IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON INTELLIGENT DATA SENSING, COLLECTION AND DISSEMINATION IN MOBILE COMPUTING

Received October 7, 2019, accepted October 19, 2019, date of publication October 28, 2019, date of current version January 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2949878

Drift Calibration Using Constrained Extreme Learning Machine and Kalman Filter in Clustered Wireless Sensor Networks

JIAWEN WU[1] AND GUANGHUI LI[1,2]

[1]School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China
[2]Engineering Research Center of Internet of Things Technology Applications (MOE), Wuxi 214122, China

Corresponding author: Guanghui Li (ghli@jiangnan.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61472368, in part by the Key Project of the Jiangsu Provincial Research and Development under Grant BE2016627, in part by the Fundamental Research Funds for the Central Universities under Grant RP51635B, and in part by the Wuxi International Science and Technology Research and Development Cooperative Project under Grant CZE02H1706, and in part by the 111 Project under Grant B12018.

ABSTRACT Wireless sensor networks (WSNs) have been increasingly applied for environmental monitoring in recent years. However, the sensor data drift is a serious issue affecting the reliability of monitoring system. Based on the assumption that the neighboring nodes have correlated measurements, this paper presents a novel algorithm using constrained extreme learning machine and Kalman filter (CELM-KF) for tracking and calibrating drift of sensor data. CELM-KF has two phases: training phase and calibrating phase. In the training phase, the wavelet denoising method is used for data preprocessing. Then the cluster head trains the model of the constrained extreme learning machine (CELM) using the measurements of its neighbors to obtain the prediction data of the target sensor. In the calibrating phase, we track and correct the data drift of target sensor using Kalman filter. To evaluate the performance of CELM-KF, simulation experiments on different datasets are conducted, and two parameters including decision coefficient and mean square error (MSE) of CELM-KF are compared with those of existing algorithms. The simulation results show that CELM-KF can successfully calibrate the sensor data drifts.

INDEX TERMS Wireless sensor network, drift calibration, extreme learning machine, Kalman filters, environmental monitoring.

I. INTRODUCTION

Environmental monitoring is a typical application of wireless sensor network (WSN) [1], [2]. A WSN node can be used in a harsh environment to measure the environmental parameters periodically, such as humidity, temperature, light and wind speed [3]–[6]. In the field of environmental monitoring, sensors are often deployed in unattended places for a long time, which makes data drift become a serious issue affecting the reliability of sensor data [7]–[10]. For example, Ni *et al.* [11] found that the drift data of a soil CO_2 sensor is 200% of the real data, which is a serious problem for the final users who need accurate data. Therefore, an automatic drift calibration technique is critical for improve the data quality [12].

The traditional calibration techniques include two main categories: Non-blind and blind calibration. Non-blind calibration technique relies on the known reference information [13]. A significant step of calibration technique is to measure the response by applying a known stimulus to the sensor network. Then the gain or offset can be obtained by comparing the ground truth with the response. The reference information is generally high-fidelity measurement of an observed quantity [14]–[16]. Another type of non-blind calibration technique is based on a manual calibration of a set of sensors, and other sensors are calibrated by those calibrated manually [15]. The Non-blind calibration technique is suitable for the scenario of small-scale networks in a specific space (such as indoor). However, it is not practical in deployment of large-scale networks [16].

Usually, it's almost impossible to measure the ground-truth data of sensing region, so sensors can only be calibrated

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

without ground-truth data. This calibration method is called blind calibration [17]. Many previous blind calibration approaches are based on the assumption that sensors are deployed intensively, so that the measurements from neighboring nodes are slightly correlated among each other [18]. Takruri *et al.* [19] utilize an interacting multiple model (IMM) to train the physical model of the surrounding sensors. Many other works apply different prediction functions in the similar framework, including support vector regression (SVR) [20] and Kriging interpolation [21]. These methods exploit the correlation of sensors, and Kalman filter is used to track the drift. However, once the predicted data becomes inaccurate, the erroneous prediction value will also be taken as the true data. As a result, the accuracy of these approaches is limited by the accumulative prediction error [22].

In this paper, we propose a blind calibration algorithm (CELM-KF) using constrained extreme learning machine and Kalman filter to track and calibrate sensor drift. Firstly, we adopt wavelet denoising approach to improve the accuracy of calibration method. Then we utilize the constrained extreme learning machine in each sensor to predict its reading. Both the predicted values and measured values of sensors are fed to the Kalman filter to detect and correct the drift of sensor measurements.

The rest of this paper is organized as follows. We present fundamental concepts and problem formulation in Section II. Then the proposed CELM-KF algorithm is described in detail in Section III. In Section IV, we present the simulation results of CELM-KF using real-world datasets. Conclusions are drawn in Section V.

