Anomaly Detection for Internet of Things Based on Compressed Sensing and Online Extreme Learning Machine Autoencoder

Yun Yu\textsuperscript{a}, Xiaojun Wu\textsuperscript{*}, Sheng Yuan\textsuperscript{b}

School of Software Engineering, Xi'an Jiaotong University, China
\textsuperscript{a}yuyun1117@stu.xjtu.edu.cn, \textsuperscript{b}ysheng@xjtu.edu.cn
\textsuperscript{*}xiaojunw@xjtu.edu.cn

ABSTRACT Abnormal events refer to specific events, such as forest fire, occurring in the wireless sensor networks of the Internet of Things (IoT), whose behaviors are quite different from normal events. By learning the underlying structure of sensor data, users can be helped to learn about the occurrence of these events as soon as possible. Due to the large number of sensors in the IoT and the periodic collections of data, sensor data has the problems of high dimensions and large amount, and the transmission of a large amount of data in the network is not a small challenge for bandwidth. In addition, the sensor data is unlabeled, so it is time-consuming and unrealistic to manually label all the data. Abnormal events in the IoT require low delay, such as gas concentration monitoring. In the IoT, data is generated continuously, so a well-trained model cannot remain unchanged, and features of new data need to be continuously learned. On account of the limitation of hardware, edge nodes cannot undertake the complicated and time-consuming task of model training and detection. None of the existing algorithms can meet the above requirements well, so this paper proposed an algorithm based on Compressed Sensing and Online Extreme Learning Machine Autoencoder named COELMAE. The proposed algorithm can carry out anomaly detection in low delay, unsupervised and online learning, and also has low computational complexity. What's more, the algorithm can reduce the amount of data transferred about 60%.

CCS Concepts •Computing methodologies → Machine learning → Learning paradigms → Unsupervised learning → Anomaly detection.

1. INTRODUCTION
With the rapid development of IoT related applications and the progress of communication technology, IoT applications (such as smart home, smart power grid, smart city, etc.) are widely used around the world. According to Statista, the number of devices connected to the Internet of things will jump from 20.35 billion in 2017 to 75.44 billion in 2025. In recent years, with the popularization of IoT, a large number of sensor devices have been widely deployed in different fields, such as chemical industry, financial industry, biomedical industry, etc. These sensor devices not only affect people's lifestyle, but also generate a lot of time series data. In the IoT scenarios, the sensors monitor the environmental data at a certain frequency and send the collected data to the data receivers of the edge nodes. A large number of data in the IoT not only accurately reflect the real-time change of environmental states, but also explain the development trend and change rule of the state of the IoT within a certain time range. Therefore, the sensor data collected based on a large number of sensor devices not only provide an
important data source for real-time data visualization monitoring, but also server as the basis and premise for further data mining.

Abnormal detection refers to finding data from the sample set whose behavior characteristics are significantly different from other samples. Reasons for abnormal data include: specific events in the node area (such as elevated temperature sensor reading in case of fire), sensor failure, and external factors [1]. As one of the main tasks of the state monitoring of the IoT system, abnormal event detection has gradually become a widely concerned research direction of researchers and some research achievements have been obtained. In order to satisfy the availability of the Internet of things system, a large number of wireless sensors are deployed in places with limited bandwidth and energy consumption. The dense deployment of sensor nodes in the IoT provides environmental monitoring in collaboration [2].

According to the time feature of abnormal detection data, abnormal event detection technology can be divided into online and offline. Since the offline anomaly event detection method is not suitable for data flow, our research focuses on online learning. Online anomaly event detection is to carry out anomaly detection on the data stream through the regression prediction of the time series data. Ahmad et al [3] conducted real-time unsupervised processing of data flow based on an online sequential storage algorithm hierarchical time memory (HTM). In the practical application process, the value read by the sensor in the anomaly detection algorithm model can be seen as a random process composed of both the state and the time of discrete values [4]. In literature [5], the minimum spanning tree was used to approximate and capture manifold structures, which improved the detection capability, and the proposed algorithm was applied to hydropower stations. The size of the sliding window is an important factor affecting the efficiency and accuracy of the model. It is difficult to determine the size of the sliding window because the acquisition frequency of different types of sensors is different. In literature [6], according to the sliding basic window sampling algorithm and Gaussian process regression, an anomaly detection algorithm for uncertain multi-data flow based on SBWS_GPR prediction model was proposed. Zhang et al [7] built a distributed sensor data model, and proposed a time-serie-based anomaly detection algorithm based on the continuity of single-source time series and the correlation of multi-source time series. Janjua et al [8] built an unsupervised abnormal event detection system based on sliding time window. Wu et al [9] proposed an anomaly detection algorithm based on multi-window mechanism and high-dimensional big data to solve the problem of misjudgment caused by real-time detection of data flow in a single window.

