A Data Fusion Scheme for Wireless Sensor Networks Using Clustering and Prediction

Xiulan Yu, Hongyu Li(✉), Chenquan Gan, and Zufan Zhang

Chongqing Key Labs of Mobile Communications Technology,
Chongqing University of Posts and Telecommunications, Chongqing 400065, China
849645620@qq.com

Abstract. This paper intends to reduce the communication cost, while ensuring data prediction accuracy and data transmission efficiency in wireless sensor networks (WSNs). A data fusion scheme using clustering and prediction algorithms is proposed. Initially, nodes are clustered by using historical data, and then they are linked based on the actual geographical distance. Next, during the data fusion process, the base station and sensor nodes both use the online recurrent extreme learning machine (OR-ELM) to predict the future sensing data, which can guarantee that the data sequence in the base station and sensor nodes are synchronous. If the prediction fails, data will be transmitted to other nodes in the link and forwarded. Finally, experimental results reveal that the proposed data fusion scheme not only can effectively predict the sensor data, but also can reduce spatial and temporal redundant transmissions with low computational cost.

Keywords: Wireless sensor network · Data fusion · Clustering · Time-series prediction

1 Introduction

WSNs have been successfully applied in many fields [1], such as surveillance, environmental monitoring, smart city, and health care, so a large number of sensor nodes will perceive the information of various monitored objects and send this information to the base station and end-users [2]. Meanwhile, the requirements of data collection and data fusion are becoming increasingly higher.

In big data WSNs, the sensed data have a massive similarity and redundancy [3], and the spatial and temporal correlation of data can lead to large amounts of redundant transmissions and the waste of node energy [4]. As a result, clustered WSNs are the prominent network architecture for data reduction strategies [5]. In particular, each cluster uses a specific data compression strategy to perform a

Hongyu Li, M.S., Chongqing University of Posts and Telecommunications. His current main research interest includes: wireless sensor networks and extreme learning machine.

Q. Liang et al. (Eds.): CSPS 2018, LNEE 515, pp. 350–358, 2019.
https://doi.org/10.1007/978-981-13-6264-4_43

series of compression operations on the cluster head [6]. Additionally, a number of compression methods were proposed, such as compressed sensing (CS) [7]. Due to the limited storage space and computation capacity of sensors, CS is inefficient for data collection at cluster head nodes. Then, data prediction was discussed as the more efficient way to obtain data reduction in WSNs [8]. Autoregressive (AR) model was used to calculate the estimation of future perceived data. However, the shortcoming of this model is that communication cost would be very high when the error threshold was set to a small value. Afterward, the GM-OP-ELM scheme, which combined gray model (GM) and optimally pruned extreme learning machine (OP-ELM) was proposed [9], and [10] considered the GM-KRLS scheme, which consisted of GM and kernel recursive least squares (KRLS). It is easy to see that both of these algorithms are based on grey model prediction. Online sequential extreme learning machine (OS-ELM) is known to be a viable solution to time-series prediction [11]. And recently, a new variant of OS-ELM, called online recurrent extreme learning machine (OR-ELM), was proposed [12].

Inspired by the above-mentioned work, a data fusion scheme for WSNs using clustering and OR-ELM prediction algorithms is proposed. First, nodes are clustered by considering both the distance of nodes and trend similarity between the node data series, and strong links between them are established. Second, OR-ELM prediction algorithm is used to reduce sampling data points. Third, prediction-failed nodes find similar nodes through different links, and only one node sends the data in the link. Finally, experimental results show that the proposed data fusion scheme not only can effectively predict the sensor data, but also can reduce spatial and temporal redundant transmissions with low computational cost.

The organization of the rest of this paper is as follows: Sect. 2 introduces the proposed scheme. In Sect. 3, some experimental results are explained. Finally, Sect. 4 summarizes this work.

2 Description of the Proposed Data Fusion Scheme

As shown in Fig. 1, the main steps of the proposed scheme are as follows:

Step 1: Through the historical data sequence, the similarity of nodes is calculated. Then, a similar node cluster is obtained using the clustering algorithm.

