Deep Unsupervised Anomaly Detection

Tangqing Li[1], Zheng Wang [2], Siying Liu[2], and Wen-Yan Lin[3]

[1]National University of Singapore, [2] Institute for Infocomm Research, Singapore,
[3]Singapore Management University

li_tangqing@u.nus.edu, {zhwang, liusy1}@i2r.a-star.edu.sg, daniellin@smu.edu.sg

Abstract

This paper proposes a novel method to detect anomalies in large datasets under a fully unsupervised setting. The key idea behind our algorithm is to learn the representation underlying normal data. To this end, we leverage the latest clustering technique suitable for handling high dimensional data. This hypothesis provides a reliable starting point for normal data selection. We train an autoencoder from the normal data subset, and iterate between hypothesizing normal candidate subset based on clustering and representation learning. The reconstruction error from the learned autoencoder serves as a scoring function to assess the normality of the data. Experimental results on several public benchmark datasets show that the proposed method outperforms state-of-the-art unsupervised techniques and is comparable to semi-supervised techniques in most cases.

1. Introduction

Anomaly detection refers to the identification of patterns that do not conform to expected normal behavior [6]. It is a critical task in diverse application domains such as fraud detection [23], intrusion detection [16] and surveillance video profiling [31, 25]. While the concept of an anomaly is intuitively easy for humans to understand, it is hard to define mathematically. Fundamentally, an anomaly is something with insufficient similarity to the rest of the data. This similarity can be computed on the basis of some feature difference. However, what makes an ideal feature representation for the data depends on what constitutes an anomaly. This forces anomaly detection into a chicken-or-egg problem in which there are a pair of problems, neither of which can be solved before the other.

To date, a number of works have attempted this problem by training an autoencoder to create low-dimensional representations for anomaly detection [5, 33, 35]. The anomalies are rejected and the autoencoder retrained [22, 29]. While this gives reasonable results, it is fundamentally dependent on how well the first iteration solves the problem.

We propose a solution in which anomalies can be defined using approximately correct features. This is achieved through an observation. Given a feature, anomalies approximately correspond to instances of high variance distributions. Such instances can be identified using a distribution-clustering [19] framework. This hypothesis provides a reliable starting point for normal data selection. We train an autoencoder from the normal data subset obtained from distribution-clustering, and iterate between hypothesizing normal candidate subset and representation learning. The reconstruction error serves as a scoring function to assess the normality of the data. The proposed framework does not rely on any training labels. Instead, it iteratively distills out anomalous data and improves the learned representation of normal data by incorporating clustering techniques into the process. Our method works with the least assumption on the data itself and does not use any label information, even in the training phase. The only assumption is that the anomalies are not statistically dominant in the entire dataset; and for this exact reason they are anomalies by nature.

We extensively assess the broad applicability of the proposed model on network intrusion, image and video data. Empirical results show that the proposed method outperforms the existing state-of-art approaches in terms of both accuracy and robustness to the percentage of anomalous data.

2. Related Works

Existing anomaly detection methods can be grouped into three categories.

Reconstruction-based method These methods assume that anomalies are in-compressible and thus cannot be effectively reconstructed from low-dimensional projections. Classical methods like Principle Component Analysis(PCA) [13] and Robust-PCA [4] are motivated by this assumption. In recent works, different forms of deep autoencoder are

proposed to analyze the reconstruction error. Xia et al. [30] show that by introducing a regularizing term to a convolutional autoencoder, the anomalies tend to produce a bigger reconstruction error. Variational Autoencoder (VAE) [1] and Generative Adversarial Networks (GANs) [26] have also been introduced to perform reconstruction-based anomaly detection. These methods demonstrate promising results when the anomaly ratio is fairly low. Although the reconstruction of anomalous samples, based on a reconstruction scheme optimized for normal data, tends to generate a higher error, a significant amount of anomalous samples could mislead the autoencoders to learn the correlations in the anomalous data instead. Pidhorskyi et al. have [24] adopted Adversarial Autoencoder [20] for generative probabilistic novelty detection. These methods, although claimed as unsupervised, require pre-isolation/identification of classes of normal data in the training phase, as normal data is needed to describe the inliner distribution. For example, In GAN methods [26, 24], label information is used to feed normal data into the discriminator during training.

Density estimation, representation learning and clustering Motivated by the assumption that anomalies occur less frequently, these algorithms treat anomalies as low-density regions in some feature space. Clustering analysis, such as Robust-KDE [14], is often used for density estimation and anomaly detection. Unfortunately, due to the curse of dimensionality, these methods are less applicable to analyzing high-dimensional data, where density estimation is a challenge in itself.