Online abnormal event detection in the IoT mainly faces three big challenges. 1) due to the large number of sensors in the IoT and the periodic collection of data, there exists the problem of high dimensions and large amount of data. It is obviously impractical to put all the data into the model for offline training at one time. The detection algorithm need to keep learning about new coming data. 2) wireless sensor nodes are limited in hardware computing capacity, battery capacity, storage space and communication range. 3) the generated data is unlabeled. The existing anomaly detection can't satisfy the requirement of low delay, high accuracy and low cost.

Therefore, this paper proposes a real-time, unsupervised, online learning algorithm with low computational complexity, named COELMAE. The following chapters of this paper are organized as follows: section 2 briefly introduces relevant theoretical knowledge, section 3 addresses the architecture of proposed COELMAE, section 4 is the experiment and evaluation, and section 5 is the conclusion of this paper.

2. RELATED WORK
In this section, we briefly introduce the relevant theoretical knowledge: Compressed sensing(CS) and OS-ELM.

2.1 Compressed Sensing
Compressed sensing theory states that as long as the signal is compressible or sparse in a certain transform domain, the original high-dimensional signal can be projected onto a low-dimensional space
through an observation matrix that is not related to the sparse transform. Then an optimization problem
can be solved to reconstruct the original signal from these small amounts of compressed signals. It has
been proven that such compression contains sufficient information needed to reconstruct the signal.
According to the compressed sensing theory, the data values collected by the sensor can be expressed
as formula:

\[ x = \Psi \alpha \]  

(1)

where \( \Psi \) is a sparse basis. Fourier transform, wavelet transform, and discrete cosine transform can be
selected as sparse basis. \( \alpha \) is the sparse coefficient. The sensor node transmits the observation value \( g \)
of the data to the sink node in a dimension much lower than the original dimension, where \( g \) is the
observation matrix:

\[ g = \Phi x = \Phi \Psi \alpha \]  

(2)

When the observation \( g \) is received, the receiving node can reconstruct the sensor data by formula 3:

\[ \min_{\alpha} ||\alpha||_0 \ s.t \ g = \Phi \Psi \alpha \]  

(3)

However, the \( l_0 \) norm solution is an NP-hard problem, which converts the \( l_0 \) problem to the \( l_1 \)
problem, that is:

\[ \min_{\alpha} ||\alpha||_1 \ s.t \ g = \Phi \Psi \alpha \]  

(4)

There are many compressed sensing reconstruction algorithms, mainly divided into two categories:
greedy algorithms and convex optimization algorithms. Because the SP algorithm has fast calculation
speed and high reconstruction probability, we choose the SP algorithm for reconstruction. The SP
algorithm introduces the idea of backtracking. According to the sparseness of the prior knowledge signal,
the correct subspace is selected in the observation signal \( g \), and then judge whether the signal is in the
current estimation or not. If it does not exist, remove the unavailable atoms and add the same number of
available atoms until the measurement signal \( \alpha \) is sufficiently close.

2.2 Online Sequential Extreme Learning Machine (OS-ELM)

Since it was proposed, the Extreme Learning Machine (ELM) had the advantages of strong
generalization ability, fast learning speed and high accuracy [10]. In the actual IoT scenarios, the sensors
(training data) are not obtained all at once, but arrived at the edge nodes one by one or in batches. In
order to monitor the changing data flow in real time, the traditional ELM approach was to learn all
existing new and old data when new data samples were generated. This not only increased the
computational complexity, but also needed to store all sample data. With more and more sample data,
the computational complexity increased, which was not conducive to long-term operation. In summary,
ELM algorithm is not suitable for application in real-time IoT anomaly detection. Therefore, we consider
introducing an online sequential extreme learning machine (OS-ELM) to meet the continuous learning
needs of real-time application scenarios [11]. OS-ELM is a kind of ELM that can dynamically adjust
the output layer weight \( \beta \) according to the newly collected data samples without retraining the model.
OS-ELM is divided into two parts, the initialization learning phase and the online learning phase.