II. FUNDAMENTAL CONCEPTS AND ASSUMPTIONS

We consider a WSN with a large number of sensors divided into multiple clusters. Specifically, there is one cluster head and several cluster members within each cluster, and the sensors are randomly distributed. The cluster members only measure the environmental parameters such as temperature, humidity, and atmospheric pressure. The cluster head is responsible for fusing the measurements from the cluster members, and forwards the data to the base station via multi-hops or one hop [23]. A common topology structure of clustered WSNs is shown in Fig. 1.

Affected by the hostile environment or the manufacture process of sensors, the sensors may develop drift in their reading, which reduces the quality of data [24]. In this section, we give some definitions and assumptions about data drift.

Definition 1: Sensor drift is a slow, one-way and long-term variation of the sensor readings due to their manufacturing process or environment conditions [25]. The sensor drift can be expressed as follow:

$$d = X - T - W \tag{1}$$

where d is the sensor drift, X is the sensor measurements, T is the ground truth and W is the Gaussian noise.

Definition 2: Any sensor within the sensing range of sensor i is called a neighboring node of sensor i.

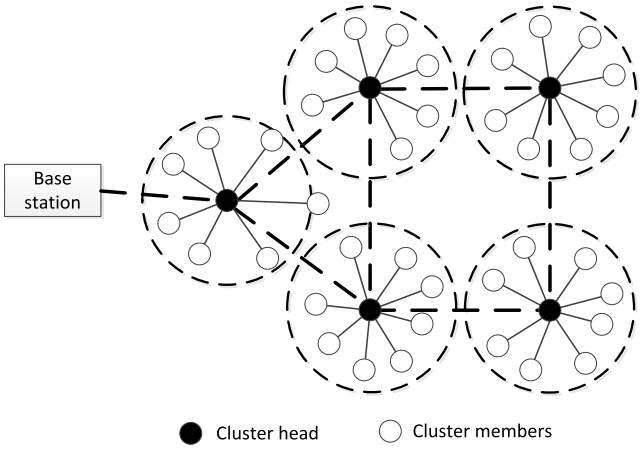


FIGURE 1. An example of the topology of clustered WSNs.

Definition 3: The change trend of different sensor readings is consistent during the same period, which is called the spatial correlation among the sensors [26].

Assumption 1: If sensor i and its neighboring nodes have spatial correlation, we can use the measurements of the neighbors to predict the measurements of sensor i:

$$\widehat{x}_i = f(neighbours\_data) \tag{2}$$

where $\widehat{x}_i$ is the predicted measurement of sensor i, and the function $f(neighbors\_\ data)$ represents the spatial correlation of the sensor measurements.

Assumption 2: The data drifts are different for different sensors. Actually, the drift is generated randomly and is strongly related to its internal structure and the environmental condition.

In this paper, we choose *MSE* and decision coefficient R^2 to evaluate the performance of drift calibration algorithms.

$$MSE = \frac{1}{n}\sum_{c=1}^{n}(x_c - \widehat{x}_c)^2 \tag{3}$$

$$R^2 = \frac{(n\sum_{c=1}^{n}\widehat{x}_c x_c - \sum_{i=1}^{n}\widehat{x}\sum_{i=1}^{n}x_c)^2}{(n\sum_{c=1}^{n}\widehat{x}_c^2 - (\sum_{c=1}^{n}\widehat{x}_c)^2)(n\sum_{c=1}^{n}x_c^2 - (\sum_{c=1}^{n}x_c)^2)} \tag{4}$$

Here,n is number of training samples, x_c is measured data of sensor i and $\widehat{x}_c$ is the predicted value of sensor i.

III. CELM-KF CALIBRATION ALGORITHM

CELM-KF runs on the cluster heads that have stronger computational capability than the cluster members. Firstly, the measurements of target sensor i and its neighbors collected during the initial deployment period are applied to model the spatial correlation among the sensors in cluster heads. Once the cluster head receives the new data, CELM-KF is used for data prediction. Afterwards, the *MSE* value between the predicted data and measured data is calculated. If *MSE* is less than a given threshold, it indicates that there is no drift in sensor i. Otherwise, it indicates that drift has occurred, and CELM-KF need to calibrate the data with drift.

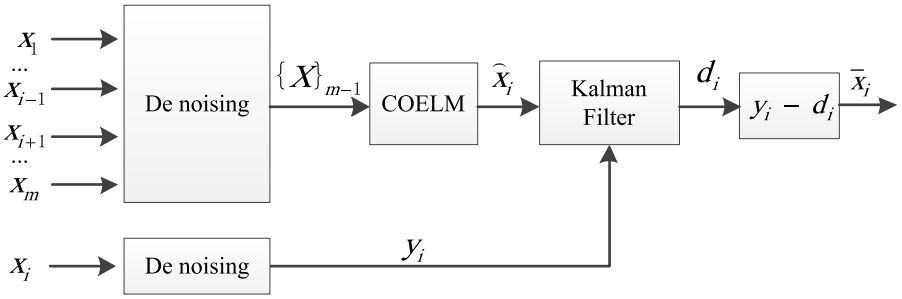


FIGURE 2. Data drift calibration framework of CELM-KF.

