Semi-supervised Anomaly Detection Algorithm Based on KL Divergence (SAD-KL)

Chong Hyun Lee
Department of Ocean System Engineering, Jeju National University
chonglee@jejunu.ac.kr

Kibae Lee
Department of Ocean System Engineering, Jeju National University
kibae0211@stu.jejunu.ac.kr

Abstract
The unlabeled data are generally assumed to be normal data in detecting abnormal data via semi-supervised learning. This assumption, however, causes inevitable detection error when distribution of unlabeled data is different from distribution of labeled normal dataset. To deal the problem caused by distribution gap between labeled and unlabeled data, we propose a semi-supervised anomaly detection algorithm using KL divergence (SAD-KL). The proposed SAD-KL is composed of two steps: (1) estimating KL divergence of probability density functions (PDFs) of the local outlier factors (LOFs) of the labeled normal data and the unlabeled data (2) estimating detection probability and threshold for detecting normal data in unlabeled data by using the KL divergence. We show that the PDFs of the LOFs follow Burr distribution and use them for detection. Once the threshold is computed, the SAD-KL runs iteratively until the labeling change rate is lower than the predefined threshold. Experiments results show that the SAD-KL shows superior detection probability over the existing algorithms even though it takes less learning time.

1 Introduction
The purpose of anomaly detection is to detect abnormal samples that deviate from the predefined normality [1]. Anomaly detection has various applications in medicine, security, and manufacturing [1]. The unsupervised anomaly detection algorithms assume that most of the samples are normal and learn features of normal samples [2, 3, 4, 5, 6]. Anomaly detection algorithms named as the one class support vector machine (OC-SVM) [2] and support vector data description (SVDD) [3] are reported. The deep learning approaches have shown outstanding performance by successively learning high-dimensional data [4, 5, 6]. The Deep SVDD [4], Deep Multi-sphere SVDD [5] and deep robust one-class classification (DROCC) [6] have been proposed to learn representation of normal samples. In spite of improved performance, these unsupervised approaches have limited detection performance since they do not train abnormal information in dataset.

Some labeled data, as well as unlabeled data, may be utilized and especially small number of anomalous samples in labeled data can be used for training. Song et al. [7] and Akvay et al. [8] proposed semi-supervised anomaly detection models which use reliable normal samples in training of unlabeled data. Nevertheless, they have limited performance because these models do not train the abnormalities yet. Ruff et al. [9] proposed a deep semi-supervised anomaly detection (Deep SAD) that learns anomalous samples in labeled data by assuming that most unlabeled samples are normal. The Deep SAD moves the normal data to the center of the latent space and labeled abnormal samples away from the center. An unsupervised outlier exposure (OE) approach for learning OE data and unlabeled data has been proposed by using similar assumption of semi-supervised anomaly detection [10]. The algorithm based on unsupervised OE learning trains a binary classifier by using OE data, the unlabeled abnormal and normal data. The hyper sphere classification (HSC) [11] algorithm based on unsupervised OE learning, uses the relative distance in latent space for training. Even the Deep SAD and HSC are advanced algorithms using abnormal samples or OE data in the training process, they cannot overcome performance degradation as the number of abnormal samples in the unlabeled dataset increases. To overcome the degradation, semi-supervised algorithm that can use both normal and abnormal data in unlabeled dataset, algorithm using probabilistic labeling (SAD-PL) is presented [12]. The SAD-PL shows superior...
to minimize the volume of the hypersphere forces the network to map the data to the center in latent space. The most representative kernel-based method for one class classification is the OC-SVM [2]. The objective of the OC-SVM finds a maximum margin hyperplane in feature space. On the other hand, the SVDD [3] attempts to separate data into hypersphere instead of hyperplane. The objective of SVDD is to find smallest hypersphere that encloses the data in feature space. Classical anomaly detection methods including OC-SVM and SVDD are limited in their scalability to large datasets. Recent deep learning approaches have shown outstanding performance by overcoming the problems which cannot solve by classical algorithms. The Deep SVDD [4], a representative deep approach, trains a neural network by minimizing the volume of a hypersphere that encloses the network representations of the data. This minimizing of the volume of the hypersphere forces the network to map the data to the center in latent space. For learning “normality” of complex data, deep one class classification algorithms such as Deep SVDD [4], Deep Multi-sphere SVDD [5] and DROCC [6], have been reported.

