Data Cleaning Based on Stacked Denoising Autoencoders and Multi-Sensor Collaborations

Xiangmao Chang[1, 2, *], Yuan Qiu[1], Shangting Su[1] and Deliang Yang[3]

Abstract: Wireless sensor networks are increasingly used in sensitive event monitoring. However, various abnormal data generated by sensors greatly decrease the accuracy of the event detection. Although many methods have been proposed to deal with the abnormal data, they generally detect and/or repair all abnormal data without further differentiate. Actually, besides the abnormal data caused by events, it is well known that sensor nodes prone to generate abnormal data due to factors such as sensor hardware drawbacks and random effects of external sources. Dealing with all abnormal data without differentiate will result in false detection or missed detection of the events. In this paper, we propose a data cleaning approach based on Stacked Denoising Autoencoders (SDAE) and multi-sensor collaborations. We detect all abnormal data by SDAE, then differentiate the abnormal data by multi-sensor collaborations. The abnormal data caused by events are unchanged, while the abnormal data caused by other factors are repaired. Real data based simulations show the efficiency of the proposed approach.

Keywords: Data cleaning, wireless sensor networks, stacked denoising autoencoders, multi-sensor collaborations.

1 Introduction

In recent years, wireless sensor networks (WSNs) have been widely used in many sensitive event monitoring applications, such as forest fire [Bolourchi and Uysal (2013)], illegal intrusion [Zhang, Meratnia and Havinga (2010)], pipeline leakage [Dai, Song, Sheng et al. (2017)] and device malfunction [Shi, Zhu, Zhang et al. (2015)]. In these applications, high detection accuracy and low false alarm rate are crucial for avoiding unnecessary losses. However, the data collected by WSNs usually contain various abnormal data, which greatly interferes with the event detection. How to deal with these abnormal data is a key issue for event detection.

The abnormal data can be produced by various factors, such as sensor hardware drawbacks, environmental factors, wireless interferences, etc., [Wang, Kundur and Yuan (2016); Gao, Wen, Zhao et al. (2013)]. We refer to this kind of abnormal data as *random-abnormal data*.

[1] Nanjing University of Aeronautics and Astronautics, Nanjing, China.

[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China.

[3] Michigan State University, Lansing, USA.

[*] Corresponding Author: Xiangmao Chang. Email: xiangmaoch@nuaa.edu.cn.

Received: 11 July 2019; Accepted: 31 July 2019.

CMC. doi:10.32604/cmc.2020.07923 www.techscience.com/journal/cmc

Meanwhile, the occurrence of some events can also produce abnormal data. We refer to this kind of abnormal data as *event-abnormal data*. The random-abnormal data can be very similar to the event-abnormal data. Many data cleaning approaches have been proposed to detect and/or repair the abnormal data [Mohamed, Kheng, Collin et al. (2011); Liu, Cheng and Huang (2017); Christopher and Divya (2015); Randive, Sneha, Singh et al. (2014); Xiao, Wang, Liu et al. (2018); Kriegel, Kröger, Schubert et al. (2009); Salehi, Leckie, Bezdek et al. (2016); Liu, Ting, Zhou et al. (2009)]. However, abnormal data are not further differentiated in these approaches. Without differentiate the random-abnormal data and the event-abnormal data, data repairing will repair all the abnormal data thus the event information will be lost, while event detection may treat random-abnormal data as event data thus the false alarm rate will be increased. Therefore, it is very necessary to differentiate the random-abnormal data and the event-abnormal data when we clean the sensor data.

There are few studies which use sliding window [Zhang, Feng and Zhou (2013)] or Stacked Denoising Autoencoders (SDAE) [Dai, Song, Sheng et al. (2017)] to differentiate the random-abnormal data and the event-abnormal data of single sensor node. However, they assume that the event-abnormal data are continuous and maintain a period of time, while in fact, event-abnormal data may occur in a sudden and last for a short time, such as partial discharge in voltage transformer and intrusion on a fence. There are also works that study the spatiotemporal correlation between sensors [Chen, Yang and Mccann (2015); He, Qiao, Zhou et al. (2018)]. These works are designed to find faulty sensors by correlation analysis, so that the faulty sensor can be replaced with a new one, which is different from abnormal data detections.

In this paper, we propose DCSM, a Data Cleaning approach based on SDAE (Stacked Denoising Autoencoders) and Multi-sensor collaborations. Specifically, DCSM first clusters all sensor nodes according to their correlations. For each cluster, DCSM detects all abnormal data by SDAE, then differentiates the random-abnormal data and event-abnormal data by multi-sensor collaborations. The event-abnormal data caused by events are unchanged, while the random-abnormal data caused by random factors are repaired. Real-data based simulations show the efficiency of DCSM.