In the initial stage, the output weight \( \beta_0 \) of the network is obtained through a small number of
samples. The input weight \( w \) and bias \( b \) of the hidden layer are randomly generated orthogonal matrices.
Once the input weights and bias are randomly determined, they can’t change through the whole
modeling, and the values of the input weights and bias remain the same during the initial learning and
online learning stages. It is supposed that there are \( n_0 \) \( (n_0 \in \mathbb{N}) \) initial training samples from
\( (x_i, t_i) \), where \( X_t = [x_{t1}, x_{t2}, ..., x_{tn}]^T \in \mathbb{R}^n, t_i = [t_{i1}, t_{i2}, ..., t_{im}]^T \in \mathbb{R}^m \), the hidden layer output matrix
is:

\[ H_0 = g(w_i \cdot x_i + b_i), i = 1, 2, ..., n_0 \]  

(5)

where \( g \) is activation function. The cost function is \( \| H_0 \beta_0 - T_0 \| \). According to the calculation method
of generalized inverse, \( \beta_0 \) is calculated as:

\[ \beta_0 = K_0^{-1} H_0^T T_0 \]  

(6)
where $K_0 = H_0^T H_0$.

In the online learning stage, when a new set of sample enters the model, assuming that $N_1$ samples enter the model, the $\beta_1$ becomes:

$$
\beta_1 = K_1^{-1} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}^T \begin{bmatrix} T_1 \\ T_0 \end{bmatrix} \tag{7}
$$

where $K_1 = \begin{bmatrix} H_0^T \\ H_1^T \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix}$, for online learning, $K_1$ is expressed as a function of $K_0$, and $K_1$ is formulated as:

$$
K_1 = \begin{bmatrix} H_0^T \\ H_1^T \end{bmatrix} \begin{bmatrix} H_0 \\ H_1 \end{bmatrix} = K_0 + H_1^T H_1 \tag{8}
$$

From the formulas 7 and 8, a recursive formula of $\beta_1$ and $\beta_0$ can be obtained, and the formula is extended to $\beta_k$. The recursive formulas for online learning are as follows:

$$
K_{k+1} = K_k + H_{k+1}^T H_{k+1} + \beta_k \tag{9}
$$

$$
\beta_{k+1} = \beta_k + K_{k+1}^{-1} H_{k+1}^T (T_{k+1} - H_{k+1} \beta_k) \tag{9}
$$

where $K_{k+1}^{-1}$ obtained from Woodbury’s formula:

$$
K_{k+1}^{-1} = (K_k + H_{k+1}^T H_{k+1})^{-1} = K_k^{-1} - K_k^{-1} H_{k+1}^T (1 + H_{k+1} K_k^{-1} H_{k+1}^T)^{-1} \times H_{k+1} K_k^{-1} \tag{10}
$$

3. PROPOSED COELMAE FOR ANOMALY DETECTION IN IOT

The IoT application is often composed of a large number of sensors, and these nodes are deployed in the same area. The nodes in the area can be divided into two types according to different functions: sensor nodes and sink nodes. The sensor nodes periodically collect sensor data and transmit the data to the sink nodes. Sensor nodes generally do not have strong computing capabilities. The sink nodes are different from the cloud center. These sink nodes are closer to the sensor nodes and have certain computing capabilities but can’t undertake complex computing tasks. The tasks of these sink nodes are to collect data from sensors node in the area and upload it to the cloud center or a higher level node.

The abnormal event detection mechanism can be deployed on the central node or on distributed nodes. In the first case, the sensor data is analyzed and detected in a centralized big data processing mode based on cloud computing, which easily cause network transmission overhead and high energy consumption and high latency. Because of the increase in the number of edge sensors, the amount of data generated has also increased enormously. On account of the limitation of the computing capacity of the data center and the requirements of real-time performance, it is no longer suitable to transfer all the original data directly to the cloud center for data mining operations. Luo et al [12] designed a two-stage distributed anomaly detection algorithm. One part of this algorithm is deployed in the cloud center and another part is deployed at the sensor nodes. Therefore, it is necessary to adjust the existing centralized data processing architecture. According to the idea of edge computing, some computing tasks are offloaded of the cloud center to the edge nodes, which can reduce the network bandwidth load, and reduce the computing pressure of the cloud center.