The sensing objects include temperature, humidity, illumination, gas concentration. So the data sequence $\mathbf{S}_i$ can be represented as an array:

$$\boldsymbol{S}_i = \left[\; a_i\left(1\right) \cdots a_i\left(t\right) \; b_i\left(1\right) \cdots \; b_i\left(t\right) \; \cdots n_i\left(1\right) \; \cdots n_i\left(t\right) \;\right], \tag{1}$$

where $a_i\left(t\right)$, $b_i\left(t\right)$, and $n_i\left(t\right)$ represent the different sampled data of node i at time slot t. In order to get a better clustering result, all data with the same attributes are normalized to the range [0,1] according to

$$s_t = \left(s_t - s_{\min}\right) / \left(s_{\max} - s_{\min}\right), \tag{2}$$

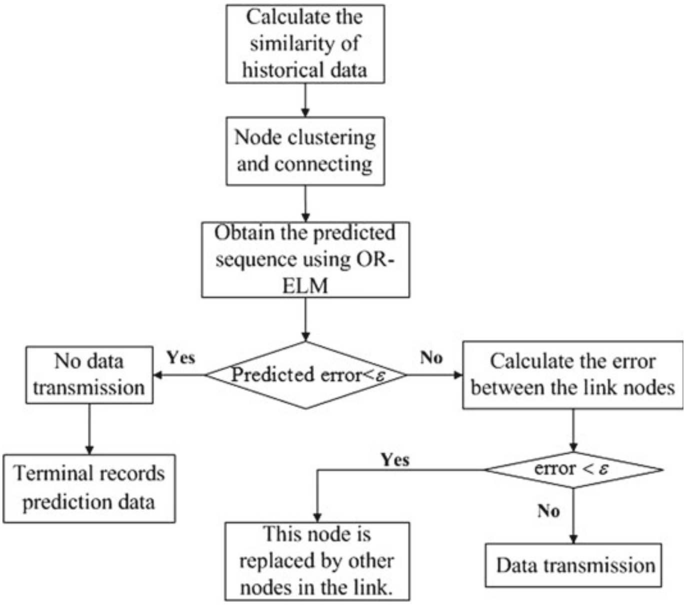
Calculate the similarity of historical data
Node clustering and connecting
Obtain the predicted sequence using OR-ELM
Predicted error<ε
Yes
No
No data transmission
Terminal records prediction data
Calculate the error between the link nodes
error < ε
Yes
No
This node is replaced by other nodes in the link.
Data transmission

Fig. 1. The flowchart of the proposed scheme.

where $s_{\min}$ and $s_{\max}$ are the maximum and minimum of the sampled data s_t during the period $[0, t]$. Then, the Euclidean distance between two nodes can be obtained as follows:

$$\Phi\left(x,y\right)=\sqrt{\left(x_1-y_1\right)^2+\left(x_2-y_2\right)^2+\cdots+\left(x_n-y_n\right)^2}=\sqrt{\sum_{i=1}^{n}\left(x_i-y_i\right)^2}, \quad (3)$$

where $\Phi\left(x,y\right)$ is similarity between nodes x and y, x_i and y_i are the corrosponding sample data. Next, the k-means method [13] is improved and applied here based on the following rules:

(R1) Define the number of randomly selected k groups (clusters).
(R2) Calculate the similarity between each point.
(R3) Classify the points where the similarity is less than a threshold into a cluster.
(R4) Update the cluster center and repeat second and third steps until the center is not changed.

Step 2: Nodes are linked based on the actual geographical distance and the strong links are obtained. If the actual geographic distance of two nodes in cluster is less than a threshold, then they will be linked. As shown in Fig. 2, the solid line are strong links, and the dotted line are weak links.

Step 3: Construct and train the OR-ELM using q training samples and predict future time series. Figure 3 shows the network topology of the OR-ELM.