The rest of this paper is organized as follows. Section 2 presents the system models and two preliminaries. Section 3 presents an overview of the DCSM system. We elaborate on the system design in Section 4. We conduct extensive real-data based simulations in Section 5. Section 6 concludes this paper.

2 System models and preliminaries

In this section, we present our system models and simply introduce the SDAE and AP algorithm which will be used later in this paper.

2.1 System models

We consider a wireless sensor network with densely deployed sensor nodes. Let the number of sensor nodes be *n*. When an event occurs, it can be detected by several sensors. Each sensor samples a data in one timeslot. Let the raw data stream sampled by s_i during

T timeslots be $X_i = x_{i1}, x_{i2}, \ldots, x_{iT}$. All the raw data sampled by the *n* sensor nodes, X_1, X_2, $\ldots$, X_n are collected by a server.

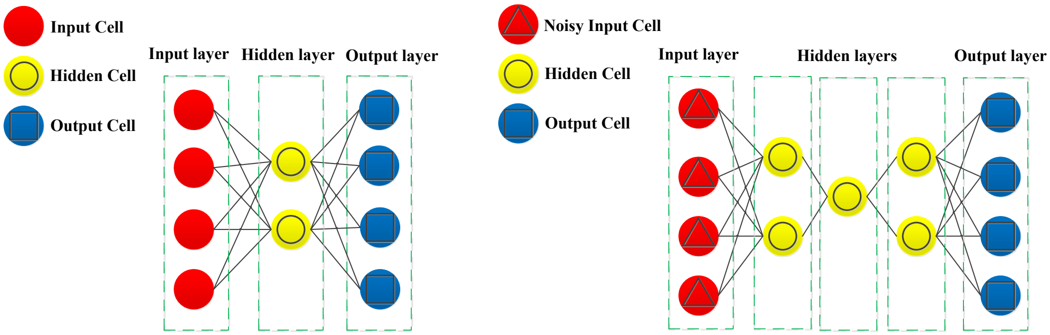
We call the abnormal data caused by random factors such as sensor hardware drawbacks, environmental factors and wireless interferences as the *random-abnormal data*, and call the abnormal data caused by the occurrence of any events as the *event-abnormal data*. The main problem of this paper is how to detect all abnormal data of $\{X_1, X_2, \ldots, X_n\}$ and differentiate them into random-abnormal data and event-abnormal data, such that the random-abnormal data can be repaired while the event-abnormal data can be used to detect the events.

2.2 Stacked denoising autoencoders (SDAE)

An autoencoder(AE) [Hinton and Salakhutdinov (2006)] is a neural network which includes an encoder and a decoder. The input data are mapped into hidden representations by the encoder, and the hidden representations are reconstructed to the input data by the decoder. The training process of the AE is to minimize the reconstruction error. If corrupted input data are used to feed the encode layer, then AE becomes DAE (Denoising Autoencoder) [Vincent, Larochelle, Bengio et al. (2008)]. When the number of hidden layers is greater than 1, DAE is called Stacked Denoising AutoEncoder (SDAE). The functions of encoder and decoder of DAE are as follows:

$$\begin{cases} y = h(\tilde{x}) = \sigma_1(W_1\tilde{x} + b_1) \\ \hat{x} = g(y) = \sigma_2(W_2 y + b_2) \end{cases}, \tag{1}$$

where $\tilde{x}$ is the corrupted input data, y is the output of the encoder, σ_1 is the encoding function and σ_2 is the decoding function, W_1 is the weight matrix between the input layer and the hidden layer, W_2 is the weight matrix between the hidden layer and the output layer, b_1 and b_2 are the bias vectors of the hidden layer and the output layer respectively, $\hat{x}$ is the output of the decoder. An example of architectures of a DAE and a SDAE is shown in Fig. 1.



(a) DAE (b) SDAE

Figure 1: Architectures of a DAE and a SDAE

2.3 Affinity propagation algorithm

Affinity Propagation (AP) [Frey and Dueck (2007)] algorithm is proposed by Frey in 2007. The AP algorithm takes the similarity matrix S=($s(i, j)$) as an input, where $s(i, j)$ is the similarity value between the node i and the node j. The AP algorithm computes two values for each node pair, the degree of availability $a(i, j)$ and the degree of responsibility $r(i, j)$. $a(i,j)$ indicates the degree that node i select the node j as a cluster center, and $r(i,j)$ indicates the degree that node j is suitable to be a cluster center of node i. $a(i, j)$ and $r(i, j)$ are iteratively computed as follows:

$$r(i,j) \leftarrow s(i,j) - \max_{j',j'\neq j}\{a(i,j') + s(i,j')\} \tag{2}$$

$$\begin{cases} a(i,j) \leftarrow \min\{0, r(j,j) + \sum\limits_{i',i'\notin\{i,j\}} \max\{0, r(i',j)\}\} \\ a(j,j) \leftarrow \sum\limits_{i',i'\notin j} \max\{0, r(i',j)\} \end{cases} \tag{3}$$

The iteration stops when it converges or reaches an iteration threshold. Then, the cluster center is selected by $r(i,i) + a(i,i) > 0$, and each node selects the cluster center by $\max\{a(i,j) + r(i,j)\}$.

