MILAD: Robust Anomaly Detection for Electric Vehicles with Label Noise

Yu Ye*, Bailin Feng¹ and Wujun Tao¹

¹ School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan Province, 611731, China
*Corresponding author’s e-mail: ye_yu@std.uestc.edu.cn

Abstract. One of the bottlenecks restricting the development of electric vehicle industry is the safety problem. Although numerous of anomaly detection algorithms for electric vehicles have been proposed, most of them may perform poorly due to the complexity and unpredictability of real scenes. We consider that there may be a certain degree of potential safety hazard in the battery system of electric vehicles before, during and after the process of faults in the real scenes, that is, label noise. In order to solve this problem, we propose a Multi-Instance Learning based Anomaly Detection (MILAD) framework, to perform anomaly detection for electric vehicles with label noise problem. Extensive cross validation experiments fully verify that the framework can effectively detect the existence of abnormal conditions in the presence of label noise in multivariate time series data.

1. Introduction

Electric vehicles have been developed rapidly for the past few years and have received numerous attentions. However, accidents caused by thermal runaway of lithium batteries in electric vehicles are common, which raises many discussions about the safety of electric vehicles. Scholars all over the world have proposed a large number of methods for the anomaly detection problem of electric vehicles. These methods can be roughly divided into two categories: model based approaches and data-driven approaches [11][13]. Model based approaches attempt to construct a model for the internal state of each power battery, through which we can accurately monitor the health status of each battery, and then realize abnormal detection of electric vehicles. The data-driven approaches do not explicitly construct a model for each battery. They use machine learning technology to explore patterns in battery data, and then provide information on the safety of electric vehicles.

Although previous algorithms have achieved certain success on their data set, a large part of these algorithms are based on ideal data constructed under laboratory conditions, and their reliability are verified by manually constructing abnormal conditions, like heating the battery or loosening battery component screws. Such algorithms have great limitations when transferred to real-world applications, because they do not take into account that electric vehicles will show a huge difference from the laboratory scene due to the complexity and unpredictability of the real scene, nor do they consider that the failure of the power battery should be a gradual accumulation rather than a sudden process.

We consider that the battery system itself has a certain degree of safety hazards right before the failure of the electric vehicle in the real scene, i.e. the failure of the power battery of the electric vehicle should be a gradual accumulation rather than a sudden process. In nature, this is a unidirectional label noise problem for multivariate time series data, which means that the normal data (negative samples)
within a period of time around the failure (positive samples) may have implicit anomalies, that is, noisy negative labels.

In this paper, we propose MILAD, a Multi-Instance Learning based Anomaly Detection framework to solve the label noise problem in precision equipment such as electric vehicles in real scenarios. Based on the idea of multi-instance learning, we let the data in a time serial bag share the same label, forcing the model to pay attention to the label noise of a segment of data before the failure, instead of simply treating it as completely normal data, and then learn a more compact decision bound between the normal mode and the failure mode from the data. Cross-validation experiments have fully confirmed that this framework can effectively detect the existence of abnormal conditions in multi-variable time series data with label noise.

2. Related Work
Multi-variable time series anomaly detection is an active topic in both academia and industry. Most of existing algorithms of anomaly detection are unsupervised or semi-supervised, owing to the lack of anomalies [4]. For these label-free algorithms technologically, they can be mainly divided into following categories: discord detection, dissimilarity-based and prediction model-based [2]. The first and most straightforward idea, i.e. discord detection is to compare each subsequence with the others [5]. Dissimilarity-based methods are based on the direct comparison of subsequences or their representations, using a reference of normality [3]. Prediction model-based methods intend to build a prediction model that captures the dynamics of the series using past data and thus detect abnormal temporal segments whose prediction error is higher than a selected threshold [7].

These label-free algorithms have achieved superior performances on various public datasets and real-world applications. However, considering that one of the most challenging problems in anomaly detection is how to define "anomaly" with little prior information about it, recently some work which is implemented in a weakly-supervised fashion and utilizes a small amount of known anomalies has been proposed. [8][10] show that by well leveraging few known anomalies can enable much more accurate anomaly detection. Similar to our work, some scholars have applied weakly-supervised MIL based anomaly detection technique on surveillance videos [12][14], and [14] devised a graph convolutional network to correct noisy labels in the MIL framework. However, these methods are specifically designed for videos, and cannot be simply transferred for electric vehicles.

