Hybrid Method for Automatic Anomaly Detection in Heterogeneous Wireless Sensor Networks

Wenhui Yu,1* Rongfu Qiu,2 and Fanzhu Jin3**

1China Southern Power Grid Company Co., Ltd., Guang Dong 510530, China
2Digital Grid Research Institute, China Southern Power Grid, Guang Dong 510530, China
3College of Mathematics and Information Engineering, Longyan University, Long Yan City 364012, China

(Received January 6, 2020; accepted June 9, 2020)

Keywords: edge data, cloud data, heterogeneous sensor networks, anomaly detection, wireless sensor

Anomaly detection is very important in the data analysis of heterogeneous sensors. The analysis of anomalies of sensors in detection systems of networks is essential to manage and maintain them. We propose a hybrid method of edge data analysis and cloud data analysis to improve the accuracy and efficiency of previous methods. The algorithm combining edge data analysis and cloud data analysis in heterogeneous sensor networks performs automatic anomaly detection satisfactorily. The new method can be applied to any ambient changes to identify short-term and long-term anomalies. This method overcomes the shortcomings of edge data analysis and cloud data analysis when each method is used by itself.

1. Introduction

A wireless sensor network (WSN) is a distributed network structure, which consists of groups of autonomous networked electronic devices (sensor nodes) that collect data from the surrounding environment. The data processed by a WSN includes temperature, humidity, light, noise, current, voltage, power, and so forth. The improvement of electronic technologies and computing power has increased the use of WSNs. The increasing number of WSNs has generated the need for an effective management technology to deal with the complexity of networks and the huge amount and diversity of sensor data. In the field of WSNs, sensor data analysis for automatic anomaly detection is a challenging research topic. As anomaly detection is important for finding outliers in a system and for managing a system more economically in terms of cost and effort, data analysis to find anomalies has been widely studied in statistics and machine learning. There are several ways to analyze data, but we focus on how a sensor system and its environment produce an anomaly, that is, unexpected changes in sensor data.

Abnormal conditions and operation of a WSN originate from device power shortage, deviations from unexpected operations, and device failure. However, it is very difficult to identify abnormal conditions and operation of a WSN from normal ones in a sensor
environment. Most methods of detecting anomalies analyze the data flow generated by a single device using various technologies to find exceptional signals. The technologies are mainly statistical methods based on the complex mathematical analysis of the data flow.\textsuperscript{6–9} These are appropriate for detecting specific anomalies of data in general, but not the data from heterogeneous WSNs. Several researchers considered the anomalies as numerical values or data with the assumption that all the data were normal. For example, Xie et al. researched the description and detection of anomalies by considering graphs. They provided a general structure overview and classifications under various settings to distinguish anomalies from normal data in financial auctions and social networks.\textsuperscript{(9)} On the basis of mathematical analysis or machine learning algorithms for different data levels, several studies showed how to detect intrusions in a security system and credit card fraud,\textsuperscript{(10–13)} compare an input data package with existing ones, and identify behavioral characteristics and spatial anomalies.\textsuperscript{(14–17)} Mocanu et al. proposed a method of monitoring heterogeneous WSNs and identifying hidden correlations between heterogeneous sensors without any definition of anomalies or their detection.\textsuperscript{(18)} On the whole, the existing detection methods only rely on available resources in the network, and thus, they cannot effectively monitor the whole data flow to discriminate anomalies.\textsuperscript{(19–26)}

Considering the above limitations of the previous methods, we propose a new monitoring framework for the anomaly detection of heterogeneous WSNs. The new framework combines two different methods: (1) an edge data analysis method to analyze sensor data from each node in a network locally and (2) a cloud data analysis method to compare the data from multiple heterogeneous sensors throughout the network. This combination improves the accuracy of anomaly detection even when it is applied to large and complex WSNs, overcoming the limitations of each method when it is used alone. We provide a new method of detecting short- and long-term anomalies that cannot even be captured by sensor nodes that can be used for WSNs of all sizes since they can be used for cloud-based systems.