3 System overview

In order to solve the problems proposed in Section 2.1, we propose the DCSM (Data Cleaning based on SDAE and Multi-sensor collaboration) system. Fig. 2 shows the architecture of the DCSM system.

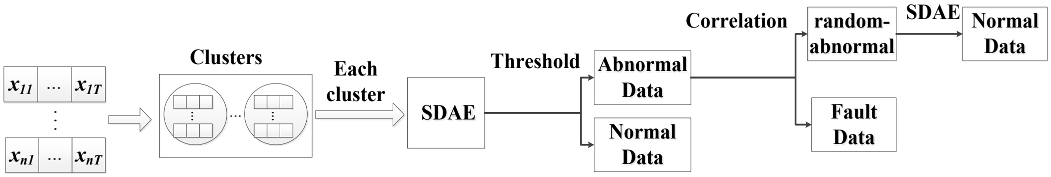


Figure 2: The architecture of the DCSM

Firstly, $X_1, X_2, \ldots, X_n$ are grouped into clusters according to their correlations. For each cluster, feed each data stream to a trained SDAE to get the expected data stream. With a carefully selected threshold, we can detect all abnormal data by comparing the difference of the expected data stream and the original data stream with a threshold. Then, by analyzing the correlations of different data streams, all abnormal data can be classified into random-abnormal data and event-abnormal data. At last, the random-abnormal data are repaired by SDAE.

There are several challenges in the above processes. Firstly, in order to differentiate the random-abnormal data and event-abnormal data, data streams must be carefully clustered to get high correlations. Secondly, the threshold to differentiate the abnormal data and the normal data must be carefully selected to get a low false alarm rate and a high detection rate in abnormal data detection. Lastly, random-abnormal data and event-abnormal data must be carefully differentiated according to the correlations of data streams.

4 System design

In this section, we elaborate on the design of DCSM from three aspects: data streams clustering, abnormal data detection and differentiation of random-abnormal data and event-abnormal data.

4.1 Data streams clustering

When $\{X_1, X_2, \ldots, X_n\}$ reaches the server, we first cluster all data streams such that the correlation of trends of all data streams in the same cluster is high, while the correlation of trends of all data streams in different clusters is low.

We use Affinity Propagation (AP) algorithms referred in Section 2.3 to cluster data streams. Unlike most prototype-based clustering algorithms (e.g., k-means), AP clusters samples only by their similarity instead of extracting features from the samples. Moreover, AP does not need to randomly select original cluster centers, which makes AP more stable. These characteristics make AP more suitable for clustering data streams on trends.

The relative trust [Zhang and Li (2017)] is used to compute the similarity matrix S=($s(i, j)$) of AP. The relative trust is proportional to the similarity between two data streams, which is computed as follows:

$$s(i,j) = 1 - \frac{\sum_{t=1}^{T} |x_{it} - x_{jt}|}{\sum_{t=1}^{T} (|x_{it}| + |x_{jt}|)} \tag{4}$$

where x_{it} and x_{jt} come from X_i and X_j respectively. Then we can compute $a(i, j)$ and $r(i, j)$ according to formulas (2) and (3) iteratively. If $r(i,i) + a(i,i) > 0$, X_i can be selected as a cluster center, and each node selects X_j as its cluster center if $a(i, j) + r(i, j)$ is the maximal data among $\{a(i, j) + r(i, j) \mid j = 1, 2, ..., n\}$. When all nodes accomplish the cluster center selection, the clustering process of data streams is accomplished.

4.2 Abnormal data detection

The normal data collected from sensors close to a non-linear low dimensional manifold, while the abnormal data caused by random factors or events deviate from the manifold distribution of normal data, as shown in Fig. 3 [Vincent, Larochelle, Bengio et al. (2008)]. In the training process of the SDAE, parts of the input normal data are randomly corrupted. The SDAE model learns features such as the deep structure and the distribution characteristics of the normal data from the undamaged parts of input data, and the SDAE predicts the real value of the corrupted parts based on the features which it learns. Therefore, the SDAE model has the ability to map the input samples to the desired manifold or around the manifold.

The training processes of the SDAE are summarized as follows:

(1) Set the number of encoder layers L, training iteration threshold h, learning rate s, noise factor α, weight decay d and the fine-tuning iteration threshold l.

