An energy-efficient and trustworthy unsupervised anomaly detection framework (EATU) for IIoT

ZIJIE HUANG, Department of Computer Science, Faculty of ESE, University of Exeter, United Kingdom
YULEI WU*, Department of Computer Science, Faculty of ESE, University of Exeter, EX4 4QF, United Kingdom
NICCOLÒ TEMPINI, Department of Sociology, Philosophy and Anthropology, Faculty of HASS, University of Exeter, EX4 4RJ, United Kingdom
HUI LIN, College of Computer and Cyber Security, Fujian Normal University, 350117, China
HAO YIN, Research Institute of Information Technology, Tsinghua University, 100084, China

Many anomaly detection techniques have been adopted by Industrial Internet of Things (IIoT) for improving self-diagnosing efficiency and infrastructures security. However, they are usually associated with the issues of computational-hungry and "black box". Thus, it becomes important to ensure that the detection is not only accurate but also energy-efficient and trustworthy. In this paper, we propose an Energy-efficient And Trustworthy Unsupervised anomaly detection framework (EATU) for IIoT. The framework consists of two levels of feature extraction: 1) Autoencoder-based feature extraction and 2) Efficient DeepExplainer-based explainable feature selection. We propose an Efficient DeepExplainer model based on perturbation-focused sampling which demonstrates the most computational efficiency, amongst state-of-the-art explainable models. With the important features selected by Efficient DeepExplainer, the rationale of why an anomaly detection decision was made is given, enhancing the trustworthiness of the detection as well as improving the accuracy of anomaly detection. Three real-world IIoT datasets with high-dimensional features are used to validate the effectiveness of the proposed framework. Extensive experimental results demonstrate that in comparison with the state-of-the-art, our framework has the attributes of improved accuracy, trustworthiness (in terms of correctness and stability of the explanation) and energy-efficiency (in terms of wall-clock-time and resource usage).

CCS Concepts: • **Security and privacy → Intrusion detection systems**; • **Computing methodologies → Machine learning**.

Additional Key Words and Phrases: Energy-efficiency, Variational autoencoder, Explainable AI, Industrial Internet of Things, Feature extraction, Anomaly detection

*Corresponding author.

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) Project (Grant No. EP/R030863/1), the National Key Research and Development Program of China (Grant No. 2018YFB2100804), and the National Natural Science Foundation of China (Grant No. 92067206 and 61972222).
Authors' addresses: Zijie Huang, Department of Computer Science, Faculty of ESE, University of Exeter, Harrison Building, Streatham Campus, N Park Rd, Exeter, United Kingdom, zh314@exerer.ac.uk; Yulei Wu, Department of Computer Science, Faculty of ESE, University of Exeter, EX4 4QF, Harrison Building, Streatham Campus, N Park Rd, Exeter, United Kingdom, y.l.wu@exeter.ac.uk; Niccolò Tempini, Department of Sociology, Philosophy and Anthropology, Faculty of HASS, University of Exeter, EX4 4RJ, Amory Building, Rennes Drive, Exeter, United Kingdom, N.Tempini@exeter.ac.uk; Hui Lin, College of Computer and Cyber Security, Fujian Normal University, 350117, 8 Shangsan Road, Fuzhou, China, linhui@fjnu.edu.cn; Hao Yin, Research Institute of Information Technology, Tsinghua University, 100084, Haidian District, Beijing, China, h-yin@mail.tsinghua.edu.cn.

ACM Reference Format:
Zijie Huang, Yulei Wu, Niccolò Tempini, Hui Lin, and Hao Yin. 2022. An energy-efficient and trustworthy unsupervised anomaly detection framework (EATU) for IIoT. 1, 1 (October 2022), 18 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Along with the digital transformation of traditional manufacturing and industries towards Industry 4.0, the Industrial Internet of Things (IIoT) has been widely used to connect huge numbers of sensors, instruments and other Internet of Things (IoT) devices with the Internet in order to achieve increased automation and productivity [5][35][30] [32]. Connected devices, and the communication enabled between them, have been widely used for system monitoring, and data collection, exchanging, and analysis for IIoT [33] [8]. Although Industry 4.0 brings increasing automation, self-monitoring and diagnosing abilities than traditional manufacturing and industrial practices, a small number of anomalies during any stages of Industry 4.0 can make a huge impact on the IIoT infrastructure [24]. For example, in 2017, hackers turned on 156 severe weather sirens in Dallas US in the middle of the night, causing a surge of 911 calls and distress [18]. Thus, addressing security problems of IIoT infrastructure is always essential. Anomaly detection is one of the promising techniques in the process of addressing the security issues in IIoT [36] [39] [31].

IIoT data are diverse and dynamic due to the factors of a large volume of transactions within applications, a high velocity of sensor-generated data, and a wide variety of data formats [29]. Most anomaly detection techniques have mainly focused on generating an optimal machine learning model to achieve high accuracy in IIoT scenarios [37]. The Sustainable Development Goals (SDGs) set up in 2015 by the United Nations General Assembly show that we need to "ensure good use of resources, improving energy efficiency, sustainable infrastructure, and providing access to basic services, green and decent jobs and ensuring a better quality of life for all." [22]. However, anomaly detection processes usually exhibit considerable energy-consumption as well as increase in the carbon footprint, which go against the sustainable development goal. Therefore, to achieve a sustainable future in IIoT, more efforts towards energy efficiency should be considered [15]. Furthermore, it is significant to consider the trade-off between accuracy and energy consumption [38].

On the other hand, not only researchers but also engineers and laypeople are diving deeper into the question of 'how the prediction was made from machine learning models', instead of only being satisfied of knowing 'what is predicted'. To this end, several interpretable machine learning models have been proposed to unveil the so-called "black box" issue. For example, model-agnostic interpretation methods such as LIME [16] and Anchors [17] have been presented. These explainers have made huge contributions for interpreting the output of black box machine learning models, and most existing explainers work well for both supervised and unsupervised machine learning models. However, there is little work on applying the explainer on unsupervised anomaly detection for IIoT in order to enhance trustworthiness, as well as reduce computational complexity for the explainer itself. Thus, an anomaly detection model with the characteristics of high accuracy, energy-efficiency and explanability is very interesting and promising.

To address the above challenges, we propose an Energy-efficient And Trustworthy Unsupervised (EATU) anomaly detection framework for IIoT. The framework consists of an energy efficient two-level feature extraction process: (i) feature extraction by autoencoders and (ii) feature selection by the feature importance coming from Efficient DeepExplainer. The main contributions of our work are summarized as follows:

- We devise an autoencoder-based approach for the first level feature extraction in our proposed EATU framework, which not only can extract unseen features, but also reduce the dimensionality of the IIoT data in order to initially reduce energy cost.

- We present an Efficient DeepExplainer-based approach for the second level feature selection in the EATU framework, which provides explanation of the extracted features from the first level, as well as reduces energy consumption.
- Extensive experimental results are carried out to evaluate the proposed framework on three real-world IIoT datasets. Results demonstrate that, with the Autoencoder feature extraction and the feature selection from the Efficient DeepExplainer, the accuracy of anomaly detection can be improved by about 10%, the energy-efficiency can be improved by 4-fold. Moreover, the framework presents trustworthiness in terms of the correctness and stability of explanation.

The remainder of this paper is organized as follows. Section 2 briefly introduces preliminaries of Variational Autoencoder, SHapley Additive exPlanation and Bayesian Local Explanations. Section 3 presents related work. In Section 4, we elaborate our Energy-efficient and Trustworthy Unsupervised anomaly detection framework for IIoT. Experimental results and analysis are conducted in Section 5. Finally, Section 6 concludes this paper.

