Pattern Recognition Letters 171 (2023) 131–137

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Autoencoders with exponential deviation loss for weakly supervised anomaly detection

Min-Seong Kwon[c], Yong-Geun Moon[a], Byungju Lee[b], Jung-Hoon Noh[a,*]

[a] Department of Aeronautics, Mechanical and Electronic Convergence Engineering, Kumoh National Institute of Technology, Gumi, South Korea
[b] Department of Information and Telecommunication Engineering, Incheon National University, Incheon, South Korea
[c] Department of Electrical Engineering, Ulsan National Institute of Science & Technology, Ulsan, South Korea

ARTICLE INFO

Article history:
Received 15 October 2022
Revised 29 March 2023
Accepted 16 May 2023
Available online 18 May 2023

Edited by: Jiwen Lu

Keywords:
Anomaly detection
Deep learning
Weakly supervised learning

ABSTRACT

Weakly supervised anomaly detection aims to detect anomalies using a small number of labeled anomalies and a large amount of unlabeled data. However, existing methods have limitations: unsuitable setting of the anomaly threshold, using an inefficient loss function, and being vulnerable to contaminated data. To address these limitations, this paper proposes a novel framework called the Exponential Deviation Autoencoder (EDAE), which consists of two stages. In the first stage, EDAE pre-trains an autoencoder (AE) to learn a compressed representation of the input data and estimates the anomaly score distribution of the training data to determine an appropriate anomaly threshold. In the second stage, EDAE fine-tunes the AE with a novel Exponential Deviation Loss (EDL) function that provides continuous and nonlinear penalties according to anomaly scores and enables more effective training using labeled anomalies. EDAE also uses batch sampling based on empirical distribution to create batches of data that are more robust to contaminated data. We conduct extensive experiments on various datasets and show that EDAE outperforms state-of-the-art weakly supervised methods with up to a 26% improvement in accuracy.

1. Introduction

Anomaly detection (AD) is a crucial branch of data mining and machine learning, widely used in various scenarios such as cybersecurity, disease detection in healthcare, and fraudulent financial transactions [1]. The objective of AD is to identify unusual instances within a general data distribution. However, AD often encounters a critical challenge due to the high cost and time required to obtain large-scale labeled anomalies. Consequently, fully supervised methods that rely on both normal and abnormal data for training a detector are often impractical. In contrast, unsupervised methods have been widely investigated due to their ability to avoid the need for large-scale labeled anomalies [2–11]. Nevertheless, such methods often suffer from a key limitation: the lack of knowledge about labeled data makes it challenging to determine what constitutes a true anomaly, leading to higher false alarm rates [12].

To address this limitation, weakly supervised methods have been proposed, which leverage a small number of labeled anomalies along with a large amount of unlabeled data. Several weakly supervised methods have been proposed, which have demonstrated superior performance over unsupervised methods [13–16]. Notably, most of these methods assume that anomaly scores in real-world datasets follow a normal (Gaussian) distribution, from which an anomaly threshold is established. A deviation loss function [15] then is employed to elevate the anomaly scores of labeled anomalies significantly above the threshold while minimizing the anomaly scores of unlabeled data (typically normal instances).

However, previous research has some limitations that need to be addressed. One of the main limitations is the assumption that anomaly scores in real-world datasets follow a normal (Gaussian) distribution, which cannot be guaranteed for all datasets. As a result, anomaly thresholds obtained based on a normal distribution may be inefficient for some datasets, either being too small or too large, which can lead to poor performance. Another limitation is the inefficiency of the deviation loss function. The deviation loss function has a discontinuity, which means that it does not penalize anomaly instances whose anomaly scores are larger than the threshold. This results in a very small amount of anomaly data that contributes to model training, which can lead to suboptimal performance. Finally, previous studies have not adequately addressed the presence of anomalies in unlabeled datasets, such as contaminated data, which can also lead to suboptimal performance.

* Corresponding authors.
E-mail addresses: minseongkwon@unist.ac.kr (M.-S. Kwon), masuriji@kumoh.ac.kr (Y.-G. Moon), bjlee@inu.ac.kr (B. Lee), jhnoh@kumoh.ac.kr (J.-H. Noh).

https://doi.org/10.1016/j.patrec.2023.05.016
0167-8655/

To overcome the limitations of previous research, we propose a novel framework for weakly supervised anomaly detection called the Exponential Deviation Autoencoder (EDAE). EDAE is a two-step approach designed to address the challenges of anomaly detection. In the first step, we pre-train an autoencoder (AE) using only unlabeled data as input. Once the pre-training AE has been trained, we estimate the distribution of anomaly scores based on the unlabeled dataset, which enables us to find a more appropriate anomaly threshold even in the presence of data that does not follow a normal distribution.

In the second step, we fine-tune the AE by utilizing labeled anomalies in the training phase. This step incorporates the exponential deviation loss (EDL), a novel loss function that forces the anomaly scores of labeled anomalies to significantly exceed the obtained threshold. EDL provides continuous and nonlinear penalties according to anomaly scores and allows outlier data with anomaly scores greater than a threshold to be used for training, making it more effective than the deviation loss.

We also propose a batch sampling method based on the empirical distribution, which creates batches of data that are more robust to contaminated data. This method uses inverse transform sampling technique based on its empirical distribution to extract as much normal data as possible and excludes data with high anomaly score. As a result, EDAE facilitates the creation of an efficient and contamination-resistant network.