2 Related work

One Class Classification Typically, anomaly detection task is treated as an unsupervised learning problem assuming that most training data are normal. The one class classification aims to learn a model that accurately describes “normality”. The most representative kernel-based method for one class classification is the OC-SVM [2]. The objective of the OC-SVM finds a maximum margin hyperplane in feature space. On the other hand, the SVDD [3] attempts to separate data into hypersphere instead of hyperplane. The objective of SVDD is to find smallest hypersphere that encloses the data in feature space. Classical anomaly detection methods including OC-SVM and SVDD are limited in their scalability to large datasets. Recent deep learning approaches have shown outstanding performance by overcoming the problems which cannot solve by classical algorithms. The Deep SVDD [4], a representative deep approach, trains a neural network by minimizing the volume of a hypersphere that encloses the network representations of the data. This minimizing of the volume of the hypersphere forces the network to map the data to the center in latent space. For learning “normality” of complex data, deep one class classification algorithms such as Deep SVDD [4], Deep Multi-sphere SVDD [5] and DROCC [6], have been reported.

Semi-supervised Anomaly Detection based on deterministic labeling Unlike the assumption of the typical anomaly detection task, some labeled data includes a small fraction of anomalous samples that can be used in real-world applications. The hybrid SAD [7] and GANomaly [8] learn reliable normal samples in unlabeled data as well as labeled data. However, these methods have limitations in detecting anomalies since they do not learn the anomalous samples. The Deep SAD [9] related to Deep SVDD is proposed for semi-supervised anomaly detection. By assuming that the majority of the unlabeled data consists of normal samples, the Deep SAD learns by forcing the normal data to be concentrated and anomalous samples in labeled data to move away from the center of the latent space. The HSC [11] is a classification algorithm based on unsupervised OE learning that uses the relative distance in latent space for training the model. The HSC can be modified as semi-supervised anomaly detection method by replacing OE data with labeled data. Semi-supervised anomaly detection approaches including Deep SAD and HSC find a compact description of the normality while also discriminating the labeled anomalies. However, the assumption of most unlabeled samples being normal inevitably causes performance degradation as the number of abnormal samples in the unlabeled dataset increases.

Semi-supervised Anomaly Detection based on probabilistic labeling The positive and unlabeled metric learning for anomaly detection (PUMAD) [13] learns reliable normal samples and potential anomalies among the unlabeled data using triplet loss. This algorithm identifies normal and abnormal samples in unlabeled data with distance hashing-based method by assuming that anomalies are clustered in latent space. The log-based anomaly detection via probabilistic label estimation (PLELog) [14] learns unlabeled data with the stability score of each cluster via hierarchical density-based spatial clustering of application with noise (HDBSCAN) [15]. These methods, however, are only applicable to group or collective anomalies. The SAD-PL [12] can handle both group and collective anomalies by using LOF score and learning unlabeled data with estimated probabilistic labels obtained by the NP criterion. However, the SAD-PL employs ensemble networks to find the best threshold satisfying the give detection probability so that it requires heavy computation. In addition, because the SAD-PL estimates the threshold by using labeled normal data, its detection performance may degrade when distribution of unlabeled data is different from distribution of labeled normal dataset.

3 SAD-KL

In this section, we introduce the SAD-KL along with the existing semi-supervised algorithms of SAD-PL as shown in Figure 1. The proposed SAD-KL uses feature representations \( \phi(x) \) through autoencoder pretraining and learns the encoded normal samples to close centroid \( c \) in latent space without the decoding network. Then, the SAD-KL is trained according to the estimated KL divergence, which uses the PDF of \( s_c \) from labeled normal and the PDF of \( s_u \) from unlabeled data among \( s = \text{LOF}_k(\phi(x)) \). The LOF scoring based on the relative distance between \( k \) neighbor samples is known to show robust performance in the multimodal normality case [16]. As shown in Figure 1, the \( s \) of the object \( p \)
is computed as follows: (1) computing k-distances of p with the k neighbor samples $o_i$ (2) computing k-distances of $o_i$ with the k neighbor samples (3) computing ratio of average of the k-distances obtained by (2) to the k-distances by (1).

To detect a small number of group anomaly samples, the SAD-KL sets k large enough to cover the relative distance between normal and abnormal samples.