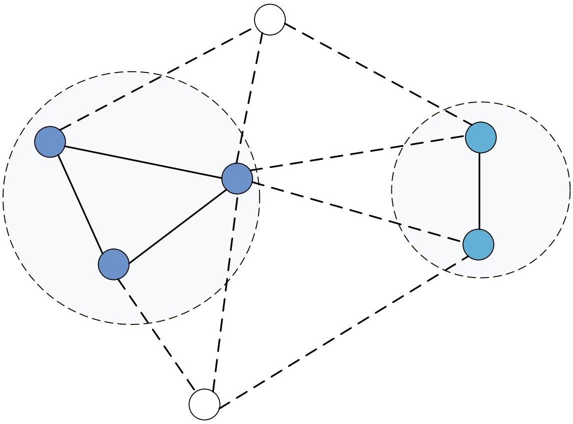

Fig. 2. Link state.

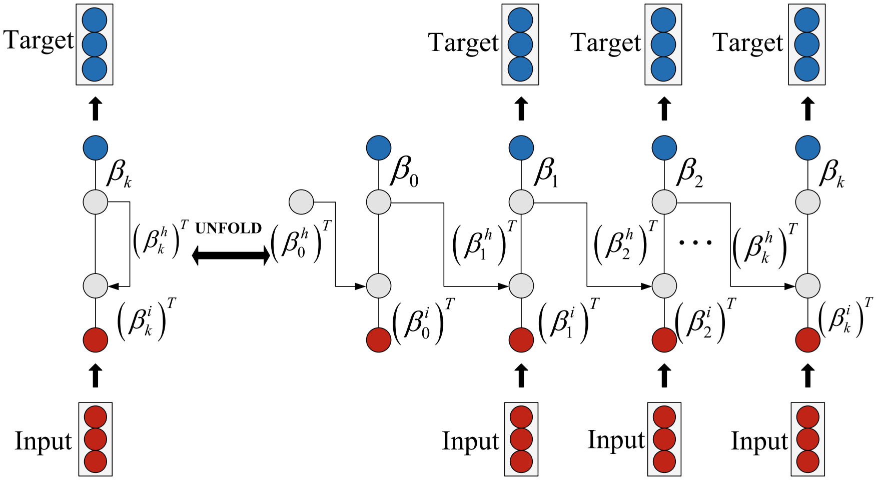
Target
Target
Target
Target
β_k
$\left(\beta_k^h\right)^T$
UNFOLD
$\left(\beta_0^h\right)^T$
β_0
β_1
β_2
β_k
$\left(\beta_1^h\right)^T$
$\left(\beta_2^h\right)^T$
$\cdots$
$\left(\beta_k^h\right)^T$
$\left(\beta_k^i\right)^T$
$\left(\beta_0^i\right)^T$
$\left(\beta_1^i\right)^T$
$\left(\beta_2^i\right)^T$
$\left(\beta_k^i\right)^T$
Input
Input
Input
Input

Fig. 3. The network structure of the OR-ELM.

ELM is a learning algorithm for training an single-hidden-layer feedforward network (SLFN). For N training samples (x_j, t_j), the output of the ELM with L hidden nodes can be denoted by

$$\sum_{i=1}^{L} \beta_i g\left(\omega_i \cdot x_j + b_i\right) = t_j, j = 1, 2, \cdots, N, \tag{4}$$

where β_i is the output weight, ω_i and b_i severally denote input weights and bias values of hidden nodes, $g(\cdot)$ is a nonlinear activation function. Therefore, it can be further expressed compactly as a matrix:

$$\mathbf{H}\beta = \mathbf{T}, \tag{5}$$

where $\mathbf{H}$ denotes the hidden-layer output matrix of the $SLFN$ and the output weights is given as follows:

$$\beta = \mathbf{H}^{\dagger}\mathbf{T}, \mathbf{H}^{\dagger} = \left(\mathbf{H}^{T}\mathbf{H} + \frac{\mathbf{I}}{C}\right)^{-1}\mathbf{H}^{T}, \tag{6}$$

where $\mathbf{H}^{\dagger}$ is the Moore–Penrose generalized inverse of matrix C and $\mathbf{H}$ is a regularization constant. The output weight β_0 for an initial training dataset with N_0 training samples is calculated as

$$\beta_0 = \mathbf{P}_0\mathbf{H}_0\mathbf{T}_0, \quad \mathbf{P}_0 = \left(\mathbf{H}_0^{T}\mathbf{H}_0 + \frac{\mathbf{I}}{C}\right)^{-1}. \tag{7}$$