2 PRELIMINARIES

This section illustrates the general knowledge about the techniques we adopt in the proposed framework: Variational Autoencoder (VAE), SHapley Additive exPlanation (SHAP) and Bayesian Local Explanations.

2.1 Variational Autoencoder

VAE [6] is a generative model whose training is regularized and the latent space enables the generative process. Moreover, VAE is a stochastic Variational inference and learning algorithm that works efficiently in the scenario of neural network intractability and large scale datasets. The VAE architecture consists of an encoder neural network and a decoder neural network. The encoder input is data x, and output is a hidden representation z. In the encoding process, the encoder network encodes data x into the latent (hidden) representation space z. Encoder can be denoted as $q_\phi(\mathbf{z}|\mathbf{x})$. Since the encoder has to compress data into a stochastic lower-dimensional space, the values of representations z can be sampled from the data distribution. The decoder's input is the representation z, and it outputs the parameters to the probability distribution of the data. The decoder can be denoted by $p_\theta(\mathbf{x}|\mathbf{z})$. As the decoder only has access to a summary of the data information, the data decoded in the decoder cannot be perfectly transmitted. Hence, a measurement is used to measure how much the information is lost. The reconstruction log-likelihood $logp_\phi(x|z)$ can indicate how effectively the decoder has learned to reconstruct the input data x given its latent representation z.

2.2 Shapley Additive Explanation

SHAP [10] is a unified framework to interpret the predictions of machine learning models. The class of additive feature attribution methods has been defined in SHAP, which is a linear function of binary variables. The rationale of SHAP is that it computes the contribution of each feature to the prediction (model output) to explain which feature is important to the prediction of an instance. The contribution denotes to Shapley values which are calculated based on game theory. Each feature value of the instance is a "player" in the game where the prediction is payout. Shapely values show how to fairly distribute the "payout" among the players (features). SHAP specifies the explanation as shown in Eq. (1), where $f(x)$ is the model to be explained, and $g(z')$ denotes the explanation model. The effect ϕ_i is attributed to each input feature from $f(x)$, and then the effects of all feature attributions are added up to approximate the $f(x)$, which can be expressed by

$$f(x) = g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{1}$$

2.3 Bayesian Local Explanations

Bayesian Local Explanations [26] is a framework designed to capture the uncertainty associated with local explanations of black box models. In the framework, a Bayesian Local Explanations model is firstly built for constructing local linear model based explanations, and capturing associated uncertainty. As Eq. (2) shows, each perturbation z is used to model the explanation-needed black box's prediction as a linear combination of the corresponding feature values ($\phi^T z$) and an error term ϵ. ϕ is the weights of the linear combination which captures the feature importance, as well as constructing explanations. The error term ϵ denotes the distance between the Bayesian explanation ϕ and the explanation-needed model f (definition - refer to Section 2.2), which is modeled as a Gaussian distribution and its variance relies on the proximity function $\pi_x(z)$. This Bayesian Local Explanations framework can be instantiated to the Bayesian version of LIME and SHAP, i.e., BayesLIME and BayesSHAP, with alterations on proximity functions. This framework proposed perturbations-to-go, and Focused Sampling approaches, to address one of the major drawbacks in LIME and KernelSHAP having high computational complexity with the random choice of the number of perturbations. Perturbations-to-go estimates the number of perturbations that are required to obtain a desired level of certainty for explanations. Focused Sampling leverages uncertainty estimates to query the black box with the most informative perturbations and thereby gains an accurate explanation with less queries. The variance of posterior predictive distribution for any instance z is shown as Eq. (3), which captures how uncertain the explanation ϕ is about the explanation-needed model prediction.

$$y|z, \phi, \epsilon \sim \phi^T z + \epsilon \quad \epsilon \sim \mathcal{N}(0, \frac{\sigma^2}{\pi_x(z)}) \tag{2}$$

$$var(\hat{y}(z)) = ((z^T V_\phi z + 1)s^2)(N/(N-2)) \tag{3}$$

3 RELATED WORK

In this section, we present an overview of recent work on energy-efficient anomaly detection techniques in intelligent IoT and explainable anomaly detection techniques.

3.1 Energy-efficient anomaly detection techniques in intelligent IoT

In [28], Wang, Sun and Xu proposed a scalable and energy-efficient anomaly detection scheme (SEE-ADS) for massive machine-type communications (mMTC) applications in IoT, capable of detecting attacks dynamically and effectively, without the discontinuous activation of energy-inefficient heavy-weight detection. In their work, an activation scheme was proposed to define the position of detection with different types of attacks in order to balance the detection effectiveness and energy consumption. Moreover, the structure constructed in their work supports the existing noncluster-based heavyweight detection, which not only improves the network flexibility, but reduces the network complexity.

Another energy-efficient anomaly detection IoT research proposed by Lydia, et al. [11], presented a new green energy-efficient routing with Deep Learning based anomaly detection (GEER-DLAD) technique for IoT applications. The GEER-DLAD applied the error lossy compression (ELC) technique to lessen the quantity of data communication over the IoT network. In addition, in order to have a more energy efficient performance, the moth flame swarm optimization (MSO) algorithm was applied for selecting optimal routes in the network.

Recently, Sater and Hamza [20] proposed a novel privacy-by-design federated learning model using a stacked long short-time memory (LSTM) model, which demonstrates that it is more than twice as fast compared to the centralized LSTM. Their experiments were carried out on the General Electric Current smart building IoT production system. The experimental results presented the effectiveness of their framework in reducing the overall training cost, without compromising the prediction performance.

3.2 Explainable Anomaly Detection

Many existing works have used interpretable methods to explain supervised machine learning models, but there is little work for the explanation of unsupervised machine learning models for anomaly detection in IIoT.

In [14], Nguyen et al. proposed a Gradient-based Explainable VAEs, named GEE, to detect and explain anomalies in network traffic. In this framework, they firstly use VAEs to detect anomalies in network traffic, then develop a gradient-based fingerprinting technique to provide explanations for the anomalies that are detected. This framework is shown to be robust in detecting network anomalies compared to Gaussian based methods, and has a good performance on clustering anomalies with similar behaviors even without ground truth labels. However, as GEE uses L_n^2 distance to calculate the average of normalized gradients fingerprint obtained from data, some attack types such as botnet attacks have many unseen patterns that cannot be reveled by calculating gradients. In turn, GEE would be hard to explain in such cases.

Another method proposed by Antwarg, Shapira and Rokach [3] was using kernel SHAP to explain anomalies detected by Autoencoder (AE). They use AE to detect anomalies and extract features with highest reconstruction error, then explain the reconstruction error (anomalies) by calculating SHAP values. As SHAP has properties of polarity, anomalies can be explained in terms of contributing and offsetting. Moreover, with the SHAP library, the visual explanation can help interpret anomalies. While during the AE training process, they need to manually set a threshold for choosing the top anomalies. In the real-world IIoT scenarios, it is impossible to find the threshold value precisely.

More recently, VAEs-LIME, a novel approach for local data-driven model interpretability, applied to the ironmaking industry, was proposed [21]. In this work, a VAE is used to learn data characteristics and generate optimal samples. LIME is used as a local interpretable model to better represent the VAE black-box model. This approach shows an improved fidelity of the local explainer and a robust model interpretability. Nevertheless, LIME may face the problem of instability of the explanation, which can cause dangerous accidents in ironmaking industry.

