Enforced Isolation Deep Network For Anomaly Detection In Images

Demetris Lappas, Vasileios Argyriou, Dimitrios Makris

School of Computer Science and Mathematics, Kingston University, London, UK

Keywords: Anomaly Detection, Triple Loss Function, Deep Learning, Dimensional Reduction

Abstract

Challenges in anomaly detection include the implicit definition of anomaly, benchmarking against human intuition and scarcity of anomalous examples. We introduce a novel approach designed to enforce separation of normal and abnormal samples in an embedded space using a refined Triple Loss Function, within the paradigm of Deep Networks. Training is based on randomly sampled triplets to manage datasets with small proportion of anomalous data. Results for a range of proportions between normal and anomalous data are presented on the MNIST, CIFAR10 and Concrete Cracks datasets and compared against the current state of the art.

1 Introduction

Anomaly detection is the task of detecting rare patterns, which deviate beyond the normal distribution of data. Automating this task would be valuable as they have the potential to significantly reduce human operators' time and therefore minimise errors and costs for a wide range of applications.

Since anomaly detection can be considered a classification task, supervised learning methods tend to outperform unsupervised learning practices [1], assuming that a large enough, balanced dataset is available for training. However, anomalies lack explicit definition without normality and are infrequent by definition. Consequently, datasets have heavily unbalanced data sample sizes. Therefore, supervised learning an uncommon approach to anomaly detection. Instead, recent approaches include unsupervised approaches that do not require any ground-truth labels or semi-supervised approaches that are trained on normal samples only.

This paper introduces the Enforced Isolation Deep Network (EIDN), based on a refined Triplet Loss function designed to train an arbitrary network into distinctly clustering normal and anomalous samples far apart, by using a small fraction of labeled anomalous samples along with the normal ones. Section 2 briefly explores previous methods and explicate on some of their intuitive flaws. Section 3 describes the proposed methodology and section 4 presents the evaluation results, analysis and comparison against the current state of the art. Finally section 5 concludes the paper.

2 Previous Work

Classical machine learning approaches tackle the problem of anomaly detection by measuring the distance between data points with respect to the area occupied by the majority of data within a vector space; examples of such methodologies are Isolation Forests [2], Local Outlier Factor [3, 4], Gaussian Mixture Models [5, 6] and One-Class SVMs [7]. These unsupervised methods have demonstrated to be effective and hold various advantages and disadvantages depending on their application. However, their effectiveness heavily depends on anomalous samples being distinctly far from any cluster of normal samples. For complex data such as images this is not always the case as there tends to be much overlap and noise between normal and anomalous samples. Images are often riddled with noisy data, nevertheless human intuition is capable of detecting anomalies in a set of images. More modern approaches tackle this challenge by using semi-supervised learning methods such as AutoEncoders (or Variational AutoEncoders) [8, 9, 10, 11]. An AutoEncoder is trained on normal samples and consists of two parts, an Encoder followed by a Decoder. The Encoder maps a dataset to a lower dimensional latent space from which the Decoder then attempts to reconstruct the original dataset. The intuition behind this is twofold: First, the Encoder will eliminate the noise and only keep the features in the latent space required for the Decoder to reconstruct the original data. Secondly, since the AutoEncoder is trained only on normal samples, an anomalous sample will not be effectively reconstructed. Although AutoEncoders have been shown to work for many anomaly detection datasets, anomalous samples are not guaranteed to be any different to normal samples in the latent space.

In current state-of-the-art methods, Generative Adversarial Networks (GANs) are commonly used [12, 13, 14]. AnoGAN [14] and EGBAD [13] detect anomalies by training a random input Generator and a Discriminator. The Generator mimics normal samples under the logical assumption anomalies will resemble fake samples and the Discriminator is trained to distinguish between real and fake samples. GANomaly [12] utilises GANs to better tune an AutoEncoder by using a Discriminator to distinguish the differences between the input and output. For the datasets used in section 4, GANomaly appears to outperform AnoGAN and EGBAD [15].

A One-Class Neural Network (OC-NN) [16] is a deep learning unsupervised learning method which encompasses an SVM loss function, designed to create a boundary around normal data samples in an embedded space. Although normal and

abnormal samples are not expected to be explicitly labelled, their ratio is required for effective training.

Supervised learning methods significantly outperform all of the mentioned unsupervised and semi-supervised methods [1], still they suffer the weakness of requiring a large, balanced dataset to be trained effectively, which is a scarce luxury in most anomaly detection datasets.