2. Design of Hybrid Anomaly Detection Method

2.1 Design of anomaly detection framework based on edge data analysis and cloud data analysis

We created the hybrid anomaly detection framework for heterogeneous sensor networks by combining edge data analysis (to identify short-term anomalies) and cloud data analysis (long-term anomalies). This hybrid method aims to avoid the defects and adopt the advantages of each method.

Edge data analysis misses changes in sensor data over a relatively long period, while cloud data analysis identifies long-term anomalies. Edge data analysis detects single and short-lived abnormalities that are mainly caused by local problems or noises and does not use historical data. However, it fails to detect local abnormalities caused by a long-term problem in a sensor. In contrast, cloud data analysis deals with data over a longer period of time and detects abnormalities that cannot be detected by edge data analysis. The combination of the two types of analysis not only reduces the load on the network and cloud, but also detects previously
hidden long-term abnormalities on the nodes. Figure 1 shows the hybrid framework topology of anomaly detection based on edge and cloud data computing methods. The edge data analysis runs continuously on each node of a WSN and does not require communication between nodes. In this case, the detection of short-term exceptions is performed at the node level. If the edge data analysis finds any short-term exceptions on a node, it issues a short-term alert and sends it to the cloud for further processing by the cloud data analysis.

The cloud data analysis identifies the sensors with anomalous information, that is, nodes of possible concern. Then, it processes the data in the existing data window of the cloud. In the processing, there is no need to wait for data generation. When long-term abnormal data are detected in the previous data window, a long-term alarm is issued. Then, tasks on each sensor run repeatedly. A task is suspended until a new long-term exception is detected from a sensor within a given period of time and a new short-term alarm is received from the sensor.

2.2 Optimization technology of edge data analysis

Generative replay technology has been proposed in several studies. However, online anomaly detection was not performed in these studies and the data distribution was imported online. Therefore, we used the generation model and replay technology of the restricted Boltzmann machine (RBM) to build a short-term anomaly detection of edge data analysis that can perform online anomaly detection. Similarly to any other RBM variant of offline training, during learning, minimizes the reconstructed version of the input data (expressed as the error between \( \hat{v} \) and the input data \( v \)). The transformation data \( \hat{x} \) of the input data follows the Gibbs sampling method only after approaching the visible neuron \( v \) from the original data point, then the activation of visible neuron \( x \) is calculated by inferring

Fig. 1. Anomaly detection framework and workflow based on edge data and cloud data.
the activation of hidden neuron \(h\). Finally, the activation of visible neuron \(\hat{v}\) is calculated by inferring the activation of the hidden neuron, and the activation value of the latter is given as \(\hat{x}\).

Sensor measurements occur at a specific time interval. At any specific time \(t\), all sensors of the node give and collect new measurements expressed in the vector \(x^t\). Starting with \(t = 0\) in a continuous cycle, \(RBM^t_{OCD}\) carries out online training to simulate all measurements of \(t\). Mocanu et al.\(^{(22)}\) and Vega et al.\(^{(23)}\) suggested the method of calculating the reconstruction error of unknown data points and the similarity measurement of training data points for offline training of the RBM. For example, if \(t\) is abnormal at a certain time, the reconstruction error of measurement \(X^t\) is very different from that of measurement \(X^{t-1}\), both of which are reconstructed with \(RBM^t_{OCD}\). The greater the difference, the greater the probability of abnormality. To quantify \(RBM^{t-1}_{OCD}\), it is necessary to use \(m_{AnGe}\) as the metric, which is a root mean square error (RMSE), to reconstruct the error, which can be calculated as

\[
m_{AnGe} = \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \left( \hat{x}_{i}^t - x_{i}^t \right)^2} - \sqrt{\frac{1}{n_v} \sum_{i=1}^{n_v} \left( \hat{x}_{i}^{t-1} - x_{i}^{t-1} \right)^2}.
\] (1)

In addition, if more similar \(X^t\) measurements occur, \(RBM^t_{OCD}\) expands its coding data distribution to include these types of measurements so that they are no longer considered exceptions. It must be emphasized here that \(RBM_{OCD}\)-weighted connections need to be stored in the device memory to be suitable for the online detection of exceptions in wireless nodes.