Then the cluster head reports the calibrated data to the base station.

The goal of CELM-KF is to accurately predict the measurements of sensor i under the interference of random error (noise) and systematic error (drift). CLEM-KF includes two phases: training phase and the calibrating phase. During the training phase, we utilize wavelet denoising method to clean the noise of sensors' datasets. Then the sensor measurements collected during the initial deployment period (training data set) are taken as the input of CELM to predict the measurements of the target sensor. During the calibrating phase, both the predicted measurements and the de-noised measurements of sensor i are fed to the Kalman filter to track and correct drift d_i. Figure 2 shows the data drift calibration framework of CELM-KF.

A. DATA PREPROCESSING USING WAVELET DENOISING

The measurements of sensors are often corrupted by noise. So it is necessary to eliminate the noise before calibrating drift. We utilize wavelet threshold denoising method [29] to improve the accuracy of calibration.

Firstly, the noisy data is decomposed, then we can get the wavelet decomposition coefficients, including the low and high frequency coefficients.

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}}\,\psi(\frac{t-b}{a})$$
$$W = \int y_i \psi_{a,b}(t)dt \tag{5}$$

where ψ is the wavelet basis, a is the scale parameter, b is the translation parameter and W is the wavelet coefficients. y_i is the measurements of sensor i with noise.

Afterwards, the wavelet coefficients are estimated.

$$\bar{W} = \begin{cases} sign(W)(W-\lambda) & |W| \geq \lambda \\ 0 & |W| < \lambda \end{cases} \tag{6}$$

Finally, the wavelet coefficients are used to reconstruct signal with the inverse transformation of wavelet transform.

$$\widehat{\psi}(\omega) = \int \psi(t)dt$$
$$C_\omega = \int \frac{\left|\widehat{\psi}(\omega)\right|^2}{|\omega|}\mathrm{d}\omega$$
$$\bar{y}_i = C_\omega^{-1} \iint \bar{W}\psi_{a,b}(t)\mathrm{d}a\mathrm{d}b \tag{7}$$

where $\widehat{\psi}(\omega)$ is Fourier transform, and y_i is the measurements of sensor i after denoising.

B. MODELING AND ANALYSIS OF CELM

Given sensor i and its neighbor node set $\{1, 2, \ldots, i-1, i+1, \ldots, m\}$, we use a model function to predict the measurements of sensors [30]. The measurements collected during the initial deployment period are taken as the training datasets, which consists of $TX = \{x_{j,t}: j = 1, 2, \ldots, i-1, i+1, \ldots, m, t = 1, 2, \ldots, n\}$ and $TY = \{x_{i,t}: t = 1, 2, \ldots, n\}$. Here, TX and TY denote data series, and t represents the time instant of sampling. All the data are normalized into $[-1, 1]$. The training goal is to minimize the MSE between the predicted value and measurement of sensor i.

For ELM, it is not necessary to dynamically update the weights of each layer, and it can produce an unique global optimal solution [31]. However, the input weights and biases of ELM are randomly generated which leads to unstable precision of regression [32].

We extend the ELM model to CELM by constraining the weight vector parameter. Let $t1$ and $t2$ represent two time instants respectively. Let $X_{i,t2}$ represents the minimum sample and $X_{i,t1}$ represents any sample other than $X_{i,t2}$ in TY. $X_{j,t2}$ represents the sample of time instant $t2$ and $X_{j,t1}$ represents the sample of time instant is $t1$ in TX. The weight w of the input layer can be calculated by equations (8-14).

$$w = a(X_{j,t1} - X_{j,t2}) \tag{8}$$
$$Xw + b = aX(X_{j,t1} - X_{j,t2}) + b \tag{9}$$

Here, a is a normalized factor, b is the bias of the hidden layer. We label samples $X_{j,t2}$ and $X_{j,t1}$, as -1 and 1 respectively.

$$aX_{j,t1}(X_{j,t1} - X_{j,t2}) + b = 1 \tag{10}$$
$$aX_{j,t2}(X_{j,t1} - X_{j,t2}) + b = 1 \tag{11}$$

We can obtain b and a by solving equations (9) and (10):

$$b = \frac{(X_{j,t2} + X_{j,t1})(X_{j,t2} - X_{j,t1})}{\left\|X_{j,t1} - X_{j,t2}\right\|^2} \tag{12}$$
$$a = \frac{2}{\left\|X_{j,t1} - X_{j,t2}\right\|^2} \tag{13}$$

Now we can get the weight w:

$$w = \frac{2(X_{j,t1} - X_{j,t2})}{\left\|X_{j,t1} - X_{j,t2}\right\|^2} \tag{14}$$

With the parameter w and b, the measurement of sensor i can be predicted.