In summary, the proposed framework has three main contributions: (i) estimating the distribution of anomaly scores to determine the anomaly threshold, which addresses the limitation of assuming a normal distribution for anomaly scores in real-world datasets; (ii) introducing a novel loss function called EDL, which provides continuous and nonlinear penalties according to anomaly scores and enables more effective training using labeled anomalies; and (iii) proposing a batch sampling method based on the empirical distribution, which creates batches of data that are more robust to contaminated data.

2. Related work

AD can be categorized into three groups depending on whether the anomaly samples are available at the training stage: supervised, unsupervised, and weakly supervised methods. However, obtaining a significant number of labeled anomalies can be challenging or infeasible, making fully supervised methods impractical. Hence, we primarily focus on introducing related unsupervised and weakly supervised AD methods.

2.1. Unsupervised anomaly detection

Unsupervised learning methods can be applied to any dataset without the need for labeled data, making them more widely applicable. Many unsupervised methods have been proposed in the last few decades, which can be roughly categorized into shallow and deep methods according to the depth of the model. Compared to shallow AD methods, deep AD methods have the advantage of being able to capture more complex and subtle patterns in high-dimensional data, making them more effective in detecting anomalies [17].

Deep unsupervised AD methods can be classified into several categories based on their detection methods, including AE-based methods, generative models, clustering-based methods, and graph-based methods. AE-based methods [2–4] use deep neural networks to learn a compressed representation of the input data, and then use reconstruction error to identify anomalies. Generative models [5,6] learn the probability distribution of the input data and use the probability density function to identify anomalies. Clustering-based methods [7,8] group similar data points together and identify anomalies as points that do not belong to any cluster. Graph-based methods [9] use graph theory to model the relationships between data points and identify anomalies as nodes with unusual connections.

In recent years, there have been several studies combining different AD models and applying them to various applications. Huang et al. [10] introduced a self-supervised learning approach to a deep AE, resulting in superior performance for visual anomaly detection. Meanwhile, Zhu et al. [11] proposed an adversarial training framework with Long Short-Term Memory Encoder-Decoder (LSTM-ED), demonstrating significant improvement in the detection of anomalies in time-series cyber system datasets.

In general, these models are based on different assumptions and yield superior results on certain datasets when the corresponding assumptions are met. However, the unsupervised approaches may face limitation in identifying the true nature of anomalies, due to the lack of knowledge of anomalies.

2.2. Weakly supervised anomaly detection

Weakly supervised methods address the scenario where a few annotated anomaly samples are available during training. Recent works have demonstrated that leveraging the limited number of abnormal samples can substantially improve the performance over the conventional unsupervised approaches. Zhao and Hryniewicki [13] proposed a three-phase system called XGBOD, which first uses unsupervised outlier detectors to extract representation of the data and then concatenates the newly generated features to the original feature for augmentation. An XGBoost classifier is then applied to this augmented feature space. Ruff et al. [14] modified an unsupervised deep SVDD [7] by exploiting an additional term in the existing objective function, thereby guaranteeing a large margin between the centroid and labeled anomalies. The most closely related works to our article are [15] and [16]. Pang et al. [15] proposed a deviation network (DevNet) that achieves end-to-end learning for anomaly scores using Z-score-based deviation loss. Zhou et al. [16] proposed feature encoding with an AE for weakly supervised anomaly detection (FEAWAD), which leverages an AE to encode input data, and utilizes hidden representation, reconstruction residual vector, and reconstruction error. The deviation loss used in DevNet and FEAWAD forces deviations in the anomaly scores between the abnormal and normal data. More details about DevNet and deviation loss are presented in Section 3.2.

Recently, numerous studies have explored the applications of weakly supervised anomaly detection across various fields, owing to its effectiveness in training scenarios with limited labeled data. Elaziz et al. [18] proposed a deep reinforcement learning approach that leverages a self-attention mechanism for identifying anomalies in business processes using sparse labeled data. In the field of medical image detection, Li et al. [19] investigated AD by employing weakly supervised learning in conjunction with generative adversarial networks. Furthermore, Tian et al. [20] developed an innovative technique for weakly supervised video AD that harnesses robust temporal feature magnitude learning to pinpoint abnormal segments within videos labeled as anomalous.

3. Preliminary

In this section, we provide a brief introduction to the AE for anomaly detection, serving as the base model. Furthermore, we introduce DevNet and its associated deviation loss.

3.1. Autoencoder for anomaly detection

An AE is a powerful tool in unsupervised AD [2–4]. The purpose of an AE is to reconstruct the original input accurately. The encoder maps the original data onto a low-dimensional feature space, whereas the decoder attempts to recover data from the projected low-dimensional space.

Specifically, the encoder maps the input vector $\mathbf{x}_i \in \mathbb{R}^m$ to its latent representation $\mathbf{z}_i \in \mathbb{R}^d (d < m)$ as follows:

$$\mathbf{z}_i = \mathcal{E}(\mathbf{x}_i; \Theta_{\mathcal{E}}) \tag{1}$$

where $\mathcal{E}(\cdot)$ is the encoder with the parameters $\Theta_{\mathcal{E}}$. The decoder maps the latent representation to the output reconstruction $\hat{\mathbf{x}}_i \in \mathbb{R}^m$ as follows:

$$\hat{\mathbf{x}}_i = \mathcal{D}(\mathbf{z}_i; \Theta_{\mathcal{D}}) \tag{2}$$

where $\mathcal{D}(\cdot)$ is the decoder with the parameters $\Theta_{\mathcal{D}}$. The network parameters of both neural networks $\Theta = \{\Theta_{\mathcal{E}}, \Theta_{\mathcal{D}}\}$ are trained to minimize the reconstruction error. The reconstruction error, which measures the square of the Euclidean distance between the input and output, can be expressed as:

$$\mathcal{A}_\Theta(\mathbf{x}_i) = \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2^2 \tag{3}$$

where $\mathcal{A}_\Theta(\mathbf{x}_i)$ is the anomaly score of input data $\mathbf{x}_i$. In general, AE-based AD uses the reconstruction error as the anomaly score. This method assumes that the AE cannot reconstruct data not utlized in the training, *i.e.*, unknown anomalous data.