A two-step approach is normally adopted to counter this issue, where dimensionality reduction is conducted first, followed by clustering analysis as a separate step. One drawback of this approach is that dimensionality reduction is trained without the guidance from the subsequent clustering analysis; hence the key information for clustering analysis could be lost during dimensionality reduction. Recently, Ionescu et al. [12] proposed to train autoencoders on tracked objects in videos to detect anomalous events. The latent representations from autoencoders are clustered, followed by a one-versus-rest classifier to discriminate between the formed clusters. There are also works that jointly learn dimensionality reduction and clustering components based on deep autoencoder [33, 35]. Notably, DAGMM [35] utilizes an autoencoder to generate a low-dimensional representation and its reconstruction error, which is further fed into an estimation network based on Gaussian Mixture Model(GMM). However, as its autoencoder was trained on the whole dataset, it is vulnerable to a high percentage of anomalous samples and may learn wrong correlations. In contrast, our proposed method addresses this issue by first finding a normal candidate subset to train an autoencoder and then iterating between representation learning and refinement of the normal candidate.

One-class classification One-class SVM [9, 5] is widely used. Under this framework, a discriminative boundary surrounding the normal instances is learned by algorithms. However, when dimensionality goes higher, such techniques often suffer from suboptimal performance due to the curse of dimensionality. OCNN [5] attempts to circumvent this problem by using an autoencoder for dimensionality reduction. However, OCNN requires training data with relatively low anomaly ratio, in order to obtain an optimized NN model to differentiate anomalies from single-class normal data. Zenati et al. [32] use GAN to learn a generative model from the normal data, and leverage the latent representation of the generator input or from the encoder in the discriminator learning. Label information of the normal data is required for training.

3. Problem Formulation

Let $\mathbf{X} = \{\mathbf{x}_i\}, i = 1, \ldots, N, \mathbf{x} \in \mathbb{R}^k$ be the set of input data points that contains a certain percentage of anomaly. The goal of anomaly detection is to learn a scoring function $h(\mathbf{x})$, $h : \mathbb{R}^k \mapsto \mathbb{R}$, to classify samples $\mathbf{x}_i$ based on some threshold λ:

$$y_i = \begin{cases} 0, & \text{if } h(\mathbf{x}_i) < \lambda \\ 1, & \text{if } h(\mathbf{x}_i) \geq \lambda \end{cases} \tag{1}$$

where y_i are the labels. $y_i = 0$ indicates $\mathbf{x}_i$ is normal and $y_i = 1$ indicates anomalous.

An overview of the proposed end-to-end anomaly detection system is presented in Fig. 1. The major component of this system is an autoencoder that learns a low-dimensional representation of the input data that are often of high dimensions, to enable simplified modeling of the underlying distribution of the data. Under a fully unsupervised setting, the only information we are given is the set of input data $\mathbf{X}$, without any label information. As an initialization, we leverage the latest clustering technique for high-dimensional data [19] to provide soft supervisory signals.

Since our input data is unlabelled, we derive a "training" set $\mathbf{S}_{train}$, where $\mathbf{S}_{train} \subset \mathbf{X}$ based on the following:

$$\mathbf{S}_{train} = \mathcal{C}(\mathbf{X}, p_0) \tag{2}$$

where $\mathcal{C}$ represents a selection process based on clustering output, and p_0 represents the percentage of anomaly, it controls which are the clusters to be accepted into the "training" set. In our experiments, we compute the threshold as the $(100 - p_0)^{th}$ percentile of cluster variance, and accept clusters with variance smaller than this threshold. The assumption here is that clusters with large variance are likely to contain anomalous members.

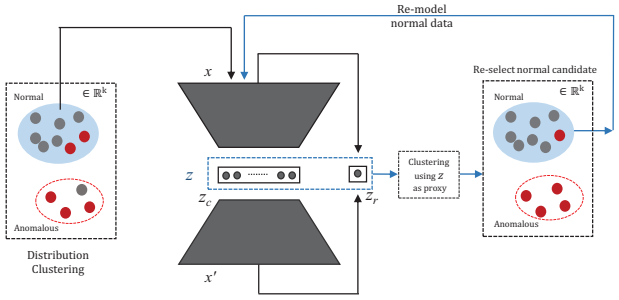


Figure 1: Flow-chart of the proposed end-to-end anomaly detection system.

3.1. Scoring Function Learning

The autoencoder network provides two sources of features: (1) a low-dimensional representation of the original input data; and (2) the reconstruction error by comparing the input with its decoded counter-part. Using the training set $\mathbf{S}_{train} = \{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_M\}$, the autoencoder learns the encoding function f_{en}:

$$\mathbf{z}_c = f_{en}(\mathbf{s}; \Theta_{en}), \quad \forall \mathbf{s} \in \mathbf{S}_{train}, \quad \mathbf{z}_c \in \mathbb{R}^{k_{bn}} \tag{3}$$

where Θ_{en} are the learned parameters for the encoder. $\mathbf{z}_c$ are known as the bottle-neck features of dimension k_{bn}.

Similarly, for the decoding part, we have:

$$\mathbf{x}' = f_{de}(\mathbf{z}_c; \Theta_{de}), \tag{4}$$

where Θ_{de} are the learned parameters for decoding. $\mathbf{x}'$ are the reconstructed features.