The NP criterion is used to determine the threshold that satisfies the detection probability under a given constraint [17]. The threshold $\eta$ is computed by using the detection probability $P_D$ as

$$P_D = \int_{0}^{\eta} p(s|H_0)ds \quad (1)$$

where $p(s|H_0)$ denotes the PDF of s obtained from data under condition $H_0$. Note that the $P_D$ depends on changes of threshold $\eta$. Therefore, when the distribution of unlabeled data is almost the same as the distribution of labeled normal data, threshold $\eta$ obtained from labeled normal data can gives almost the same detection probability on the unlabeled data. However, if unlabeled data contains diverse normal samples so that it has different distribution, then same detection probability $P_D$ cannot be guaranteed with the threshold $\eta$ pre-obtained from the labeled data. To consider the distribution difference in computing $P_D$, we use KL divergence and adopt exponential function mapping the obtained KL divergence to probability value. Consequently, a detection probability formula for the unlabeled data is proposed as follows:

$$P_D = \exp\left(\frac{KL(P||Q)}{\beta}\right) \quad (2)$$

where $KL(P||Q) = \int_{-\infty}^{\infty} p(s|Q) \log \frac{p(s|P)}{q(s)}$ denotes the KL divergence of PDF $p(s)$ of labeled normal data and $q(s)$ of unlabeled data and the $\beta$ is a parameter to be determined. Since the $KL(P||Q)$ represents difference between the labeled and the unlabeled data distributions, the detection probability of normal data in unlabeled data can be estimated by proper choice of $\beta$. Once appropriate $\beta$ is chosen, the threshold $\eta$ can be found by setting (1) equals to (2).

As shown in the following simulations, the $\beta$ is not insensitive to dataset so that sole choice of $\beta = 2500$ is sufficient to have the best detection performance on three datasets. Furthermore, the PDF of s can be assumed as Burr distribution so that we can use the following PDF $p(s)$ and CDF $F(s)$ [18, 19]:

$$p(s) = \frac{kcs^{-c-1}}{(1 + s)^{-k}} \quad (3)$$
$$F(s) = 1 - (1 + s)^{-k} \quad (4)$$

where $c$ and $k$ are shape parameters that can cover a broad set of skewness and kurtosis [18, 19]. The threshold $\eta$ can be computed by finding s satisfying $P_D$ in (2) as follows:

$$\eta = \left((1 - P_D)^{-\frac{1}{k}} - 1\right)^{\frac{1}{c}} \quad (5)$$
Note that $\eta$ changes because the PDF of $s$ varies as SAD-KL learns. Once $\eta$ is estimated by using equation (2), the label $y(t)$ at $t$ can be found as follows in [12]. The labels $y_l(t), t \geq n$ for $n$ labeled samples $x_l$ and the labels $y_u(t), u > n$ for unlabeled data $x_u$, are as follows:

$$y_l(t) = \begin{cases} 
1, & \forall t \text{ for normal} \\
0, & \forall t \text{ for abnormal}
\end{cases} \quad (6)$$

$$y_u(t) = \begin{cases} 
P_D, & s_u \leq \eta \\
1 - P_D, & s_u > \eta
\end{cases} \quad (7)$$

where $s_u$ represents the LOF score of $x_u$. Note that $y_u(t)$ implies the probabilistic label for the unlabeled $x_u$. For network $\phi$ training, the SAD-KL uses datasets $x_l, y_l(t), l \leq n$ and $x_u, y_u(t), u > n$, which are composed of $n$ labeled and $(m - n)$ unlabeled data from the corresponding probabilistic labels estimated in (6) and (7). The SAD-KL use the same objective function of the SAD-PL described as follows:

$$\min_{W} \frac{1}{m} \sum_{i=1}^{m} y_l(t) d(\phi(x_i)) + (1 - y_l(t))(1 - d(\phi(x_i))) + \lambda J \sum_{j=1}^{J} \|W_j\|_F \quad (8)$$
where $W^j$ is the weights of layer $j \in \{1, \cdots, J\}$, $\| \cdot \|_F$ denotes the Frobenius norm and $d(\phi(x_i))$ is the function of the distance from $c$ using Geman-McClure loss defined as

$$d(\phi(x_i)) = \frac{\|\phi(x_i) - c\|^2}{\|\phi(x_i) - c\|^2 + 1} \quad (9)$$

The properties of the proposed SAD-KL follow those of SAD-PL [12] and runs until the labeling change rate $\Delta y_u(t)$ at iteration $t$ and is lower than the preset threshold $\epsilon$. The $\Delta y_u(t)$ is computed as

$$\Delta y_u(t) = \frac{1}{m-n} \sum_{u=n+1}^{m} \frac{|y_u(t) - y_u(t-1)|}{2P_D - 1} \quad (10)$$