Table 1 provides a general overview of the above related work. We systematically categorize them in terms of their focused topics: energy-efficient anomaly detection and explainable anomaly detection. Furthermore, we include our proposed work in the table highlighted, which presents that our framework considers both topics.

4 THE ENERGY-EFFICIENT AND TRUSTWORTHY UNSUPERVISED ANOMALY DETECTION FRAMEWORK (EATU)

In this section, we elaborate our proposed framework which aims to provide energy-efficient, accurate and trustworthy unsupervised anomaly detection approach for IIoT. To achieve this goal, we present an energy efficient trustworthy two-level feature extraction, the corresponding workflow is shown in Fig. 1. The first level feature extraction is via autoencoders to generate compressed features while reducing anomaly detection model's computation complexity. The second level leverages Efficient DeepExplainer's feature importance which is based on game theory to select relative features with higher importance. Meanwhile, Efficient DeepExplainer maintains the best computational performance

Table 1. A summary of existing work

Focus	Method	Reference
Energy-efficient anomaly detection techniques in intelligent IoT	A Scalable and Energy-Efficient Anomaly Detection Scheme in Wireless SDN-Based mMTC Networks for IoT	[28]
Energy-efficient anomaly detection techniques in intelligent IoT	Green Energy Efficient Routing with Deep Learning Based Anomaly Detection for Internet of Things (IoT) Communications	[11]
Energy-efficient anomaly detection techniques in intelligent IoT	A Federated Learning Approach to Anomaly Detection in Smart Buildings	[20]
Explainable anomaly detection	A Gradient-based Explainable VAEs (GEE)	[14]
Explainable anomaly detection	Explaining anomalies detected by autoencoders using SHAP	[3]
Explainable anomaly detection	VAE-LIME: A deep generative model based approach for local data-Driven model interpretability applied to the ironmaking industry	[21]
Energy-efficient and explainable anomaly detection	An energy-efficient trustworthy automatic unsupervised anomaly detection framework (ETAU) for IIoT	Our framework

with Focused Sampling of perturbations instead of perturbing a number of points randomly. More details will be demonstrated in the following sections.

Problem formulation. Given a dataset $D = (F_n, I_m)$, where F_n represents n features and I_m regards to m instances, our approach aims to find the best trade-off between energy-consumption reduction and anomaly detection accuracy improvement as well as making anomaly detection trustworthy.

4.1 Autoencoder-based Feature Extraction

Since IIoT data are normally high-dimensional, in order to reduce the computational consumption as well as extract the hidden relationships among original data, we present the first level of the proposed approach with autoencoder-based feature extraction. The structure of the Autoencoder is shown in Fig. 2: the Encoder net picks the crucial features from the input data, and the Decoder net attempts to recreate the original data using the critical components. The bottleneck only retains the characteristics that would be needed to reconstruct the data in the Decode net. In our framework, the Autoencoder structure only builds with Encoder net and bottleneck as shown in red dash rectangle in Fig 2. There is no need to build the Decoder net for feature extraction because the extracted features we needed would come from the bottleneck. We can see from Fig. 2, that bottleneck has less nodes than the input layer (Encoder), by which the flow of the original data information, through the network, is limited. Thus, we can get latent representation of original input data which enables to capture the most notable features of the original input data. Compared with traditional feature extraction methods PCA [7], Autoencoder can learn nonlinear relationships in data, and is more powerful to deal with high-dimensional data because of the neural networks structure. The mathematical rationale is shown as following:

The Encoder net is represented by the standard neural network function passed through an activation function as shown by Eq. 4, where z is the latent dimension (the number of features we manage to decrease), W is a weight matrix, and b is a bias vector. Given a hidden layer, the encoder takes the input x and maps it to z. Weights and biases are usually initialized randomly, and then updated iteratively during training through backpropagation, the bottleneck would generate the most representative latent space with the most informative features of the original data.

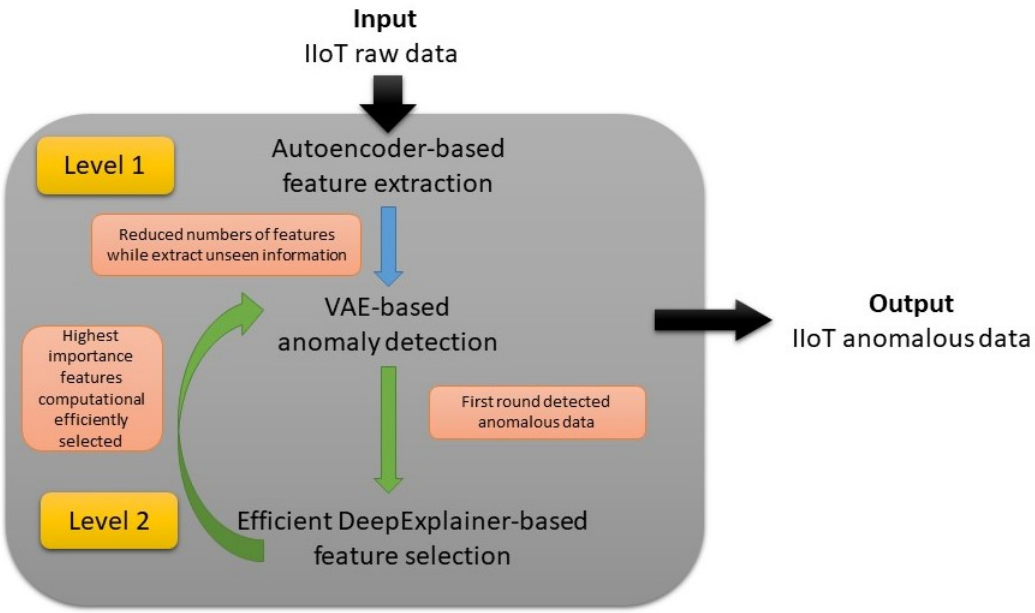

Fig. 1. The Energy-efficient And Trustworthy Unsupervised anomaly detection Framework (EATU)

$$
\begin{aligned}
z &= \sigma(Wx + b) \\
x &\in \mathbb{R}^d = \mathcal{X} \\
z &\in \mathbb{R}^p = \mathcal{F}
\end{aligned} \tag{4}
$$

4.2 Efficient DeepExplainer-based Feature Selection

The second level of our approach is Efficient DeepExplainer-based feature selection, according to feature importance. Although some deep learning anomaly detection techniques, which have been mentioned in Section 3, have low energy consumption as well as high prediction accuracy, lack of interpretability is one of the biggest challenges in adopting unsupervised learning models (as have been mentioned in Section 1). Many state-of-the-art explainable models can be used to gain interpretability and intuition from feature importance, such as LIME [16] and SHAP [10]. However, these methods are highly computationally complex and require significant computing memory. For instance, the core of LIME and SHAP constructs local approximation of original models by randomly perturbing a large number of points, which usually cause prohibitively computational inefficiency. Therefore, to solve the issue of the trade-off between model energy consumption and accuracy, as well as the issue of model's trustworthiness, we present an Efficient DeepExplainer-based feature selection in the second level, which is inspired by Bayesian Local Explanations Framework [26]. We instantiate Bayesian Local Explanations Framework to build an Efficient DeepExplainer, which outputs stable and reliable explanations with computational efficiency.