C. DRIFT CALIBRATION USING KALMAN FILTER

Let $d_{i,t}$ denote the smoothing drift of sensor i. Here t is the time index. At each time instant t, sensor i measures a reading, and the ground truth is $r_{i,t}$. To estimate the drift of sensor, the mathematical model given in (15) is used.

$$d_{i,t} = d_{i,t-1} + w_{i,t}, \quad w_{i,t} \sim N(0, Q_{i,t}) \tag{15}$$

where $w_{i,t}$ is Gaussian noise and $Q_{i,t}$ is state noise covariance.

Equation (15) is a function that tracks the state of the sensor, and can used to track the change of drift over time. In target tracking, it is necessary to establish a measurement equation for the estimation procedure [33]. Suppose that there is a measuring instrument in reality, which can measure the sensor drift, then there must be an error associated with the measurements:

$$z_{i,t} = d_{i,t} + v_{i,t}, \quad v_{i,t} \sim N(0, R_{i,t}) \tag{16}$$

Here, $v_{i,t}$ is Gaussian noise, and $R_{i,t}$ is the covariance of measurement noise.

Actually, there is no instrument that can directly measure the sensor drift. In this case, the ideal drift is equal to the measured value minus the actual value:

$$d_{i,t} = r_{i,t} - T_{i,t} \tag{17}$$

Then we use $\widehat{x}_{i,t}$ as a estimation of $T_{i,t}$.

$$d_{i,t} = r_{i,t} - \widehat{x}_{i,t} \tag{18}$$

Equation (15) and (16) form a Kalman Filter tracking set of equations. This modified KF algorithm runs in each sensor to estimate its drift with an iterative procedure, which is shown in Algorithm 1.

In Algorithm 1, n is the number of measurements of sensor i. At time t, the drift estimation d_pre and the covariance estimationp_pre are predicted based on the drift estimation and the covariance estimation at time t-1. Then update the Kalman filter gain K and the covariance estimation at time t. After the algorithm is completed iteratively, we can obtain the predicted data $\bar{x}_{i,t}$ of sensor i.

$$\bar{x}_{i,t} = r_{i,t} - d_{i,t} \tag{19}$$

Algorithm 1 Drift Calibration Algorithm

input: Real measurement Z_t; Process variance Q_t; Initial drift d_0
Measurement variance R_t; Initial covariance P_0;
Initial drift value: d

Calculate:

1: **for** $t = 1$ to n **do**
2: $d\_pre = d_{t-1}$;
3: $P\_pre = P_{t-1} + Q_t$;
4: $K = P\_pre/(P\_pre + R_t)$;
5: $d_t = d\_pre + K^*(Z_t - d\_pre)$;
6: $P_t = -(1 - K)^*P\_pre$;
7: $X_{t-1} = Z_{t-1} - d_{t-1}$
8: **end for**

Output: estimated drift: X_t

IV. EVALUATION

To evaluate the performance of CELM-KF algorithm, several experiments were carried out on real sensor network datasets in LUCE and JNSN systems. We compare the experimental results of CELM-KF with SVR-KF [20] and SSP-KF [34].

A. DATASETS

The LUCE dataset comes from a WSN deployed in EPFL [35]. The network consists of 97 sensors grouped in 10 clusters according to their spatial proximity. They recorded ambient temperature, surface temperature, relative humidity, solar radiation, soil moisture and wind direction at 31 seconds intervals during the period from 1st October 2006 to 9th May 2007.

We selected three clusters from LUCE system. The sensor IDs in the first dataset LUCE_1 were 10,14,15,17,18,19. The sensor IDs in the second dataset LUCE_2 were 21,23,24,25,26,27,28. The sensor IDs in the third dataset LUCE_3 were 81,82,85,86,87, 89. All datasets were collected from 5th October 2006 to 13th October, 2006.

The JNSN dataset comes from a WSN deployed in the campus of Jiangnan University (refer to Figure 3). This network consists of 31 sensors including one sink, and recorded ambient temperature, relative humidity, solar radiation at 10 minutes intervals during the period starting from 25th April 2018 to 11th July 2018.

We selected two clusters from JNSN system. The sensor IDs in the fourth dataset JNSN_1 were 1,2,3,5,6,7. The sensor IDs in the fifth dataset JNSN_2 were 8,9,12,13,14,16. All datasets were collected from 15th May, 2018 to 11th July, 2018.