More recently, a traditional Triplet Loss function has also been used to detect anomalies from the output data of vehicles [17], however, it maps normal and anomalous samples to arbitrarily located clusters in an embedded space, never allowing a uniform measurable distinction between normal and anomalous samples.

The main contribution of our methodology is the refined Triplet Loss Function, a novel loss function designed to distinctly enforce isolation between anomalous and normal samples in the latent space using Deep Networks. Instead of using a GAN to cope with unbalanced training datasets, we introduce the Random Triplet Sampling.

3 Enforced Isolation Deep Network (EIDN)

Our method uses a modified Triplet Loss function which enforces the mapping of normal samples near the origin and anomalous samples near the one-vector, in the embedded space. The network is trained using triplets which are randomly sampled. A Euclidean distance metric on the embedded space is sufficient to distinguish how far a sample deviates from normality.

An anomaly detection dataset can be abstractly described by a set D containing normal and anomalous samples X and $\hat{X}$ respectfully, such that $X \subset D$, $\hat{X} \subset D$, $X \cap \hat{X} = \emptyset$, $X \cup \hat{X} = D$, assuming that the dataset is imbalanced towards the normal samples $|X| \gg |\hat{X}|$. We define an arbitrary neural network f where R^k represents the input dimensional space and R^l represents a lower dimensional embedded space.

$$f : R^k \rightarrow R^l \text{ s.t. } k \gg l \tag{1}$$

The neural network f can be comprised of any sequence of Convolutional layers, Dense Layers, or any other differentiable layer of choice.

We refine the Triplet Loss Function $\mathcal{L}$ so that it trains f to map different classes to specific coordinates, as shown in equations 2, 3, 4 and 5. Three size-M batches of vectors are passed through the loss function, an anchor value y, a positive value x^p and a negative value x^n.

$$\mathcal{L}(y, x^p, x^n) = \sum_{i}^{M} [\|y_i - f(x_i^p)\|^2 - \|y_i - f(x_i^n)\|^2 + \alpha]_+ \tag{2}$$

$$y_i \in \{\vec{0}, \vec{1}\} \subset R^l \tag{3}$$

$$x_i^p \in \begin{cases} X & \text{if } y_i = \vec{0} \\ \hat{X} & \text{if } y_i = \vec{1} \end{cases} \tag{4}$$

$$x_i^n \in \begin{cases} \hat{X} & \text{if } y_i = \vec{0} \\ X & \text{if } y_i = \vec{1} \end{cases} \tag{5}$$

The $[V]_+$ symbol is notation for $\max(V, 0)$ where $V \in R$. The loss function $\mathcal{L}$ is designed to map all normal samples to the $\vec{0}$ vector, the origin of the embedded space, via f and all anomalous samples to the $\vec{1}$ coordinate, the one-vector; thus maximizing the distance between the two. It does so by minimizing the Euclidean distance between y and $f(x^p)$ and maximizing the Euclidean distance between y and $f(x^n)$. The real value $\alpha \in R$ represents the minimum distance f should attempt to attain between normal and anomalous samples. If $\forall i$, $\|y_i - f(x_i^n)\|^2 \geq \alpha$ and $\|y_i - f(x_i^p)\|^2 = 0$ then the network is completely trained and $\mathcal{L}(y, x^p, x^n) = 0$.

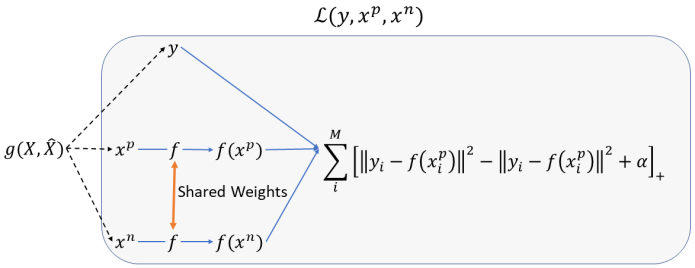

Figure 1. Architecture of generator passing randomly sampled triplets through f and to the modified Triplet Loss function. y values consisting of $\vec{1}$ or $\vec{0}$ are passed directly to the loss function, while their corresponding x^p and x^n values are filtered through f into the loss function.

Instead of creating a tremendous dataset of triplets to train our model, we developed a generator g which randomly samples from X and $\hat{X}$ to create batches of triplets to train the networks. Therefore, the network is utilised more effectively and did not require downsizing the data sample to save memory.