(2) Initialize the parameters of encoder layers and decoder layers, x is normalized to $\bar{x}$ by $\bar{x}_i = (x_i - x_{\min}) / (x_{\max} - x_{\min})$, and $\bar{x}$ is stochastic mapped to $\tilde{x}$ by $\tilde{x} = \bar{x} + \alpha N(0,1)$.

(3) Pre-training: Forward propagation $\tilde{x}$ through all network layers to compute $\hat{x}$. The output of a DAE is fed to the next DAE as an input. In this step, use a cost function and the gradient descent method to update the parameters. Iterate this process until meeting the training iteration threshold *h*.

(4) Fine-tuning: Compute the reconstruction error of the output layer. From back to front, compute the reconstruction error of each layer and update the parameters of each layer from front to back by gradient descent method to minimize the reconstruction error. Iterate this process until meeting the fine-tuning iteration threshold *l*.

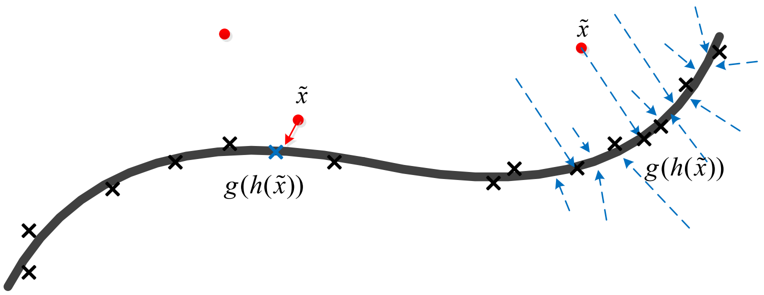


Figure 3: The manifold learning of SDAE

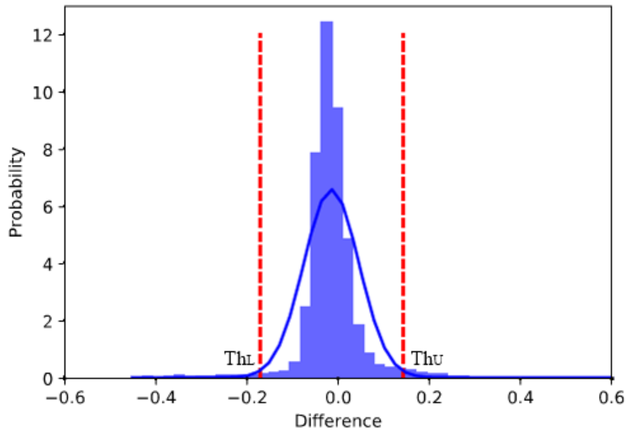


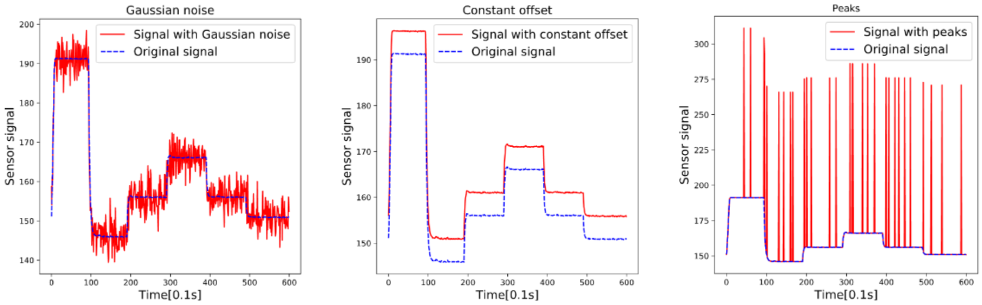
Figure 4: Thresholds for detecting abnormal data

After the SDAE is trained, feed each original data series X_i to the SDAE. The SDAE outputs the reconstructed data series $\hat{X}_i$ of X_i. Then, a difference value series can be computed by $D_i = \{d_{i1}, d_{i2}, \ldots, d_{iT}\} = X_i - \hat{X}_i$. The abnormal data can be detected by two thresholds Th_L and Th_U. If $Th_L < d_{it} < Th_U$, x_{it} is normal, else, x_{it} is abnormal.

The selection of the two thresholds is essential for the abnormal data detection. We carefully select the two thresholds Th_L and Th_U as follows. Feed a normal data series X_{normal} to the SDAE, we can get a reconstruction data series $\hat{X}_{normal}$ and $D_{normal} = X_{normal} - \hat{X}_{normal}$. Then we fit the probability distribution of D_{normal} with a normal distribution and obtain the confidence intervals with a high confidence level, e.g., 99%. Then we can get two thresholds Th_L and Th_U, as shown in Fig. 4.