During the online sequential learning phase, whenever the new input data of the N_{k+1} training sample arrives, the output weight is updated:

$$\beta_{k+1} = \beta_k + \mathbf{P}_{k+1}\mathbf{H}_{k+1}^{T}\left(\mathbf{T}_{k+1} - \mathbf{H}_{k+1}\beta_k\right), \tag{8}$$

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k\mathbf{H}_{k+1}^{T}\left(\mathbf{I} + \mathbf{H}_{k+1}\mathbf{P}_k\mathbf{H}_{k+1}^{T}\right)^{-1}\mathbf{H}_{k+1}\mathbf{P}_k, \tag{9}$$

where $k+1$ is the $(k+1)$th chunk of input data with k increasing from zero, and H_{k+1} indicates the hidden-layer output for the $(k+1)$th chunk of input data. The input sample $x\left(k+1\right) \in R^{n\times 1}$ is propagated to the hidden layer so hidden-layer output matrix $\mathbf{H}_{k+1}^{i}$ can be calculated as follows:

$$\mathbf{H}_{k+1}^{i} = g\left(norm\left(\mathbf{X}_{k+1}^{i}x\left(k+1\right)\right)\right). \tag{10}$$

Note that the *norm* function is added before the nonlinear activation as a layer normalization procedure to prevent the problem of internal covariate shift. Then, one can calculate output weight β_{k+1}^{i} using RLS:

$$\beta_{k+1}^{i} = \beta_k^{i} + \mathbf{P}_{k+1}^{i}{\mathbf{H}_{k+1}^{i}}^{T}\left(x\left(k+1\right) - \mathbf{H}_{k+1}^{i}\beta_k^{i}\right), \tag{11}$$

$$\mathbf{P}_{k+1}^{i} = \frac{1}{\lambda}\mathbf{P}_k^{i} - \mathbf{P}_k^{i}{\mathbf{H}_{k+1}^{i}}^{T}\left(\lambda^2 + \lambda\mathbf{H}_{k+1}^{i}P_k^{i}{\mathbf{H}_{k+1}^{i}}^{T}\right)^{-1}\mathbf{H}_{k+1}^{i}\mathbf{P}_k^{i}. \tag{12}$$

Set the forgetting factor $\lambda \in (0,1]$. The transpose of β_{k+1}^{i} serves as the input weight of OR-ELM $\mathbf{X}_{k+1}$:

$$\mathbf{X}_{k+1} = {\beta_{k+1}^{i}}^{T}. \tag{13}$$

Hidden-layer output matrix $\mathbf{H}_{k+1}^{h}$ is given by

$$\mathbf{H}_{k+1}^{h} = g\left(norm\left(\mathbf{X}_{k+1}^{h}\mathbf{H}_k\right)\right). \tag{14}$$

Then, using RLS calculates output weight β_{k+1}^{h} and the transpose of β_{k+1}^{h} is used as the hidden weight of OR-ELM $\mathbf{Y}_{k+1}$:

$$\mathbf{Y}_{k+1} = {\beta_{k+1}^{h}}^{T}. \tag{15}$$

Now, for the $(k+1)$th input $x(k+1)$, the OR-ELM hidden-layer output matrix $\mathbf{H}_{k+1}$ is calculated as follows:

$$\mathbf{H}_{k+1} = g\left(norm\left(\mathbf{X}_{k+1}x\left(k+1\right) + \mathbf{Y}_{k+1}\mathbf{H}_{k}\right)\right). \tag{16}$$

Step 4: The base station and sensor nodes both use the OR-ELM to predict the future perceived data, which can guarantee that the data sequence in the sensor nodes and base station are synchronous. If the error between them is below the threshold, then it is unnecessary for the sensor node to transmit data to the base station. When the data prediction target is achieved, the data need to be saved. At the same time, the terminal treats the predicted data as perceived data for the current period.

Step 5: If the prediction fails, data will be transmitted to other nodes in the link and forwarded. Figure 4 displays the nodes connection diagram, and the blue nodes represent nodes that need to transmit data.