The data from LUCE_1, LUCE_2, LUCE_3, JNSN_1 and JNSN_2 are resampled at 70 seconds intervals, 48 seconds intervals, 35 seconds intervals, 16 minutes intervals, 27 minutes intervals, respectively. In LUCE_1, LUCE_2 and LUCE_3, we used the data from the first 4 days as the training datasets, and the data from the next 4 days as testing dataset in the calibrating phase. In JNSN_1 and JNSN_2, we used the data from the first 29 days as the training datasets, and the data from the next 28 days as testing dataset. It is noted that all the datasets in our experiments consist of only temperature data, and the details of datasets are shown in Table 1.

B. EXPERIMENTAL RESULTS OF WAVELET DENOISING

Since the original datasets have no significant Gaussian noise, we introduced different white Gaussian noise in datasets. The *MSE* index is used to evaluate the denoising performance. For example, we introduced noise to 2300 samples of sensor 10 from dataset 1 and the noise variance is 0.8. We can

(a) the sensor node of JNSN

(b) the deployment field of JNSN

FIGURE 3. Sensors deployment of JNSN system.

TABLE 1. Experimental datasets.

S/N	Dataset	Training Samples	Test Samples	Sensor numbers
I	LUCE_1	4600	4600	6
II	LUCE_2	7150	7150	7
III	LUCE_3	9700	9700	6
IV	JNSN_1	2500	2400	6
V	JNSN_2	1500	1500	6

observe that the denoising data curve (black color) is basically close to the initial data curve (red color) in figure 4. The *MSE* between the initial data and the denoising data in figure 4 is only 0.076. The result shows that wavelet denoising method has good effect on noise suppression.

C. COMPARISON RESULTS OF TRAINING TIME

To verify the advantages of CELM in training time, we selected 1000, 3000, 5000, 7000, 9000, 11000 samples in the dataset III. We use different samples for SSP, SVR, ELM and CELM to train model, respectively. The experimental results are shown in Table 2.

It can be seen from Table 2 that SSP has the shortest training time when the number of samples is 1000. However, as the number of samples increases, the training time increases exponentially. The reason is that SSP needs to perform singular value decomposition on the training data. Compared with ELM and SVR, CELM has the shortest training time. Actually, with the increase of the number of samples,

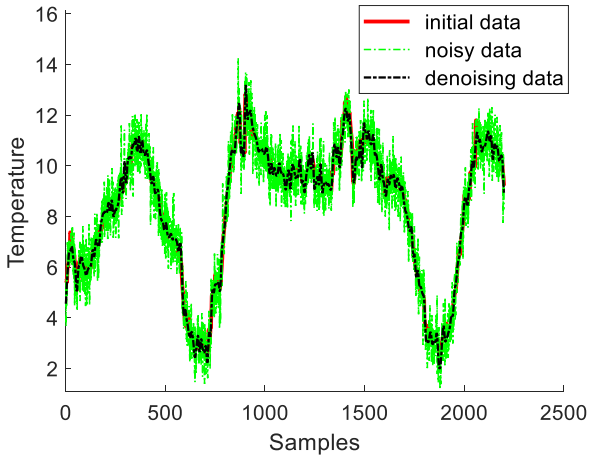


FIGURE 4. Experimental results of wavelet denoising.

TABLE 2. Comparison results of training time among four algorithms.

Samples	SSP/s	SVR/s	ELM/s	CELM/s
1000	0.05	0.089	0.122	0.125
3000	0.45	0.532	0.143	0.126
5000	1.01	1.555	0.150	0.135
7000	2.04	2.755	0.153	0.137
9000	139	4.789	0.154	0.139
11000	7340	7.159	0.154	0.139
Average	1247.1	2.813	0.146	0.134

the more support vectors of SVR need to train, resulting in the longer training times. CELM speeds up the calculation of output weights and thus it takes less training time than ELM.

D. COMPARISON OF MODEL FITTING DEGREE

In the training phase of CELM-KF, the measurements of sensor i and its neighbors are taken as the input of CELM. Then the predicted data and measured data of sensor i are fed into the Kalman filter to estimate the drift. Therefore, the key of blind calibration algorithm is to select the mathematical model with the highest model fitting degree.

We randomly selected one sensor's data from the five datasets in Table 1 to compare the model fitting degree of CELM-KF with other algorithms. We repeated five experiments and compared the average results of different algorithms.

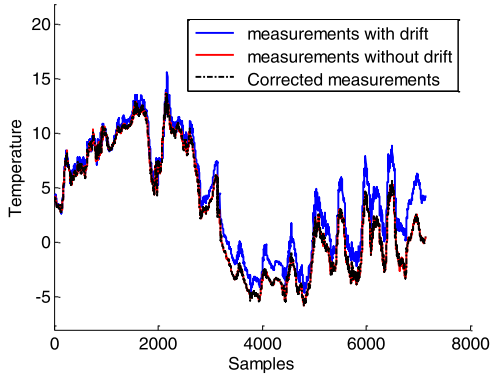
Table 3 shows that the *MSE* of CELM is lower than SVR, SSP and ELM. The determination coefficient R^2 of CELM is higher than SVR, SSP and ELM.