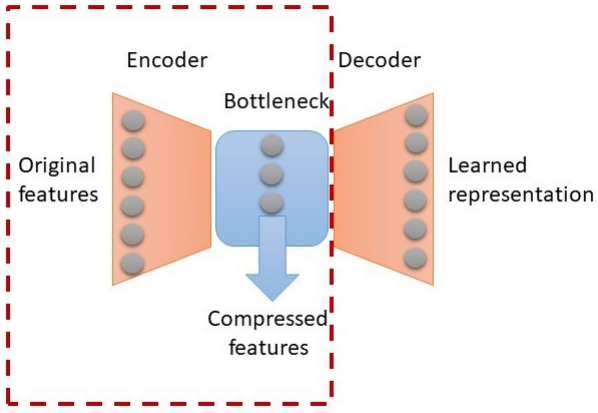

Fig. 2. Autoencoder for feature extraction in Level 1

The efficiency of the Efficient DeepExplainer is demonstrated as follows. Firstly, DeepExplainer combines SHAP values and DeepLIFT [25], which is a high-speed approximation algorithm for SHAP values in deep learning models, and performs an improved computational performance versus Kernel SHAP, because of the recursively passing of DeepLIFT's multipliers backwards through the network. However, the core of DeepExplainer is the same as KernelSHAP which is randomly choosing estimated points to predict local probability and then approximates the explanation-needed model via using game theoretic principles. Thus, DeepExplainer might be prohibitively slow in some complex deep learning models. Secondly, in order to address the drawback of DeepExplainer mentioned above, we refer to the Focus Sampling procedure from Bayesian Local Explanations Framework. A pseudocode is provided for Efficient DeepExplainer in Algorithm 1. Current perturbations *rdata* are firstly calculated by the Focus Sampling function with customized parameters: sample number N, *batch_size*, *initial_points* and anomalous data *data* detected from VAE network *vae*. Since the Focused Sampling approach uses the predictive variance (Equation 3) to strategically choose perturbations that have the lowest uncertainty to be labeled by the explanation-needed model, the current perturbations *rdata* calculated in this step will have the highest disturbance to converge to the explanation-needed model, which avoids the inefficiency of disturbance explanation-needed model in less-related perturbations. Then current perturbations *rdata* and the shape of current perturbation data M are output to calculate the weights with Combinations Calculator formulation (Equation 5). Lastly, labels y from the Focus Sampling function and *weights* are fed into the VAE model to gain coefficients for the anomaly predictions of each feature. The value of coefficients for each feature is feature importance. Higher coefficient values represent a more important feature.

$$C(n, r) = \frac{n!}{r!(n-r)!} \tag{5}$$

The trustworthy explanation of an anomaly detection model can thus be obtained from this level. Meanwhile, we propose to leverage the feature ranking coming from Efficient DeepExplainer to re-select important features. We devise a process which iterates features, from 2 to the number of extracted features, to do anomaly detection. In every iteration,

Algorithm 1 Efficient DeepExplainer feature selection

Input: sample number N; anomalous data $data$; classifier vae; focus sample batch size $batch\_size$; focus sample initial points $initial\_points$

Output: feature importance $importance$

1: Current perturbations $rdata$ and label $y \leftarrow$ Call function Focus Sampling with Input $data$, vae, N, $batch\_size$ and $initial\_points$. ▷ Using Equation (3)

2: DeepExplainer's perturbations weights $weights \leftarrow$ Output current perturbations $rdata$ and shape of perturbations M to Combinations Calculator formulation. ▷ Using Equation (5)

3: Feature importance $importance \leftarrow$ VAE model vae training with current perturbation $rdata$, label y and DeepExplainer's weights $weights$.

we calculate the weighted F1-score (see Eq. (11)), and the anomaly detection model in the iteration with the highest weighted F1-score is seen as the trustworthy anomaly detection.

4.3 VAE-based unsupervised Anomaly Detection

To address the diversity and dynamism characteristics of IIoT data, during the procedure of anomaly detection, we adopt deep generative VAE to detect anomalies through the reconstruction error. Generative models can learn normal patterns of data, i.e., they can express the relationships between variables and be easily applied to complicated data. Thus, the usage of reconstruction log-likelihood $logp_\phi(x|z)$ in VAE architecture can indicate how effectively the decoder has learned to reconstruct the input data x given its latent representation z. Moreover, we build the loss function in our anomaly detection procedure, which consists of a Kullback–Leibler (KL) divergence term and an expectation term, as shown in Eq. (6). The first term in the equation represents the reconstruction error of the hidden variable z to the sample x, and takes the expectation in the encoder $q_\theta(z|x)$ space. The KL divergence is to measure how much information is lost if the prior distribution $p(z)$ is used to represent the encoder $q_\theta(z|x)$.

$$\mathbf{L}(\theta, \phi) = -\mathbf{E}_{Z\sim q_\phi(z|x_i)}[logp_\phi(x_i|z)] + \mathbf{D_{KL}}[q_\theta(z|x_i)||p(z)] \tag{6}$$

where θ is the encoder's biases, and ϕ represents the decoder's biases.

Fig. 3 presents the process of the proposed unsupervised anomaly detection model. Data are recognized as abnormal when the reconstruction error of its input features is high. We use Mean Square Error (MSE) to calculate the distance between the observed features, and the expectation of the reconstructed ones. This is because MSE can provide a quadratic loss which can measure the uncertainty in the prediction model. The process of anomaly detection is described as followed: We first take an instance $\overline{X}$ with a set of features $\overline{x}_1, \overline{x}_2, ..., \overline{x}_n$ as the input to the unsupervised VAE anomaly detection model, and then we get the corresponding output $\overline{X}'$ and the reconstructed values $\overline{x}'_1, \overline{x}'_2, ..., \overline{x}'_n$. Next, the reconstruction error of the instance will be calculated by MSE with the sum of errors of each feature, denoted as: $\sum_{i=1}^{n}(\overline{x}_i - \overline{x}'_i)^2$.

Algorithm 2 summarizes the overall training procedure of our efficient two-level feature extraction approach. The inputs include an IIoT dataset $D = (F_n, I_m)$, an Autoencoder model AE, an VAE model vae. In Line 1, features extracted by Autoencoder model AE will be generated $F_{AE} = \{F_1, F_2, ..., Fn\}$. Next, a new dataset $D = (F_{AE}, I_m)$ is used to train the model vae. Then, the Efficient DeepExplainer explains the vae model, weights of features will be obtained in this stage as $|W| = \{|W_1|, |W_2|, ..., |W_{AE}|\}$. Next, features are sorted in descending order according to their $weights$ calculated from the last step, which are presented as $E_{AE} = \{E_1, E_2, ..., E_n\}$. From Line 4 to Line 7, this **for** loop processes the

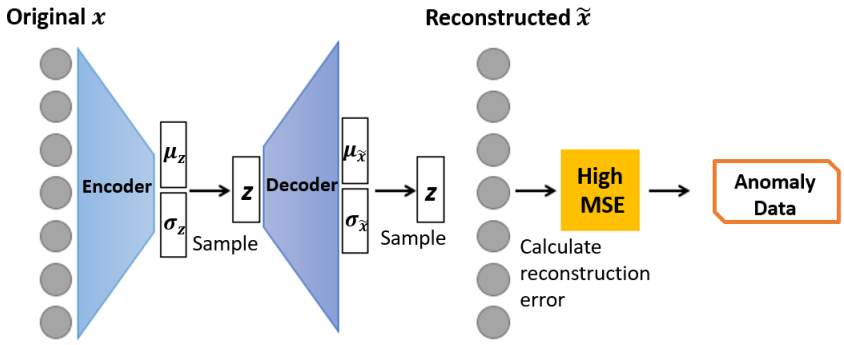

Fig. 3. The Unsupervised Anomaly Detection Model

number of features from 2 to n in order to gain the best number of important features from DeepExpaliner. After the **for** loop ends, the number of features $num\_F_{AE} = \{num\_F_1, num\_F_2, ..., num\_F_{AE}\}$ that can provide the highest weighted F1-score is generated. Finally, the dataset $D = (num\_F_{AE}, Im)$ is input to the *vae* model to perform the training of the anomaly detection model.