3.2. DevNet

DevNet [15] is a weakly supervised AD that utilizes a few labeled anomalies to separate anomalies from the normal data instances. DevNet uses an end-to-end anomaly scoring network $\phi : \mathcal{X} \mapsto \mathbb{R}$, which directly maps the input data vectors to the anomaly scores of the scalar. Assuming that the anomaly score follows a normal distribution, DevNet randomly samples a sufficiently large number of normal variables to obtain reference scores for the mean μ_R and standard deviation σ_R. Having obtained the anomaly scoring network $\phi(x; \Theta)$ and the reference scores, the Z-score-based deviation can be introduced as:

$$\text{dev}(x) = \frac{\phi(x; \Theta) - \mu_R}{\sigma_R} \tag{4}$$

Based on contrastive loss [21], deviation loss can be represented as

$$\mathcal{L}_{\text{dev}} = (1 - y)|\text{dev}(x)| + y[\max(0, a - \text{dev}(x))] \tag{5}$$

where $y = 1$ if x is an anomaly, $y = 0$ if x is a normal object, and a is the confidence margin parameter, which is equivalent to the threshold in this study. In short, DevNet tries to yield the output of anomalies close to a and allows the output of normal data to be close to zero. DevNet has shown that using small amounts of anomaly data can significantly improve the detection performance.

However, there is room for further improvement. DevNet misses two crucial points: (i) there is no guarantee that the anomaly scores of every dataset follow a normal distribution. The confidence margin or threshold should be modified by a different value depending on the actual distribution of anomaly scores. Otherwise, the performance improvements can be limited to datasets whose anomaly scores do not follow the normal distribution; and (ii) in (5), the loss function excludes anomaly scores higher than a, but only considers anomaly scores lower than a. As anomalies are highly likely to have scores higher than a, the effective anomalies for loss function (5) are limited.

4. Proposed framework

4.1. Framework overview

We propose an enhanced weakly supervised anomaly detection framework named EDAE. As shown in Fig. 1, our framework is trained in two steps. Both steps use the reconstruction error as an anomaly score based on the AE. However, each step has a different use. Firstly, they differ in the type of data used for training. The pre-training AE (the first stage) is trained on unlabeled data to learn the distribution of data in advance, while the main training AE (the second stage) is trained on both labeled anomalies and an unlabeled data to make use of a few annotated anomalies. Secondly, their objective functions are also different. The pre-training AE uses reconstruction error as an objective function to learn the features of the unlabeled dataset. In contrast, the main training AE employs the novel EDL function, which aims to force the network to distribute the anomaly scores of the labeled anomalies above the threshold.

Specifically, given a training data set $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N, (\mathbf{x}_{N+1}, y_{N+1}), \ldots, (\mathbf{x}_{N+K}, y_{N+K})\}$ of size $N+K$ with $\mathbf{x}_i \in \mathbb{R}^m$ and $y \in \{0, 1\}$. Let $y = 0$ indicate a normal sample, and $y = 1$ indicate an abnormal sample. Dataset $\mathcal{X}$ is divided into two parts: $\mathcal{U} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is an unlabeled data set and $\mathcal{K} = \{(\mathbf{x}_{N+1}, y_{N+1}), \ldots, (\mathbf{x}_{N+K}, y_{N+K})\}$ is labeled anomalies that provide some prior knowledge of anomalies where $K \ll N$.

4.2. Pre-training autoencoder

Pre-training AE attempts to estimate the probability distribution of the anomaly scores of the normal data from the unlabeled data. The reason for estimating the probability distribution is to identify some contaminated data in training dataset, and to exclude contaminated data in the main training. In addition, a threshold is obtained from the estimated probability distribution, which is the reference point for determining an abnormality.

Specifically, the pre-training AE uses only the unlabeled dataset $\mathcal{U}$ as the training data. The pre-training AE uses the reconstruction error as an anomaly score. The network is trained to minimize the reconstruction error of the unlabeled dataset. The objective function of the pre-training AE is given by:

$$\Theta^* = \arg\min_\Theta \sum_{i=1}^{N} \mathcal{A}_\Theta(\mathbf{x}_i) = \arg\min_\Theta \sum_{i=1}^{N} \left\| \mathbf{x}_i - \hat{\mathbf{x}}_i \right\|_2^2. \tag{6}$$

Once the pre-training AE has been trained, network with Θ^* extracts the anomaly scores from the training dataset. The empirical distribution is then estimated based on the extracted anomaly scores. Since the loss function utilized in this stage is based on the squared value of the Euclidean distance, we assume that the distribution of the anomaly scores follows a gamma distribution, which can be expressed as

$$\mathcal{A}_{\Theta^*}(X) \sim \Gamma(\alpha, \beta), \tag{7}$$

where α and β correspond to the shape and scale parameters of the gamma distribution, respectively. These parameters are estimated from the anomaly scores of the training dataset. Before fitting the probability distribution to observed anomaly scores, we exclude data within a percentage at the top (5% in this work). This is because these data points are more likely to correspond to unlabeled anomaly data. Otherwise, even a small amount of anomalous data with high anomaly scores can distort the estimation of the anomaly score distribution.