We compare the anomaly detection models described above with two moons dataset. The training data consist of a total of 10000 samples, of which 10% are labeled data and 90% are unlabeled data. Five percent of labeled data and 1% of unlabeled data consist of abnormal samples. The test data consists of 1000 normal and abnormal samples. The normal samples contain Gaussian noise with a variance of 0.3 in big moon and small moon patterns. The abnormal samples are generated with a uniform distribution. The models for comparison use the same encoding network of the pretrained autoencoder. The encoding network consists of two hidden layers of 100 nodes, followed by ELU activations that represent a two-dimensional input in a two-dimensional latent space. For autoencoder pretraining, we employ the above architectures for the encoding networks and then construct decoding networks symmetrically. We set $\zeta = 1$ for the Deep SAD, and $k = 100$ and $\epsilon = 0.0001$ for the SAD-PL and the SAD-KL. For training of SAD-PL with soft labeling, we use $Q = 20$ by setting $P_M = 0$ to $P_M = 0.1$ with an $\delta = 0.05$ interval. To determine $P_D$ in SAD-KL learning, we set $\beta = 500$. We set $\lambda = 10^{-6}$ and use the Adam optimizer with a learning rate of $10^{-5}$ and train the Deep SAD with 200 epochs.

We plot the histogram, the estimated PDF and the CDF of $s_u$ of unlabeled data in Figure 2, which validates the Burr distribution assumption. Figure 3 shows the decision boundaries with training data and the test AUC (area under the receiver operating characteristic curve) of anomaly detection models. The decision boundaries in Figure 3 are expressed by an upper bound on 10% of the anomaly score normalized via min-max scaling. The anomaly score is computed $\|\phi(x) - c\|^2$ for all models. The training times normalized by SAD-KL is shown in Figure 4. As shown in Figure 3, the
SAD-KL and SAD-PL represent the tighter decision boundaries than the Deep SAD which assumes that all unlabeled data are normal. Even the SAD-KL has tight decision boundaries and high AUC of 98.82%, it takes the least training time as shown in Figure 4. Figure 5 shows the training AUCs and the label change rate $\Delta y_u(t)$ of the comparative models according to epoch. As shown in Figure 5, the proposed SAD-KL shows high learning efficiency and faster convergence of $\Delta y_u(t)$ due to KL divergence based efficient learning whereas the Deep SAD shows slower learning.

### 4 Experiments

We evaluate the proposed SAD-KL on the well-known MNIST [20], CIFAR10 [21], and MNIST-C [22] datasets. The SAD-KL is compared with the methods based on one class classification. We present results from unsupervised methods of SVDD [3] and the Deep SVDD [4] and semi-supervised methods of the Deep SAD [9], HSC [11] and SAD-PL [12]. We implement the semi-supervised HSC by replacing OE data with the labeled data. We run all experiments for $\nu \in \{0.1, 0.25, 0.5\}$ of SVDD with a Gaussian kernel and show the corresponding results. The deep models use the same encoding network structure in the pretrained autoencoder. We employ LeNet-type convolutional neural networks (CNNs), where each convolutional module consists of a convolutional layer followed by leaky ReLU activations and 2×2 max-pooling. In the MNIST and MNIST-C experiments, we employ a CNN with two modules, 8x(5x5) filters, and a final dense layer of 32 units. In the CIFAR10 experiments, we employ a CNN with three modules, 32x(5x5) filters, 64x(5x5) filters, and 128x(5x5) filters, followed by a final dense layer of 128 units. For the pretraining autoencoder, we employ identical encoding networks and then construct the decoding networks symmetrically, where we replace max-pooling with simple upsampling and convolutions with deconvolutions. We use a batch size of 200 and set $\lambda = 10^{-6}$. We also use the Adam optimizer with a learning rate of $10^{-5}$. For experiments on the Deep SVDD, we use the one-class Deep SVDD model [4]. We run all experiments for $\zeta \in \{0.01, 0.1, 1, 0.10\}$ of the Deep SAD and show the best results. We set $k = 200$ and stop learning when $\Delta y_u(t)$ is less than $\epsilon = 0.001$ for both SAD-KL and SAD-PL. The SAD-PL is evaluated via two separate hard and soft labels. In soft labeling, we set $Q = 101$ by setting $P_M = 0$ to $P_M = 0.2$ with an $\delta = 0.002$ interval and select the $P_M(o)$ in which $\Delta s$ is maximum in $T = 10$. For SAD-KL learning, we use the same $\beta = 2500$ for all datasets and train the deep models for 300 epochs. We use a typical one vs. rest evaluation method on the MNIST and CIFAR10 datasets [23]. On MNIST and CIFAR 10, we set the ten classes to be normal classes and let the remaining nine classes represent anomalies. We use the original training and test data. In the training data, we constitute most of the data as the normal class and replace a small amount with the data from the abnormal class according to the experimental scenario. We also divide the training data into labeled and unlabeled data, while organizing the abnormal samples in labeled data from a single anomalous class. However, the abnormal samples in unlabeled data equally contain samples of all anomalous classes. The class for abnormal samples in labeled data is randomly determined in each experiment. This gives training set sizes of approximately 6000 for MNIST and 5000 for CIFAR10. Both test sets have 10000 samples, including samples from the nine anomalous classes for each setup. On MNIST-C, we set original images to be normal and corrupted images to be abnormal. We use pre-configured training and test data. We also make the training data into most normal samples and a few abnormal samples according to the scenario. The training data are divided into labeled and unlabeled data. We organize the abnormal samples in labeled data using corrupted images of the same type. The abnormal samples in unlabeled data include all kinds of corrupted images equally. The type of corrupted images for abnormal samples in labeled data is randomly determined in each experiment. This configuration gives a training set size of approximately 60000 and a test set size of 160000 for MNIST-C. We use 5% as labeled data and 95% as unlabeled data in the training data as Ruff et al. [9] do for evaluation of the Deep SAD. We constituted 98% normal samples and 2% abnormal samples for our evaluation.

