sink nodes will increase significantly together with network size explosion, which possibly leads to data congestion, and accordingly high loss rate of sensory data and low signal-noise ratio [3–5]. Using data prediction methods to reduce unnecessary data transmission is an effective way to improve the quality of data collection and increase the network lifetime. The current methods usually use the periodicity and redundancy to predict the specific sensory data based on historical data, which often results in low prediction stability and biased predictions [6–11].

Data correlation among the sensory data is helpful to recover the lost data. For example, the temporal correlation can be observed in case that the physical

environment condition changes in a continuous way. On the one hand, the value of sequential sensory data for one single node is generally continuous when the collection duration is small enough. On the other hand, the sensors are deployed to observe the similar physical or environmental conditions; the collected data is generally spatial correlated. This similarity among the data tendency can be used to support the prediction process in a more relatively accurate and stable way. By exploiting these correlations among the sensory data, the impact of abnormal data on the prediction can also be weakened. The prediction process can support the end-users to predict the periodic change of the monitoring object or area and thus makes it possible to control the potential risk of the monitored object or area.

The prediction model needs to take into account the structure of sensory data and find the main factors which play important roles during the prediction process. These factors can be described as following: (1) time correlation—sensory data has periodicity and it has a dependency on its historical data; (2) spatial correlation—sensory data of wireless sensor node has a dependency on its surrounding node data; (3) data quality control—some of the sensory data is lost or a

\* Correspondence: [yuzhiyong@fzu.edu.cn](mailto:yuzhiyong@fzu.edu.cn)

<sup>1</sup>College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China

Full list of author information is available at the end of the article

noised version compared with the original value, and data quality can be improved by data preprocessing.

In recent years, deep learning has developed rapidly. The recurrent neural network (RNN) has many good applications in speech recognition, machine translation, and time-series data prediction because of its memory ability. Long short-term memory (LSTM) neural network is based on the development of RNN. It has good performance in processing long-term dependencies of time series data and predicting long-interval events [12–15]. Using the LSTM neural network to extract and fuse high-quality sensory data with spatial-temporal correlation can improve the efficiency and accuracy of the prediction model. Therefore, how to use or improve the above three factors and select good neural network model architecture to improve prediction accuracy has become an important issue that needs further study.

## 2 Related works

Data prediction can be used in many applications including data prediction in wireless sensor networks, traffic flow prediction, weather prediction, financial prediction, and disaster early warning.

Song et al. [6] proposed a wireless sensor network data prediction model PLB based on periodicity and linear relationship. The model used the large amount of redundancy in the data to predict future data and reduce the transmission of predictable data. Yang and Tsai [7] proposed a link stability prediction model based on current link relationships and user information. The prediction result could be used for link performance prediction, system performance analysis, service quality prediction, and route search applications. Kolodziej and Xhafa [8] proposed an activity-based method Markov chain model to define and predict the human movement patterns. Then, they used the Nonparametric Belief Propagation technique for prediction of the areas that would be visited and those that would not in the future. Liu et al. [9] proposed a microclimate data prediction model based on the extreme learning machine. The model is oriented to improve the prediction speed while ensuring accurate prediction. Sinha et al. [10] proposed a data aggregation model TDPA based on time data prediction. The model generates an estimate of future data to analyze the prediction error and uses the predicted value to save transmission energy consumption when the prediction meets a predefined threshold. Spenza et al. [11] proposed an energy prediction model called Pro-Energy. The prediction model gets good results in short and medium-term predictions by using historical energy observation. The above methods do not utilize the spatial-temporal correlation among the sensory data and do not make a quantitative analysis for the dependencies between nodes.

Weather prediction or disaster early-warning models based on deep learning have become popular in recent years. Tian and Chen proposed a neural network-based multivariate correspondence analysis model (MCA-NN) for disaster information monitoring. The model aims to improve the detection results by combining features from multivariate shallow learning models [16]. Zhang et al. use cellular neural networks to predict the degree of desertification. The Ruoqiang Basin is used as an example to predict the trend of land desertification from 2000 to 2011, and the experiment shows that the model is better than others [17]. Traore et al. proposed a method based on artificial neural network to predict the recent irrigation requirement. The paper uses the multi-layer perceptron model to extract the climate information retrieved from the public weather forecast to predict the recent crop evapotranspiration [18]. Biswas et al. proposed a multi-weather attribute model to predict weather based on nonlinear autoregressive neural networks. In this paper, the weather seasonality is captured, and the nonlinear autoregressive neural network is used to map the nonlinear relationship of weather data to obtain reliable prediction results [19].

Financial prediction has become popular in recent years, and it provides a method different from traditional financial models. Shah and Liao proposed a stock forecasting method based on event sentiment analysis. The model extracts the emotional sentiment of stock events in social media and aggregates daily sentiment trends to predict subsequent stock market trends [20]. Dong et al. proposed an error constraint algorithm based on the single-step prediction model by finding better weights and deviations. The experiment shows the proposed model accumulates less error in multi-step predictions than others [21]. Chen and Du proposed a stock forecasting method that combines sentiment analysis and online social behavior analysis. By constructing social behavior graphs and calculating key features, it finds the correlation between transaction volume or price and these features [22]. Wang et al. used delayed neural network models to predict public housing prices in Singapore. Nine independent economic and demographic variables are used to predict the trend of the resale price index (RPI). The results show that the proposed prediction model produces a good fit [23]. Teye and Ahelegbey used the Bayesian Graphical Vector Autoregression to research the spatial-temporal relationship between house prices in twelve provinces of the Netherlands. The result shows the house price diffusion patterns in the Netherlands and the patterns [24]. Che-Yu Lee used recurrent convolutional neural networks (RCN) to predict stock price. The proposed prediction model combines convolutions, word embedding, and sequence modeling to

extract information from financial news, then the technical analysis indicators are added to predict the stock price [25].

Traffic flow prediction needs to consider the surrounding environment and the periodicity of traffic flow. The traffic flow data has strong spatial-temporal correlations, and this correlation is similar to the spatial-temporal correlation in wireless sensor networks. Lv et al. proposed a self-encoder based on spatial-temporal correlation to learn the traffic flow feature. The experiments show that the spatial-temporal-based prediction model has better performance [26]. Huang et al. proposed a depth framework that combines multi-task learning. By using the weight sharing in the depth framework, a grouping method based on top-level weights is proposed to make the prediction model more efficient [27]. Fu et al. proposed a model for predicting traffic flow using long short-term memory networks (LSTM). They compare the performance of ARIMA and LSTM in predicting traffic flow problems and prove that LSTM has certain advantages in traffic flow prediction [28]. Dai et al. proposed a deep learning model Deep-Trend for traffic flow prediction. The model consists of an extraction layer and a prediction layer, in which the extraction layer is used to extract the trends of raw data and the trends are used by the prediction layer to make predictions [29].