Upon training, we have a learned autoencoder with optimized parameters $\{\Theta_{en}, \Theta_{de}\}$. We apply the encoder network on the entire input set $\mathbf{X}$ to produce a new set of features $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_N\}$. Each data point in this set is formed by concatenating the bottle-neck feature with the reconstruction error:

$$\mathbf{z} = [\mathbf{z}_c;\ \mathbf{z}_r], \quad \mathbf{z} \in \mathbb{R}^{k_{bn}+1}, \tag{5}$$

where the reconstruction error $\mathbf{z}_r$ is measured in terms of cosine similarity between $\mathbf{x}$ and its decoded counter-part:

$$\mathbf{z}_r = d(\mathbf{x}, \mathbf{x}') = \cos^{-1}\left(\frac{\mathbf{x}^T\mathbf{x}'}{\| \mathbf{x} \| \| \mathbf{x}' \|}\right) \tag{6}$$

ℓ_2 normalization is applied on each data point $\mathbf{z}$. The inclusion of reconstruction loss helps to make anomalous data points more distinguishable. $\mathbf{Z}$ is now of a much lower dimension than the input data $\mathbf{X}$. Hence, traditional clustering techniques such as Gaussian Mixture Model would suffice for subsequent training set selection. To ensure the initial training set can capture most of the normal samples, we adopt a more conservative cluster variance threshold.

With the new encoding scheme, the entire input set $\mathbf{X}$ is now represented as $\mathbf{Z}$. We can "re-label" the training set $\mathbf{X}$ by using $\mathbf{Z}$ as a proxy, and an assumed anomaly percentage p to determine the threshold. Similar to the initial training set selection, we select members that belong to low-variance clusters in $\mathbf{Z}$. The process of *Training set selection* $\rightarrow$ *Autoencoder training* $\rightarrow$ *New feature computation* is performed iteratively. The training set is updated as follows:

$$\mathbf{Z}_{train}^{t+1} = \mathcal{C}(\mathbf{Z}^t, p), \tag{7}$$

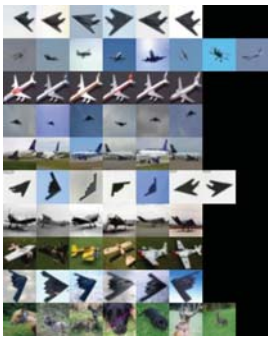
$$\mathbf{S}_{train}^{t+1} = \{\mathbf{x}_j : \forall \mathbf{z}_j \in \mathbf{Z}_{train}^{t+1}\}, \tag{8}$$

where the superscript t here refers to the t^{th} iteration.

Finally, the training process terminates when there is no change in the set of selected normal samples between two successive iterations. After the last iteration, $t = t_F$, we obtain the autoencoder parameters $\{\Theta_{en}^{t_F}, \Theta_{de}^{t_F}\}$, and use it to construct the scoring function:

$$h(\mathbf{x}) = d(\mathbf{x}, \mathbf{x}') = d(\mathbf{x}, f_{de}(f_{en}(\mathbf{x}; \Theta_{en}^{t_F});\ \Theta_{de}^{t_F})), \tag{9}$$

where $\mathbf{x}'$ is the result of going through the encoding-decoding process according to the trained autoencoder.



(a) Clustering result of CIFAR-10 ("airplanes" class forms the normal group) : cluster 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, with increasing cluster variance.

(b) Clustering result of MNIST (digit '4' forms the normal group): cluster 5, 10, 15, 20, 25, 30, 35, 40, 45, 50 with increasing cluster variance.

Figure 2: Results from Distribution Clustering.

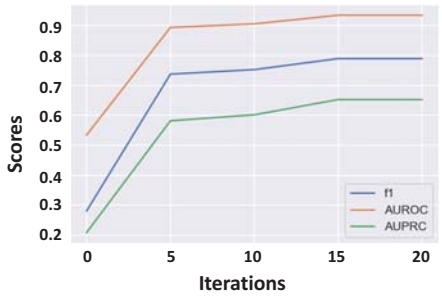


Figure 3: F1, AUROC, AUPRC scores for 20 iterations on KDDCUP dataset, with 20% anomaly.

3.2. Algorithm

The proposed framework is summarized in Algorithm 1. We obtain an initial split of the data into normal and

abnormal subsets through clustering (i.e. GMM for KDD-CUP data and Distribution Clustering [19] for image and video data). Candidate normal samples are then passed into the autoencoder for representation learning. All examples' memberships are re-evaluated based on its low-dimensional representation every r epochs, where the new normal candidates are fed into the autoencoder for learning. Finally, when there is no change in all samples' memberships, an encoder that learns the low-dimensional projection of the normal data is finalized, and its reconstruction loss will be used for scoring

Initialization We use Distribution Clustering [19] to make an educated guess about the normal data subset for image and video data. Selected clustering outputs for CIFAR-10 and MNIST datasets are shown in Fig. 2. Observe that as cluster variance increases, the samples' appearance become more anomalous.