In the IoT scenario, abnormal situations occur less frequently than normal cases, and the number of positive samples is much larger than the negative samples. Therefore, most of the data feature learning from the encoder describes normal data. Autoencoder is a special type of neural network whose goal is to reconstruct the input, rather than predicting certain target variables. The main idea of using autoencoder is that, the association between variables is affected, whenever abnormal events occur in the monitored environment (such as changes in temperature, pressure, vibration, etc.). When this happens, there is an increase in errors in network reconstruction of input variables. By monitoring the reconstruction error, we can get an "abnormal" indication of the monitored system. In fact, this reconstruction error will increase with the occurrence of abnormal events.
The online extreme learning machine autoencoder is improved on the basis of OS-ELM. The number of neurons in the input layer is equal to the number of neurons in the output layer. The output target matrix is equal to the input matrix. The number of input neurons is equal to the number of data’s feature. The online extreme learning machine autoencoder is a single hidden layer feedforward neural network. The biggest difference between this network and the traditional feedforward neural network (back propagation BP network) is that iterative training is not performed using gradient descent. The online extreme learning machine autoencoder has three advantages: 1) all parameters of online extreme learning machine autoencoder need to be calculated only once to achieve the optimal training speed, so it is suitable for application in IoT anomaly detection; 2) the online extreme learning machine autoencoder does not need to re-enter the newly generated data and the original data into the network for training to obtain the parameters of model, which is suitable for the continuous generation of sensor data; 3) unsupervised learning data characteristics. The online extreme learning machine autoencoder has two phases: the initialization phase and the online learning phase. First, the online extreme learning machine autoencoder is initialized with a small amount of sample data, and then the data generated by the sensor is continuously used to update the output weight without training the entire network.

Combining the data compression capability of compressed sensing and the online feature extraction capability of online extreme learning machine autoencoder, we proposed a Compressed Online Extreme Learning Machine AutoEncoder (COELMAE) anomaly detection algorithm. The architecture of COELMAE is shown in Figure 1. It can be seen from the figure that the sensing data generated by the sensor is not directly put into the autoencoder for training, but the data is compressed first. And the reconstructed data is entered into the network model for training. Compressed sensing not only reduces the amount of data transmitted, but also removes some noise during the compression process. The reconstructed data retains the most important features of the original data. The online extreme learning machine autoencoder does not require back-propagation and iterative training to obtain network parameters. This scheme is fast, and can obtain new network parameters without retraining. Unlike traditional anomaly detection algorithms, COELMAE is not all deployed at the central node. Inspired by edge computing, the algorithm is divided into two parts and deployed at the sensor node and the sink node respectively, where data compression is deployed at the sensor node, and data reconstruction and online extreme learning machine autoencoder is deployed at the sink node. The sensor nodes perform data compression steps, and the sink nodes perform data reconstruction, online learning, and detection tasks. In order to explain clearly the activities of each participating part of COELMAE, a COELMAE activity diagram of (unified modeling language, UML) is given, as shown in Figure 2. The COELMAE algorithm is divided into two phases: the initialization phase and the online learning phase.

![Figure 1. The architecture of COELMAE.](image-url)
1) Initialization Phase

During the initialization phase, the sink node uses the collected historical data to train online extreme learning machine autoencoder, and obtains the initial output weight $\beta_0$. This process uses offline batch processing to load training data into memory at one time. It should be noted that the data here is historical data that already exists in the sink node, and no data compression operation is required. The weights $w$ and biases $b$ of the hidden nodes are randomly generated at this stage to generate the output layer matrix $H_0$ by formula 5, and remain unchanged in the subsequent online learning stage. Calculate $\beta_0$ using following formula:

$$\beta_0 = P_0H_0^TX_0$$

$$P_0 = (H_0^TH_0)^{-1}$$  \hspace{1cm} (11)