| Normal class | Unsupervised learning | Semi-supervised learning | Training time (s) |
|--------------|-----------------------|--------------------------|-------------------|
|              | SVDD                  | Deep SVDD                | HSC              | SAD-PL (Hard) | SAD-PL (Soft) | SAD-KL | Average | 10,000 | 20,000 | 60,000 | 160,000 |
| 0            | 89.91                 | 90.38                    | 91.84            | 93.59        | 95.22        | 97.29   | 97.94   | 12.62  | 36.45  | 37.52  | 35.48   | 32.62  | 776.45 | 25.04   |
| 1            | 95.39                 | 98.30                    | 99.73            | 99.57        | 98.58        | 99.70   | 99.42   | 77.08  | 79.95  | 80.51  | 83.06   | 86.72  | 88.77  | 88.82   |
| 2            | 77.08                 | 79.95                    | 80.51            | 83.06        | 86.72        | 88.77   | 88.82   | 31.64  | 83.10  | 84.34  | 84.49   | 89.64  | 91.72  | 91.40   |
| 3            | 89.09                 | 91.49                    | 92.26            | 91.73        | 94.34        | 96.43   | 96.82   | 89.09  | 91.49  | 92.26  | 91.73   | 94.34  | 96.43  | 96.82   |
| 4            | 80.96                 | 81.38                    | 82.31            | 86.72        | 91.09        | 93.18   | 92.55   | 80.96  | 81.38  | 82.31  | 86.72   | 91.09  | 93.18  | 92.55   |
| 5            | 91.50                 | 92.77                    | 93.82            | 96.07        | 96.48        | 98.53   | 96.75   | 91.50  | 92.77  | 93.82  | 96.07   | 96.48  | 98.53  | 96.75   |
| 6            | 88.36                 | 91.10                    | 92.55            | 97.28        | 96.61        | 98.62   | 97.36   | 88.36  | 91.10  | 92.55  | 97.28   | 96.61  | 98.62  | 97.36   |
| 7            | 81.74                 | 84.12                    | 85.04            | 77.23        | 92.45        | 94.50   | 94.68   | 81.74  | 84.12  | 85.04  | 77.23   | 92.45  | 94.50  | 94.68   |
| 8            | 87.36                 | 90.24                    | 91.73            | 88.59        | 95.32        | 97.42   | 95.45   | 87.36  | 90.24  | 91.73  | 88.59   | 95.32  | 97.42  | 95.45   |
| 9            | 86.30                 | 88.28                    | 89.41            | 89.83        | 93.65        | 95.62   | 95.12   | 86.30  | 88.28  | 89.41  | 89.83   | 93.65  | 95.62  | 95.12   |

Table 1: Average AUCs in % and training time for one vs. rest evaluation on MNIST dataset.
Table 2: Average AUCs in % and training time for one vs. rest evaluation on CIFAR10 dataset.