Convergence Assuming a $p\%$ anomaly percentage, our algorithm starts with a tight cut-off, accepting clusters with variances below $(100-p_0)^{th}$ percentile as an initial training set, where $p_0 > p$. This ensures the initial training set is as pure as possible. Our assumption is that given partial normal data, the autoencoder would be able to learn a representation and generalize well on the "unseen" normal data that was discarded, and progressively recover them as iterations go on. Empirically, we plotted the AUROC, AUPRC and F-score for 20 iterations for the KDDCUP experiment, presented in Fig. 3 . It demonstrates the convergence as iteration progresses. The same behavior was observed throughout our experiments on other datasets.

Algorithm 1 Deep end-to-end Unsupervised Anomaly Detection

Input: $\mathbf{X} = \{\mathbf{x}_i\}$, $i = 1, 2..., N$: set of normal and anomalous input examples. r: number of epochs required for re-evaluation of the membership of the entire input set $\mathbf{X}$. p_0 and p: thresholds for initial and subsequent training set selection, respectively

Output: Reconstruction-based anomaly score function $h(\mathbf{x})$ and trained autoencoder $\{\Theta_{en}^{t_F}, \Theta_{de}^{t_F}\}$,

procedure GET_DECISION_SCORE($\mathbf{X}, r, p, f_{en}, f_{de}$)
  $\mathbf{S}_{train} \leftarrow \mathcal{C}(\mathbf{X}, p_0)$ ▷ Run clustering, select instances from low-variance clusters
  $\mathbf{L} = \{k : \forall \mathbf{x}_k \in \mathbf{S}_{train}\}$ ▷ $\mathbf{L}$ is the set of indices of selected normal training samples
  $\mathbf{L}^{old} := \emptyset$
  while setdiff$(\mathbf{L}^{old}, \mathbf{L}) \neq \emptyset$ **do**
    for each epoch **do**
      if $((CurrentEpoch + 1) \bmod \mathbf{r}) == 0$ **then** ▷ Re-evaluate normality every r epochs
        $\mathbf{Z}_c \leftarrow f_{en}(\mathbf{X}, \Theta_{en})$ ▷ Bottle-neck features
        $\mathbf{X}' \leftarrow f_{de}(\mathbf{Z}_c, \Theta_{de})$
        $\mathbf{Z}_r \leftarrow d(\mathbf{X}, \mathbf{X}')$ ▷ Reconstruction error
        $\mathbf{Z} \leftarrow [\mathbf{Z}_c; \mathbf{Z}_r]$
        $\mathbf{S}_{train} \leftarrow \mathcal{C}(\mathbf{Z}, p)$ ▷ Get new training set according to threshold p
        $\mathbf{L}^{old} := \mathbf{L}$
        $\mathbf{L} \leftarrow \mathbf{S}_{train}$ ▷ Update set of indices for training samples
      else
        Train f_{en}, f_{de} on $\mathbf{S}_{train}$ to obtain $\{\Theta_{en}, \Theta_{de}\}$
    end for
  end while
  $\Theta_{en}^{t_F} = \Theta_{en}, \quad \Theta_{de}^{t_F} = \Theta_{de}$
  Output $h(\mathbf{x})$ according to finalized autoencoder $\{\Theta_{en}^{t_F}, \Theta_{de}^{t_F}\}$ base on Eq. (9)
end procedure

4. Experiments

4.1. Baseline Methods

On the topic of anomaly detection, there are different terminologies concerning the nature of supervision: (a) Algorithm uses label information of the normal class for training (label information could be used in part, or all of the stages of an algorithm); (b) No training labels are given, algorithm treats the entire dataset with both normal and anomalous classes as input. For the purpose of this paper, we term type (a) semi-supervised and type (b) unsupervised. We evaluate our method against the following state-of-the-art methods:

OC-NN*(semi-supervised)* One-class neural networks (OC-NN) [5] contains 2 major components: a deep autoencoder and a feed-forward convolutional network. The deep encoder is trained on normal data for representation learning. The trained encoder, with its parameters frozen, is subsequently used as the input layers of a feed-forward network with 1 extra hidden layer. Variants of OC-NN employ different activation functions (i.e. linear, sigmoid, relu) in the hidden layer. We report the best score attained among all possible activation functions in our experiments.

OC-SVM*(unsupervised)* One-class support vector machine (OC-SVM) [9] is a kernel-based method for anomaly detection. The algorithm searches for best-performing hyper-parameters γ (kernel coefficient) and ν (upper bound of the fraction of training errors and lower bound of the fraction of support vectors) to obtain the optimal AUROC [3].

DAGMM*(unsupervised)* Deep autoencoding Gaussian mixture model (DAGMM) [35], comprised of one compression net and one estimation net, is a method based on representation learning. The compression network provides

low-dimensional representations of input samples and the reconstruction error features. They are fed into the estimation network, which functions as a Gaussian Mixture Model, to predict the mixture membership for each sample. We modify the original DAGMM algorithm by adding a small value to the diagonal elements of the covariance matrix. The model achieves better results than the reported score from the original work.