3. Method

3.1. Problem Statement
Supposed that we are given a normal time series set \( \mathcal{N} = \{d_1, d_2, \cdots, d_n\} \) and an abnormal time series set \( \mathcal{A} = \{d_{n+1}, \cdots, d_{n+m}\} \). For a time series \( d = (X, y) \), we have \( X = \{x_1, \cdots, x_T\} \in \mathbb{R}^{T \times D} \) as its features and \( y \in \{0,1\} \) as its labels, where \( D \) is the feature dimension and \( T \) is its length. For a time frame \( i \), \( y_i = 0,1 \) indicates this time frame is normal or abnormal respectively. For convenience, we denote \( B_t = \{x_{t-w+1}, \cdots, x_{t-1}, x_t\} \) as a time window (or temporal segment) of fixed length \( w \) at time \( t \), and its label is \( y_t \). We now define a bag (or batch) with start time \( t \) is \( B = \{B_t, B_{t+1}, \cdots, B_{t+b-1}\} \), where \( b \) is the bag (or batch) size, the bag-level label is \( Y = \max(y_{t+1}, y_{t+2}, \cdots, y_{t+b-1}) \).

Our goal is to train an anomaly score ranking model \( f: \mathbb{R}^{w \times D} \rightarrow \{0,1\} \) by the given data set \( \mathcal{D} = \mathcal{A} \cup \mathcal{N} \). For a testing time window \( B_t \), we have \( f(B_t) \in \{0,1\} \) indicating how abnormal \( B_t \) is.

3.2. Proposed Framework
In response to this problem, we proposed a novel anomaly detection framework. Our main idea is derived from multi-instance learning. We let the data in a bag share the same label, forcing the model to pay attention to the label noise of a segment of data around the failure, instead of simply treating it as completely normal data. Then the model will learn the more compact decision bound between a normal mode and the failure mode from the data.
Figure 1. Overview of our proposed framework. The bag generation module first divides the original time series of electric vehicle data into temporal segments and packs continuous ones into a bag/batch. Then the anomaly scoring networks, composed of an encoder, a decoder and a regressor, capture the temporal dependence of features, yield discriminant representations and calculate the anomaly score for each temporal segment. Finally, all the anomaly scores of time series segments in the bag will be compared with the label of the bag to calculate the multi-instance loss.

Our propose framework is shown in Figure 1. It is composed of three parts: a bag generation module, an anomaly scoring module, and a multi-instance loss module. It works as follows: (1) The bag generation module first receives the original real-world data of the electric vehicle as input, and outputs a bag to downstream tasks through operations such as data processing, data sampling, and packing data into a bag; (2) The anomaly scoring module takes each time series segment in the bag from upstream as its input, and calculates an anomaly score for each of these segments through the recurrent neural network architecture; (3) Finally, all the anomaly scores of time series segments in the bag will be integrated and compared with the label of the bag to calculate the multi-instance loss. And the parameters of the neural network will be updated by gradient descent back propagation.

3.2.1 Bag Generation
To solve the problem of potential label noise around fault data, we adopt a multi-instance learning based method. The first step is to pack the original real-world data of electric vehicles into bags.

We consider that we need to input time series into the recurrent neural network, which requires a clip of time series for each input to extract the temporal representations from it. Therefore, we divide each time series into overlapping temporal segments of fixed length for the input shape of the recurrent neural network. Formally, consider a time series \( X = \{x_1, x_2, \ldots, x_T\} \) where each step \( x_t \in \mathbb{R}^D \) in the time series is a \( D \)-dimension vector. For each time step \( t \), data that does not exceed \( w \) steps before \( x_t \), that is \( B^t \), along with the its corresponding label \( y^t \) will be sent to the anomaly scoring network to model the temporal mapping mode, where \( w \) is the fixed window length. Considering the label noise problem, we treat each batch as a bag, in which all temporal segments share the same label, which is the maximum value of the labels of each time steps.