2.3 Optimization technology of cloud data analysis

We introduced a string similarity measure in the cloud data analysis, which is called the multiparametric edit distance (MPED), to identify long-term anomalies.\(^{(24)}\) The MPED calculates the minimum edit distance between two strings under one constraint to find the best matching pattern. To describe the cloud data analysis, it is first necessary to illustrate the data flow. Let \(N\) be a group of nodes and \(S\) be a group of sensors. Each sensor \(s \in S\) is equipped on a node \(n \in N\) that accommodates multiple sensors. To simplify the representation, it is assumed that each sensor \(s \in S\) is uniquely identified in the set, and if necessary, the function \(\gamma: S \to N\) returns node \(n\) provided by sensor \(s\).

General-purpose sensors collect data on a regular basis: the observed value is defined as the value \(v\) collected by the sensor at a specific time \(t\) and is expressed as \(a_{i}^{(0)}\), supposing that \(t\) stores the complete timestamp (date/time) of the collection.

A group of sensors operate for any amount of time \(T\). Any time series \(t_i, t_{i+1}, \ldots, t_{i+k-1}\) defines an interval in which a large section of data (an ordered observation sequence) is collected, and it must be converted into a string to apply the MPED. In addition, it is necessary to carry out the observation at a specific time interval to analyze the behavior of sensors.

According to the context of the application, we analyzed data by the hour or day. The functions \(\rho(t_d)\) and \(\delta(t_d)\) for the day \(d\) and the time sequence of a specific hour \(h\) belong to the set \(\gamma(d, h)\) defined as Eq. (2) with conditions of \(d \in [1, D]\), \(h \in [1, 24]\).
The set $\gamma$ forms the basis of the string construction provided for the MPED and is formalized as follows. Given sensor $S_i$, day of interest $d$, and hour of interest $h$, the corresponding observation sequence is an ordered sequence,

$$q(S_i, d, h) = \{\Psi(a_{S_i}(t_k)) | t_k \in \gamma(d, h)\}, \quad (3)$$

where $\Psi$ transforms each individual observation into the corresponding symbolic representation. This means that observations can be completed in designated hours or days.

Then, the cloud data analysis and optimization technology are designed into two stages: training and operation. The training stage is dedicated to training the system under “normal” operating conditions to familiarize it with the expected information for each sensor and each time period. The operation stage involves learning the information in the training stage to identify potentially abnormal conditions. A novel feature of the long-term anomaly detection method in this study is that it does not calculate the expected values of various sensors in the training stage but the expected correlation between sensors. In the training stage, for each sensor and each time period, the most relevant sensor in a certain time period is selected for analysis. In fact, the significant correlation between the two changes makes it easier to detect potential anomalies.

(1) Training stage

At the beginning, the average correlation between each pair of sensors per hour is calculated within a fixed period of the training day $D_T$. For each pair of sensors $S_i$, $S_j$, and $h \in [1, 24]$, $C(S_i, S_j, h)$ is defined as the average correlation of days in $D_T$, and the calculation formula is

$$\forall S_i, S_j, h \quad C(S_i, S_j, h) = \frac{1}{D_T} \sum_{d \in D_T} 1 - L\left(q(S_i, d, h), q(S_j, d, h)\right). \quad (4)$$

Using $C$, for each sensor $S_i$ and hour $h$, the matching sensor $S_{i,h}^*$ of $S_i$ can be defined as

$$S_{i,h}^* = \tau(S_i, h) = \arg\max_{S_j} \left\{C(S_i, S_j, h)\right\}. \quad (5)$$

Finally, for each sensor $S_i$ and hour $h$, the expected correlation of the sensor is defined, and its pairing is $\eta(S_i, h) = C(S_i, \tau(S_{i,h}), h)$.