Deep anomaly detection using geometric transformations*(semi-supervised)* This method [11] employs a deep neural model to identify out-of-distribution samples of image data, given only the examples from the normal class. A series of geometric transformations are applied to the normal class to create a multi-class dataset. A deep neural net, trained using this dataset, is then employed to discriminate the transformations applied. Subsequently, given an unseen instance, the model applies each transformation on it and assigns membership scores. The final normality score is determined based on the combined log-likelihood of softmax response vectors.

4.2. Datasets

We employ five benchmark datasets, namely, KDDCUP, MNIST, CIFAR-10, CatVsDog and UCF-Crime, to evaluate our proposed method, together with other methods described above.

• **KDDCUP:** The KDDCUP network intrusion dataset [18] contains samples of 41 dimensions. Similar to [35], categorical features are prepared by applying one-hot encoding. 20% of the "normal" samples form the minority group, while the rest 80% are treated as "attackers". As "normal" samples are the minorities, they are treated as anomalies

• **MNIST:** The MNIST dataset [17] consists of 60,000 gray-scale 28×28 images of handwritten digits from 0 to 9. We formulate an anomaly detection task as per described in [5] and [34], where 4,859 images of digit 4 are randomly sampled as normal instances and 265 images are evenly sampled from all other categories as anomalies.

• **CIFAR-10:** The CIFAR-10 dataset [15] contains 60,000 color images of size 32×32 from 10 classes. We formulate an anomaly detection task with 5050 examples from class airplane (category 0) being the normal group and 450 images evenly sampled from the rest of the categories as anomalous instances.

• **CatVsDog:** The CatVsDog dataset consists of dogs and cats images of varying sizes, which are extracted from the ASIRRA dataset [8] following the settings specified in [11]. 12,500 images of dogs and 2,500 images of cats are sampled to form an anomaly detection task. The cat images are treated as anomalies.

• **UCF-Crime:** The UCF-Crime dataset [27] contains 1,900 long and untrimmed videos captured from CCTV cameras. It covers 13 categories of real-world crimes under diverse conditions, e.g., indoor and outdoor, day and night times. Example crime categories include fighting, burglary, etc. In both the training and testing sets, videos are of different lengths and anomalies happens at various temporal locations. Videos within the same category may contain diverse background scenes. Some of the videos may have multiple anomaly events.

4.3. Evaluations

We adopt Area Under the curve of the Receiver Operating Characteristic (AUROC) as the main evaluation metric to measure the discrimination power of different models. AUROC is a standard method to assess the effectiveness of a classifier [10]. It can be interpreted as the probability that an anomalous instance is assigned to a higher anomaly score than a normal instance [7]. In this section, we compare the performance of our method against other baseline methods.

KDDCUP: Network Intrusion Data In this experiment, we divide the KDDCUP dataset following the setting in [35]. 50% of the data is reserved for testing by random sampling. From the remaining 50% of the data reserved for training, we take all samples from the normal class and mix them with different percentages of samples from the anomaly class to form the training set. Parameters for this experiment (see Algorithm 1) are set to: $p_0 = 35\%$, $p = 30\%$, $r = 10$.

Table 2 and 3 reports the AUROC and AUPRC of OC-SVM, DAGMM and our model on the KDDCUP dataset after 200 epochs, with anomaly percentage in training set being 5%, 10% and 20%, respectively. It can be observed that an increase in the percentage of anomalous data undermines the detection performance of OC-SVM and DAGMM more severely, while our method remains robust to such changes.

Figure 4 shows the Receiver Operating Characteristic (ROC) curves of different models when the anomaly percentage of the training data is 20%. In our unsupervised setting where no prior knowledge of normal class is known, our method is clearly more robust to contaminated training data.

Image Data In table 4, we compare the AUROC scores obtained from OC-NN, OC-SVM, DAGMM, Geometric Transformation and our model, based on multiple image datasets. It should be noted that Geometric Transformation approach trains on data from the normal class only (hence classified as semi-supervised). Our method, on the other hand, does not require label information. Unless otherwise specified, we use NetVLAD [2] as feature representation for all the image datasets.

The parameters used (refer to Algorithm 1) for each image dataset are as follows: MNIST ($p_0 = 25\%$, $p = 20\%$, $r = 5$), CIFAR-10 ($p_0 = 15\%$, $p = 10\%$, $r = 5$) and CatVsDog ($p_0 = 25\%$, $p = 20\%$, $r = 10$). Detailed parameters for the experiments are presented in the appendix.

Table 1: Summary statistics of datasets.

Dataset	Normal Class	Input Dimension	# Instances	Anomaly Percentage (%)
KDDcup	attack	1×120	494,021	20
MNIST	digit 4	28×28	5,124	5
CIFAR-10	airplane (category 0)	32×32	5,500	8
CatVsDog	dog	128×128	15,000	17
UCF-Crime	non-crime scenes	varying	dep. on video	< 35

Table 2: AUROC (in %) of different models with different anomaly percentage based on KDDCUP dataset. Our proposed method is much more immune to increase in anomaly percentage.