### 3 Data preprocessing

This paper uses Intel indoor dataset [30] to study the data prediction problem in the wireless sensor networks.

The dataset was collected by Intel Berkeley Research Laboratory using Mica2Dot sensors in 2004 with the TinyDB in-network query processing system built on the TinyOS platform. The dataset contains 2.3 million pieces of sensory data collected by 54 nodes, including date, time, timestamp, node id, temperature, humidity, light, and voltage. Figure 1 shows the location distribution of 54 sensor nodes. Each sub-area has multiple sensor nodes to collect sensory data.

Node failure or data transmission errors sometimes occur in wireless sensor networks. In order to avoid the impact of abnormal data on the data prediction problem, this paper mainly deals with two forms of data outliers to improve data quality:

- *Global outlier*: The data deviates from the range of the entire dataset.
- *Local outlier*: The data is with the range of entire dataset, but abnormal compared with its neighborhood.

Figure 2 shows an example to demonstrate these two kinds of data outliers. We consider the sine function with the normal data range as  $[-1, 1]$ . The value of node A is 1.2, which falls out of range  $[-1, 1]$ . In this case, we can say that the value of node A is a global outlier. On the other hand, the value of node B, 0, is regarded as a local outlier because it is in range of  $[-1, 1]$  and is abnormal with its neighboring data.

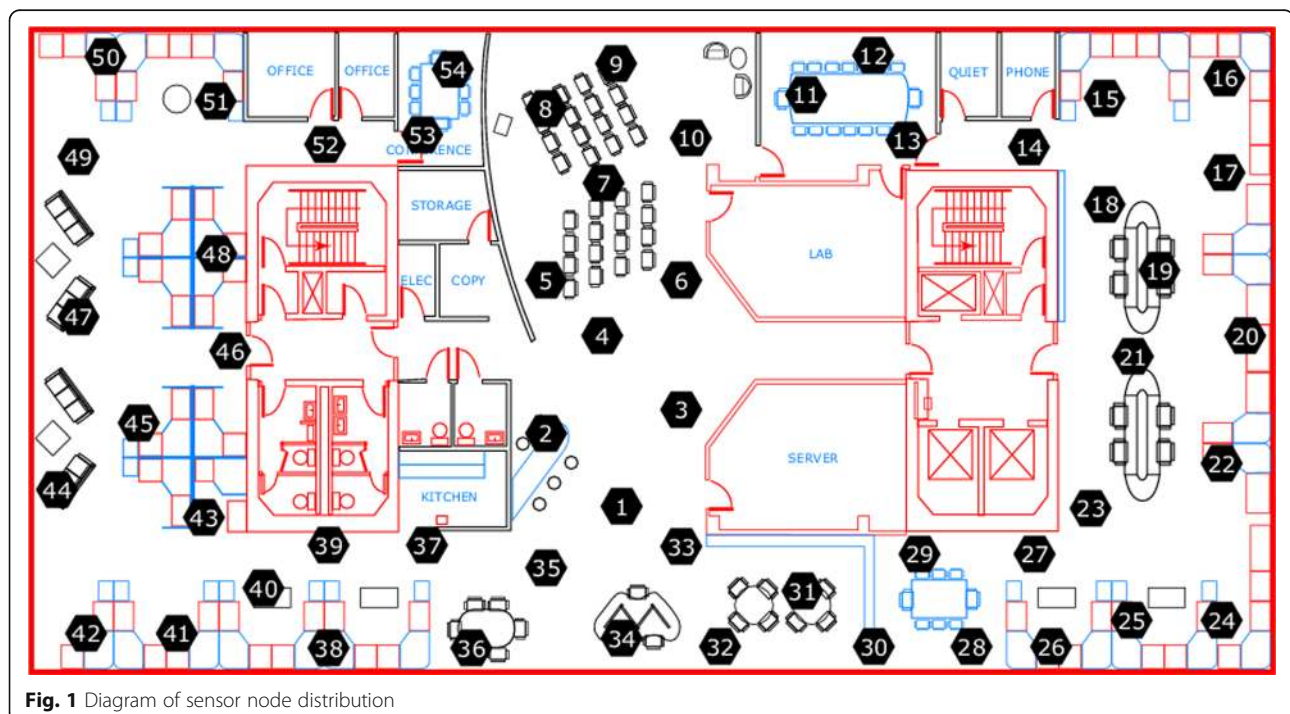
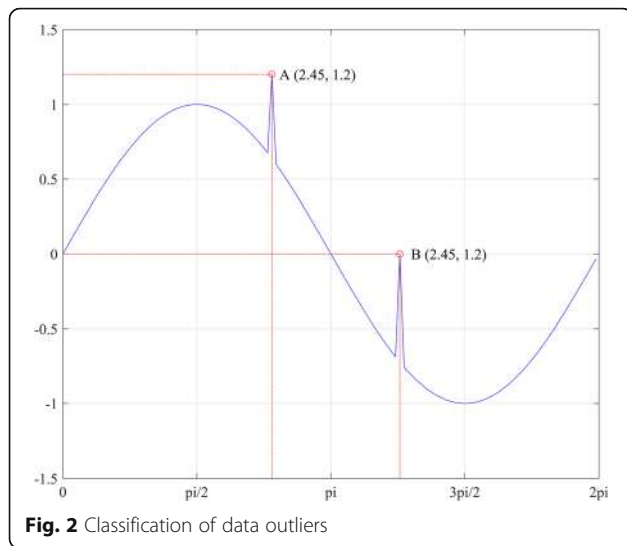


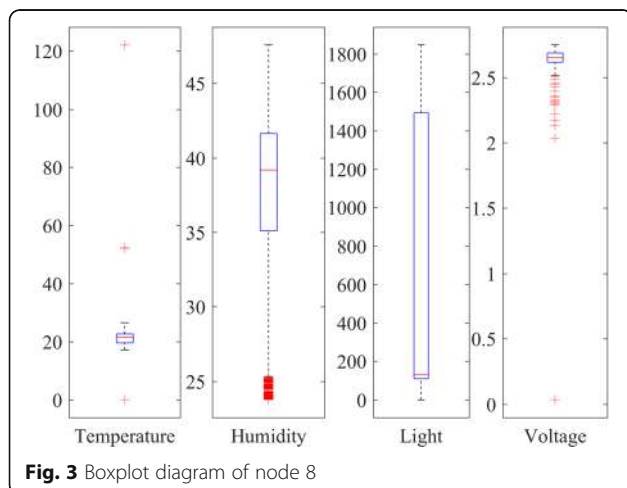
Fig. 1 Diagram of sensor node distribution



### 3.1 Global outliers processing

Global outlier has a great influence on [data normalization](#) and feature extraction, so it must be *removed* before using the neural network to extract features of sensory data. In this paper, the quartile method is used to process the global outliers. First, find the lower quartile (Q1), median (Q2), and upper quartile (Q3) in the sensory data. Then, calculate the interquartile range (IQR), where  $IQR = Q3 - Q1$ . Finally, calculate the lower fence ( $Q1 - 1.5IQR$ ) and upper fence ( $Q3 + 1.5IQR$ ), which are regarded as the lower and upper bound for the range of the entire dataset. In this way, the data which falls out of range  $[Q1 - 1.5IQR, Q3 + 1.5IQR]$  are considered as global outliers. In the Intel indoor dataset [30], there are four different types of data collected by the sensors, i.e., temperature, humidity, voltage, and light. We can obtain IQR and the upper and lower fences accordingly by calculating Q1, Q2, and Q3 for each given attribute.