The estimated anomaly score distribution is utilized when generating mini-batches in the following process. In addition, the anomaly score threshold η is determined as the point of the pre-

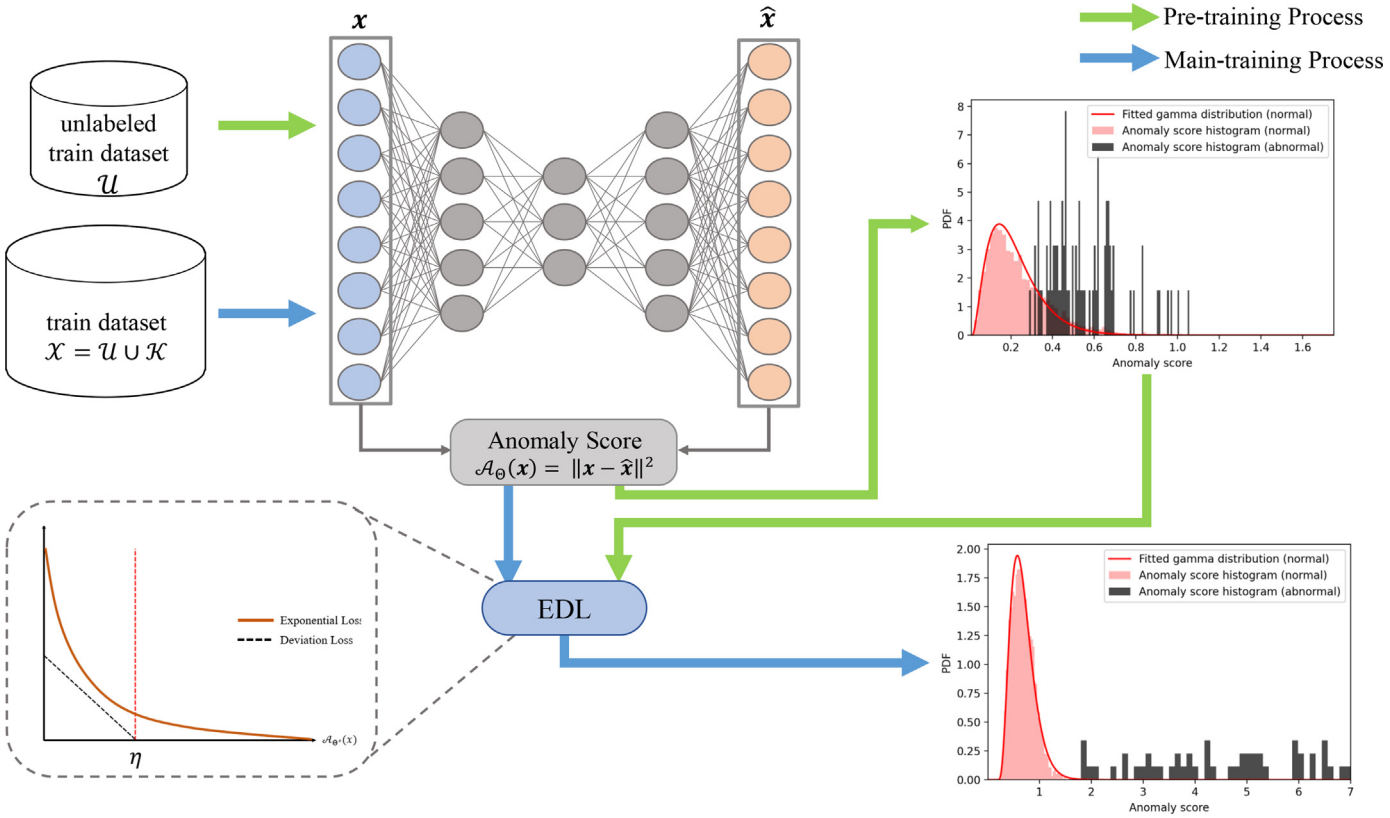

Fig. 1. An illustration of the proposed model. The model is trained in two steps, *i.e.*, the pre-training process (the green line in the figure) and the main training process (the blue line in the figure). Both steps exploit the reconstruction error as an anomaly score. The pre-training AE learns unlabeled dataset as train data, estimates the distribution of anomaly scores of unlabeled data, and finds a threshold. The main-training process uses EDL to map normal data close to zero and abnormal data higher than the threshold.

given percentile (99% in this study) of the cumulative density function (CDF) of the estimated anomaly score distribution expressed as

$$\exists \eta \in \mathbb{R} \qquad s.t. \qquad P(\mathcal{A}_{\Theta^*}(X) \leq \eta) = 0.99. \tag{8}$$

4.3. Weakly supervised autoencoder for anomaly detection

The existing unsupervised AE-based AD tests are directly based on reconstruction errors. These methods assume that anomalies cannot be properly reconstructed and that the anomaly score of abnormal data is highly likely to be larger than that of normal data. However, unsupervised learning may encounter performance degradation when the boundaries between the normal and abnormal data are unclear or complex. To address this issue, we propose a novel network, which forces the AE to assign higher anomaly scores to abnormal data using a limited number of known anomalies.