In a batch size of M, for each sample i, the generator $g(X, \hat{X})$ would randomly make a choice between $y_i = \vec{0}$ or $y_i = \vec{1}$ be normal or anomalous. If $y_i = \vec{0}$ then $x_i^p \in X$ and $x_i^n \in \hat{X}$, while if $y_i = \vec{1}$ then $x_i^p \in \hat{X}$ and $x_i^n \in X$.

Most semi-supervised learning methods tend to use only normal samples while training, therefore their training set sample size is $|X|$, while our network has the potential training sample size of $|X| \cdot |\hat{X}|$.

Our approach uses a manipulation of the Triplet Loss Function to train f into clustering normal and anomalous samples within an embedded space. (see Fig,1). Intuitively, each sample is "attracted" to their respective class coordinates while being "repelled" from samples of other classes. The training of a two dimensional embedded space is visualised in Fig.2.

The projections of the normal and abnormal samples on the embedded space should hover near the origin and the one-vector, respectively. This makes Euclidean distance an ideal metric for measuring how near or far a sample lies to normality. Normal samples would have a near zero Euclidean distance from the origin while anomalous samples would have a large Euclidean distance. Such an embedded space can be effectively used even by other anomaly detection methods, as demonstrated later in the next section.

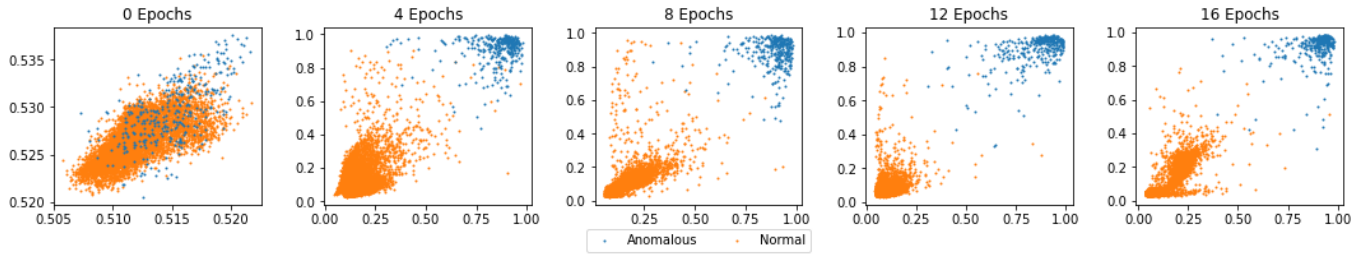

Figure 2. Demonstration of the embedded space of Class 1 of the MNIST dataset using EIDN, as described in section 4, gradually clustering normal samples (orange) to the origin and anomalies (blue) to the one-vector with each epoch.

Class	GANomaly	AnoGAN	EGBAD	DenseNet-169	ResNet-152	Inception-V4	OC-NN(.01)	EIDN(.01)
Class 0	0.881	0.623	0.783	0.998265	0.99805	0.997676	0.976	0.982818
Class 1	0.675	0.31	0.294	0.994258	0.994176	0.994609	0.9953	0.98322
Class 2	0.953	0.521	0.523	0.984126	0.982025	0.983431	0.8732	0.952041
Class 3	0.801	0.458	0.506	0.98075	0.981253	0.980548	0.8652	0.972594
Class 4	0.827	0.442	0.453	0.983918	0.984338	0.984617	0.9325	0.973357
Class 5	0.864	0.431	0.436	0.992295	0.989994	0.992676	0.8648	0.923452
Class 6	0.849	0.492	0.593	0.984011	0.98097	0.983624	0.9712	0.943311
Class 7	0.682	0.401	0.398	0.997476	0.99894	0.997108	0.9364	0.98485
Class 8	0.856	0.392	0.523	0.991551	0.989815	0.994305	0.8854	0.96304
Class 9	0.558	0.368	0.358	0.999386	0.998982	0.99908	0.9354	0.944933
Mean	0.7946	0.4438	0.4867	0.9906036	0.9898543	0.9907674	0.92354	0.962362

Table 1. Area Under Curve results on MNIST dataset from various competitor models [15, 16] and EIDN.