Figure 3 shows the boxplot for these four different attributes of node 8. The whiskers represent lower and



upper fences for each given attribute, and the red line inside the box is the median value. The collected data which falls out of the fences are marked with label + which means that they are outliers. As we can see from Fig. 3, there are less outliers with the temperature attribute, while many outliers can be found with the voltage and humidity. Especially, most of the outliers of the humidity is close to the lower fence, while the outlier distribution of voltage is relatively scattered.

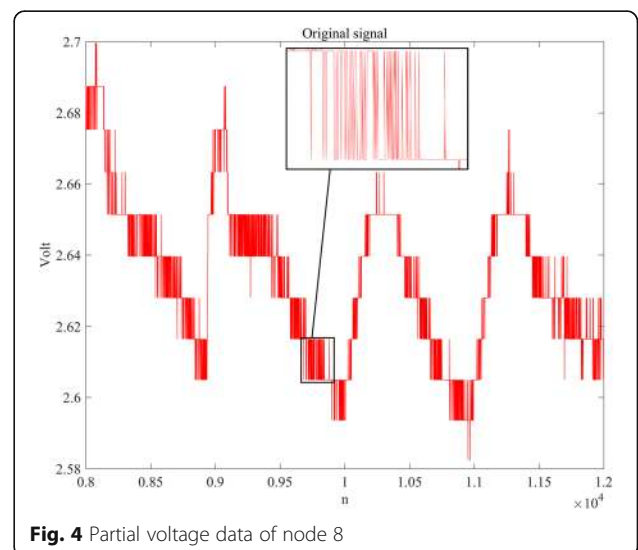
### 3.2 Local outliers processing

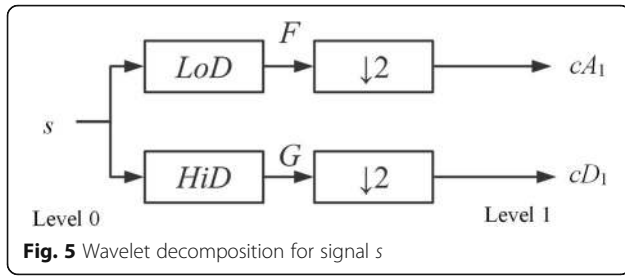
After the sensory dataset is processed by the quartile method, there are still a large number of local outliers, which generally appear different from their adjacent data although they are collected by the same node with the same attribute. The local outlier occurs sometimes due to the environmental noise which will influence the collected data in a random manner. Figure 4 shows the impact of noise on the data of node 8.

In order to reduce the noise influence on the data prediction problem, we adopt the wavelet threshold denoising to illuminate the noise in the original data. Wavelet threshold denoising can be divided into three sequential steps: wavelet decomposition, threshold acquisition, and wavelet denoising.

#### 3.2.1 Wavelet decomposition

Given a 1-dimensional signal, in this paper, we use the multi-level wavelet decomposition in which the decomposition level is set to 4 (which is generally proposed in [31]), to obtain the wavelet decomposition coefficient  $C$  and the coefficient length  $L$  which is used to calculate the threshold and while the multi-level decomposition is completed.





**Fig. 5** Wavelet decomposition for signal  $s$

Assuming that the input signal is  $s$ , the first step in the wavelet decomposition process of the signal  $s$  is shown in Fig. 5. HiD and LoD represent the high-pass and low-pass decomposition filter separately, and  $\downarrow 2$  presents the down-sampling process. In this way, the input signal  $s$  is converted to two outputs as  $cA_1$  and  $cD_1$ .

The decomposition process continues four times with the previous output  $cA_j$  as the input (Fig. 6). Finally, we can obtain the coefficients  $[cA_4, cD_4, cD_3, cD_2, cD_1]$  and the length  $L$  of each decomposition coefficient.

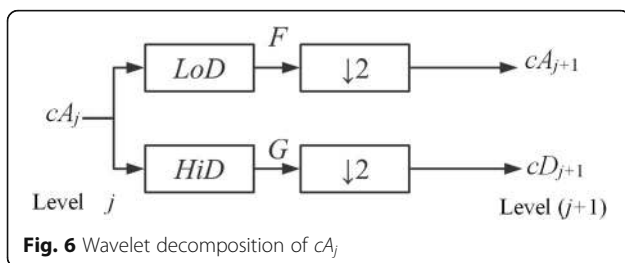
### 3.2.2 Threshold acquisition

In this paper, we use the unbiased risk estimation model to get the threshold of the one-dimensional wavelet transform. The threshold is calculated by the following steps:

- a. Obtain the absolute value of each element in the signal; then, sort all the absolute values from small to large; finally, square the sorted data to get a new signal  $f(k)$ , ( $k = 0, 1, 2, \dots, N - 1$ ).
- b. Calculate Risk ( $k$ ) with Eq. (1) for  $k = 0, 1, 2, \dots, N - 1$ :

$$\text{Risk}(k) = \frac{N-2k + \sum_{i=1}^k f(i) + (N-k)f(N-k)}{N} \quad (1)$$

- c. Find the minimum one among these Risk( $k$ ),  $k = 0, 1, 2, \dots, N-1$ , and let its square root be the final threshold  $\lambda$ .