Specifically, we use $N + K$ datasets $\mathcal{X}$ for training. We assign $y = 1$ to the K labeled anomalies, and $y = 0$ to the N unlabeled dataset to define the loss function. The weakly supervised AE takes network parameters of the pre-training AE, estimated anomaly score distribution, and threshold η from the pre-training AE. In order to effectively distribute anomalies above the threshold, we define an exponential loss function for abnormal data as follows:

$$\mathcal{L}_{exp}(\mathbf{x}_i) = \lambda \exp(\gamma(\eta - \mathcal{A}_{\Theta^*}(\mathbf{x}_i)) \tag{9}$$

where λ and γ are hyperparameter of the exponential loss, η is the threshold obtained in the pre-training AE, and $\mathcal{A}_{\Theta^*}(\mathbf{x}_i)$ is the anomaly score of the input data $\mathbf{x}_i \in \mathcal{X}$. It should be noted that λ and γ must be positive. Then we propose the following EDL based on contrastive loss [21], which can be given by

$$\mathcal{L}_{EDL}(\mathbf{x}_i) = (1 - y_i)\mathcal{A}_{\Theta^*}(\mathbf{x}_i) + y_i \mathcal{L}_{exp}(\mathbf{x}_i) \tag{10}$$

where y_i is label, $\mathcal{A}_{\Theta^*}(\mathbf{x}_i)$ is the general anomaly score of AE, and $\mathcal{L}_{\exp}(\mathbf{x}_i)$ is the exponential loss of $\mathbf{x}_i$. EDL differs from deviation loss (5) in two primary ways. Firstly, EDL utilizes an exponential penalty mechanism based on the anomaly score, which enhances the network's sensitivity towards low anomaly scores. Secondly, EDL imposes a penalty even on anomalous data instances with scores exceeding the anomaly threshold, in contrast to deviation loss that refrains from penalizing such instances. As a result, the EDL approach enables the network to effectively utilize anomalous instances during the training process.

To better utilize the limited number of anomalies, we construct mini-batch by sampling an equal number of normal (unlabeled) and anomaly (labeled) samples as shown in Fig. 2. While the sampling from $\mathcal{K}$ is completely random, the sampling from $\mathcal{U}$ uses an inverse transform sampling technique to minimize the possibility of selecting contaminated data. Inverse transform sampling is a well-established technique for generating random numbers from any probability density function whose inverse can be computed. By selecting samples according to the empirical distribution of anomaly scores of normal data obtained in the pre-training step, the probability of selecting data outside this distribution (i.e., presumably contaminated data) is minimized.

The process of inverse transform sampling comprises the following steps: first, calculate the inverse CDF of the anomaly scores from (7) as $F^{-1}_{\mathcal{A}_{\Theta^*}(X)}(u; \alpha, \beta)$. Next, generate a uniform random variable $u_k \sim U(0, 1)$ where $k \in \{1, 2, \ldots, B/2\}$ and B is the size of the mini-batch. Then, input u_k into the $F^{-1}_{\mathcal{A}_{\Theta^*}(X)}(u; \alpha, \beta)$ function to obtain a random sample of anomaly scores following the empirical

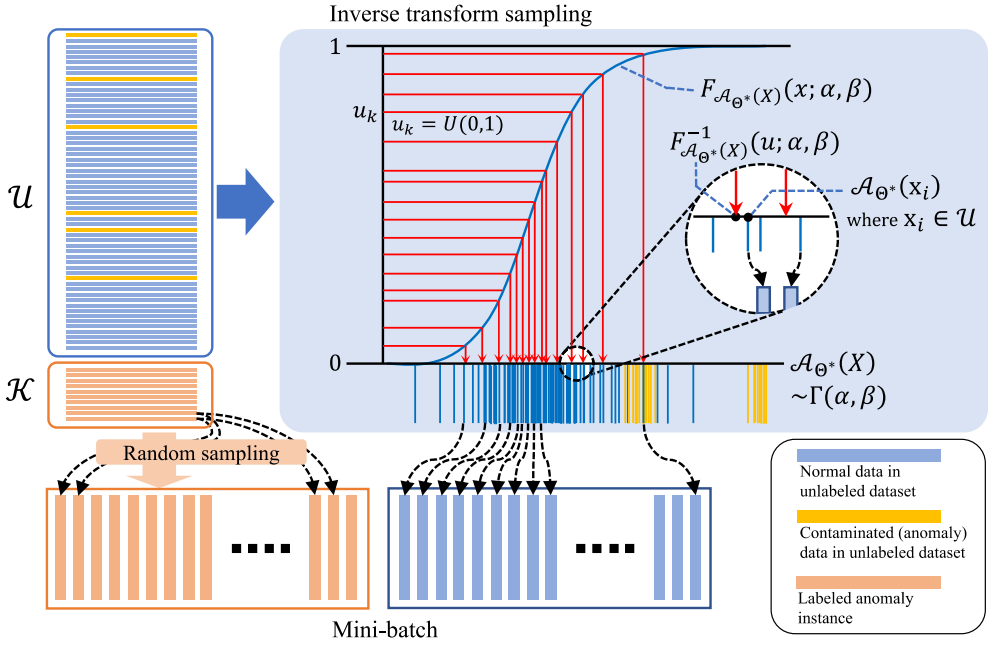

Fig. 2. An illustration of generating the mini-batch.

distribution of anomaly scores. Finally, select $\mathbf{x}_i \in \mathcal{U}$ such that

$$i = \underset{j}{\operatorname{argmin}} \left| \mathcal{A}_{\Theta^*}(\mathbf{x}_j) - F^{-1}_{\mathcal{A}_{\Theta^*}(X)}(u_k;\alpha,\beta) \right|, \text{ for } k \in \{1,2,\ldots,B/2\}. \tag{11}$$

Thus, by selecting the data whose pre-training anomaly score is closest to $F^{-1}_{\mathcal{A}_{\Theta^*}(X)}(u_k;\alpha,\beta)$, we can generate a data sample that conforms to the empirical distribution of anomaly scores while minimizing the selection of data that deviates from the distribution. Algorithm 1 presents the training procedure of EDAE.

Algorithm 1 EDAE Training Procedure.