Algorithm 2 The two-level feature extraction based unsupervised anomaly detection approach

Input: An IIoT dataset $D = (F_n, I_m)$; Autoencoder model AE; VAE model vae; number of iterations 100.
Output: Anomalous instances I_a

```
Extract features F_AE = {F_1, F_2, ..., Fn} by Autoencoder model AE
VAE model vae training with D = (F_AE, I_m)
weights of features |W| = {|W_1|, |W_2|, ..., |W_AE|} ← Efficient DeepExplainer explains VAE model vae
E_AE = {E_1, E_2, ..., E_n} ← Sort features F_AE = {F_1, F_2, ..., Fn} in descending order according to weights W = {W_1, W_2, ..., W_AE}
for i = 2 to E_AE do
    VAE model vae training with D = (E_AE, I_m)
    Calculate the weighted F1-score.
end for
Extract the number of features num_F_AE with the highest weighted F1-score
VAE model vae training with D = (num_F_AE, Im)
```

5 EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we report on experiments we conducted to validate the effectiveness of the proposed energy-efficient and trustworthy unsupervised anomaly detection framework (EATU). First, we use three real-world IIoT datasets to assess the improvement of the accuracy of the proposed approach. Second, we demonstrate the energy-efficiency of our framework comparing with state-of-the-art baselines. Finally, we present the trustworthiness of our framework.

5.1 Datasets

We perform the evaluation using the following three real-world IIoT datasets. Table 2 summarizes the quantitative characteristics of the datasets.

SECOM Dataset. This dataset was collected from 590 sensors in the semiconductor manufacturing process [12]. The dataset consists of manufacturing operation data and semiconductor quality data. The aim of SECOM dataset is to detect the anomalous semi-conductor from manufacturing process.

Wafer Manufacturing Anomaly Detection Dataset. This dataset was collected from one of India's leading manufacturers of wafers (semiconductors) [4]. The wafer manufacturing system needs to be monitored every 10 milliseconds to capture their abnormal behaviors.

Air Pressure System (APS) Failure at Scania Trucks Dataset. This dataset consists of data collected from heavy Scania trucks in everyday usage [27]. The technology under consideration is the Air Pressure System (APS), which creates pressurised air for use in different vehicle tasks such as braking and gear changes.

Table 2. Description of the Datasets

Dataset	Instances Number	Features Number	Anomalies Number
SECOM	1567	590	104
Wafer	1763	1558	143
APS	60000	171	16000

5.2 Baselines

We benchmark the performance of our proposed framework against the following state-of-the-art baselines in terms of energy efficiency, anomaly detection accuracy and trustworthiness: 1) Kernel SHAP, 2) Gradient-based Explainable VAE (GEE) [14], and 3) DeepExapliner. The three models are all model-agnostic, perturbation-based explanation techniques. Kernel SHAP estimates for an instance x the contributions of each feature value to the prediction. It has a solid theoretical foundation in game theory (refer to Section 2.2), where its prediction is fairly distributed among the feature values, and have contrastive explanations that compare the prediction with the average prediction. GEE is a framework using gradient-based fingerprinting technique for explaining anomalies. It computes the derivative of the variational lower bound for each feature of every data point. In order to achieve a fair comparison, we use the same VAE network to train anomaly detection for GEE and our proposed framework. DeepExplainer is an improved DeepLIFT algorithm with a game theory based explanation, which can be used not only on deep models but any others.

5.3 Experimental Setup

5.3.1 Data Preprocessing. Since IIoT data are collected from various industrial facilities via different protocols such as Zigbee, Bluetooth and MQTT (Message Queuing Telemetry Transport) [9] etc., features of IIoT data that have irregular values are largely repetitive and noisy and will therefore affect the performance of underlying machine learning models. Data preprocessing is a necessary and integral step to make sure data can reserve its quality and be best positioned for the subsequent machine learning models. In order to make input data compatible with the Autoencoder-based feature extraction approach, the unsupervised anomaly detection model (in Section 4.3), and the interpreter model (in Section 4.2), we process the data with following rules:

- **Handle missing values.** The real-world dataset often has a lot of missing values. There are many ways to handle missing values for a dataset, e.g., deductive imputation [19]. We impute missing values with mean value of that feature column, denoted by $C_n = (C_1 + C_2 + C_3 + ... + C_n)/n$, where C_i denotes the i_{th} value in that feature

column and n is the number of rows in that feature column; this can prevent the loss of hidden information behind the data compared to deleting the missing values.

- **Handle categorical variables.** Some features and labels in a dataset are shown as categorical values (e.g., text data), and neural network models require input data to be in the numerical format. Hence, we use LableEncoder class from the sklearn library to convert categorical data into model-understandable numerical data. LabelEncoder can encode labels with a value between 0 and $m - 1$, where m denotes the number of distinct labels.
- **Standardize the data.** Due to different features and different measurement units, the range of input data has large differences. Standardization can make data be internally consistent across different inputs. Standardization scales the values that are centered around the mean with a unit standard deviation. We use the sklearn library to standardize data, which scales the mean of the attribute to zero and the resultant distribution has a unit standard deviation, denoted by $Z = \frac{x-\mu}{\sigma}$, where x represents input data, μ is the mean of input data, and σ denotes the standard deviation of input data.

5.3.2 Hyperparameters. For the autoencoder feature extraction model, we employ batch normalization and leaky Relu function as activation function in both the Encoder network and the Decoder network. For different datasets, we set different bottleneck dimension around 50 to 100. For feature extraction, there is no need to consider the decoder network's capacity and the limitation of regenerating data. On the other hand, if too many features are considered, it would raise the energy consumption issue due to the needs of higher computing resources.

For the VAE-based anomaly detection model, we build the encoder and decoder neural networks using two hidden layers and a four dimensional latent variable. We set 100 training epochs, and the input shape is dependent on the number of features in the dataset. The activation function we employ in the encoder network is ReLu and Linear, and in the decoder network is Relu and Sigmoid. As there are hidden layers before the decoder network, the Sigmoid function is more suitable in the decoder due to its capability of activating nonlinear layers (hidden layers). Due to the need for real-time training in many IIoT scenarios, Adam is employed as the optimization algorithm for achieving real-time training, where a small minibatch size of data is selected to train and then discard in every training epoch.

5.3.3 Compute Used. In this work, we ran all experiments on a single NVIDIA TITAN RTX GPU.

5.4 Performance Evaluation

In this section, we evaluate our proposed framework in terms of three important aspects, i.e., anomaly detection accuracy, energy consumption and trustworthiness of framework. The experimental results and analysis are presented below.