**Fig. 6** Wavelet decomposition of  $cA_j$

### 3.2.3 Wavelet denoising

The soft wavelet threshold denoising method uses different thresholds for denoising in each layer. The calculation process is:

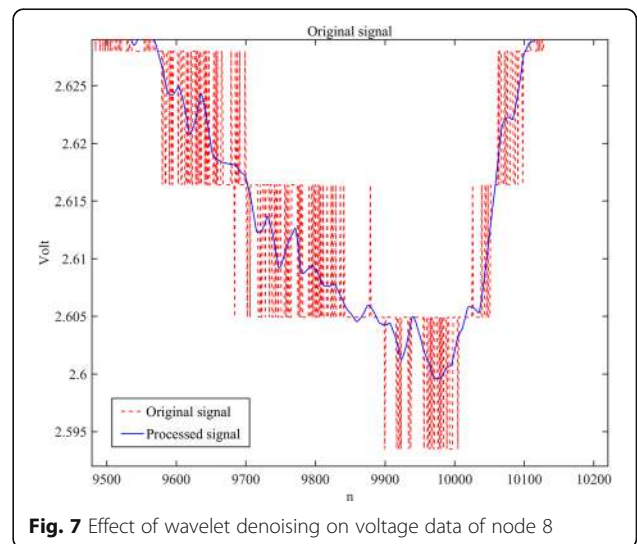
$$w_\lambda = \begin{cases} \text{sgn}(w)(|w|-\lambda), & |w| \geq \lambda \\ 0, & |w| < \lambda \end{cases} \quad (2)$$

where  $w$  is the wavelet coefficient,  $\lambda$  is the pre-selected threshold, and  $\text{sgn}(\cdot)$  is the sign function.  $w_\lambda$  is the wavelet coefficient filtered by the threshold function. Experiments have shown that the local outliers present are controlled after wavelet denoising [32–34]. Figure 7 shows the comparison between the original data and the data after wavelet denoising. Similar wavelet denoising process can be applied to different attributes of nodes in the network. In this way, we can finally get a wireless sensor network dataset with better data quality.

## 4 Correlation analysis

### 4.1 Data correlation in a single node

The Intel Indoor Dataset [23] contains a variety of sensory data collected by 54 nodes. In order to select appropriate sensory data to training the neural network and making the predictions reasonable, this paper takes the node 8 as an example to study the correlation of various sensory data and quantify the correlation. The sensory data used in this paper includes temperature, humidity, voltage, and light. Temperature is in degrees Celsius. Humidity is expressed in temperature corrected relative humidity, ranging from 0 to 100%. Light is in Lux, ranging from 0 to 2000. Voltage is in Volt, ranging from 2 to 3. Considering the different range of several sensory data, in order to extract the correlation features, this paper uses min-max normalization to linearly transform the sensory data to  $[0,$



**Fig. 7** Effect of wavelet denoising on voltage data of node 8

1]. The min-max normalization is calculated as shown in Eq. (3):

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

where  $x$  is the raw data,  $x_{\min}$  is the minimum of the dataset,  $x_{\max}$  is the maximum of the dataset, and  $x'$  is the normalized data. After the normalization process, all sensory data is mapped to [0, 1]. Figure 8 shows the normalized temperature and humidity data.

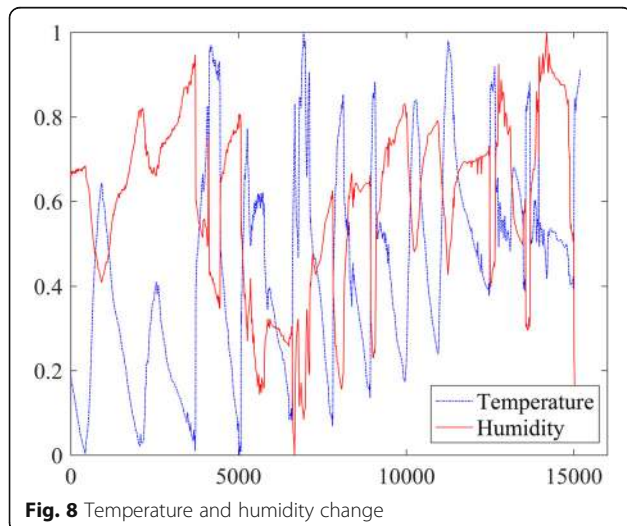
According to the Fig. 8, there is a correlation between temperature and humidity. In order to improve the accuracy of the correlation analysis, this paper uses the Spearman correlation coefficient to quantify the correlation. The Spearman correlation coefficient is calculated as shown in Eq. (4):

$$\rho = 1 - \frac{\sum 6d_i^2}{n(n^2 - 1)} \tag{4}$$

where  $d_i$  is the difference between the two ranks of each observation.  $n$  is the number of observations.

Using the calculation Eq. (4) of the Spearman correlation coefficient, the correlation coefficient of temperature and humidity in node 8 is  $\rho = -0.4830$ . The correlation between various types of sensory data according to this method is shown in Table 1.

Table 1 shows a strong correlation between temperature and humidity, temperature, and light, where temperature is negatively correlated with humidity and positively correlated with light. Humidity has a strong correlation with temperature and voltage, and humidity is negatively correlated with voltage.



**Fig. 8** Temperature and humidity change

**Table 1** Correlation coefficient of sensory data

Correlation	Temperature	Humidity	Voltage	Light
Temperature	1.0000	-0.4830	0.0415	0.5617
Humidity	-0.4830	1.0000	-0.5026	-0.2742
Voltage	0.0415	-0.5026	1.0000	0.2088
Light	0.5617	-0.2742	0.2088	1.0000

#### 4.2 Data correlation between multiple nodes

In order to get the spatial-temporal correlation between multi-node sensory data for neural network learning, this paper takes the node 8 as the center and selects the nearest node 7 and 9 to study the correlation of multi-node sensory data and quantizes it by Spearman’s correlation coefficient. The correlation of various types of sensory data among the three nodes is shown in Table 2.

The same type of sensory data under multiple nodes has a strong correlation, and the temperature, humidity, and voltage are most obvious. From the position of node 8 and node 7 and node 9 in the wireless sensor network, the correlation of light data is mainly affected by the distance between light source, the position of the shelter, and the orientation of the room. It is not suitable as a feature to train the spatial-temporal correlation-based prediction model.

Considering the sensory data correlation analysis of single-node and multi-node, this paper selects the temperature and humidity data of node 8 and the temperature data of nodes 7 and 9 as the input parameters of the spatial-temporal correlation-based prediction model, which is used to predict temperature data of node 8.