As an example, Table 1 shows the correlation calculated over an $N$-day period for eight heterogeneous sensors, where $h = 1$.

Since sensors $S_1$ and $S_2$ are optimally correlated over an hour interval, it is reasonable to expect a similar level of correlation for the corresponding time interval over $N$ different days. The matchings between the sensors with $h = 1$ extracted from the analysis in Table 1 is are $(S_1, S_2)$, $(S_3, S_4)$, $(S_5, S_7)$, and $(S_6, S_8)$. A similar calculation is performed here for the other values of $H$. 

\[
\gamma(d, h) = \{t_i | t_i \in T, \rho(t_i) = h, \delta(t_i) = d\} \quad (2)
\]
(2) Operation stage

The operation stage starts after the completion of the training stage. Here, each sensor has been associated with its fit. Therefore, the operation phase works as follows.

Given the threshold value $\theta \in [0, 1]$, for each sensor, each day $d$, and each hour $h$, the actual correlation, which is represented by $\chi(S_i, H, d)$, can be calculated as the correlation between sensor $S_i$ and its matching $\tau(S_i, h)$ in the form of

$$\forall S_i, d, h \quad \chi(S_i, h, d) = 1 - L^*\left(q(S_i, d, h), q(S_j, d, h)\right).$$

(6)

Now, when the actual correlation between $S_i$ and its coordination is significantly different from the expected value, potentially abnormal behaviors are detected under the following condition:

$$\left|\chi(S_i, h, d) - \eta(S_i, h)\right| > \theta.$$  

(7)

To reduce false positives, an alarm is issued if the average difference in the conditions is verified to be greater than the threshold value in a fixed number of hours $H^*$. The alarm calculation is performed with

$$\text{alert}(S_i, h, d) \leftarrow \text{avg}_{h' \in [h-H^*, h]} \left|\chi(S_i, h', d) - \eta(S_i, h')\right| > \theta.$$ 

(8)

In fact, when it is necessary to verify the behavior of a sensor in a certain period of time, its data must only be compared with the data from another sensor in the same slot, which can provide good performance in detecting the abnormal behavior.

2.4 Design of experiment

We designed a test experiment on a heterogeneous WSN with sensors operating in different areas of a building and different types of synthetic interference. The experimental steps are as follows: (1) determine the anomalies that can be identified by the edge data analysis in a given

<table>
<thead>
<tr>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>—</td>
<td><strong>0.9</strong></td>
<td>0.3</td>
<td>0.8</td>
<td>0.5</td>
<td>0.3</td>
<td>0.5</td>
<td>0.8</td>
</tr>
<tr>
<td><strong>0.9</strong></td>
<td>—</td>
<td>0.5</td>
<td>0.9</td>
<td>0.5</td>
<td>0.4</td>
<td>0.8</td>
<td>0.2</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>—</td>
<td><strong>0.9</strong></td>
<td>0.4</td>
<td>0.4</td>
<td>0.6</td>
<td>0.8</td>
</tr>
<tr>
<td>0.8</td>
<td>0.8</td>
<td><strong>0.9</strong></td>
<td>—</td>
<td>0.7</td>
<td>0.5</td>
<td>0.7</td>
<td>0.1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.4</td>
<td>0.7</td>
<td>—</td>
<td>0.3</td>
<td><strong>0.8</strong></td>
<td>0.4</td>
</tr>
<tr>
<td>0.3</td>
<td>0.4</td>
<td>0.4</td>
<td>0.5</td>
<td>0.3</td>
<td>—</td>
<td>0.3</td>
<td><strong>0.6</strong></td>
</tr>
<tr>
<td>0.5</td>
<td>0.8</td>
<td>0.6</td>
<td>0.7</td>
<td><strong>0.8</strong></td>
<td>0.3</td>
<td>—</td>
<td>0.5</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
<td>0.8</td>
<td>0.1</td>
<td>0.4</td>
<td><strong>0.6</strong></td>
<td>0.5</td>
<td>—</td>
</tr>
</tbody>
</table>
test case, (2) determine the anomalies that can be identified by the cloud data analysis in a given test case, (3) determine whether the edge data analysis can detect all anomalies of interest, (4) determine whether the alarm sent by the edge data analysis can activate the cloud data analysis.