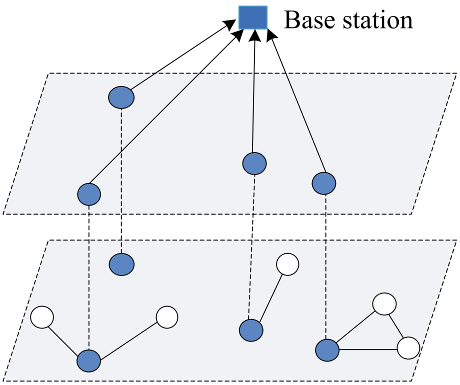

Fig. 4. Network topology.

3 Experimental Results

In order to evaluate the proposed scheme, some simulations are examined based on Intel Labs' data collected from Intel Labs during a 1-month period. The real data set is collected from 54 distributed sensors.

The OR-ELM input dimension is configured to 250 steps with a 250-step time delay. The output dimension of the OR-ELM is set to 1. That is, the proposed scheme will continue to accept sequences that take the past 250 steps as input, and as a target output, there are 5 steps of future values. Because 250 data sampling points for each node are used for clustering, the statistical data analysis is also performed for the data after 250 sampling points. Set the forgetting factor $\lambda = 0.96$, and $L = 25$ denotes the number of hidden nodes.

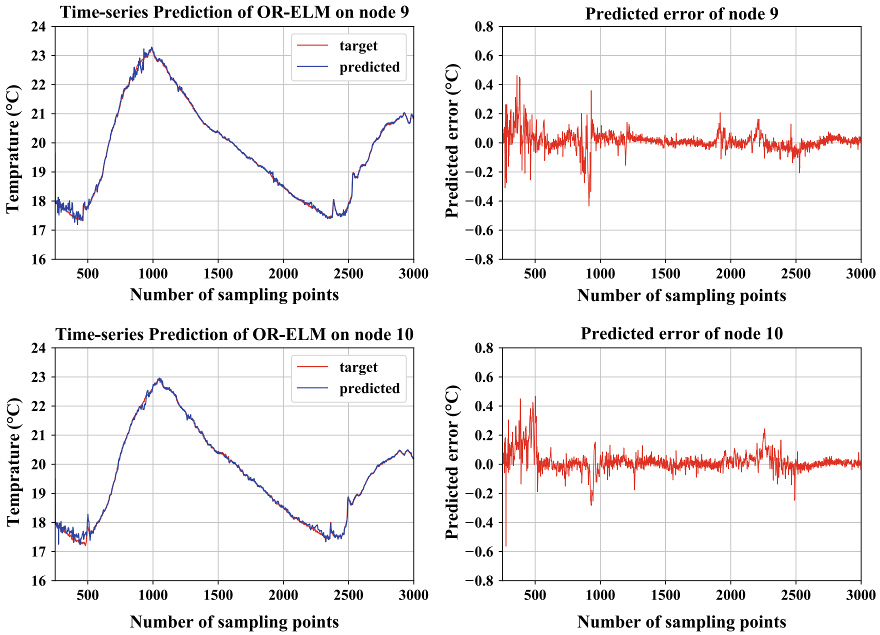

Fig. 5. Prediction results.

Two nodes are randomly selected to prove the performance of the algorithm. Figure 5 shows nodes 9 and 10 time-series prediction of OR-ELM on dataset. Comparing the prediction success rates for nodes 9 and 10 as the error threshold e-max changes from 0° to 0.22°. As shown in Fig. 6, when e-max = 0.1, the algorithm achieves about 90% successful prediction for all nodes. In other words, over 90% of the collected sensor data is not to be transmitted to the base station. With the increase of threshold, the algorithm gets a significant improvement in reducing communication. Clustering results prove that nodes 9 and 10

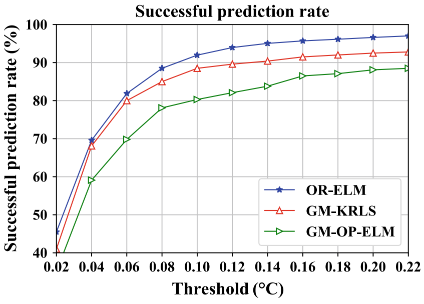

Fig. 6. The comparison of the prediction success rates.