### 5 Method

This section describes the features learning process of prediction model based on the two-directional LSTM neural network which is named as multi-node multi-feature (MNMF) prediction model in this paper. As a special form of recurrent neural network (RNN), bidirectional LSTM neural network has a natural advantage in long-term memory [12–14]. Both LSTM and RNN have a chain structure consisting of a certain neural network module, which is called cell in LSTM. The cell consists of three gates: input gate, output gate, and forget gate. The structure of the cell used in this paper is as follows:

**Table 2** Correlation coefficient of multi-node

Correlation	Temperature	Humidity	Voltage	Light
No.8, no.7	0.9633	0.9726	0.9724	0.8674
No.8, no.9	0.9945	0.9909	0.9880	0.9180
No.7, no.9	0.9575	0.9661	0.9690	0.8461

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{6}$$

$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{7}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{8}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{9}$$

$$h_t = o_t * \tanh(C_t) \tag{10}$$

Equation (5) is the input gate process,  $h_{t-1}$  is the output of the previous cell,  $x_t$  is the current cell input,  $\sigma$  is the sigmoid function, and  $W_i$  and  $U_i$  are the input gate weights. Equation (6) is the function of forgot gate, which determines the information discarded in the cell, and  $W_f$  and  $U_f$  are the forgot gate weights. Equation (7) is a candidate memory unit that generates alternative updates. Equation (8) is the function of updating the cell state. The forgot gate decides what to be discarded in the old state information and adds the updated information to get the new state.  $W_c$  and  $U_c$  are the weights of the alternative new state, and  $*$  is the Hadamard product. Equations (9) and (10) are the output gate functions. Firstly, the sigmoid layer is used to determine the state of the cells to be output, then the updated cell state is processed by the tanh layer. Finally, the two parameters are multiplied to get the output, where  $W_o$  and  $U_o$  are weights of the output gate.

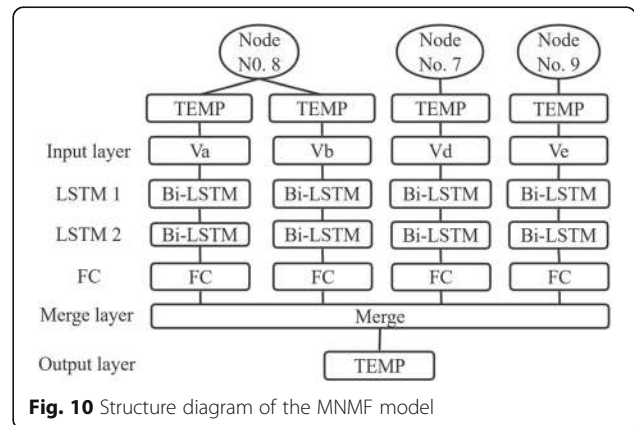
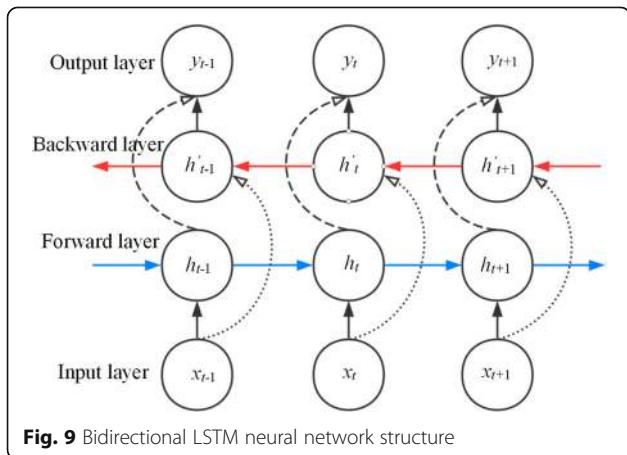
With the cell as the basic structure, this paper uses two-layer bidirectional LSTM neural network to construct the prediction model. Compared with the ordinary LSTM neural network, the bidirectional LSTM provides more local information to the network, which uses the forward and backward time series to get available information of timestamps in the past and future, so that it has better prediction result [15]. There is no direct connection between the backward layer and the forward layer in Fig. 9, ensuring that the expansion is acyclic. For the input layer

data  $x_t$ , the results of the forward and backward layers are combined at the output layer to get the output  $y_t$ . The basic structure of the bidirectional LSTM is shown in Fig. 9.

In wireless sensor networks, the sensory data collected by nodes has regional characteristics; in this way, the sensory data of different nodes have similar distribution patterns. Similarly, there is a correlation between different sensory data originated from the same node, which is represented by a positive or negative correlation between various sensory data. In this paper, the spatial-temporal correlation of multi-node sensory data is used to construct a wireless sensor network data prediction model. As an example, the MNMF model structure is shown in Fig. 10.

In Fig. 10, Va and Vb are the temperature and humidity data of node 8. Vd and Ve are the temperature data of nodes 7 and 9. To extract the spatial correlation between nodes, the timestamps of the node 8 needs to be exactly the same as nodes 9 and 7. LSTM1 is the first layer of bidirectional LSTM neural network that processes the input layer features and transmits them to the next layer. LSTM2 is the second layer of bidirectional LSTM neural network, which extracts abstract features from the previous layer. The FC is a fully connected layer, which performs the nonlinear transformation on the high-dimensional data in the previous layer. Merge is a fusion layer, which combines the abstract features of each node in the previous layer to predict temperature.

Since LSTM is used as the main structure of the prediction model, the shape of the input layer data needs to suit the parameter shape of the LSTM neural network, including the number of features input, the length of the time step, and the number of data. The stability and training speed of the prediction model need to be considered when choosing the number of bidirectional LSTM neural network nodes. Too few neural network nodes are likely to cause insufficient training and under-fitting, and too many neural network nodes are likely to cause over-fitting and increase the duration of the model



training. The length of the time step also has an effect on the prediction. The model dimensions adopted in this paper are shown in Table 3.

In Table 3, the first dimension of the input layer and the LSTM1 layer is determined by the time steps of the specified feature, the time steps of Va are 50, and the time steps of Vb, Vd, Ve are 10. The length of the data sequence used by the LSTM model is determined by the time step. Using the appropriate time steps for different features can make the prediction model get a relatively good prediction. In this paper, the mean square error (MSE) is used as a loss function to estimate the deviation. The calculation process is as shown in Eq. (11):

$$L(\theta) = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \tag{11}$$

where  $\theta$  represents all parameters in the model,  $y_i$  represents the true value, and  $\hat{y}_i$  represents the predicted value. In this paper, the model uses the backpropagation algorithm to train and uses the Adam algorithm as the optimizer to calculate and update the network parameters. The adopted batch\_size is set as 50, and the training ends when the epochs of the training exceed 200. The model training process is described in Table 4.