2) Online Learning Phase

In the online learning phase, the sensor uses some kind of sparse transformation basis to compress the data $x \in \mathbb{R}^{m \times c_1}$. This data generated in a short period and needed to be transmitted it to the sink node through the network. At the sink node, the reconstruction algorithm SP is used to reconstruct original data $x$. Let the reconstructed data be $x'$. The reconstructed data is then put into the initialized online extreme learning machine autoencoder. The output value $x''$ of network is:

$$X_i'' = g(w_i \cdot x_i' + b_i) \beta_i \hspace{0.5cm} i = 1, 2, ..., m$$ \hspace{1cm} (12)

The reconstruction error MSE is calculated for the output values $x''$ and $x'$. If the reconstruction error is greater than the set threshold $\alpha$, the cloud center is notified to issue a warning to the user. The reconstruction error MSE calculation formula is formula 13:

$$\text{MSE} = \frac{1}{m} \sum_{i=1}^{m} (x_i'' - x_i')^2$$ \hspace{1cm} (13)

If the reconstruction error is not greater than the threshold, it means that the data is normal, and it is put back into the model for training. The calculation formula is as follows:

$$P_{k+1} = P_k - P_kH_{k+1}^T(I + H_{k+1}P_kH_{k+1}^T)^{-1} \times H_{k+1}P_k$$

$$\beta_{k+1} = \beta_k + P_{k+1}H_{k+1}^T(x_{k+1}' - H_{k+1}\beta_k)$$ \hspace{1cm} (14)

In the initialization phase, there is still a very small number of abnormal samples in the historical data putting into the network training. Although the small number of these samples has a small impact on the generation of model parameters, the errors they bring still exist. Therefore, it is considered that only normal samples are put into the online learning phase for online learning, so that the network model can better learn the features of normal samples and detect abnormal samples more accurately.
4. EXPERIMENT AND EVALUATION

We will introduce the data set, experimental settings, evaluation criteria, and experiment results in this section.

4.1 Dataset and Evaluation Criteria

The dataset we used is the Harvard unsupervised anomaly detection standard dataset shuttle dataset [13]. The dataset has 9 numerical features of the shuttle. The number of total samples is 46,464. The number of abnormal samples is 878 and the number of normal samples is 45,586. It is a standard dataset specifically for unsupervised anomaly detection learning. Using labeled anomaly detection datasets for unsupervised anomaly detection learning can determine whether the model training results are correct. The experimental dataset is normalized to [0,1] for each feature value, and is divided into a training data set and a test data set according to 8:2. The experiment was simulated using python. We used 0.4 Gaussian rate in compressed phase, which means only 40% data are sent to sink node.

A classic algorithm with significance performance in unsupervised anomaly detection is IsolatedForest. COELMAE will conduct a comparison test with IsolationForest. At the same time, in order to verify the effectiveness of the compression technology, COELMAE also compares the algorithm performance with the online extreme learning machine autoencoder. In order to prevent accidental errors, all tests are repeated three times.

Because the distribution of positive and negative samples of the data set is very uneven, the accuracy alone cannot fully measure the performance of the algorithm. Therefore, the experiment uses accuracy, precision, recall and F1 score to measure algorithm performance. Table 1 shows the confusion matrix for anomaly detection.

| Predict Result | Actual Result |
|----------------|--------------|
| Abnormal       | TP           |
| Normal         | FN           |
|                | FP           |
|                | TN           |
The calculation formula of accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \quad (15)$$

The calculation formula of precision is as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (16)$$

The calculation formula of recall is as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (17)$$

The calculation formula of $F_1$ score is as follows:

$$F_1 - \text{Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

4.2 Result Discussion

It can be seen from the Table 2 that the accuracy and precision of the two algorithms based on ELM are higher than those of the classic anomaly detection algorithm IsolationForest, which reflects the advantages of ELM. The accuracy of COELMAE is higher than that of the online extreme learning machine autoencoder model, which indicates that compressed sensing can remove redundant data and some noise. What’s more, the reconstructed data can better reflect the original data, which improves the accuracy and reduces about 60% of the transmitted data volume. The reason why IsolationForest is not suitable for the anomaly detection in IoT scenarios is that it needs to train all the training data on the model, which does not meet the characteristics of the continuous generation of the data stream and cannot continuously learn the characteristics of the new data. COELMAE is not training all the data at once, but learning the data stream incrementally.