E. DRIFT TRACKING AND CALIBRATION

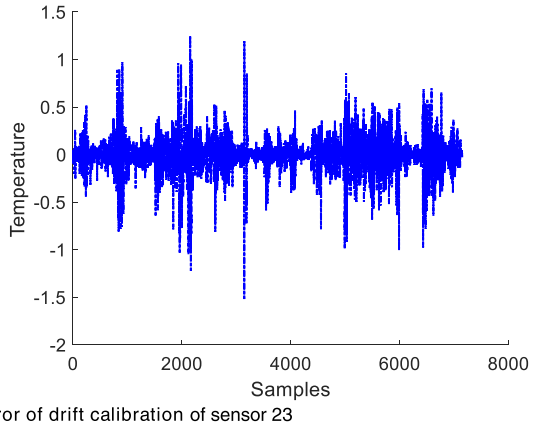
Different exponential drift are added to the original dataset of each sensor, starting randomly after the first 20 samples of

TABLE 3. Comparison results of Model fitting degree among four algorithms.

Dataset	SSP		SVR		ELM		CELM	
	MSE	R^2	MSE	R^2	MSE	R^2	MSE	R^2
I	1.4993	88.79%	0.4411	93.91%	0.2786	95.79%	0.1693	97.41%
II	1.8892	81.15%	1.8571	86.67%	0.9287	96.36%	0.6618	98.47%
III	1.7857	88.76%	0.5311	93.23%	0.4862	93.27%	0.3131	96.27%
IV	1.2518	87.89%	2.1214	81.11%	0.4949	94.55%	0.3615	97.44%
V	2.0406	80.11%	0.5177	90.38%	0.4907	94.68%	0.2140	96.76%
Average	1.6932	85.34%	1.0937	89.06%	0.5358	94.93%	0.3439	97.27%

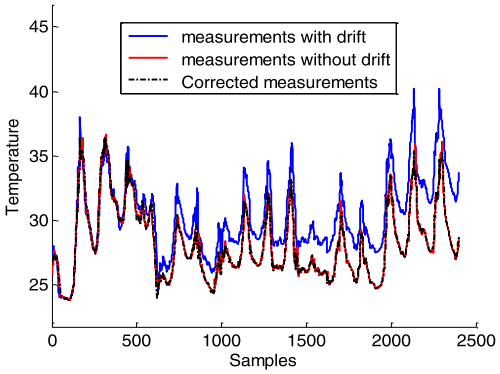


(a) Experimental results of drift calibration of sensor 23

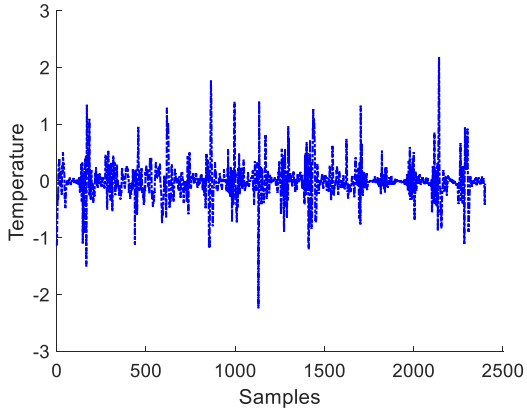


(b) Error of drift calibration of sensor 23

FIGURE 5. Drift calibration of sensor 23 in dataset II.



(a) Experimental results of drift calibration of sensor 16



(b) Experimental results of drift calibration of sensor 16

FIGURE 6. Drift calibration of sensor 16 in dataset V.

test dataset, which means that, each sensor develops drift at different time, the magnitude of drift is different, and the drift is independent. The measurements of sensor 23 in dataset II and sensor 16 in dataset V are used for this experiment.

Besides the drift, the measurement noise is introduced to the datasets of different sensors to evaluate the effect of noise calibrating of CELM-KF. In our experiments, we set Kalman filter parameters $Q = 0.1$ and $R = 0.01$.

Figure 5(a) and Figure 6(a) show the calibration results of CELM-KF of sensor 23 in dataset II and sensor 16 in dataset V, respectively. Figure 5(b) and Figure 6(b) present the error between the measurements without drift and the calibrated measurements of sensor 23 and sensor 16, respectively. From Fig.5 and Fig.6, we can observe that the calibrated curve is very close to that without drift.

Table 4 and Table 5 show the calibration results of CELM-KF on datasets I and V. It can be seen that, for all the sensors in both datasets, the average of *MSE* values are less than 0.4, and the average of R^2 values are more than 96%. This experiment results demonstrate that CELM-KF can effectively calibrate the sensor drift and measurement noise.