5.4.1 ***Anomaly Detection Accuracy.*** To measure the accuracy performance of anomaly detection, we use AUC-ROC [13] and weighted F1-score [34] metrics. AUC-ROC can reflect the performance of a classification problem at various threshold settings, which shows how well the anomaly detection model is able to distinguish normal and abnormal data. The mathematics behind AUC-ROC is that it is calculated by True Positive Rate (TPR) and False Positive Rate (FPR) using Eqs. (8) and 9), respectively. The AUC-ROC rate is formulated as $\frac{TPR}{FPR}$, the higher AUC-ROC rate, the better the ability of model classification. Since the real-world datasets are normally imbalanced, weighted F1-score can calculate metrics for each label (normal and abnormal in our experiment), and find their average weighted value by the number of true instances for each label. Weighted F1-score can be computed by Eq. (11), where w is the class weight, i is the ith class, and n is the total number of classes.

$$Precision = \frac{True\ Positives}{(True\ Positives + False\ Positives)} \tag{7}$$

$$TPR = Recall = \frac{True\ Positives}{(True\ Positives + False\ Negatives)} \tag{8}$$

$$FPR = \frac{FalsePositive}{FalsePositive + TrueNegative} \tag{9}$$

$$F1 - score = 2 \times \frac{(precision \times recall)}{(precision + recall)} \tag{10}$$

$$weighted\ F1 - score = \frac{\sum_n^i w \times (F1 - score^{(i)})}{n} \tag{11}$$

In our two-level feature extraction approach, it firstly extracts features by Autoencoder models, then leverages feature importance from Efficient DeepExplainer to re-select features. In order to improve anomaly detection accuracy, we iterate features from Efficient DeepExplainer's feature importance ranking, to choose the best numbers of important features. Table 3 shows the improvement of anomaly detection accuracy after using the the best numbers of important features comparing with only extracting features by Autoencoder to certain numbers of features. The improvement range from approximately 6.52% to 12.96%. The improvement shows that features selected from Efficient DeepExplainer-based feature importance have significant influence on the performance of model training. In addition, Efficient DeepExplainer is supported by game theory, which considers the global connection among all the features. Thus, Efficient DeepExplainer can make a bigger improvement in terms of accuracy owing to the better extracted features than the Autoencoder model. In other situations, GEE shows an average improvement of 4.98%. The comparison demonstrates that our proposed framework can not only improve anomaly detection accuracy, but also perform better than gradient-based explanation (GEE: Section 5.2). GEE is using a gradient-based fingerprints technique, but it cannot revel unseen patterns when the dimensionality of dataset is high.

Table 3. Comparison the accuracy of our approach EATU with only using Autoencoder feature extraction and GEE in anomaly detection

Dataset	Method	AUC-ROC ↑	weighted F1-score ↑
SECOM	Autoencoder	72.16	75.34
	GEE	78.41 (+6.25)	80.49 (+5.15)
	EATU	82.72 (+10.56)	84.53 (+9.19)
Wafer	Autoencoder	76.83	79.24
	GEE	79.39 (+2.56)	81.26 (+2.02)
	EATU	88.16 (+11.33)	89.37 (+10.13)
APS	Autoencoder	80.29	91.24
	GEE	88.68 (+8.39)	94.60 (+3.36)
	EATU	93.25 (+12.96)	97.76 (+6.52)

5.4.2 ***Energy consumption of framework.*** To measure the energy consumption of our framework, we use metrics wall clock time and resource usage.

Wall clock time – the elapsed time between when a process starts to run and when it is finished. Intuitively, it is the time we can get if we measure the computer run-time with a stopwatch. It consists of CPU time, I/O time and the communication channel delay. We include an explanation efficiency comparison of Efficient-DeepExplainer, with baselines KernelSHAP, DeepExplainer and Gradient-based fingerprint, in terms of wall clock time. We also provide results demonstrating the wall clock time comparison of the explainable anomaly detection task between our framework EATU and GEE (Section: 5.2) on three datasets. The result provided in Fig.4 demonstrates that Efficient-DeepExplainer has the fastest explanation procedure (near 30 seconds in three datasets) over KernelSHAP (above around 350 seconds), DeepExplainer (above around 250 seconds), and Gradient-based fingerprint (above around 190 seconds), which improves on the baselines by 8-fold. The comparison of explainable anomaly detection tasks includes the anomaly detection wall clock time and explanation wall clock time, which is the wall clock time of entire framework procedure. The results in Fig. 5 show that our proposed framework EATU has the most energy-efficient explanable anomaly detection procedure, which is about 4 times energy saving than GEE. These results clearly demonstrate that the Efficient-DeepExplainer approach, and our two-level feature extraction framework, can significantly speed up the process of detecting anomalous and generating explanation.

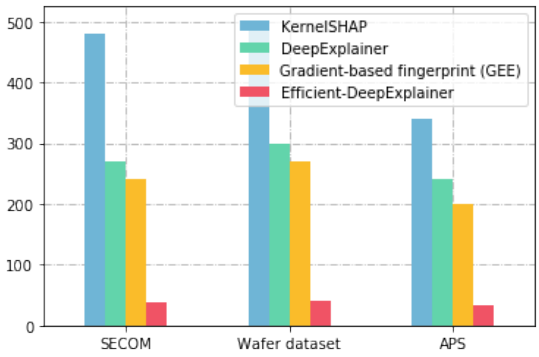

Fig. 4. Explanation efficiency Wall Clock Time comparison

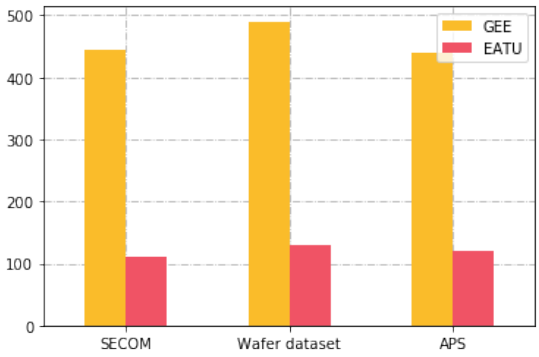

Fig. 5. Framework Wall Clock Time comparison

Resource usage Resource usage metric describes how the system resource has been used, e.g. the number of times the process is swapped out of main memory. Table 4 demonstrates resource usage among Kernel SHAP, GEE, DeepExplainer and our proposed framework EATU. We can see that although GEE performs better on SECOM dataset, EATU has smaller resource usage when datasets have higher numbers of features, which means EATU performs energy efficiently on high dimensional IIoT data.

5.4.3 ***Trustworthiness of framework.*** We evaluate the trustworthiness of our framework by assessing if the feature importance returned by the proposed explainer is ground-truth correct and the stability of explanation.

Results of explanations Fig. 6 shows the explanation of the anomalous data detected from the VAE model in the Wafer dataset. The figure presents the feature importance which provides a visual depiction for explaining the extracted features' influence. Features causing the increase in prediction value are in blue, and the visual size shows the magnitude of the feature's effect. Features having the decreasing effect on the prediction are in red.

Correctness of explanation The approach of measuring the correctness of explanations is to compute the correlations between the importance assigned by the explainer to the features and the effect of each of the features on

Table 4. Resource usage of EATU and other baselines

Dataset	Method	Resource usage (MByte)
SECOM	Kernel SHAP	3.6 MByte
	GEE	2.0 MByte
	DeepExplainer	3.4 MByte
	EATU	2.2 MByte
Wafer	Kernel SHAP	4.8 MByte
	GEE	3.6 MByte
	DeepExplainer	4.2 MByte
	EATU	2.6 MByte
APS	Kernel SHAP	3.6 MByte
	GEE	3.6 MByte
	DeepExplainer	3.4 MByte
	EATU	2.6 MByte