Stage 1: Pre-training AE
Input: unlabeled dataset $\mathcal{U} \in \mathbb{R}^m$
Output: $\Theta^* = \{\Theta^*_{\mathcal{E}}, \Theta^*_{\mathcal{D}}\}$, $\Gamma(\alpha,\beta)$, threshold η

Randomly initialize $\Theta = \{\Theta_{\mathcal{E}}, \Theta_{\mathcal{D}}\}$
repeat
  Randomly sample one batch of unlabeled data
  Compute the reconstruction error $\mathcal{A}_\Theta(\mathbf{x}_i)$ and update parameters Θ
until converge
return $\Theta^* = \{\Theta^*_{\mathcal{E}}, \Theta^*_{\mathcal{D}}\} = \arg\min_{\Theta} \sum_{i=1}^{N} \mathcal{A}_\Theta(\mathbf{x}_i)$
Compute the anomaly scores $\{\mathcal{A}_{\Theta^*}(\mathbf{x}_i)\}_{i=1}^N$ and fit into a Gamma distribution $\Gamma(\alpha,\beta)$
Compute the threshold η by (**??**)

Stage 2: Main-training AE
Input: dataset $\mathcal{X} \in \mathbb{R}^m$ with $\mathcal{X} = \mathcal{U} \cup \mathcal{K}$ and $\varnothing = \mathcal{U} \cap \mathcal{K}$
Output: $\hat{\Theta}$

load $\Theta^*, \Gamma(\alpha,\beta), \eta$ from the pre-trained network
repeat
  inverse transform sample half from $\mathcal{U}$ and half of samples from $\mathcal{K}$ to form a batch
  Compute the EDL $\mathcal{L}_{EDL}$ and update the network parameters Θ^*
until converge
return $\hat{\Theta} = \arg\min_{\Theta^*} \sum_{i=1}^{N+K} \mathcal{L}_{EDL}(\mathbf{x}_i)$

5. Experiments

5.1. Datasets and experiment setting

To verify the performance of the proposed method, we used ten real-world datasets covering broad application domains, including healthcare, astronautics, and botany [22]. Details of the datasets are described in Table 1. We used 70% of the data for training and the remaining 30% for the testing set. We followed the ADBench [22] experimental settings. Each experiment was repeated five times, and the averages and standard deviations were reported. Accordingly, we assumed that we only knew approximately 5% of the anomalies in the training data.

5.2. Performance evaluation methods

We evaluated the anomaly detectors with two performance metrics: the area under the receiver operating characteristic curve (AUC-ROC) and the area under the precision-recall curve (AUC-PR). AUC-ROC is the area under the ROC curve describing the relationship between true and false-positive rates. AUC-PR is the area under the PR curve representing the relationship between precision and recall. AUC-ROC and AUC-PR lie in the range [0, 1], and a higher value indicates better performance. However, when dealing with an extreme positive-negative class imbalance, the AUC-PR is more informative than the AUC-ROC in reflecting the detector performance [23]. The AUC-PR and AUC-ROC performance compared in this paper are averaged over the results of five independent repeated experiments.

5.3. Experimental setup

To demonstrate the superiority of our proposed framework, we conduct two groups of experiments on 10 datasets. In the first group of experiments, EDAE[1] is compared with three ad-

[1] The experimental code is made available at https://github.com/min-seong-kwon/EDAE.

Table 1
Dataset description. D is the dimension of the data, N is the number of samples, and f_1 denotes the percentage of labeled anomalies in the training datasets.

Dataset	D	N	f_1	Description
abalone	7	4,177	2.49%	predicting the age of abalone from physical measurements
annthyroid	6	7,200	0.37%	detecting thyroid dysfunction from various patient characteristics
Cardiotocography	21	2,114	1.1%	detecting abnormal fetal heart rate
cover	10	286,048	0.05%	predicting the forest cover type from various cartographic variables
Hepatitis	19	80	1.79%	predicting the outcome (live or die) of patients with hepatitis
Ionosphere	32	351	1.79%	classifying “good” or “bad” based on radar signal
landsat	36	6,435	1.04%	sensory data consisting of multispectral satellite imagery of the Earth's surface
magic.gamma	10	19,020	1.76%	measurement of high-energy gamma particles interacting with gamma-ray telescopes
WDBC	30	367	0.28%	diagnosing breast cancer based on breast tissue images
Wilt	5	4,819	0.27%	predicting the presence of Wilt disease in tomatoes

Table 2
Experimental results (mean ± standard deviation) of EDAE and other three competitive methods. The best performance is boldfaced.

Dataset	AUC-ROC Performance				AUC-PR Performance			
	Proposed	DevNet	FEAWAD	AE	Proposed	DevNet	FEAWAD	AE
abalone	**0.821**±0.005	0.810±0.001	0.782±0.028	0.609±0.041	**0.819**±0.007	0.804±0.002	0.759±0.032	0.643±0.045
annthyroid	0.951±0.036	0.823±0.006	**0.958**±0.022	0.622±0.009	**0.753**±0.103	0.475±0.010	0.683±0.063	0.143±0.004
Cardiotocography	**0.929**±0.009	0.927±0.002	0.818±0.041	0.723±0.003	0.796±0.022	**0.801**±0.005	0.689±0.025	0.469±0.015
cover	**0.999**±0.001	0.997±0.001	0.980±0.014	0.859±0.051	0.898±0.072	**0.913**±0.021	0.761±0.192	0.054±0.019
Hepatitis	**0.888**±0.012	0.757±0.060	0.515±0.057	0.712±0.034	**0.647**±0.043	0.492±0.055	0.348±0.035	0.306±0.023
Ionosphere	**0.893**±0.025	0.779± 0.008	0.661±0.033	0.862±0.031	**0.869**±0.025	0.797±0.008	0.670±0.031	0.859±0.014
landsat	**0.791**±0.026	0.761± 0.021	0.780±0.021	0.326±0.015	**0.506**±0.105	0.401±0.051	0.470±0.016	0.146±0.003
magic.gamma	**0.838**±0.004	0.828±0.001	0.804±0.009	0.720±0.011	**0.764**±0.025	0.742±0.001	0.727±0.015	0.653±0.006
WDBC	**0.992**±0.002	0.797±0.071	0.684±0.083	0.970±0.003	**0.840**±0.049	0.284±0.099	0.319±0.028	0.491±0.012
Wilt	**0.901**±0.006	0.678±0.001	0.761±0.121	0.272±0.003	**0.351**±0.035	0.082±0.001	0.333±0.282	0.034±0.002
Average	**0.900**±0.013	0.816±0.017	0.774±0.043	0.668±0.020	**0.724**±0.049	0.579±0.025	0.576±0.072	0.380±0.014