Anomaly Percentage (%)	OC-SVM	DAGMM	Ours
5	96.8 ± 0.5	96.6 ± 1.1	$\mathbf{98.2 \pm 1.0}$
10	89.7 ± 0.1	88.6 ± 2.0	$\mathbf{98.4 \pm 0.8}$
20	61.6 ± 0.1	79.5 ± 2.0	$\mathbf{93.5 \pm 1.1}$

Table 3: AUPRC (in %) of different models with different anomaly percentage based on KDDCUP dataset. Our proposed method is much more immune to increase in anomaly percentage.

Anomaly Percentage (%)	OC-SVM	DAGMM	Ours
5	77.8 ± 0.1	75.4 ± 0.7	$\mathbf{94.5 \pm 0.1}$
10	68.1 ± 0.1	53.4 ± 2.7	$\mathbf{93.9 \pm 0.1}$
20	45.4 ± 0.0	40.7 ± 2.8	$\mathbf{90.3 \pm 0.5}$

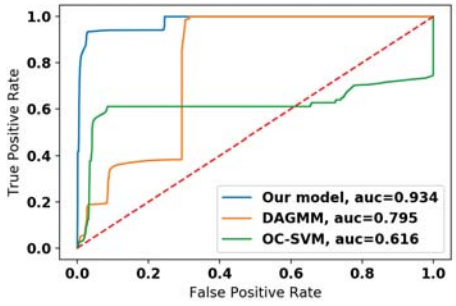


Figure 4: ROC comparison of our proposed mothod, DAGMM, and OC-SVM. Results are obtained based on the KDDCUP dataset, with 20% anomaly.

Results in Table 4 demonstrate an outstanding performance of our method over other unsupervised approaches. In addition, on CIFAR-10, the performance of our proposed algorithm is comparable to that of Geometric Transformation, a semi-supervised method. In the last column of Table 4, we report results obtained using distribution clustering alone. The combination of distribution clustering and autoencoder significantly improves discrimination against anomalies. Details of the network parameters used in the experiments are reported in the supplementary material.

We make several key observations based on the results in Table 4. Unlike all other methods that tend to perform better on simpler datasets (e.g. MNIST), the advantage of our method becomes more evident on datasets with higher complexity. Notably, our method outperforms other unsupervised approaches on CIFAR-10 and CatVsDog. The shortfall of our method on MNIST dataset could be due to the adoption of NetVLAD feature extractor (4096-d feature vectors), which may not be an ideal choice of feature representation since the images are pre-aligned and hence of low dimensionality. We repeat the same experiment using raw image pixel values from MNIST. It shows improved AUROC score, suggesting that raw feature is a better representation.

While our method is able to produce results comparable to semi-supervised approaches, the gap is wider on CatVsDog dataset as compared to CIFAR-10. We attribute this to the high noise level in the CatVsDog dataset. For example, some images consist of both dog and cats. Moreover, training on normal data (with augmentation through geometric transformation) gives Geometric Transform a natural advantage. According to [8], ASIRRA dataset, from which the CatVsDog is extracted, is deemed extremely challenging for computers. Sample images of the dataset are presented in the supplementary material.

Video Data We apply the proposed approach on UCF-Crime dataset [27], with features extracted using C3D [28] descriptor. In default C3D settings, every 16-frame segment is aggregated to generate 1 feature vector.

In [27], although the AUROC score of each video category is not reported, the AUROC averaged across the entire test set of UCF-Crime dataset is 75.4%. It is achieved by adopting a weakly-supervised method called multiple instance learning (MIL). We applied our method on each crime category by combining all test videos in each category as a single dataset for training. The results are tabulated in Table 5. Our method is able to score 72.9%, losing by just a small margin of 2.5% to [27]. We note that crimes

Table 4: AUROC in %. Highest score among all methods and highest score among all unsupervised methods are highlighted. On complex datasets such as CIFAR-10 and CatVsDog, our proposed method has higher performance gain among all unsupervised methods. The last column presents results obtained using distribution clustering alone.

Dataset	OC-NN (semi-supervised)	Geom. Transform. (semi-supervised)	DAGMM (unsupervised)	OC-SVM (unsupervised)	Ours (unsupervised)	Distrib. Clust. only
MNIST	70.0	**98.2**	50.3	**90.2**	82.4 ± 1.8 (raw) / 70.5 ± 2.1 (NetVLAD)	63.3
CIFAR-10	63.8	73.3	49.0	69.7	$\mathbf{73.6 \pm 0.6}$	48.7
CatVsDog	50.8	**88.3**	43.4	56.2	$\mathbf{78.0 \pm 1.2}$	56.1

Table 5: AUROC for each crime scene category UCF-Crime.