Finally, we considered the problem of class imbalance. We solved this problem with a simple down-sampling trick. In each epoch, we traverse all abnormal bags, randomly sample an equal number of normal bags from all normal bags and output them for downstream tasks.
3.2.2 Anomaly Scoring Network
The anomaly scoring network is fed with a temporal segment with \( w \) length and outputs a value \( y_t \in [0,1] \), named anomaly score, indicating the degree of the input segment being abnormal. The anomaly scoring network maps each temporal segment in the bag from upstream to an anomaly score. We converts the anomaly detection problem into a regression problem by this anomaly scoring module. We formulate the mapping function of anomaly scoring network as:

\[
s_t = f_\theta(B_t)
\]

(1)

where \( \theta \) is the parameters of anomaly scoring network, \( B_t \) is the input temporal segment or time window at time \( t \) from the bag generated from upstream, and \( s_t \) is the anomaly score.

To extract discriminant representations and achieve better regression performance simultaneously, we adopt the GRU-AE architecture, which is shown in Figure 2. The anomaly scoring network can further be divided into three parts: an encoder, a decoder and a regressor, all of which have GRU layer inside to model the temporal dependence. The encoder first encodes input batch/bag into a low-dimension space as embedded representations to compress the features in raw data. Then the decoder decodes the dense representations into initial feature spaces to reconstruct the last time frame in \( B_t \). Finally, the regressor maps the extracted temporal representations to a scalar for each time frame as the anomaly score.

We formulate these parts as following:

\[
Z_t = E(B_t), \tilde{x}_t = D(Z_t), s_t = R(Z_t)
\]

(2)

where \( E, D, R \) denote the encoder, decoder and regressor, respectively, \( Z_t \) is the dense representations extracted from \( B_t \) by the encoder, \( \tilde{x}_t \) is the reconstructed last time frame in \( B_t \), and \( s_t \) is the output scalar. The output scalar is then normalized to \( f_\theta(B_k) \in [0,1] \), which can be seen as the ranking score or anomaly score indicating the degree of being abnormal of the time frame.

3.2.3 Multi-Instance Loss
As mentioned above, we adopt multi-instance loss to make the learning model to focus on the overall maximum anomaly score. Consider a batch (or a bag) of temporal segments \( B = \{B_t, B_{t+1}, \ldots, B_{t+b-1}\} \) and the corresponding label sequence \( Y = \{y_t, \ldots, y_{t+b-1}\} \), then the bag-level label is \( \max(Y) \). To ensure that the model can learn both normal patterns and abnormal patterns in bags with different bag label, we adopt different forms of loss function w.r.t bag label as:

\[
\mathcal{L}(B) = \mathcal{L}(B| \max(Y) = 1) + \mathcal{L}(B| \max(Y) = 0)
\]

(3)

If the bag label is positive, our proposed multi-instance loss is

\[
\mathcal{L}(B| \max(Y) = 1) = \mathcal{L}_{AD}(B) + \lambda_1 \mathcal{L}_{MIL}(B)
\]

(4)

\[
\mathcal{L}_{AD}(B) = \frac{1}{b} \sum_{k=t}^{t+b-1} y_k(s_k - y_k)^2
\]

(5)

\[
\mathcal{L}_{MIL}(B) = \max(0, C - \max_{0 \leq k \leq \max(s_k) | y_k = 0, k = t, \ldots, t + b - 1})
\]

(6)

where \( \mathcal{L}_{AD} \) denotes Anomaly Discrimination Loss, and \( \mathcal{L}_{MIL} \) denotes Multi-Instance Learning Loss. As we expect that the model should have superior discrimination on anomalies, we directly optimize the square error for all labeled anomalies, which is formulated as \( \mathcal{L}_{AD}(B) \). As for normal samples in abnormal bags, we consider that it is dangerous to minimize the output anomaly scores, because there is underlying label noise in normal samples around anomalies, i.e. in abnormal bags. Inspired by the success of MIL in surveillance videos \([12][14]\), we optimize their anomaly score by the MIL loss.