| Normal class | Normal class | Unsupervised learning | Deep SVDD | Deep SAD | HSC | Semi-supervised learning | SAD-PL (Hard) | SAD-PL (Soft) | SAD-KL |
|--------------|--------------|-----------------------|-----------|----------|-----|--------------------------|---------------|---------------|--------|
| Airplane     | 59.82        | 62.56                 | 63.64     | 67.98    | 73.70 | 78.92                    | 79.21         |
| Automobile   | 48.48        | 50.61                 | 52.09     | 42.81    | 52.21 | 57.29                    | 64.77         |
| Bird         | 61.28        | 62.63                 | 64.41     | 64.39    | 58.41 | 63.82                    | 64.62         |
| Cat          | 50.39        | 52.76                 | 54.31     | 59.48    | 59.31 | 64.52                    | 66.39         |
| Deer         | 64.73        | 65.96                 | 68.37     | 72.89    | 77.35 | 82.64                    | 77.83         |
| Dog          | 58.57        | 60.59                 | 61.67     | 58.26    | 58.29 | 63.47                    | 61.79         |
| Frog         | 67.04        | 69.20                 | 70.32     | 68.31    | 71.68 | 76.89                    | 73.38         |
| Horse        | 54.63        | 56.72                 | 58.18     | 55.48    | 60.72 | 65.98                    | 63.85         |
| Ship         | 64.33        | 66.89                 | 68.29     | 76.78    | 69.35 | 74.52                    | 73.87         |
| Truck        | 53.79        | 56.28                 | 57.76     | 60.40    | 62.69 | 67.87                    | 73.47         |
| Average      | **58.30**    | **60.42**             | **61.90** | **62.68**| **64.37**| **69.59**               | **69.91**     |
| Training time (s) | 70.27 | 186.45 | 188.52 | 184.48 | 179.67 | 1436.67 | 167.97 |

Table 3: Average AUCs in % and training time for evaluation on MNIST-C dataset.

| Model Learning | AUC  | Training time (s) |
|----------------|------|--------------------|
| Unsupervised learning |     |                    |
| SVDD            | 67.59| 2970               |
| Deep SVDD       | 82.79| 7927               |
| Deep SAD        | 89.89| 7945               |
| Semi-supervised learning | | |
| HSC             | 85.10| 7894               |
| SAD-PL (Hard)   | 90.05| 7562               |
| SAD-PL (Soft)   | 92.43| 83999              |
| SAD-KL          | 95.26| 6192               |

Figure 6: The ROC curve obtained from MNIST-C dataset.

in the labeled data, along with 99% normal samples and 1% abnormal samples in the unlabeled data. We present the evaluation results for the models with an average AUC of 30 times.

Table 1 shows the evaluation results using MNIST dataset. The SAD-KL achieves a better performance than the unsupervised and semi-supervised anomaly detection models except for SAD-PL(Soft) with an average AUC of 95.12%. However, note that SAD-KL has 31 times faster training with little performance degradation of 0.50% and is at least 7.25sec faster than other deep anomaly detection models. Tables 2 shows the evaluation results using CIFAR10 dataset. Similarly, the SAD-KL shows better performance than existing anomaly detection models with an average AUC of 69.91% and fastest training time. Tables 3 shows anomaly detection evaluation results using MNIST-C dataset. The SAD-KL achieves the best performance over all unsupervised and semi-supervised models with an average AUC of 95.26% and faster training time than the semi-supervised models.

Figure 6 shows the receiver operation characteristic (ROC) curve obtained from MNIST-C dataset which contains anomalous images of multimodal normality. Note that the SAD-KL shows the best ROC performance even the unlabeled data has diverse statistics. We present anomaly scores of normal and abnormal images obtained from 4 anomaly detection models in Figure 7. We can observe that anomaly scores of the SAD-KL are the biggest with abnormal image and compatible to the SAD-PL(soft) with normal image.

Next, we investigate the effect of including labeled data during training on the MNIST-C datasets by increasing the ratio of labeled training data from 0% to 10% and presenting the averaged AUCs of 30 times. The ratio of abnormal samples in labeled and unlabeled data is maintained at 5% and 1%, as in the previous experiments. Figure 8 (a) shows variations of performance in the average AUCs for the semi-supervised models according to the ratio of labeled data during training. Note that the SAD-KL achieve the best AUC performance even with a high proportion of labeled data during training. Similarly, we investigate the effect of including abnormal samples in the unlabeled data during the training with the MNIST-C datasets. To do this, we increase the anomaly ratio, which is the proportion of abnormal samples in unlabeled training data, from 0% to 10% and represent the average AUCs of 30 times. For the experiment, we keep the labeled data at 95% of the training data and include 98% and 2% of normal and abnormal samples, respectively. Figure 8 (b) shows the variations of performance for the average AUCs according to the anomaly ratio in unlabeled data. Note than existing unsupervised and semi-supervised algorithms show performance degradation in the average AUCs as the...
anomaly ratio in unlabeled data increases. However, the SAD-KL shows improved AUC performance as the anomaly ratio in unlabeled data increases, which implies that the proposed KL divergence-based detection algorithm is robust to ratio of normal data to abnormal data.