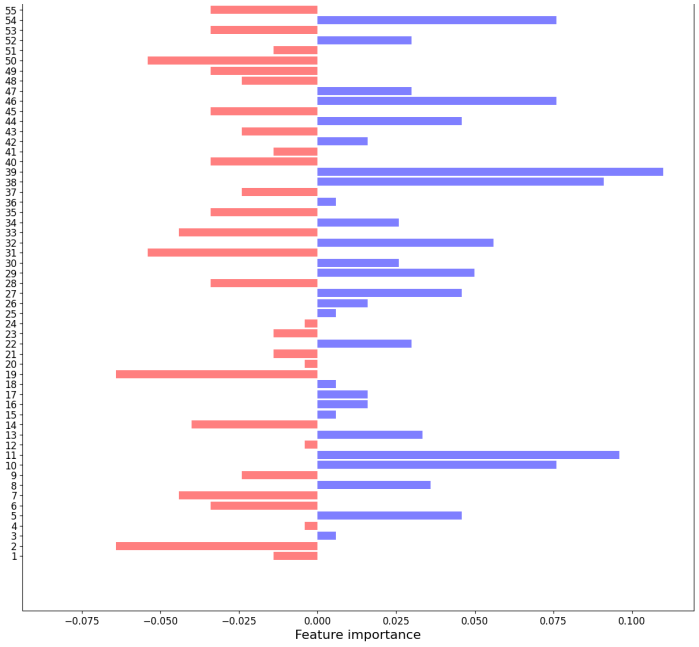

Fig. 6. Results of explanations on the Wafer dataset

the performance of the anomaly detection model. This method is inspired by [2]. Intuitively, this method replaces the feature value with the base value, and then passes it over to the anomaly detection model. Removing the features with high importance would degrade the performance of the anomaly detection model. Given a feature importance vector θ and the corresponding prediction probabilities $\mathbf{p}$, with the use of Pearsons correlation [23], the correction of explanations can be computed as follows:

$$\phi = -\rho(\theta, \mathbf{p}) \tag{12}$$

where θ denotes the feature importance vector and $\mathbf{p}$ represents the corresponding prediction probabilities.

The higher the value of ϕ, the better the explanation. Fig. 7 depicts our approach and the baseline correctness distribution of explanation. As shown in the figure, the explanation from Efficient-DeepExplainer has a mean value of around 0.095 in the three datasets separately, and is higher than Kernel SHAP and GEE, which means features explained by Efficient-DeepExplainer have a higher importance. Furthermore, the high deviation indicates that the explanation from Efficient-DeepExplainer has closer correlations with the ground-truth explanations. The advantage of using Efficient DeepExplainer-based explainer in our approach is that it has a solid theoretical foundation in game theory which can guarantee a high accuracy explanation, compared with other explainers. Meanwhile, Kernel SHAP has issues of feature dependence ignorance. Since Kernel SHAP randomly samples from the marginal distribution by replacing feature values from random instances, if features are dependent that would lead to putting too much weight on unlikely data points.

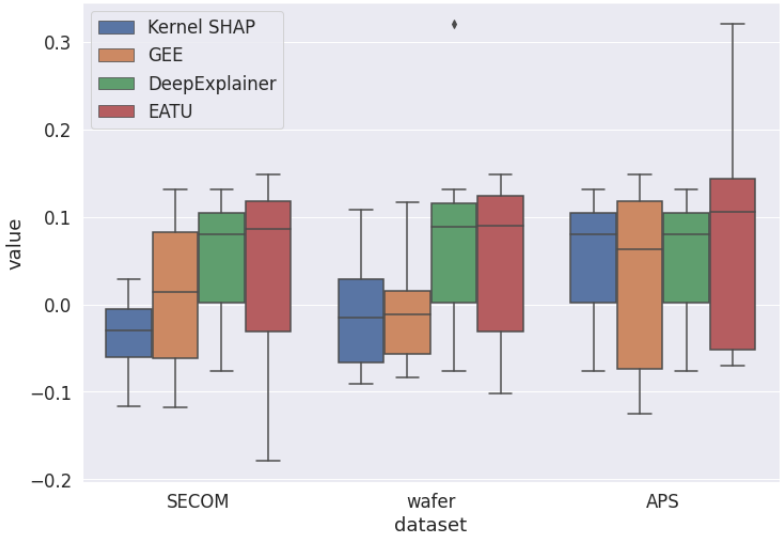

Fig. 7. Correctness of explanation

Stability of explanations To perform the stability analysis, we use the local Lipschitz metric [1]:

$$\hat{L}(x_i) = \underset{x_j \in N_\epsilon(x_i)}{\operatorname{argmax}} \frac{||\phi_i - \phi_j||_2}{||x_i - x_j||_2} \tag{13}$$

where x_i presents a data instance, $N_\epsilon(x_i)$ is a ball centered at x_i with ϵ radius, and ϕ_i and ϕ_j are the explanations for x_i and x_j. Lower local Lipschitz values indicate the explanation is more stable. We follow the calculation setting in [1] to compute the local Lipschitz values, comparing DeepExplainer and our framework EATU across SECOM, Wafer and APS datasets. We perform the comparison using the same number of perturbations (100) in DeepExplainer and our framework EATU, with batch size 2500. The results presenting in Fig. 8 show a significant improvement (about 50%) in stability in all datasets for EATU.

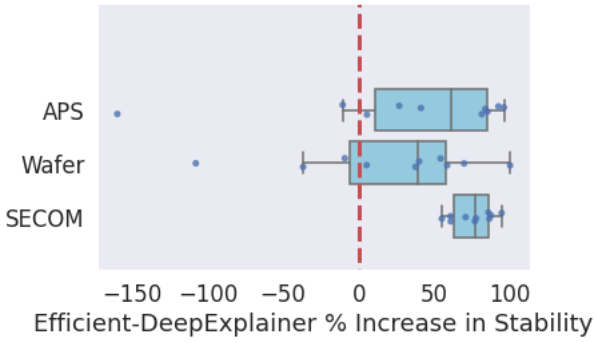

Fig. 8. Stability of explanation

6 CONCLUSIONS

In this paper, we proposed an energy-efficient and trustworthy unsupervised anomaly detection framework (EATU) which not only presents low energy consumption, but also improves upon the accuracy, as well as the trustworthiness of anomaly detection in IIoT. Our framework processes an energy efficient two-level feature extraction. The first level feature extraction was based on Autoencoders, and the second level was dependent on the Efficient-DeepExplainer. Both levels reduce computational complexity, in terms of features dimension reduction, and the presence of Efficient-DeepExplainer. In addition, our framework improved the quality of extracted features, hence enhancing the accuracy of anomaly detection. The explanation from the Efficient-DeepExplainer also improved the trustworthiness of the anomaly detection. Experimental results on real-world IIoT datasets showed that, comparing with the baselines, our framework EATU demonstrates the features of low energy consumption that is 4 times faster, high anomaly detection accuracy with about 10% improvement, and enhanced trustworthiness.

REFERENCES

[1] David Alvarez-Melis and Tommi S Jaakkola. 2018. On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049* (2018).

[2] David Alvarez-Melis and Tommi S. Jaakkola. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems* (Montréal, Canada) *(NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, 7786–7795.

[3] Liat Antwarg, Bracha Shapira, and Lior Rokach. 2019. Explaining anomalies detected by autoencoders using SHAP. *arXiv preprint arXiv:1903.02407* (2019).

[4] ask9. 2020. Detecting Anomalies in Wafer Manufacturing. https://www.kaggle.com/arbazkhan971/anomaly-detection. Accessed: 28.09.2021.