Crime Scene	No. of videos	AUROC (%)
Abuse	2	66.4
Arrest	5	63.2
Arson	9	65.6
Assault	3	76.9
Burglary	13	76.8
Explosion	21	72.8
Fighting	5	76.4
Road Accidents	23	79.5
Robbery	5	78.1
Shooting	23	73.3
Shoplifting	21	63.5
Stealing	5	73.3
Vandalism	5	83.5
		Ave. over all categories: 73.0
Average	10.77	Ave. over all videos: 72.9

Table 6: Performance comparison based on AUROC scores on selected videos with DAGMM [35]

Crime scene	# Video selected	Ours	DAGMM
Arrest	2	70.6	52.2
Arson	3	67.8	60.1
Burglary	4	79.2	67.4
Fighting	3	77.1	57.0

such as "shoplifting" has subtle actions that are less distinguishable from normal behavior, hence the AUROC is lower. Despite being fully unsupervised (without label information), our method is as effective as its semi-supervised competitor, demonstrating its strength in handling complex and high-dimensional data.

To benchmark against DAGMM, the state-of-the-art unsupervised method, we conducted another evaluation on 12 randomly selected videos from 4 categories. This experiment takes individual video file as input, instead of taking all test videos within a category as one single dataset as in the previous experiment. As both methods need a sufficient amount of feature vectors for training, we selected videos with at least 65 feature vectors (i.e. 1040 frames). The AUROC scores on 4 randomly selected video categories, using our method and DAGMM, are reported in Table 6. Detailed parameter settings for this experiment are reported in the supplementary material.

We also plotted the segment-wise anomaly scores against ground truth in Fig. 5. A good correspondence between the ground truth and our anomaly scores can be observed, where frames with anomalous events under the orange lines are assigned to higher anomaly scores. Our method significantly out-performs DAGMM.

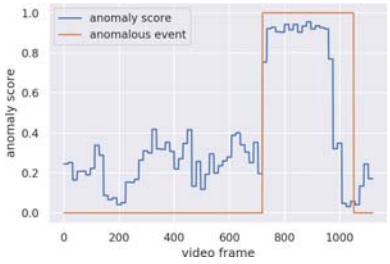
Run Time Excluding feature extraction and clustering process, on a single NVIDIA Tesla P100 GPU, our method takes 4 minutes 20 seconds on average to complete the CIFAR-10 experiment described above (consisting of 5,500 instances). This timing is averaged over 5 runs.

4.4. Ablation Study

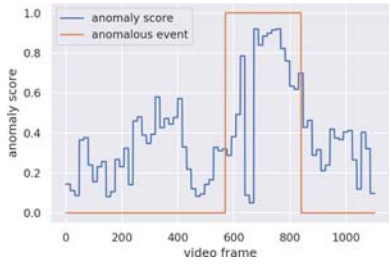
Initialization To examine the effect of adopting distribution clustering as the initialization method for high-dimensional data, a variety of other mainstream clustering methods, including K-means, HDBSCAN [21] and Gaussian Mixture Model (GMM) are used to replace the distribution clustering component in the initial normal subset selection. We compare results on the CIFAR-10 task.

For K-means and GMM, the number of clusters/components is set to 20, which is consistent with the setting of the GMM employed in the proposed model. For HDBSCAN, the minimum size of a cluster is set to 5, that follows the setting as distribution clustering. Table 8 reports the AUROC scores obtained from CIFAR-10 anomaly detection task with different clustering techniques. The results demonstrate that using distribution clustering initialization provides better supervisory signals and leads to favorable performance.

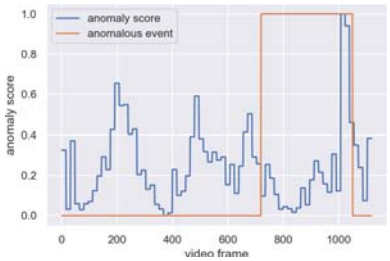
To further understand the effectiveness of distribution clustering, we tabulated the AUROC achieved using distribution clustering alone, for the experiments on image data (refer to right-most column of Table 4). Surprisingly, this



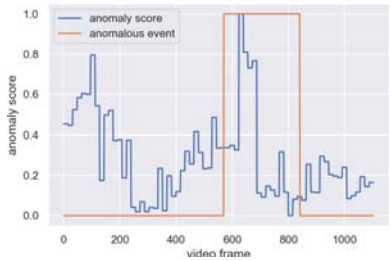
(a) Burglary018 anomaly score by our method



(b) Fighting033 anomaly score by our method



(c) Burglary018 anomaly score by DAGMM



(d) Fighting033 anomaly score by DAGMM

Figure 5: Anomaly scores (normalized) plotted against ground truth (flagged by orange lines). Compared to DAGMM, our method shows much better correspondence to the ground truth.

Table 7: AUROC score when varying anomaly percentage p. Ground truth anomaly pecentages are shown on the left column.