5 Conclusion

To detect normal data in unlabeled data, we propose KL divergence-based algorithm. The proposed SAD-KL uses KL divergence of the PDFs of LOF scores obtained from both labeled and unlabeled data and then estimates the threshold for normal data detection. The SAD-KL takes less learning time than existing semi-supervised anomaly detection algorithms since it estimates PDFs of the data and then evaluate the detection threshold from the computed CDF. Through experiments, we show that PDFs of LOFs follow Burr distribution and that the SAD-KL presents a higher performance in the average AUCs, displays tighter decision boundaries and achieves higher learning efficiency than the existing algorithms. Therefore, the proposed SAD-KL can be a strong candidate for efficient semi-supervised learning algorithm requiring stable detection performance regardless of distribution variation between the labeled and unlabeled data. The inclusion of labeled abnormal data in detection probability estimation and the extension of the SAD-KL to explainable anomaly detection become interesting topics for future research.
6 References

[1] L. Ruff, J. R. Kauffmann, R. A. Vandermeulen, G. Monatavon, W. Samek, M. Kloft, T. G. Dietterich, and K. R. Müller. A unifying review of deep and shallow anomaly detection. Proceeding of the IEEE, 109(5):756-795, 2021.

[2] B. Schölkopf, R. Williamson, A. Smola, J. S. Taylor, and J. 850 Platt. Support vector method for novelty detection. In NIPS, 851 1999.

[3] D. M. Tax, and R. P. Duin. Support vector data description. Machine learning, 59(1):45-66, 2004 .

[4] L. Ruff, R. A. Vandermeulen, N. Görnitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft. Deep one-class classification. In ICML, 2018.

[5] Z. Ghafoori, and C. Leckie. Deep multi-sphere support vector data description. In SIAM, 2020.

[6] S. Goyal, A. Raghunathan, M. Jain, H. Simhadri, and P. Jain. DROCC: Deep robust one-class classification. In ICML, 2020.

[7] H. Song, Z. Jiang, A. Men, and B. Yang. A hybrid semisupervised anomaly detection model for high-dimensional data. Computational Intelligence and Neuroscience, vol. 2017.

[8] S. Akvay, A. A. Abarghouel, and T. P. Breckon. GANomaly: semi-supervised anomaly detection via adversarial training. In ACCV, 2018.

[9] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. R. Müller, and M. Kloft. Deep semi-supervised anomaly detection. In ICLR, 2020.

[10] D. Hendrycks, M. Mazeika, and T. Dietterich. Deep anomaly detection with outlier exposure. In ICLR, 2019.

[11] L. Ruff, R. A. Vandermeulen, B. J. Franks, K. R. Müller, and M. Kloft. Rethinking assumptions in deep anomaly detection. In ICML, 2021.

[12] K. Lee, C. H. Lee, and J. Lee. Semi-supervised anomaly detection algorithm using probabilistic labeling (SAD-PL). IEEE Access, 9:142972-142981, 2021.

[13] H. Ju, D. Lee, J. Hwang, J. Namkung, and H. Yu. PUMAD: PU metric learning for anomaly detection. Information Sciences, 523:167-183, 2020.

[14] L. Yang, J. Chen, Z. Wang, W. Wang, J. Jiang, X. Dong, and W. Zhang. PLELog: Semi-supervised log-based anomaly detection via probabilistic label estimation. In ICSE, 2021.

[15] L. McInnes, J. Healy, and S. Astels. hdbscan: Hierarchical density based clustering. J. Open Source Softw., 2(11):205, 2017.

[16] M. M. Breuning, H. P. Kriegel, R. T. Ng, and J. Sander. LOF: Identifying density-based local outliers. In ACM SIGMOD, 2000.

[17] J. Neyman, and E. S. Pearson. On the problem of the most efficient tests of statistical hypotheses. Philosophical Transactions of the Royal Society of London. Series A, 231:289-337, 1933.

[18] I. W. Burr. Cumulative frequency functions. Annals of Mathematical Statistics. 13(2):215-232, 1942.

[19] A. R. Hakim, I Fithriani, and M. Novita. Properties of Burr distribution and its application to heavy-tailed survival time data. J. Phys.: Conf. Ser., 2021.