TABLE 4. Drift calibration performance based on dataset I.

dataset I	Sensor 10	Sensor 14	Sensor 15	Sensor 17	Sensor 18	Sensor 19
MSE	0.335	0.363	0.502	0.295	0.288	0.355
R^2	95.14%	96.98%	95.37%	96.77%	97.35%	96.14%

TABLE 5. Drift calibration performance based on dataset V.

dataset V	Sensor 8	Sensor 9	Sensor 12	Sensor 13	Sensor 14	Sensor 16
MSE	0.181	0.323	0.279	0.398	0.270	0.253
R^2	97.91%	96.17%	98.90%	97.91%	98.55%	96.45%

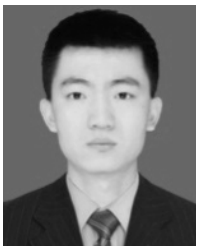
V. CONCLUSION

In this paper, we proposed a blind calibration algorithm CELM-KF for calibrating sensor drift. CELM models the spatial correlation among neighboring nodes and then predict the future measurements. The predicted measurements are fed to the Kalman filter to track and calibrate the drift in the sensor readings. The simulation results show that CELM-KF has obvious advantages over the existing similar algorithms in terms of training time, model fitting degree and calibration accuracy. In the future, we will implement CELM-KF on the sensor nodes of JNSN system for practical application.