Dataset	G.T.	p (%)									
	%	5	10	15	20	25	30	35	40	45	50
MNIST (raw)	5	42.7 ± 1.0	55.8 ± 1.4	63.0 ± 1.3	70.5 ± 2.1	70.3 ± 0.8	70.2 ± 1.5	71.2 ± 1.0	73.7 ± 0.3	70.0 ± 1.9	71.1 ± 1.5
Cifar10	8	72.8 ± 0.7	73.6 ± 0.6	71.9 ± 0.7	70.9 ± 0.5	67.5 ± 0.5	71.6 ± 0.4	69.0 ± 0.3	68.8 ± 1.0	66.0 ± 1.0	64.0 ± 0.9
CatVsDog	17	60.4 ± 1.7	66.5 ± 2.1	74.0 ± 4.4	78.0 ± 1.2	74.0 ± 2.5	70.1 ± 3.2	69.5 ± 2.7	62.4 ± 4.0	60.3 ± 1.6	61.0 ± 3.3

Table 8: Comparison on initialization methods on CIFAR-10.

Clustering method	AUROC
K-means	60.3
HDBSCAN	61.2
GMM	50.0
Distribution Clustering	**73.6**

result is even better than those of DAGMM and OC-NN on the challenging CatVsDog dataset.

Influence of Anomaly Percentage p is the assumed anomaly percentage that serves as a threshold for normal candidate selection from the clustering output. Table 7 shows the AUROC scores with varying p values. We observe that, except for the MNIST dataset, a small overestimation above the ground truth values have little impact on performance. AUROC scores degrade gracefully as p increases to be more than 10% above the ground truth. This implies the autoencoder was still able to generalize well on the "unseen" normal data that was discarded.

5. Discussion and Conclusion

This paper presents an end-to-end method for anomaly detection under a fully unsupervised setting. The key insight of our algorithm is to model normal data. We first leverage distribution clustering technique to make an educated guess on the normal data. By incorporating clustering to provide supervisory signals, we iterate between hypothesizing normal candidate subset and representation learning. This framework iteratively distills out anomalous data and improves the learned representation of normal data. Extensive experiments on benchmark datasets demonstrate our proposed method outperforms existing unsupervised approaches and is comparable to semi-supervised solutions in most cases.

Limitations and future work: Using only an autoencoder may be insufficient to handle highly complex patterns and hence falls short on difficult dataset such as CatVsDog. For future work, we seek to explore more sophisticated generative frameworks for representation learning.

Acknowledgment This research is supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 1 grant.

References

[1] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. Technical report, 2015.

[2] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016.

[3] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[4] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3), June 2011.

[5] Raghavendra Chalapathy, Aditya Krishna Menon, and Sanjay Chawla. Anomaly detection using one-class neural networks. *arXiv preprint arXiv:1802.06360*, 2018.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[7] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[8] Jeremy Elson, John JD Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. 2007.

[9] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134.

[10] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[11] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *arXiv preprint arXiv:1805.10917*, 2018.

[12] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7842–7851, 2019.

[13] I.T.Jolliffe. Principle component analysis and factor analysis. *In Principal Component Analysis*, pages 115–128, 1986.

[14] JooSeuk Kim and C. Scott. Robust kernel density estimation. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3381–3384, March 2008.

[15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[16] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM International Conference on Data Mining*, pages 25–36. SIAM, 2003.

[17] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[18] Moshe Lichman et al. Uci machine learning repository, 2013.

[19] Wen-Yan Lin, Siying Liu, Jian-Huang Lai, and Yasuyuki Matsushita. Dimensionality's blessing: Clustering images by underlying distribution. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5784–5793, 2018.

[20] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders, 2015.

[21] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. *J. Open Source Software*, 2(11):205, 2017.

[22] G. Mishne and I. Cohen. Iterative diffusion-based anomaly detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1682–1686, March 2017.

[23] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

[24] Stanislav Pidhorskyi, Ranya Almohsen, and Gianfranco Doretto. Generative probabilistic novelty detection with adversarial autoencoders. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 6822–6833. Curran Associates, Inc., 2018.

[25] Venkatesh Saligrama, Janusz Konrad, and Pierre-Marc Jodoin. Video anomaly identification. *IEEE Signal Processing Magazine*, 27(5):18–33, 2010.

[26] Thomas Schlegl, Philipp Seebck, Sebastian M. Waldstein, Georg Langs, and Ursula Schmidt-Erfurth. f-anogan: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 54, 01 2019.

[27] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

[28] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[29] Y. Wang, B. Xue, L. Wang, H. Li, L. Lee, C. Yu, M. Song, S. Li, and C. Chang. Iterative anomaly detection. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 586–589, July 2017.

[30] Yan Xia, Xudong Cao, Fang Wen, Gang Hua, and Jian Sun. Learning discriminative reconstructions for unsupervised outlier removal. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[31] Tao Xiang and Shaogang Gong. Video behavior profiling for anomaly detection. *IEEE transactions on pattern analysis and machine intelligence*, 30(5):893–908, 2008.

[32] Houssam Zenati, Chuan Sheng Foo, Bruno Lecouat, Gaurav Manek, and Vijay Ramaseshan Chandrasekhar. Efficient gan-based anomaly detection, 2018.

[33] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1100–1109. JMLR.org, 2016.

[34] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 665–674. ACM, 2017.

[35] Bo Zong, Qiankun Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Dae ki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2018.