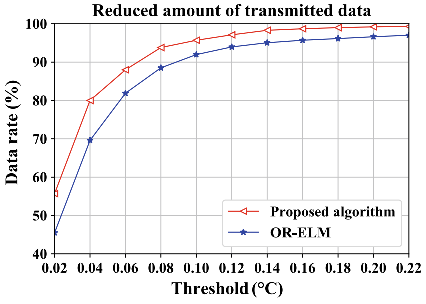

Fig. 7. The comparison of the reduced amount of transmitted data.

are strong links. The threshold is set to 0.1. So the successful rate of nodes 9 and 10 is 91.96% and 88.76%, respectively. The error of 65.5% of their real data is less than 0.1. The calculation yields that there are only 85 sample points that node 9 predicts failed and then is not similar to node 10. In other words, it senses the 2750 data, only 85 sampling data need to be sent to the terminal. There are also 135 sampling data that are predicted to fail. But they are similar to node 10 and can be replaced by values of node 10. Figure 7 shows the reduced rate of transmitted data. Although the results only report each node, the data reduction of the sensor nodes will result in a reduction of each cluster.

4 Conclusions

This paper has improved the performance of a data fusion scheme of a wireless sensor data transmission using clustering and prediction. The OR-ELM has shown higher accuracy than other algorithms, and highly correlated data between nodes have been clustered. Additionally, nodes can compare with connected nodes before transmission to further reduce the amount of data. Finally, experimental results have shown that the proposed scheme improved the prediction accuracy and reduced spatial and temporal redundant transmissions with low computational cost.

References

1. Durisic, M.P., Tafa, Z., Dimic, G., et al.: A survey of military applications of wireless sensor networks. Embedded Comput. pp. 196–199. IEEE (2012)
2. Tan, R., Xing, G., Liu, B., et al.: Exploiting data fusion to improve the coverage of wireless sensor networks. IEEE/ACM Trans. Netw. **20**(2), 450–462 (2012)
3. Zhang, C., Wang, B., Fang, S., et al.: Clustering algorithms for wireless sensor networks using spatial data correlation. In: International Conference on Information and Automation, pp. 53–58. IEEE (2008)
4. Vuran, M.C., Akyildiz, I.F.: Spatio-temporal correlation: theory and applications for wireless sensor networks. Comput. Netw. **45**(3), 245–259 (2004)
5. Arunraja, M., Malathi, V., Sakthivel, E.: Distributed similarity based clustering and compressed forwarding for wireless sensor networks. ISA Trans. **59**, 180–192 (2015)
6. Cheng, L., Guo, S., Wang, Y., Yang, Y.: Lifting wavelet compression based data aggregation in big data wireless sensor networks. In: IEEE, International Conference on Parallel and Distributed Systems, pp. 561–568. IEEE (2017)
7. Zheng, H., Yang, F., Tian, X., et al.: Data gathering with compressive sensing in wireless sensor networks: a random walk based approach. IEEE Trans. Parallel Distrib. Syst. **26**(1), 35–44 (2014)
8. Tulone, D., Madden, S.: PAQ: time series forecasting for approximate query answering in sensor networks. In: European Workshop on Wireless Sensor Networks, Vol. 3868, pp. 21–37. Springer, Berlin, Heidelberg (2006)
9. Luo, X., Chang, X.: A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks. Int. J. Control Autom. Syst. **13**(3), 539–546 (2015)

10. Luo, X., Zhang, D., Yang, L.T., et al.: A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems. Future Gener. Comput. Syst. **61**, 85–96 (2016)
11. Liang, N., Huan, G., Saratchandran, P., Sundararajan, N.: A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans. Neural Netw. **17**(6), 1411–1423 (2006)
12. Park, J.M., Kim, J.H.: Online recurrent extreme learning machine and its application to time-series prediction. In: International Joint Conference on Neural Networks, pp. 1983–1990. IEEE (2017)
13. Aznaoui, H., Raghay, S., Aziz, L.: New smart nodes distribution using kmeans approach to enhance routing in wsn. Indian J. Sci. Technol. **9**(46) (2016)