vanced deep AD algorithms in recent years, including DevNet [15], FEAWAD [16], and unsupervised AE. DevNet and FEAWAD are chosen because they are state-of-the-art models for weakly supervised learning. The unsupervised AE is chosen to show how much greater performance gain can be obtained by an additional process. For all the competing algorithms in our experiments, we use their original default hyperparameter settings for a fair comparison.

In the second group of experiments, we conduct an ablation study to explore the impact of the proposed EDL. Specifically, we compare our proposed model, which follows the EDAE structure and utilizes EDL, with a model that utilizes deviation loss (5) instead of EDL, denoted as EDAE+DL. To ensure a fair comparison, we set the confidence margin parameter of deviation loss to the threshold obtained from the empirical distribution.

For both experiments, EDAE applies the same fully connected AE structure for pre-training AE and main-training AE: a two-layer encoder, latent layer, and two-layer decoder. Dropout is applied to each layer to prevent overfitting. We use leaky ReLU as the activation function for each layer. Adam (learning rate:0.0001) is used as the optimizer, and the batch size is set to 64. The pre-epoch and epoch values are set to 50. λ and γ in (9) are set to 13 and 2, respectively. We evaluate performance using AUC-ROC and AUC-PR metrics, and the results are presented in Tables 2 and 3, respectively.

5.4. Experimental results

As shown in Table 2, the proposed method performed best on nine and eight datasets in the respective AUC-ROC and AUC-PR performance. In addition, although the proposed method does not achieve the best results for annothyroid, cardiotocography, and cover, the gap between the best methods and EDAE is relatively small. In terms of AUC-ROC, EDAE outperformed DevNet (10.3%), FEAWAD (16.3%), and unsupervised AE (34.8%) with a substantially better average improvement. In terms of AUC-PR, EDAE outperformed DevNet (25.0%), FEAWAD (25.7%), and unsupervised AE (90.5%) with a substantially better average improvement. Compared with the unsupervised method, the proposed method outperformed the unsupervised method on all datasets in both AUC-ROC and AUC-PR. In particular, for the WDBC dataset, the AUC-PR performance increased by 71% compared with the unsupervised AE, even though only one known outlier was given. In addition, compared with two weakly supervised methods, DevNet and FEAWAD, our method shows substantially higher detection performance in most datasets. Table 3 shows that our proposed model achieves a significant improvement compared to the model using deviation loss by 9.0%(average) and 28.1%(average) in AUC-ROC and AUC-PR, respectively.

The superiority of our proposed method is attributed to two factors. First, the proposed EDAE enables more efficient training by the EDL, giving more penalties to abnormal data with lower anomaly scores than the conventional deviation loss. In addition, the proposed EDL imposes a loss even if the anomaly score of the data is greater than the threshold, which makes better use of the anomalies. However, the conventional deviation loss imposes zero loss for abnormal data with an anomaly score greater than the confidence margin. Hence, a substantial number of anomalies did not affect the training.

Second, EDAE employs the inverse transform sampling technique based on the empirical distribution when constructing mini-batches. This approach reduces the risk of training with contaminated data. Additionally, the anomaly threshold is determined using the estimated anomaly score distribution. Hence, the proposed method exhibits improved performance compared

Table 3
Experimental results (mean ± standard deviation) to investigate the efficiency of the loss function. The best performance is boldfaced.

Dataset	AUC-ROC Performance		AUC-PR Performance	
	EDAE+EDL	EDAE+DL	EDAE+EDL	EDAE+DL
abalone	**0.821**±0.005	0.725±0.011	**0.819**±0.007	0.740±0.013
annthyroid	**0.951**±0.036	0.765±0.065	**0.753**±0.103	0.350±0.099
Cardiotocography	**0.929**±0.009	0.878±0.009	**0.796**±0.022	0.682±0.023
cover	**0.999**±0.001	0.985±0.001	**0.898**±0.072	0.338±0.008
Hepatitis	**0.888**±0.012	0.858±0.082	**0.647**±0.043	0.641±0.126
Ionosphere	0.893±0.025	**0.968**±0.006	0.869±0.025	**0.950**±0.008
landsat	**0.791**±0.026	0.741±0.025	**0.506**±0.105	0.377±0.039
magic.gamma	**0.838**±0.004	0.822±0.008	**0.822**±0.025	0.764±0.001
WDBC	**0.992**±0.002	0.985±0.002	**0.840**±0.049	0.736±0.035
Wilt	**0.901**±0.006	0.536±0.178	**0.351**±0.035	0.073±0.061
Average	**0.900**±0.013	0.826±0.039	0.724±0.049	0.565±0.041
XSXS				

to conventional methods by leveraging more precise prior knowledge.