4.3 Differentiation of random-abnormal data and event-abnormal data

The abnormal data detected by Section 4.2 are consist of random-abnormal data and event abnormal data. In this section, we present our method to differentiate the two types of abnormal data.



(a) Gaussian noise (b) Constant offset (c) Peaks

Figure 5: Three types of random-abnormal data in sensor data

The sensors can generate several types of random-abnormal data when they are operating in unideal conditions, such as Gaussian noise, constant offset and peaks, as shown in Fig. 5 [Zhang, Szabo and Sheng (2015); Helwig, Pignanelli and Schütze (2015); Sharma, Golubchik and Govindan (2010)]. Since the random-abnormal data are caused by random factors, they have no correlations between different sensors. On the other side, the event-abnormal data are generated by events that can be detected by several sensors simultaneously. Based on this difference between random-abnormal data and event-abnormal data, DCSM differentiates these two types of abnormal data as follows.

For each cluster grouped in Section 4.1, DCSM detects all abnormal data of each data series according to the method described in Section 4.2. All abnormal data are labeled '1', while all normal data are labeled '0'. Then, for each data stream X_i=$x_{i1}, x_{i2}, \ldots, x_{iT}$, a label stream K_i=$k_{i1}, k_{i2}, \ldots, k_{iT}$ is generated, where $k_{ij} \in \{0,1\}$. Assume there are *m* sensors in the cluster, DCSM computes $\sum_{i=1}^{m} k_{it}$ and compares it with βm, where $\beta \in (0,1]$ is a threshold for detecting the event-abnormal data, and *m* is the number of sensors in the cluster. If $\sum_{i=1}^{m} k_{it} \geq \beta m$, all abnormal data at timeslot *t* are event-abnormal data, else, all abnormal data at timeslot *t* are random-abnormal data. For each random-abnormal data $\tilde{x}_{it}$, DCSM repairs it by $g(h(\tilde{x}_{it}))$.

The algorithm of DCSM is shown in Algorithm 1.

Algorithm 1 DCSM algorithm

```
Input: Raw data streams sampled by sensors during T timeslots: $X_1, X_2, \ldots, X_n$
Output: $\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_n$ which is $X_1, X_2, \ldots, X_n$ with the repaired random-abnormal
data, the event-abnormal data of $X_1, X_2, \ldots, X_n$
Step 1) Cluster data streams
        i) Calculate the similarity matrix $RT_{ij}$ by formula (4);
        ii) Cluster data streams by applying AP algorithm;
Step 2) Detect all abnormal data for each cluster
        i) Feed each data stream $X_i$ to the SDAE model to obtain an output data stream
$\hat{X}_i$; /* the SDAE model is trained in advance by the method presented in Section 4.2*/
        ii) Compute $D_i = \{d_{i1}, d_{i2}, \ldots, d_{iT}\} = X_i - \hat{X}_i$,
           for $t$=1 to $T$
              if $Th_L<d_{it}<Th_U$
                 $x_{it}$ is normal, label it with $k_{it}$=1;
              else
                 $x_{it}$ is abnormal, label it with $k_{it}$=0;
           end for
/* the two thresholds $Th_L$ and $Th_U$ are calculated in advance by the method presented in
Section 4.2*/
Step 3) Differentiate the random-abnormal data and the event-abnormal data for each
cluster
        if $\sum_{i=1}^{m} k_{it} \geq \beta m$
          $x_{it}$ is event-abnormal data;
        else
          $x_{it}$ is random-abnormal data, replace it with $\hat{x}_{it}$;
/* $m$ is the number of sensors in the cluster, $\beta$ is a threshold for detecting the event-
abnormal data.*/
```

5 Performance evaluation

We use the temperature data collected by Intel Lab's sensors [http://db.lcs.mit.edu/labdata/labdata.html] to evaluate the performance of DCSM. The data set contains data streams sampled from 53 sensors during 12 days with a sample rate of 2 per second. We use 6 days' data to train the SDAE model and use the other 6 days' data to test the performance of the DCSM.

By the Step 1) of DCSM, all 53 sensors are grouped to clusters. For each cluster, we use the following metric to measure the correlations between cluster's members:

$$mv = mean(\sum_{i>j} var(X_i - X_j)), \tag{5}$$

where $var(\cdot)$ is the variance of a vector and $mean(\cdot)$ is the mean of a vector. The correlation is higher when mv is smaller. Fig. 6 shows the comparison of AP and k-means on data streams clustering, where each point is the average mv of 10 rounds and each round is the average mv of all clusters. We can see that AP performs better and more stable than k-means. The instability of k-means comes from the random selection of the original cluster centers.