[5] Hugh Boyes, Bil Hallaq, Joe Cunningham, and Tim Watson. 2018. The industrial internet of things (IIoT): An analysis framework. *Computers in Industry* 101 (2018), 1–12. https://doi.org/10.1016/j.compind.2018.04.015

[6] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[7] Ludmila I. Kuncheva and William J. Faithfull. 2014. PCA Feature Extraction for Change Detection in Multidimensional Unlabeled Data. *IEEE Transactions on Neural Networks and Learning Systems* 25, 1 (2014), 69–80. https://doi.org/10.1109/TNNLS.2013.2248094

[8] Wei Liang, Yiyong Hu, Xiaokang Zhou, Yi Pan, and Kevin I-Kai Wang. 2021. Variational Few-Shot Learning for Microservice-Oriented Intrusion Detection in Distributed Industrial IoT. *IEEE Transactions on Industrial Informatics* (2021), 1–1. https://doi.org/10.1109/TII.2021.3116085

[9] Roger A Light. 2017. Mosquitto: server and client implementation of the MQTT protocol. *Journal of Open Source Software* 2, 13 (2017), 265.

[10] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems* 30 (2017).

[11] E Laxmi Lydia, A Arokiaraj Jovith, A Francis Saviour Devaraj, Changho Seo, and Gyanendra Prasad Joshi. 2021. Green energy efficient routing with deep learning based anomaly detection for internet of things (IoT) communications. *Mathematics* 9, 5 (2021), 500.

[12] McCann Michael and Johnston Adrian. 2008. SECOM Data Set. http://archive.ics.uci.edu/ml/datasets/secom. Accessed: 17.11.2021.

[13] Sarang Narkhede. 2018. Understanding auc-roc curve. *Towards Data Science* 26 (2018), 220–227.

[14] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. 2019. GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection. In *2019 IEEE Conference on Communications and Network Security (CNS)*. 91–99. https://doi.org/10.1109/CNS.2019.8802833

[15] Meikang Qiu, Edwin H.-M. Sha, Meilin Liu, Man Lin, Shaoxiong Hua, and Laurence T. Yang. 2008. Energy minimization with loop fusion and multi-functional-unit scheduling for multidimensional DSP. *J. Parallel and Distrib. Comput.* 68, 4 (2008), 443–455. https://doi.org/10.1016/j.jpdc.2007.06.014

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " Why should i trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 1135–1144.

[17] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[18] Eli Rosenberg and Maya Salam. [n. d.]. Hacking Attack Woke Up Dallas With Emergency Sirens, Officials Say. https://www.nytimes.com/2017/04/08/us/dallas-emergency-sirens-hacking.html. Accessed: 22.09.2021.

[19] Manish Sarkar and Tze-Yun Leong. 2001. Fuzzy K-means clustering with missing values.. In *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 588.

[20] Raed Abdel Sater and A Ben Hamza. 2021. A federated learning approach to anomaly detection in smart buildings. *ACM Transactions on Internet of Things* 2, 4 (2021), 1–23.

[21] Cedric Schockaert, Vadim Macher, and Alexander Schmitz. 2020. VAE-LIME: deep generative model based approach for local data-Driven model interpretability applied to the ironmaking industry. *arXiv preprint arXiv:2007.10256* (2020).

[22] Sdgs.un.org. 2020. THE 17 GOALS | Sustainable Development. https://sdgs.un.org/goals. Accessed: 30.03.2022.

[23] Philip Sedgwick. 2012. Pearson's correlation coefficient. *Bmj* 345 (2012).

[24] Zili Shao, C. Xue, Q. Zhuge, M. Qiu, Bin Xiao, and E.H.-M. Sha. 2006. Security protection and checking for embedded system integration against buffer overflow attacks via hardware/software. *IEEE Trans. Comput.* 55, 4 (2006), 443–453. https://doi.org/10.1109/TC.2006.59

[25] Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713* (2016).

[26] Dylan Slack, Anna Hilgard, Sameer Singh, and Himabindu Lakkaraju. 2021. Reliable post hoc explanations: Modeling uncertainty in explainability. *Advances in Neural Information Processing Systems* 34 (2021).

[27] Lindgren Tony and Biteus Jonas. 2017. APS Failure at Scania Trucks Data Set. https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks#. Accessed: 17.11.2021.

[28] Bizhu Wang, Yan Sun, and Xiaodong Xu. 2021. A Scalable and Energy-Efficient Anomaly Detection Scheme in Wireless SDN-Based mMTC Networks for IoT. *IEEE Internet of Things Journal* 8, 3 (2021), 1388–1405. https://doi.org/10.1109/JIOT.2020.3011521

[29] Xiaokang Wang, Laurence T. Yang, Xingyu Chen, Jian-Jun Han, and Jun Feng. 2019. A Tensor Computation and Optimization Model for Cyber-Physical-Social Big Data. *IEEE Transactions on Sustainable Computing* 4, 4 (2019), 326–339. https://doi.org/10.1109/TSUSC.2017.2777503

[30] Xiaokang Wang, Laurence T. Yang, Lei Ren, Yihao Wang, and M. Jamal Deen. 2022. A Tensor-Based Computing and Optimization Model for Intelligent Edge Services. *IEEE Network* 36, 1 (2022), 40–44. https://doi.org/10.1109/MNET.011.1800508

[31] Yulei Wu, Hong-Ning Dai, and Haina Tang. 2021. Graph Neural Networks for Anomaly Detection in Industrial Internet of Things. *IEEE Internet of Things Journal* (2021), 1–1. https://doi.org/10.1109/JIOT.2021.3094295

[32] Yulei Wu, Hong-Ning Dai, Haozhe Wang, Zehui Xiong, and Song Guo. 2022. A Survey of Intelligent Network Slicing Management for Industrial IoT: Integrated Approaches for Smart Transportation, Smart Energy, and Smart Factory. *IEEE Communications Surveys & Tutorials* 24, 2 (2022), 1175–1211. https://doi.org/10.1109/COMST.2022.3158270

[33] Xingjie Yu and Huaqun Guo. 2019. A Survey on IIoT Security. In *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. 1–5. https://doi.org/10.1109/VTS-APWCS.2019.8851679

[34] Dell Zhang, Jun Wang, and Xiaoxue Zhao. 2015. Estimating the uncertainty of average F1 scores. In *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*. 317–320.

[35] Luying Zhou and Huaqun Guo. 2018. Anomaly Detection Methods for IIoT Networks. In *2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. 214–219. https://doi.org/10.1109/SOLI.2018.8476769

[36] Xiaokang Zhou, Wei Liang, Weimin Li, Ke Yan, Shohei Shimizu, and Kevin I-Kai Wang. 2021. Hierarchical Adversarial Attacks Against Graph Neural Network Based IoT Network Intrusion Detection System. *IEEE Internet of Things Journal* (2021), 1–1. https://doi.org/10.1109/JIOT.2021.3130434

[37] Xiaokang Zhou, Xuesong Xu, Wei Liang, Zhi Zeng, and Zheng Yan. 2021. Deep-Learning-Enhanced Multitarget Detection for End–Edge–Cloud Surveillance in Smart IoT. *IEEE Internet of Things Journal* 8, 16 (2021), 12588–12596. https://doi.org/10.1109/JIOT.2021.3077449

[38] Xiaokang Zhou, Xiang Yang, Jianhua Ma, and Kevin I-Kai Wang. 2021. Energy Efficient Smart Routing Based on Link Correlation Mining for Wireless Edge Computing in IoT. *IEEE Internet of Things Journal* (2021), 1–1. https://doi.org/10.1109/JIOT.2021.3077937

[39] Yuan Zuo, Yulei Wu, Geyong Min, Chengqiang Huang, and Ke Pei. 2020. An Intelligent Anomaly Detection Scheme for Micro-Services Architectures With Temporal and Spatial Data Analysis. *IEEE Transactions on Cognitive Communications and Networking* 6, 2 (2020), 548–561. https://doi.org/10.1109/TCCN.2020.2966615