6. Conclusion

In this paper, we proposed a novel framework for weakly supervised anomaly detection called EDAE that addresses the limitations of previous methods. Firstly, EDAE utilizes prior knowledge gained from the empirical distribution of anomaly scores, rather than relying on assumptions that anomaly scores follow a normal distribution. This approach guarantees the selection of an appropriate anomaly threshold across a wide range of datasets. Secondly, EDL, a novel loss function employed in EDAE, imposes an exponentially increasing penalty on anomalous data with low anomaly scores, whereas the deviation loss merely imposes a linear penalty. This approach enables more efficient use of limited labeled abnormal data compared to traditional loss functions. Thirdly, we propose a batch sampling method based on the empirical distribution that creates batches of data that are more robust to contaminated data. The experimental results show that the proposed method improves the detection accuracy compared to unsupervised AE and other advanced methods (i.e., DevNet and FEAWAD), even with very few anomalies. Additionally, an ablation study shows the superior efficiency of EDL compared to the conventional loss function. In future work, we plan to explore the robustness of EDAE with respect to different anomaly contamination levels in the unlabeled training data. We also plan to apply our framework to other domains such as image or video anomaly detection.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research was supported by Kumoh National Institute of Technology (2021).

References

[1] G. Li, J.J. Jung, Dynamic relationship identification for abnormality detection on financial time series, Pattern Recognit. Lett. 145 (2021) 194–199.
[2] S. Yan, H. Shao, Y. Xiao, B. Liu, J. Wan, Hybrid robust convolutional autoencoder for unsupervised anomaly detection of machine tools under noises, in: Robot. Comput.-Integr. Manuf., Vol. 79, 2023, p. 102441.
[3] C. Zhou, R.C. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proc. 23rd ACM SIGKDD Intl. Conf. on Knowl. Discov. & Data Min., 2017, pp. 665–674.
[4] J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability, in: Special Lecture on IE, Vol. 2, 2015, pp. 1–18.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, in: Commun. ACM, Vol. 63, 2020, pp. 139–144.
[6] C. Huang, J. Wen, Y. Xu, Q. Jiang, J. Yang, Y. Wang, D. Zhang, Self-supervised attentive generative adversarial networks for video anomaly detection, IEEE Trans. Neural. Netw. Learn. Syst. (2022).
[7] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: Proc. 35st Int. Conf. Mach. Learn., 2018, pp. 4393–4402.
[8] S. Son, S. Nah, K.M. Lee, Clustering convolutional kernels to compress deep neural networks, in: Proc. European Conference on Computer Vision (ECCV), 2018, pp. 216–232.
[9] L.-T. Li, Z.-Y. Xiong, Q.-Z. Dai, Y.-F. Zha, Y.-F. Zhang, J.P. Dan, A novel graph-based clustering method using noise cutting, Inf. Syst. 91 (2020) 101504.
[10] C. Huang, Z. Yang, J. Wen, Y. Xu, Q. Jiang, J. Yang, Y. Wang, Self-supervision-augmented deep autoencoder for unsupervised visual anomaly detection, IEEE Trans. Cybern. 52 (2021) 13834–13847.
[11] H. Zhu, S. Liu, F. Jiang, Adversarial training of LSTM-ED based anomaly detection for complex time-series in cyber-physical-social systems, Pattern Recognit. Lett. 164 (2022) 132–139.
[12] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, in: ACM Comput. Surv., Vol. 41, 2009, pp. 1–58.
[13] Y. Zhao, M.K. Hryniewicki, XGBOD: improving supervised outlier detection with unsupervised representation learning, in: Intl. Joint Conf. on Neural Netw., 2018.
[14] L. Ruff, R.A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, in: Proc. Int. Conf. Learn. Repr. (ICLR), 2019.
[15] G. Pang, C. Shen, A. van den Hengel, Deep anomaly detection with deviation networks, in: Proc. of the 25th ACM SIGKDD Intl. Conf. on Knowl. Discov. & Data Min., 2019, pp. 353–362.
[16] Y. Zhou, X. Song, Y. Zhang, F. Liu, C. Zhu, L. Liu, Feature encoding with autoencoders for weakly supervised anomaly detection, IEEE Trans. Neural Netw. Learn. Syst. 33 (6) (2022) 2454–2465.
[17] A. Boukerche, L. Zheng, O. Alfandi, Outlier detection: methods, models, and classification, ACM Comput. Surv. 53 (3) (2020) 1–37.
[18] E.A. Elaziz, R. Fathalla, M. Shaheen, Deep reinforcement learning for data-efficient weakly supervised business process anomaly detection, J. Big Data 10 (2023) 33.
[19] H. Li, Y. Iwamoto, X. Han, L. Lin, R. Tong, H. Hu, A. Furukawa, S. Kanasaki, Y.W. Chen, A weakly-supervised anomaly detection method via adversarial training for medical images, in: Proc. IEEE Int. Conf. Consum. Electron. (ICCE), 2022.
[20] Y. Tian, G. Pang, Y. Chen, R. Singh, J.W. Verjans, G. Carneiro, Weakly-supervised video anomaly detection with robust temporal feature magnitude learning, in: Proc. the IEEE/CVF Int. Conf. on Computer Vision, 2021, pp. 4975–4986.
[21] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2006, pp. 1735–1742.
[22] S. Han, X. Hu, H. Huang, M. Jiang, Y. Zhao, ADBench: anomaly detection benchmark, ArXiv: arXiv:2206.09426.
[23] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proc. 23rd Int. Conf. Mach. Learn., 2006, pp. 233–240.