## 6 Results and discussion

### 6.1 Time step selection

In this paper, Tensorflow, Keras, and Matlab are used as the primary tools of the experiment, and GPU acceleration is used to train the model. By testing a variety of parameter configurations, it is found that the time step length has a great influence on the extraction efficiency of data features in MNMF. If the sequence is too short, the prediction model will get less information and can not make an accurate prediction. If the sequence is too long, the model will get too much information to extract useful feature from data. In this paper, 10, 20, 50, 100, 200, and other specific time steps are used as the basic unit of the combination, the first 50% of the entire sensory dataset is used as the train set, and the last 50% is used as the test set to evaluate the prediction. Table 4 and 5 shows the relationship between the time step length

**Table 3** Feature shape of each layer in MNMF

Dimension	Va	Vb	Vd	Ve
Input layer	(50, 1)	(10, 1)	(10, 1)	(10, 1)
LSTM1	(50, 128)	(10, 128)	(10, 128)	(10, 128)
LSTM2	64	64	64	64
FC	16	16	16	16
Merge layer	16			
Output layer	1			

**Table 4** Training process of MNMF model

<b>Input:</b> Sensory data: [Va, Vb, Vd, Ve], Timestep data: [TVa, TVb, TVd, TVe].
<b>Output:</b> Parameters of each layer.
<b>//Input preparation</b>
Sensory data is arranged in chronological order.
Va = [Va <sup>1</sup> , Va <sup>2</sup> , ..., Va <sup>TVa-1</sup> , Va <sup>TVa</sup> ]
Vb = [Vb <sup>1</sup> , Vb <sup>2</sup> , ..., Vb <sup>TVb-1</sup> , Vb <sup>TVb</sup> ]
Vd = [Vd <sup>1</sup> , Vd <sup>2</sup> , ..., Vd <sup>TVd-1</sup> , Vd <sup>TVd</sup> ]
Ve = [Ve <sup>1</sup> , Ve <sup>2</sup> , ..., Ve <sup>TVe-1</sup> , Ve <sup>TVe</sup> ]
<b>//Initialization</b>
batch_size = 50, EpochBound = 200, currentEpoch = 0
<b>//Training model</b>
While currentEpoch < EpochBound
train_Data_num = 0;
While train_Data_num < len(trainSet)
Select (batch_size) of data to update parameters;
train_Data_num = train_Data_num + batch_size;
End While
Calculate the loss value(val_loss) on the testSet;
End While

**Table 5** Prediction deviation of multi-node multi-feature model

Timesteps	Va	Vb	Vd	Ve	RMSE
100	200	10	10	10	0.201
	100	10	10	10	0.160
	100	20	10	10	0.172
	100	20	20	20	0.248
	100	10	20	20	0.424
50	50	10	10	10	0.139
	10	10	10	10	0.290
	100	10	10	10	0.443
	50	10	10	10	0.101
	20	10	10	10	0.156
	10	10	10	10	0.261



of the multi-node multi-feature model (MNMF) and the prediction error, where  $V_a$  and  $V_b$  are the temperature and humidity of node 8, and  $V_d$  and  $V_e$  are the temperatures of nodes 7 and 9.

The RMSE in Table 4 and 5 is the root-mean-square error, which is calculated as the square root of the Eq. (10). The root-mean-square error is used to measure the accuracy of the predicted value. This paper uses a variety of batch size training models and then compares them. Through the change of time step length and RMSE, it can be known that selecting a reasonable time step length is an effective way to improve the prediction effect. The prediction deviation of multi-node multi-feature model is described in Table 5.

### 6.2 Feature selection

In the dataset used in this paper, there are many kinds of sensory data that can be used for prediction. According to the correlation between data and experimental results, the MNMF prediction model selects four features for prediction, in which  $V_b$  represents the temporal correlation between the data to be predicted and other sensory data of the same node, and  $V_d$  and  $V_e$  represent the spatial-temporal correlation between adjacent nodes.  $V_a$  represents the temporal correlation between the data to be predicted and its historical data. In addition, this paper also constructs two prediction models based on single-node multi-features and multi-node single-features. The parameter configuration is similar to the MNMF model. The combination of the chosen sensory data and the length of the time step is shown in Tables 6 and 7.

$V_a$  shown in Table 6 is the temperature data sequence of node 8,  $V_b$  is the humidity data sequence, and  $V_c$  is the light data sequence. Since the sensory data of a single node is used, this model is called a single node multi-feature model (SNMF), where `batch_size` is set to 100. Table 6 shows the experimental results of the single-node model at each time step length, two extra-sensory data are used to extract useful correlation features, and the influence of the time step on the prediction is reasonable.

**Table 6** Prediction deviation of single-node multi-feature model

$V_a$	$V_b$	$V_c$	RMSE
5	2	2	0.184
10	10	10	0.171
20	10	10	0.190
50	10	10	0.322
50	50	10	0.424

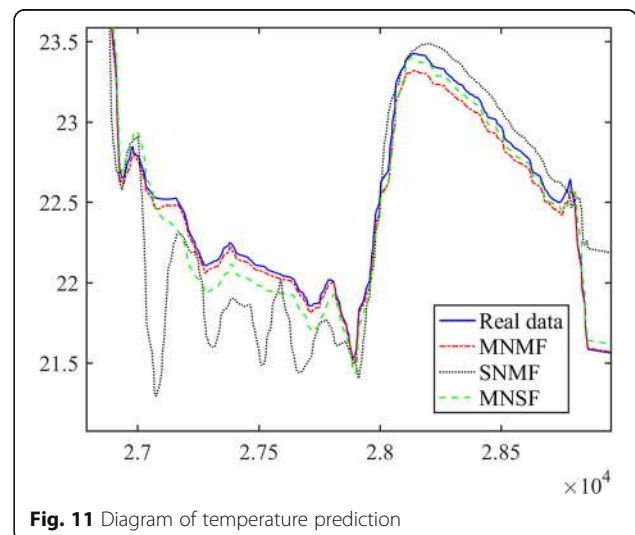
**Table 7** Prediction deviation of multi-node single-feature model

$V_a$	$V_d$	$V_e$	RMSE
5	2	2	0.174
10	10	10	0.159
20	10	10	0.140
50	10	10	0.213
50	50	10	0.305

$V_a$  in Table 7 is the temperature data of node 8,  $V_d$  is the temperature data of node 7, and  $V_e$  is the temperature data of 9. Because using the same kind of sensory data in multiple nodes to train, it is called a multi-node single-feature model (MNSF) in this paper, and the batch size is set to 100 in training. Table 7 shows a prediction result that only considering the correlation of the same kind sensory data in multiple nodes. Node 8 is in the same room with nodes 7 and 9 and they are close to each other so that the collected sensory data has a strong correlation. As shown in Table 2, the temperature data correlation coefficient between node 8 and node 7 is 0.9633, and the temperature data correlation between nodes 8 and 9 is 0.9945. The above results prove that constructing a prediction model using sensory data correlation in a wireless sensor network is an effective method. This paper combines the advantages of the above two models, constructing a multi-node multi-feature model (MNMF). Figure 11 shows the partial sensory data prediction of the above three models.

### 6.3 Comparative experiment

To verify the performance of the model (MNMF), three neural network prediction models were used to compare the performance in the simulations.