REFERENCES

[1] C. Alippi, R. Camplani, C. Galperti, and M. Roveri, "A robust, adaptive, solar-powered WSN framework for aquatic environmental monitoring," *IEEE Sensors J.*, vol. 11, no. 1, pp. 45–55, Jan. 2011.
[2] X. Liu, "Node deployment based on extra path creation for wireless sensor networks on mountain roads," *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2376–2379, Nov. 2017.
[3] M. Wu, L. Tan, and N. Xiong, "Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications," *Inf. Sci.*, vol. 329, pp. 800–818, Feb. 2016.
[4] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorsCope: Application-specific sensor network for environmental monitoring," *ACM Trans. Sens. Netw.*, vol. 6, pp. 1–32, Jan. 2010.
[5] T. Wang, Z. A. Bhuiyan, G. Wang, A. Rahman, J. Wu, and J. Cao, "Big data reduction for a smart city's critical infrastructural health monitoring," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 128–133, Mar. 2018.
[6] M. Huang, W. Liu, T. Wang, H. Song, X. Li, and A. Liu, "A queuing delay utilization scheme for on-path service aggregation in services oriented computing networks," *IEEE Access*, vol. 7, pp. 23816–23833, 2019.
[7] J. Li, W. Liu, T. Wang, H. Song, X. Li, F. Liu, and A. Liu, "Battery-friendly relay selection scheme for prolonging the lifetimes of sensor nodes in the Internet of Things," *IEEE Access*, vol. 7, pp. 33180–33201, 2019.
[8] X. Liu, T. Qiu, and T. Wang, "Load-balanced data dissemination for wireless sensor networks: A nature-inspired approach," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2019.2900763.
[9] X. Liu and P. Zhang, "Data drainage: A novel load balancing strategy for wireless sensor networks," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 125–128, Jan. 2018.
[10] X. Liu, T. Qiu, X. Zhou, T. Wang, L. Yang, and V. Chang, "Latency-aware anchor-point deployment for disconnected sensor networks with mobile sinks," *IEEE Trans. Ind. Informat.*, to be published, doi: 10.1109/TII.2019.2916300.
[11] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, p. 25, 2009.
[12] J. Tan, W. Liu, M. Xie, H. Song, A. Liu, M. Zhao, and G. Zhang, "A low redundancy data collection scheme to maximize lifetime using matrix completion technique," *EURASIP J. Wireless Commun. Netw.*, vol. 5, pp. 1–27, Jan. 2019.
[13] R. Tan, G. Xing, Z. Yuan, X. Liu, and J. Yao, "System-level calibration for data fusion in wireless sensor networks," *ACM Trans. Sensor Netw.*, vol. 9, no. 3, p. 28, 2013.
[14] N. Ramanathan, M. Burt, D. Estrin, T. Harmon, C. Harvey, J. Jay, E. Kohler, S. Rothenberg, and M. Srivastava, "Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks," Center Embedded Networked Sens., Los Angeles, CA, USA, Tech. Rep. CENS TR 62. 2006.
[15] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure IoT service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4831–4843, Jun. 2019.
[16] G. Zhang, T. Wang, G. Wang, A. Liu, and W. Jia, "Detection of hidden data attacks combined fog computing and trust evaluation method in sensor-cloud system," *Concurrency Comput. Pract. Exper.*, 2018.
[17] E. Miluzzo, N. D. Lane, A. T. Campbell, and R. Olfati-Saber, "CaliBree: A self-calibration system for mobile sensor networks," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst.*, New York, NY, USA, Springer-Verlag, 2008, pp. 314–331.
[18] C. Taylor, A. Rahimi, J. Bachrach, H. Shrobe, and A. Grue, "Simultaneous localization, calibration, and tracking in an ad hoc sensor network," in *Proc. 5th Int. Conf. Inf. Process. Sensor Netw.*, 2006, pp. 27–33.
[19] R. Rossini, E. Ferrera, D. Conzon, and C. Pastrone, "WSNs self-calibration approach for smart city applications leveraging incremental machine learning techniques," in *Proc. 8th IFIP Int. Conf. New Technol. Mobility Secur. (NTMS)*, Nov. 2016, pp. 1–7.
[20] B.-T. Lee, S.-C. Son, and K. Kang, "A blind calibration scheme exploiting mutual calibration relationships for a dense mobile sensor network," *IEEE Sensors J.*, vol. 14, no. 5, pp. 1518–1526, May 2014.
[21] M. Takruri, S. Challa, and R. Chakravorty, "Auto calibration in drift aware wireless sensor networks using the interacting multiple model algorithm," in *Proc. Mosharaka Int. Conf. Commun. Comput. Appl.*, Aug. 2008, pp. 98–103.
[22] M. Takruri, S. Rajasegarar, S. Challa, C. Leckie, and M. Palaniswami, "Online drift correction in wireless sensor networks using spatio-temporal modeling," in *Proc. 11th Int. Conf. Inf. Fusion*, Jun./Jul. 2008, pp. 1–8.
[23] D. Kumar, S. Rajasegarar, and M. Palaniswami, "Automatic sensor drift detection and correction using spatial Kriging and Kalman filtering," in *Proc. IEEE Int. Conf. Distrib. Comput. Sensor Syst.*, May 2013, pp. 183–190.
[24] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. Sensor Netw.*, vol. 6, no. 3, p. 23, 2010.
[25] M. S. Stanković, S. S. Stanković, and K. H. Johansson, "Distributed blind calibration in lossy sensor networks via output synchronization," *IEEE Trans. Autom. Control*, vol. 60, no. 12, pp. 3257–3262, Dec. 2015.
[26] M. Takruri, K. Aboura, and S. Challa, "Distributed recursive algorithm for auto calibration in drift aware wireless sensor networks," in *Innovations and Advanced Techniques in Systems, Computing Sciences and Software Engineering*. The Netherlands: Springer, 2008, pp. 21–25.
[27] M. Takruri, S. Rajasegarar, S. Challa, C. Leckie, and M. Palaniswami, "Spatio-temporal modelling-based drift-aware wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 1, no. 2, pp. 110–122, 2011.
[28] Y. Wang, A. Yang, Z. Li, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using signal space projection and Kalman filter," in *Proc. IEEE 10th Int. Conf. Intell. Sensors Sensor Netw. Inf. Process. (ISSNIP)*, Apr. 2015, pp. 1–6.
[29] V. Nourani, G. Andalib, and F. Sadikoglu, "Multi-station streamflow forecasting using wavelet denoising and artificial intelligence models," *Proc. Comput. Sci.*, vol. 120, pp. 617–624, Jan. 2017.
[30] Y. Wang, A. Yang, Z. Li, X. Chen, P. Wang, and H. Yang, "Blind drift calibration of sensor networks using sparse Bayesian learning," *IEEE Sensors J.*, vol. 16, no. 16, pp. 6249–6260, Aug. 2016.
[31] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.
[32] Y. Lan, Y. C. Soh, and G.-B. Huang, "Constructive hidden nodes selection of extreme learning machine for regression," *Neurocomputing*, vol. 73, nos. 16–18, pp. 3191–3199, 2010.

[33] Y. Wang, A. Yang, X. Chen, P. Wang, and Y. Wang, H. Yang, "A deep learning approach for blind drift calibration of sensor networks," *IEEE Sensors J.*, vol. 17, no. 13, pp. 4158–4171, Jul. 2017.
[34] Z. Li, Y. Wang, A. Yang, and H. Yang, "Drift detection and calibration of sensor networks," in *Proc. Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2015, pp. 1–6.
[35] [Online]. Available: http://lcav.epfl.ch/page-145180-en.html



JIAWEN WU received the bachelor's degree in Internet of Things engineering from the Taiyuan University of Technology, Taiyuan, China, in 2016. He is currently pursuing the master's degree with the School of Internet of Things Engineering, Jiangnan University. His current research interest includes data calibration for wireless sensor networks.

GUANGHUI LI received the M.S. degree from Xiangtan University, Xiangtan, China, in 1999, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He is currently a Professor with the Department of Computer Science, Jiangnan University, Wuxi, China. He has published over 70 articles in journal or conferences. His research interests include wireless sensor networks, fault tolerant computing, and nondestructive testing and evaluation.