When it turns to negative bag, i.e. \( Y = 0^t \), the loss function is

\[
\mathcal{L}(B| \max(Y) = 0) = \mathcal{L}_{ND}(B) + \lambda_2 \mathcal{L}_{REC}(B)
\]

(7)

\[
\mathcal{L}_{ND}(B) = \frac{1}{b} \sum_{k=t}^{t+b-1} s_k^2
\]

(8)

\[
\mathcal{L}_{REC}(B) = \sum_{k=t}^{t+b-1} y_k^2
\]
\[
L_{REC}(B) = \frac{1}{b} \sum_{k=t}^{t+b-1} \| \tilde{x}_k - x_k \|_2
\]  

where \( L_{ND} \) denotes Normality Discrimination Loss, \( L_{REC} \) denotes Reconstruction Loss, and \( \| \cdot \|_2 \) refers to the L2-norm. For a negative bag totally composed of normal instances, the prediction anomaly scores for each instance should be as low as possible. We achieve this by minimizing the square error formulated as \( L_{ND}(B) \). We also need to minimize the reconstruction loss of negative bags formulated as \( L_{REC}(B) \). Denotes that we do not optimize this objective in positive bags because we expect that the AE can be trained on completely normal instances and learn the normal patterns implied in data, so that the abnormal instances and normal instances show greater differences w.r.t. both reconstruction error and anomaly score.

4. Experiments

4.1. Dataset Details

To evaluate the performance of our proposed framework, we conduct extensive experiments on Huawei dataset. This dataset is composed of the real runtime data of 7 electric vehicles. All of these vehicles have experienced overheating fault, which is the target anomaly to be detected. The anomaly ratio of total dataset is 2.99%, a low-level to perform weakly-supervised learning.

4.2. Competing Methods

This section compares MILAD with state-of-art anomaly detection algorithms. First, we choose iForest [6] as the baseline, as it is simple and effective. However, iForest is unsupervised and stationary, and deep learning methods can acquire diverse and discriminate representations to achieve better performance [9]. For this reason, we also studied the results of DAGMM[15], USAD[1], DevNet[10], all of which are based on deep architectures. DAGMM joints Deep Autoencoder (AE) and Gaussian Mixture Model (GMM) to estimate the density of extracted latent representations, but it is not designed for multi-variable time series. USAD is based on AEs and trained within an adversarial training inspired by the Generative Adversarial Networks (GAN), which makes it effective to detect anomalies in multi-variable time series in unsupervised fashion. While DevNet is designed to leverage a few labelled anomalies with a prior to fulfil an end-to-end differentiable learning of anomaly scores.

4.3. Results of Anomaly Detection

| Dataset | AUC-PR | AUC-ROC |
|---------|--------|---------|
|         | iForest | DAGMM | USAD | DevNet | MILAD | iForest | DAGMM | USAD | DevNet | MILAD |
| 1       | 0.171   | 0.026  | 0.042 | 0.161   | 0.547  | 0.933   | 0.525  | 0.661 | 0.87    | 0.969  |
| 2       | 0.398   | 0.5    | 0.816 | 0.465   | 0.375  | 0.365   | 0.5    | 0.797 | 0.286   | 0.04   |
| 3       | 0.087   | 0.017  | 0.014 | 0.554   | 0.827  | 0.914   | 0.5    | 0.423 | 0.98    | 0.997  |
| 4       | 0.286   | 0.08   | 0.124 | 0.182   | 0.203  | 0.849   | 0.504  | 0.699 | 0.74    | 0.559  |
| 5       | 0.075   | 0.037  | 0.127 | 0.113   | 0.178  | 0.771   | 0.501  | 0.723 | 0.772   | 0.785  |
| 6       | 0.106   | 0.038  | 0.264 | 0.278   | 0.607  | 0.85    | 0.582  | 0.797 | 0.881   | 0.984  |
| 7       | 0.234   | 0.03   | 0.134 | 0.373   | 0.844  | 0.947   | 0.684  | 0.917 | 0.949   | 0.993  |
| Average | 0.153   | 0.035  | 0.147 | 0.259   | 0.566  | 0.876   | 0.577  | 0.772 | 0.873   | 0.924  |

Table 1 demonstrates the AUC-PR and AUC-ROC of MILAD and four competing methods on Huawei dataset and the average performance. These results are due to the following reasons:

- MILAD captures the temporal dependence in electric vehicles. MILAD, USAD and DevNet adopt GRU layers to model the temporal dependence of multi-variable time series and extract discriminant representations for downstream tasks.
• MILAD efficiently leverages the limited labeled anomalies. iForest, DAGMM and USAD are all unsupervised and lack of prior information about anomalies. Although we remove abnormal data in the training phase to enhance their discriminant ability, they still lose to MILAD and DevNet.