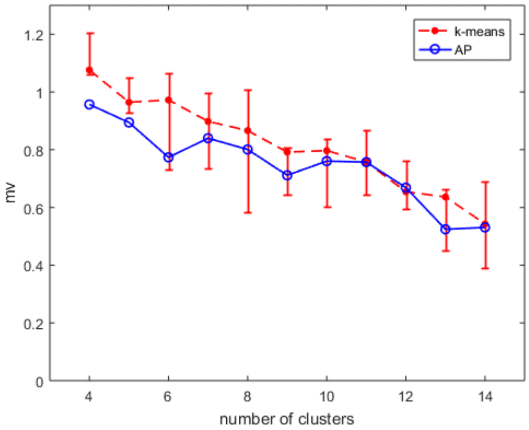


Figure 6: AP *vs.* k-means on data streams clustering

In our experiment, we use the default settings of AP in python to cluster the data streams. The result of sensor clusters is shown in Tab. 1.

Table 1: The result of sensor clusters

Cluster	1	2	3	4	5	6	7	8	9	10
Sensors	1,2,3, 4,20	6,7,8, 9,10	11,14	12,13, 53,54	14,46, 48,49, 50,51, 52	15,16, 17,18, 19,21	22,24, 25,26, 28,30	23,27, 29,33, 35,37, 39	31,32, 34,36, 38,40, 41,43	42,44, 45,47

With the training data, the SDAE model is trained according to the method presented in Section 4.2. The number of cells in the input layer of SDA is 500. There are 6 hidden layers with the number of cells of 60, 40, 40, 40, 40 and 60 respectively. The training iteration threshold is set to 500 and the noise factor is 0.001.

We use the mean absolute error (MAE) to evaluate the fitting accuracy of the trained SDAE.

$$MAE = \frac{1}{T}\sum_{t=1}^{T}(\overline{x}_t - x_t) \tag{6}$$

where x_t is the original data, $\overline{x}_t$ is the repaired data. As an example of the parameter selection, we show the *MAE vs.* different noise factors in Fig. 7. We can see that, the *MAE* is minimized when the noise factor is 0.001.

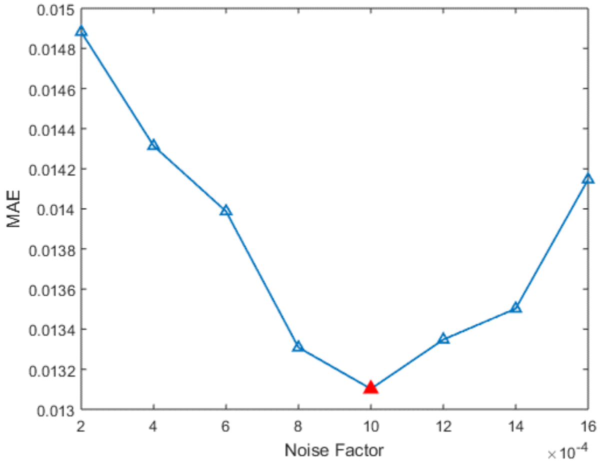


Figure 7: MAE *vs.* different noise factors

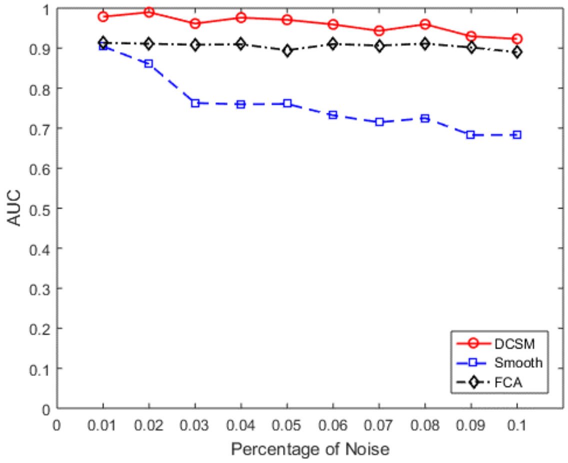
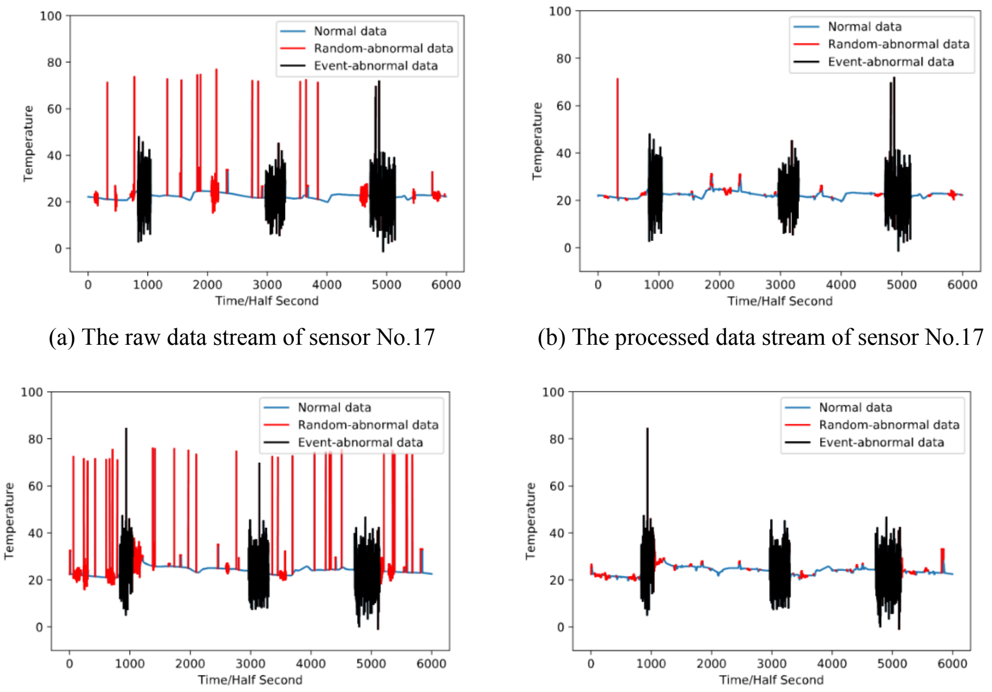