**Fig. 11** Diagram of temperature prediction

1. Elman neural network. It is a typical local forward network (global feed forward local recurrent). The Elman network can be seen as a recurrent neural network with local memory units and local feedback connections.
2. NARX (nonlinear autoregressive exogenous model). The nonlinear autoregressive exogenous model mainly consists of four layers: input layer, hidden layer, bearing layer, and output layer, wherein the bearing layer uses a nonlinear autoregressive model with exogenous input.
3. GRNN (general regression neural network). A generalized regression neural network is an artificial neural network that uses a radial basis function as an activation function, which is an improvement of the radial basis network.

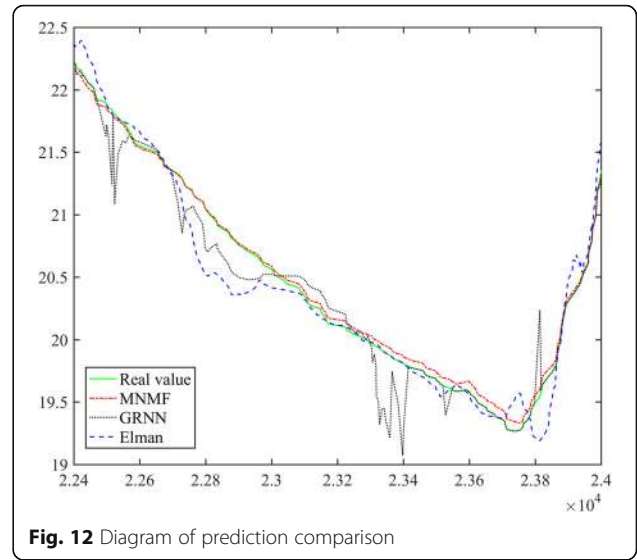
In order to improve the comprehensiveness of the evaluation, the root-mean-square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and  $R^2$  are used as evaluation indicators to evaluate the prediction model. RMSE is sensitive to outliers that appear in prediction errors, while outliers in prediction errors have a relatively small impact on MAE, so RMSE and MAE are both used to evaluate the prediction. MAPE shows the ratio between the error and the actual value, which can be used to measure errors in different orders of magnitude.  $R^2$  is used to measure how well the regression prediction approximates to actual data, which is necessary for regression. Multi-type of evaluation indicators can estimate the quality of model predictions better and avoid incomplete evaluation, so the above four indicators are both used to evaluate the predictions. Equation (12) is the calculation process of MAE, Eq. (13) is the calculation of MAPE, and Eqs. (14), (15), and (16) are the calculation of  $R$  square.

$$MAE(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \tag{12}$$

$$MAPE(y, \hat{y}) = \frac{100\%}{m} \sum_{i=1}^m \left| \frac{y_i - \hat{y}_i}{y_i} \right| \tag{13}$$

**Table 8** Predictive evaluation of multiple models

	RMSE	MAE	MAPE	$R^2$
MNMF	0.10095	0.07272	0.31804	0.99625
Elman	0.25578	0.14736	0.69806	0.98654
GRNN	0.18955	0.08076	0.32872	0.99193
NARX	0.34793	0.25057	1.17430	0.98090



**Fig. 12** Diagram of prediction comparison

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} \tag{14}$$

$$SS_{residual} = \sum_i (y_i - \hat{y}_i)^2 \tag{15}$$

$$SS_{total} = \sum_i (y_i - \bar{y})^2 \tag{16}$$

In the above equations,  $y_i$  is the true value,  $\hat{y}_i$  is the predicted value,  $\bar{y}$  is the average value, and  $m$  is the number of samples. MAPE is the percentage of prediction bias and true value. Because the data range of each type of data is different, the calculated error is very different among various types of data. The  $R^2$  can be interpreted as the ratio of the predicted mean square error to the data variance. It represents the fitness of the predicted value and the actual value. The calculated evaluation indicators are shown in Table 8.

The experiment shows that the MNMF model has a great advantage over Elman and NARX, and it has an advantage in RMSE and  $R^2$  when compared with GRNN. Figure 12 shows the partial temperature data prediction curves of MNMF, GRNN, and Elman. Since the NARX neural network is obviously weaker than other models in various indicators, no further comparison is made here. It can be seen from Figure 12 that the MNMF model has lower prediction error and the prediction is more stable than the other two models.

### 7 Conclusion

The sensory data in the wireless sensor network is collected by multiple sensors of different nodes, which shows the relative variation of several environmental factors in different regions. In this paper, we quantify the correlation features between different sensory data and construct a

sensory data prediction multi-node multi-feature (MNMF) model, based on bidirectional LSTM. The model considers three factors including the temporal correlation between the sensory data and its historical data, the spatial correlation of the sensory data between different nodes, and the low data quality caused by the transmission error of the sensor network. Firstly, the quartile method and wavelet threshold denoising method are used to improve the data quality. Then, the bidirectional LSTM neural network is used to learn the prediction features respectively. Finally, the merge layer of the neural network is used to fuse multiple data features to predict the specific sensory data. In this paper, Intel indoor dataset is used for experimentation. The experiments show that the proposed MNMF model has high prediction accuracy and reasonable prediction bias.

#### Abbreviations

GRNN: General regression neural network; IOT: Internet of things; IQR: Interquartile range; LSTM: Long short-term memory; MAE: Mean absolute error; MAPE: Mean absolute percent error; MNMF: Multi-node multi-feature data prediction model; MNSF: Multi-node single-feature data prediction model; MSE: Mean square error; NARX: Nonlinear autoregressive exogenous; RMSE: Root-mean-square error; RNN: Recurrent neural network; SNMF: Single-node multi-feature data prediction model

#### Acknowledgements

None.

#### Authors' contributions

HC proposed the framework of the data prediction strategy. Moreover, he also participated in the writing of this paper. ZX carried out the simulation and the initial idea of the MNMF. LW contributed to the data pre-processing. ZY discussed the framework and helped to improve the writing of this paper. RL provided the material and carried out the original data analysis. All authors read and approved the final manuscript.

#### Funding

This work is supported by the National Natural Science Foundation of China under Grand No. 61772136, 61370210, and the Science Foundation of Fujian Province under Grand No. 2019J01245.

#### Availability of data and materials

None.

#### Competing interest

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350116, China. <sup>2</sup>Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou, China. <sup>3</sup>Fujian College Applied Engineering Centre of Internet of Things, Minjiang Teachers College, Fuzhou 350000, China.