As the experimental conditions, eight WSN nodes are deployed on one floor of the office of China Southern Power Grid Co., Ltd., which are TelosB nodes running TinyOS 2.1.2. With the building management framework (BMF), node management is implemented in a WSN with multihop organization. Figure 2 shows an example of BMF network with layers consisting of the base station nodes.

Figure 3 shows all the nodes deployed and their locations on the floor plan of the building. According to their location, the deployed nodes are grouped in pairs as follows.

(1) Nodes 1 and 124 are attached to the window of an office. These nodes can be reached by direct sunlight.

(2) Nodes 17 and 27 are placed on the bookcase of an air-conditioned office. These nodes are less affected by sunlight than nodes 1 and 124.

(3) Nodes 25 and 31 are placed on the table of an air-conditioned and artificially lit laboratory.

The experiment is carried out over 27 days divided into three phases of 9 days.

(1) In the first phase, all nodes work normally (without interference) and are powered on.

(2) In the second phase, some interference is introduced by the experimenters at nodes 1, 17, 31, and 5. In particular, node 1 is covered with a thick paper sheet, and a bag filled with silicon is placed near node 1, a lit light bulb is placed adjacent to nodes 17 and 31, and a bag of silicon is placed near node 5.

(3) In the third phase, no nodes are disturbed. However, nodes 1, 17, 25, and 28 are powered by batteries.
3. Results and Discussion

3.1 Edge data analysis of short-term anomaly detection

The model of short-term anomaly detection (RBMOCD) based on edge data analysis is set to have three visible neurons and 10 hidden neurons. In addition, a total of 43 parameters are generated and stored in memory. After each measurement, the model parameters are continuously updated online. Before each update, the current model generates three samples to avoid unintended erasure during the learning of new data for measurement.

The graphs in Fig. 4 show the results of the edge data analysis to detect exceptions on specific nodes. In the graphs, the x-axis represents the time of measurement. The left y-axis and the right y-axis (red) represent the measured value of the sensor and the \( m_{\text{AnGe}} \) value, respectively. Figure 4 shows how the analysis detected abnormalities at each time. For example, in Fig. 4(c), \( m_{\text{AnGe}} \) is close to zero, indicating that there is no abnormality in this analysis, but a strong abnormality occurred at about 8000 min. For node 17, \( m_{\text{AnGe}} \) showed values between −100 and +100 exactly when the node was interfered with by the experimenters. This indicated a new mode of sensor measurement. \( m_{\text{AnGe}} \) sometimes showed values as if a node had been interfered with. We did not set the threshold value of \( m_{\text{AnGe}} \) because the main goal of this method is to show changes in a short-term anomaly detection model based on edge data analysis. Setting a threshold value in accordance with the requirements of the application is expected to enhance the sensitivity of the detection model.

3.2 Cloud data analysis of long-term anomaly detection

The results of the cloud data analysis are shown in Figs. 5–7, where the x-axis represents the time of measurement and the y-axis represents the value of \( |\chi(S_i, h', d) - \eta(S_i, h')| \) obtained from light, temperature, and humidity sensors. To simplify the presentation, values greater than the threshold are displayed, with each corresponding to an alarm. In this case, it is important to set a threshold value because the equation can be used to measure the difference between the expected and calculated values. If there is no threshold value, it is difficult to read the graph. Thus, the sensitivity of the method can be easily controlled by precisely adjusting the threshold value.