Figure 8: Performance comparison of three methods on detecting the abnormal data

The threshold Th_L and Th_U are calculated as Th_L=-0.0009, Th_U=0.0027 by the method presented in Section 4.2. With these two thresholds and the output of the SDAE, all abnormal data can be detected by DCSM. For evaluating the performance of detecting abnormal data of DCSM, we compare DCSM with two baseline methods: the moving smoothing and the cleanup method based on fog computing architecture [Zhang and Li (2017)], we call them Smooth and FCA for short respectively.

We use AUC (Area Under ROC Curve) to evaluate the performance of the algorithm, which is a widely used metric for evaluating the performance of outlier detections [Zimek, Campello and Sander (2014)]. The performance of the algorithm is better when AUC is larger. Fig. 8 shows the performance comparison of the three methods on detecting the abnormal data when they apply on data with different percentage of noise. We can see that, DCSM always outperforms the other two methods.

We use DCSM to process data streams of two sensors in the sixth cluster to intuitively show the effect of DCSM on differentiating the two types of abnormal data and repairing the random-abnormal data. As shown in Fig. 9, we can see that the event-abnormal data and the random-abnormal data are well differentiated and the most of random-abnormal data are well repaired. There are few random-abnormal data which are not repaired, this is because several random-abnormal data appear at the same time very coincidentally, which makes DCSM mistakenly treat them as event-abnormal data.



(a) The raw data stream of sensor No.17

(b) The processed data stream of sensor No.17

(c) The raw data stream of sensor No.18

(d) The processed data stream of sensor No.18

Figure 9: The result of differentiating the two types of abnormal data and repairing the random-abnormal data by DCSM on data streams of two sensors

6 Conclusion

In this paper, we propose DCSM, a Data Cleaning approach based on SDAE and Multi-sensor collaborations. DCSM first group all sensors into clusters. For each cluster, DCSM detects all abnormal data by SDAE, then DCSM differentiate the random-abnormal data and event-abnormal data by multi-sensor collaborations. The event-abnormal data caused by events are unchanged, while the random-abnormal data caused by random factors are repaired. We conduct extensive real-data based simulations to show the efficiency of DCSM.

Acknowledgement: This work is supported by the National Natural Science Foundation of China (Grant No. 61672282) and the Basic Research Program of Jiangsu Province (Grant No. BK20161491).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Bolourchi, P.; Uysal, S. (2013): Forest fire detection in wireless sensor network using fuzzy logic. *Fifth International Conference on Computational Intelligence, Communication Systems and Networks*, pp. 83-87.

Chen, P.; Yang, S.; Mccann, J. A. (2015): Distributed real-time anomaly detection in networked industrial sensing systems. *IEEE Transactions on Industrial Electronics*, vo1. 62, no. 6, pp. 3832-3842.

Christopher, T.; Divya, M. T. (2015): A comparative analysis of hierarchical and partitioning clustering algorithms for outlier detection in data streams. *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 12, pp. 273-281.

Dai, J.; Song, H.; Sheng, G.; Jiang, X. (2017): Cleaning method for status monitoring data of power equipment based on stacked denoising autoencoders. *IEEE Access*, vol. 5, pp. 22863-22870.

Frey, B. J.; Dueck, D. (2007): Clustering by passing messages between data points. *Science*, vol. 315, no. 5814, pp. 972-976.