Received: 22 March 2019 Accepted: 5 July 2019

Published online: 13 August 2019

#### References

- P. Rawat, K.D. Singh, H. Chaouchi, J.M. Bonnin, Wireless sensor networks: a survey on recent developments and potential synergies. *Journal of Supercomputing* **68**(1), 1–48 (2014)
- T. Raitt, A. Bouabdallah, Y. Challal, Energy efficiency in wireless sensor networks: a top-down survey. *Computer Networks* **67**(8), 104–122 (2014)
- H. Xiao, S. Lei, Y. Chen, H. Zhou, *WX-MAC: An energy efficient MAC protocol for wireless sensor networks* (2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems, Hangzhou, 2013), pp. 423–424
- M.A. Razzaque, C. Bleakley, S. Dobson, Compression in wireless sensor networks. *Acm Transactions on Sensor Networks* **10**(1), 1–44 (2013)
- C.P. Chen, S.C. Mukhopadhyay, C.L. Chuang, M.Y. Liu, J.A. Jiang, Efficient coverage and connectivity preservation With Load Balance for Wireless Sensor Networks. *Sensors Journal IEEE* **15**(1), 48–62 (2015)
- Y. Song, J. Luo, C. Liu, W. He, *Periodicity-and-Linear-Based Data Suppression Mechanism for WSN* (2015 IEEE Trustcom/BigDataSE/ISPA, Helsinki, 2015), pp. 1267–1271
- K.J. Yang, Y.R. Tsai, *WSN18-2: Link Stability Prediction for mobile ad hoc networks in shadowed environments* (IEEE Globecom 2006, San Francisco, 2006)
- J. Kolodziej, F. Xhafa, *Utilization of Markov model and non-parametric belief propagation for activity-based indoor mobility prediction in wireless networks* (2011 International Conference on Complex, Intelligent, and Software Intensive Systems, Seoul, 2011), pp. 513–518
- Q. Liu, D. Jin, J. Shen, Z. Fu, N. Linge, *A WSN-Based Prediction Model of Microclimate in a Greenhouse Using an Extreme Learning Approach* (2016 18th International Conference on Advanced Communication Technology (ICACT), Pyeongchang, 2016)
- A. Sinha, D.K. Lobiyal, Prediction models for energy efficient data aggregation in wireless sensor network. *Wireless Personal Communications* **84**(2), 1325–1343 (2015)
- D. Spenza, C. Petrioli, A. Cammarano, *Pro-Energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks* (2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012), Las Vegas, 2012), pp. 75–83
- F. Karim, S. Majumdar, H. Darabi, S. Chen, LSTM fully convolutional networks for time series classification. *IEEE Access* **6**, 1662–1669 (2017)
- A. Graves, N. Jaitly, A.R. Mohamed, *Hybrid speech recognition with Deep Bidirectional LSTM* (2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, 2013), pp. 273–278
- K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, LSTM: a search space odyssey. *IEEE Transactions on Neural Networks & Learning Systems* **28**(10), 2222–2232 (2015)
- Y. Yao, Z. Huang, *Bi-directional LSTM recurrent neural network for Chinese word segmentation* (International Conference on Neural Information Processing, 2016), pp. 345–353
- H. Tian, S.C. Chen, *MCA-NN: multiple correspondence analysis based neural network for disaster information detection* (2017 IEEE Third International Conference on Multimedia Big Data (BigMM), Laguna Hills, 2017), pp. 268–275
- F.X. Zhang, G.D. Li, W.X. Xu, *Xinjiang Desertification disaster prediction research based on cellular neural networks* (2016 International Conference on Smart City and Systems Engineering (ICSCSE), Hunan, 2016), pp. 545–548
- S. Traore, Y.F. Luo, G. Fipps, Deployment of artificial neural network for short-term forecasting of evapotranspiration using public weather forecast restricted messages. *Agricultural Water Management* **163**(1), 363–379 (2016)
- S.K. Biswas, N. Sinha, B. Purkayastha, L. Marbaniang, Weather prediction by recurrent neural network dynamics. *International Journal of Intelligent Engineering Informatics* **2**(2/3), 166–180 (2014)
- M. Makrehchi, S. Shah, W. Liao, *Stock prediction using event-based sentiment analysis* (2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), Atlanta, 2013), pp. 337–342
- G. Dong, K. Fatallyev, L. Wang, One-step and multi-step ahead stock prediction using backpropagation neural networks (2013 9th International Conference on Information, Communications & Signal Processing, Tainan, 2013).
- Z. Chen, X. Du, Study of stock prediction based on social network (2013 International Conference on Social Computing, Alexandria, 2013), pp. 913–916.
- L. Wang, F.F. Chan, Y. Wang, Q. Chang, *Predicting public housing prices using delayed neural networks* (2016 IEEE Region 10 Conference (TENCON), Singapore, 2016), pp. 3589–3592
- A.L. Teye, D.F. Ahelegbey, Detecting spatial and temporal house price diffusion in the Netherlands: a Bayesian network approach. *Regional Science and Urban Economics* **65**, 56–64 (2017)
- C.Y. Lee, V.W. Soo, *Predict stock price with financial news based on recurrent convolutional neural networks* (2017 Conference on Technologies and Applications of Artificial Intelligence (TAAI), Taipei, 2017)

26. Y. Lv, Y. Duan, W. Kang, Z. Li, F.Y. Wang, Traffic flow prediction with big data: a deep learning approach. *IEEE Transactions on Intelligent Transportation Systems* **16**(2), 865–873 (2015)
27. W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems* **15**(5), 2191–2201 (2014)
28. R. Fu, Z. Zhang, L. Li, *Using LSTM and GRU neural network methods for traffic flow prediction* (2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, 2016), pp. 324–328
29. X. Dai, R. Fu, Y. Lin, L. Li, *DeepTrend: A deep hierarchical neural network for traffic flow prediction* (2017 20th International Conference on Intelligent Transportation Systems, Yakahama, 2017)
30. Intel. Intel Lab Data. <http://db.csail.mit.edu/labdata/labdata.html>. Accessed 19 Apr 2019.
31. K. Kannan, S.A. Perumal, *Optimal decomposition level of discrete wavelet transform for pixel based fusion of multi-focused images* (International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007), Sivakasi, 2007), pp. 314–318
32. R. F. Navea, E. Dadios, Classification of wavelet-denoised musical tone stimulated EEG signals using artificial neural networks (2016 IEEE Region 10 Conference (TENCON), Singapore, 2016), pp. 1503–1508.
33. J. Zhao, J. Huang, N. Xiong, An effective exponential-based trust and reputation evaluation system in wireless sensor networks. *IEEE Access* **7**, 33859–33869 (2019)
34. W. Guo, Y. Shi, S. Wang, N. Xiong, An unsupervised embedding learning feature representation scheme for network big data analysis. *IEEE Transactions on Network Science and Engineering*, 1–14 (2018)

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---