• MILAD is aware of the label noise problem. Inspired by the success of MIL in surveillance videos, MILAD adopts MIL loss to model the label noise in normal segments around anomalies. By optimizing the MIL loss, the model gets better discrimination and finally defeat DevNet.

5. Conclusions
This paper introduces MILAD, a Multi-Instance Learning based Anomaly Detection framework for electric vehicles. We take the lead in considering the problem of label noise in real scenarios, and we propose a MIL loss to convert the label noise problem to a weakly-supervised learning problem. The GRU layer enables our model to capture the time dependence and extract discriminant representations, and the MIL loss leads our model to focus on the label noise on normal data. We compared our proposed MILAD with other four state-of-art anomaly detection methods in the Huawei dataset, and the experiment results demonstrated that MILAD got superior performance over other methods.

References
[1] Audibert, J.; Michiardi, P.; Guyard, F.; Marti, S.; and Zuluaga, M. A. 2020. Usad: Unsupervised anomaly detection on multivariate time series. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 3395–3404.
[2] Blázquez-García, A.; Conde, A.; Mori, U.; and Lozano, J. A. 2020. A review on outlier/anomaly detection in time series data. arXiv preprint arXiv:2002.04236.
[3] Carreño, A.; Inza, I.; and Lozano, J. A. 2019. Analysing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. Artificial Intelligence Review 1–20.
[4] Chandola, V.; Banerjee, A.; and Kumar, V. 2009. Anomaly detection: A survey. ACM computing surveys (CSUR) 41(3):1–58.
[5] Chau, P. M.; Duc, B. M.; and Anh, D. T. 2018. Discord discovery in streaming time series based on an improved hot sax algorithm. In Proceedings of the Ninth International Symposium on Information and Communication Technology, 24–30.
[6] Liu, F. T.; Ting, K. M.; and Zhou, Z.-H. 2008. Isolation forest. In 2008 eighth ieee international conference on data mining, 413–422. IEEE.
[7] Munir, M.; Siddiqui, S. A.; Dengel, A.; and Ahmed, S. 2018. Deepant: A deep learning approach for unsupervised anomaly detection in time series. IEEE Access 7:1991–2005.
[8] Pang, G.; Shen, C.; Jin, H.; and Hengel, A. V. d. 2019. Deep weakly-supervised anomaly detection. arXiv preprint arXiv:1910.13601.
[9] Pang, G.; Shen, C.; Cao, L.; and Hengel, A. V. D. 2021. Deep learning for anomaly detection: A review. ACM Computing Surveys (CSUR) 54(2):1–38.
[10] Pang, G.; Shen, C.; and van den Hengel, A. 2019. Deep anomaly detection with deviation networks. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 353–362.
[11] Rezvanian, S. M.; Liu, Z.; Chen, Y.; and Lee, J. 2014. Review and recent advances in battery health monitoring and prognostics technologies for electric vehicle (ev) safety and mobility. Journal of power sources 256:110–124.
[12] Sultani, W.; Chen, C.; and Shah, M. 2018. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE conference on computer vision and pattern recognition, 6479–6488.
[13] Xu, X.; Wang, H.; Zhang, N.; Liu, Z.; and Wang, X. 2017. Review of the fault mechanism and diagnostic techniques for the range extender hybrid electric vehicle. IEEE Access 5:14234–14244.
[14] Zhong, J.-X.; Li, N.; Kong, W.; Liu, S.; Li, T. H.; and Li, G. 2019. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1237–1246.

[15] Zong, B.; Song, Q.; Min, M. R.; Cheng, W.; Lumezanu, C.; Cho, D.; and Chen, H. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International Conference on Learning Representations.