| Algorithm                          | Accuracy | Precision | Recall   | F1 score |
|------------------------------------|----------|-----------|----------|----------|
| IsolationForest                    | 0.9115   | 0.2278    | 0.9118   | 0.3645   |
| online extreme learning machine autoencoder | 0.9971   | 0.9268    | 0.9101   | 0.9184   |
| COELMAE                            | **0.9975** | **0.9320** | **0.9263** | **0.9292** |

5. CONCLUSION

This paper mainly addresses the proposed anomaly detection algorithm COELMAE. There are two major improvements: 1) is compressed sensing technology is introduced in order to reducing the amount of data in the network transmission, which means generated data is compressed on the sensor node, while reconstructed data can replace the original data for training; 2) OS-ELM is promoted to unsupervised learning, which can solve the problem of unlabeled data. Data compression is carried out at the sensor node, and data reconstruction and training are carried out at the sink node. Through the experimental verification, the reconstructed data can achieve the better performance than origin data, and effectively reduce the data transmission amount about 60%, relieving the pressure of network bandwidth. Through the experimental verification of the public data set, the proposed algorithm can update the network model in real time according to the new data, and the update speed is fast, which meets the real-time requirement of abnormal events in the IoT scenario.

REFERENCES

[1] Fei, H., Xiao, F., Li, G. H., et al. An Anomaly Detection Method of Wireless Sensor Network Based on Multi-Modals Data Stream[J]. Chinese Journal of Computers, 2017, 40(8): 1829-1842.

[2] Tian, L. M., Zhang, D. M. An Anomaly Detection Method of Sensor Data Based on Information Entropy[J]. Software, 2018, 39(9): 69-73.

[3] Ahmad, S., Lavin, A., Purdy S, et al. Unsupervised real-time anomaly detection for streaming data[J]. Neurocomputing,2017,262:134–147.
[4] Zhang, T., Song, J. T. Detection method of abnormal data based on Markov model and convolutional neural network [J]. Internet of Things Technologies, 2018, 80-82. DOI : 10.16667/j.issn.2095-1302.2018.08.024
[5] Ahmed, I., Galoppo, T., Ding, Y. O-LoMST: An Online Anomaly Detection Approach And Its Application In A Hydropower Generation Plant[C]. 2019 IEEE 15th International Conference on Automation Science and Engineering(CASE),2019:762-767.
[6] Zhu, S. C, Qin, N. N. Outlier detection of uncertainly multiple data stream based on SBWS_GPR prediction model[J]. Application Research of Computers, 2018, 35(2): 381-385.
[7] Zhang, Q., Hu, Y. P., Ji, C., et al. Edge Computing Application: Real-Time Anomaly Detection Algorithm for Sensing Data [J]. Journal of Computer Research and Development, 2018, 55(3): 524-536. DOI:10.7544/issn1000-1239.2018.20170804
[8] Janjua, Z. H., Vecchio, M., Antonini, M., et al. IRESE: An intelligent rare-event detection system using unsupervised learning on the IoT edge[J]. Engineering Applications of Artificial Intelligence,2019,84:41–50.
[9] Wu, L. F., Luo, Y. L., Wang, X. H. High-dimensional Continuous Outlier Detection Method for Big Data Stream Simulation [J]. Computer Simulation, 2018, 35(10): 462-466.
[10] Huang, G. B., Zhu, Q. Y., Siew, C. K. Extreme learning machine: theory and applications, Neurocomputing, 70(1): 489–501, 2006.
[11] Liang, N. Y., Huang, G. B., Saratchandran, P., and N. Sundararajan. A fast and accurate online sequential learning algorithm for feedforward networks, IEEE Trans. Neural Networks, 17(6): 1411-1423, 2006.
[12] Luo T, Nagarajan, S. G. Distributed anomaly detection using autoencoder neural networks in WSN for IoT[C]. IEEE ICC, May 2018.
[13] Goldstein, Markus, 2015, "Unsupervised Anomaly Detection Benchmark", https://doi.org/10.7910/DVN/OPQMVF, Harvard Dataverse, V1, UNF :6:EnytiA6wCiiizHetzQQV7A== [fileUNF].