Gao, F.; Wen, H.; Zhao, L.; Chen, Y. (2013): Design and optimization of a cross-layer routing protocol for multi-hop wireless sensor networks. *International Conference on Sensor Network Security Technology and Privacy Communication System*, pp. 5-8.

Helwig, N.; Pignanelli, E.; Schütze, A. (2015): D8.1-Detecting and compensating sensor faults in a hydraulic condition monitoring system. *Association for Sensors and Measurement Conferences*, pp. 641-646.

He, W.; Qiao, P.; Zhou, Z.; Hu, G.; Feng, Z. et al. (2018): A new belief-rule-based method for fault diagnosis of wireless sensor network. *IEEE Access*, vol. 6, pp. 9404-9419.

Hinton, G. E.; Salakhutdinov, R. R. (2006): Reducing the dimensionality of data with neural network. *Science*, vol. 313, no. 5786, pp. 504-507.

Kriegel, H. P.; Kröger, P.; Schubert, E.; Zimek, A. (2009): LoOP: local outlier probabilities. *18th ACM conference on Information and Knowledge Management*, pp. 1649-1652.

Liu, F.; Ting, K.; Zhou, Z. (2009): Isolation forest. *Eighth IEEE International Conference on Data Mining*, pp. 413-422.

Liu, H.; Chen, J.; Huang, F.; Li, H. (2017): An electric power sensor data oriented data cleaning solution. *14th International Symposium on Pervasive Systems, Algorithms and*

Networks & 11th International Conference on Frontier of Computer Science and Technology & Third International Symposium of Creative Computing, vol. 1, pp. 430-435.

Mohamed, H. H.; Kheng, T. L.; Collin, C.; Lee, O. S. (2011): E-Clean: a data cleaning framework for patient data. *First International Conference on Informatics and Computational Intelligence*, pp. 63-68.

Randive, N.; Sneha; Singh, N.; Singh, R.; Abin, D. (2014): Hybrid approach for outlier detection in high dimensional data. *International Journal of Engineering Research & Applications*, vol. 4, no. 4, pp. 31-35.

Salehi, M.; Leckie, C.; Bezdek, J.; Vaithianathan, T.; Zhang, X. (2016): Fast memory efficient local outlier detection in data streams. *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 12, pp. 3246-3260.

Sharma, A. B.; Golubchik, L.; Govindan, R. (2010): Sensor faults: detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks*, vol. 6, no. 3, pp. 1-39.

Shi, W.; Zhu, Y.; Zhang, J.; Tao, X.; Sheng, G. et al. (2015): Improving power grid monitoring data quality: an efficient machine learning framework for missing data prediction. *IEEE 17th International Conference on High Performance Computing and Communications, IEEE 7th International Symposium on Cyberspace Safety and Security & IEEE 12th International Conference on Embedded Software and Systems*, pp. 417-422.

Vincent, P.; Larochelle, H.; Bengio, Y.; Manzagol, P. A.; Cohen, W. W. et al. (2008): Extracting and composing robust features with denoising autoencoders. *25th International Conference on Machine Learning*, pp. 1096-1103.

Wang, Q.; Kundur, D.; Yuan, H.; Liu, Y.; Lu, J. et al. (2016): Noise suppression of corona current measurement from HVdc transmission lines. *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 264-275.

Xiao, B.; Wang, Z.; Liu, Q.; Liu, X. (2018): SMK-means: an improved mini batch K-means algorithm based on mapreduce with big data. *Computers, Materials & Continua*, vol. 56, no. 3, pp. 365-379.

Zhang, G.; Li, R. (2017): Fog computing architecture-based data acquisition for WSN applications. *China Communications*, vol. 14, no. 11, pp. 69-81.

Zhang, P.; Feng, X.; Zhou, J. (2013): Outlier detection technique based on cluster analysis and spatial correlation in wireless sensor networks. *Application Research of Computers*, vol. 30, no. 5, pp. 1370-1373.

Zhang, Y.; Meratnia, N.; Havinga, P. (2010): Outlier detection techniques for wireless sensor networks: a survey. *IEEE Communications Surveys & Tutorials*, vol. 12, no. 2, pp. 159-170.

Zhang, Y.; Szabo, C.; Sheng, Q. (2015): Cleaning environmental sensing data streams based on individual sensor reliability. *International Conference on Web Information Systems Engineering, Lecture Notes in Computer Science*, vol. 8787, pp. 405-414.

Zimek, A.; Campello, R. J. G. B.; Sander, J. (2014): Ensembles for unsupervised outlier detection: challenges and research questions a position paper. *ACM SIGKDD Explorations Newsletter*, vo1. 15, no. 1, pp. 11-22.