For node 1, it is observed that a long-term anomaly did not issue an alarm even in the case of interference. Even when the light intensity was temporarily reduced with a thick sheet of paper, the sensor could detect long-term changes in light because the edge data analysis also recognized many small interferences. However, when disturbed by a light bulb, nodes 17 and 31 became almost unable to detect changes in light and issued a long-term alarm throughout the test. In the application of hybrid edge and cloud data analysis, the edge data analysis activated the cloud data analysis to confirm the persistence of abnormal conditions or classify occasional alarms. Node 124 did not detect any long-term anomalies as it was not subject to external interference even when adjacent sensors were interfered with. Nodes 25 and 27 were close to the sensors interfered with by the light bulb (nodes 17 and 31) and showed slight anomalies.
Fig. 4. (Color online) Short-term anomaly detection results [blue line: temperature; green line: light; light blue line: humidity] (x-axis: time (min); y-axis: value (number)]. (a) Node ID 1, (b) Node ID 5, (c) Node ID 17, (d) Node ID 25, (e) Node ID 27, (f) Node ID 28, (g) Node ID 31, and (h) Node ID 124.
The results are consistent with those of the short-term edge data analysis. Nodes 5 and 28 did not create long-term anomalies, although some anomalies were observed from time to time. As these nodes were located in corridors, they may often yield irregular data, some of which was caused by humidity produced by a bag filled with silicon. The temperature sensor did
not give out any specific alarm as there was no human interference with the measurement of
temperature. Node 25 was an exception owing to the malfunction of its batteries. Humidity
also did not cause any specific anomaly except for node 1 when a silicon bag was placed close
to it during the test. Node 5 also showed some anomalies, which were not long-term, when the
silicon bag was placed near it.

Fig. 6. Alarms (with threshold) for long-term anomaly detection of temperature sensors [x-axis: time (min),
y-axis: value (number)]. (a) Node ID 1, (b) Node ID 5, (c) Node ID 17, (d) Node ID 25, (e) Node ID 27, (f) Node ID 28, (g) Node ID 31, and (h) Node ID 124.
4. Conclusion

Automatic anomaly detection in heterogeneous WSNs is challenging. The signals produced by sensors are affected by changes in ambient circumstances, which may produce anomalies. As WSNs deal with a large amount of information, data analysis with an algorithm designed
for local data preprocessing needs to be optimized for further processing. We discussed and proposed a combination of long-term and short-term algorithms that can identify anomalies in WSNs more accurately and effectively than the individual algorithms. Our short-term method based on edge data analysis showed good performance in identifying potential anomalies from local sensors. The method transferred anomalous data for cloud data analysis for further long-term analysis. The long-term method performed well in an experiment to identify the impact of anomalies. In general, the hybrid edge and cloud data analysis method can solve problems originating from using a single method. The results verified the timeliness and accuracy of the hybrid method in detecting anomalies. However, it is still necessary to automate the overall process by using refined threshold values.

Acknowledgments

This work was supported by a science and technology project of China Southern Power Grid Co., Ltd. (Grant Nos. ZDKJXM20180009 and ZBKJXM20180347).

References


About the Authors

**Wenhui Yu** received his M.S. degree from Hunan University, China, in 1995. He works for China Southern Power Grid Corporation in the Production Technology Department, carrying out distribution network technology management as a professor-level senior engineer. His main research field is distribution network technology and management. (yuwenhui197604@163.com)

**Rongfu Qiu** received his M.S. degree from Shanghai Electric Power University, China, in 2011. He is a senior manager of China Southern Grid Digital Grid Research Institute Co., Ltd., with the title of engineer. His main research fields are power-dispatching automation systems and substations automation technology and management. (qionrongfu1984@163.com)

**Fanzhu Jin** received his Ph.D. degree from Seoul National University, Republic of Korea, in 2001, then started working at Gdansk Medical University as an assistant professor. He is affiliated with the College of Mathematics and Information Engineering, Longyan University, Fujian, China. His research interests are in IoT and its applications. (dennisbskim@hotmail.com)