[20] Y. Lecun, C. Cortes, and C. Burges. MNIST handwritten digit database. AT&T Labs, 2010.

[21] A. Krizhevsky. Learning multiple layers of features from tiny images. 2009.

[22] N. Mu, and J. Gilmer. MNIST-C: robustness benchmark for computer vision. In ICML, 2019.

[23] A. F. Emmott, S. Das, T. Dietterich, A. Fern, and W. K. Wong. Systematic construction of anomaly detection benchmarks from real data. In ACM SIGKDD, 2013.
A. Validation for Burr distribution assumption

We plot histograms, the estimated PDFs and the CDFs of $s_u$ of the MNIST, CIFAR10, and MNIST-C datasets in Figure S1, which validates the Burr distribution assumption.

![Figure S1: Histogram and the Burr PDF obtained from (a) MNIST (b) CIFAR10 (c) MNIST-C and the estimated CDF and the Burr CDF from (d) MNIST (e) CIFAR10 (f) MNIST-C dataset.](image)

We did Kolmogorov-Smirnov (KS) test and two histogram-based tests on MNIST-C and summarize results in Table S1, which validates our assumption

| KS-Test ($\alpha = 0.05$) | Histogram Intersection | Histogram Correlation |
|---------------------------|------------------------|-----------------------|
| p-value = 0.9855 > $\alpha$ | 0.9682 | 0.9981 |

B. Estimation of parameter $\beta$

We estimated KL divergences of four datasets and summarize them in Table S2.

| MNIST  | CIFAR10 | MNIST-C | Two moons |
|--------|---------|---------|-----------|
| 81.30  | 48.21   | 60.73   | 20.44     |

From $P_D = \exp(-KL(P\|Q)/\beta)$, we can obtain equation for $\beta$ as follows:

$$\beta = -KL(P\|Q)/\log(P_D)$$  \hspace{1cm} (1)

The $\beta$ can be determined once range of $P_D$ is chosen a priori as shown in Figure S2. Note that higher $P_D$ does not give higher AUC since the $P_D$ is for LOF detection not for anomaly detection. Suppose the $P_D$ ranges from 0.95 to 0.98, then $\beta$ ranges from 1600 to 3000 for MNIST. In supplementary material we showed $\beta \approx 2500$ seems to be appropriate for three datasets. When other dataset is considered, range of $P_D$ and subsequent choice of $\beta$ still can be estimated by equation (1) using new KL divergence.

We present the detection performance of the SAD-KL with respect to the $\beta$ by changing it from 2000 to 3000 with 50 increasement. The results shown in Figure S2 prove that the SAD-KL’s AUC performance is fairly stable in wide range of the $\beta$ chosen for three datasets.
C. Detection performance according to training data ratio

We investigate the AUCs obtained by changing percentage of the labeled data and percentage of the abnormal data in the MNIST and the CIFAR10 datasets. The results are presented in Figure S3, which shows the performance of the proposed SAD-KL is compatible to the SAD-PL requiring heavy computation.

![Figure S2: \(P_D\) according to \(\beta\).](image)

![Figure S3: The AUCs according to \(\beta\) on the MNIST, CIFAR10, and MNIST-C datasets.](image)

D. Training time of the SAD-KL

While the SAD-PL adopts ensemble networks to choose the best network, the SAD-KL uses single network with fast threshold computation, which yields fast training. Even the SAD-KL takes little extra computation of LOF every epoch, due to KL divergence-based labeling it shows faster convergence than the Deep SAD and HSC. The superior convergence performance is shown in Figure S4.

![Figure S4: AUC of MNIST-C according to epoch.](image)

E. Comparisons with AnoGAN and GANomaly

We performed experiments by using 1 class being “abnormal” and the others being “normal” as the AnoGAN and GANomaly do. The AUC results shown in Figure S5 shows superior AUC performance of the SAD-KL.

![Figure S5: AUC results on (a)MNIST and (b)CIFAR10](image)
F. Anomaly score comparisons with normal and abnormal images

We present normal and abnormal test images along with the corresponding anomaly scores of the SAD-KL and four models in Figure S6 and S7. The anomaly scores are computed as $\|\phi(x) - c\|^2$ for Deep SAD, SAD-PL and SAD-KL and as $\|\phi(x)\|^2$ for HSC. Note the SAD-KL’s scores are bigger in most MNIST images and is the biggest in MNIST-C images.

Figure S6: Anomaly scores of test images in MNIST dataset and the corresponding images.

Figure S7: Anomaly scores of test images in MNIST-C dataset and the corresponding images.