Class	GANomaly	AnoGAN	EGBAD	DenseNet-169	ResNet-152	Inception-V4	OC-NN(.1)	EIDN(.1)
plane	0.633	0.516	0.577	0.998449	0.998071	0.930263	0.6042	0.786571
car	0.631	0.492	0.514	0.998933	0.998203	0.971474	0.6197	0.854727
bird	0.51	0.411	0.383	0.99498	0.995249	0.84234	0.6366	0.699063
cat	0.587	0.399	0.448	0.992014	0.991605	0.853591	0.5357	0.724128
deer	0.593	0.335	0.374	0.998145	0.99848	0.895042	0.674	0.749263
frog	0.683	0.321	0.353	0.991758	0.991375	0.893674	0.6331	0.849284
horse	0.605	0.399	0.526	0.999031	0.999607	0.949273	0.6009	0.811566
ship	0.616	0.567	0.413	0.998386	0.999289	0.921899	0.6467	0.85433
truck	0.617	0.511	0.555	0.998948	0.998934	0.954804	0.6032	0.833514
dog	0.628	0.393	0.481	0.998291	0.9979	0.931945	0.5611	0.750012
Mean	0.6103	0.4344	0.4624	0.9968935	0.9968713	0.9144305	0.61152	0.791246

Table 2. Area Under Curve results on CIFAR10 dataset from various competitor models [15, 16] and EIDN.

Class	GANomaly	DenseNet-169	ResNet-152	Inception-V4	EIDN(.5)
Cracks	0.858	0.999998	0.999986	0.998462	0.999645

Table 3. Area Under Curve results on Concrete Cracks dataset from various competitor models [15] and EIDN.

4 Experiments

We tested the EIDN on three different image datasets - MNIST [18], CIFAR10 [19] and Concrete Cracks [20]. For MNIST and CIFAR10 we used a one-versus-the-rest approach [15], i.e. one class would be considered normal and the rest would be considered anomalous. The MNIST dataset (hand-written digits) consists of 60,000 28x28 grayscale hand-written training images and 10,000 test images, where all ten classes are fairly balanced. The CIFAR10 dataset consists of 50,000 32x32 RGB training images and 10,000 test images, evenly distributed on

ten object categories. The Concrete Cracks dataset consists of 40,000 227x227 RGB images of concrete, equally split between concrete images without cracks (normal class) and with cracks (anomalous class); We reduced the dimension size of the Concrete Cracks dataset to 32x32 RGB images, concatenated the normal and anomalous samples and randomly assigned 80% to be the training set and the other 20% to be the testing set. Examples of normal and anomalous class splits can be found in Fig.3.

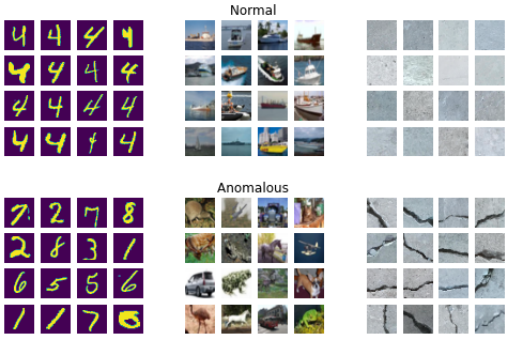

Figure 3. Dataset examples of the MNIST and CIFAR10 datasets of arbitrarly chosen normal samples (Class 4 and ship) against their respective anomalous samples (all other classes) and Concrete Cracks dataset normal samples (without cracks) against anomalous samples (with cracks).

To demonstrate EIDN's ability to cope with unbalanced data we first trained an EIDN on completely balanced data and then trained other EIDNs on further reduced anomalous data sample sizes; we reduced the training set anomalous sample sizes to be 100%, 50%, 10%, 5% and 1% of the training set normal sample sizes for each dataset.

We defined f to map colour images to a two dimensional space, to allow visualisation of results (equation 6),

$$f : R^{l \times w \times c} \rightarrow R^2 \tag{6}$$

where l, w, c represents the *length*, *width* and *channels* of an image respectively. We chose a simple network architecture (see Fig.4) to demonstrate the effectiveness of our method.

We trained a unique EIDN (f) for each anomaly detection class, for each ratio between anomalous and normal training samples, on every dataset, totalling 105 networks (50 for MNIST, 50 for CIFAR10, 5 for Concrete Cracks). To apply the Triplet Loss Anomaly Detection, we created another network with a single Triplet Loss Function Layer which inputs $[y, x^p, x^n]$ and outputs nothing. We trained the networks for 50 epochs, at 100 steps per epoch with a mini-batch size of 128 triplets, using an Adam Optimizer with $\alpha = 0.2$.

We calculated the Euclidean distance of each sample from the origin on the embedded space and scored those values on an Area Under Curve (AUC) for all 105 f EIDNs. For the MNIST and CIFAR10 datasets we compare our results to other models

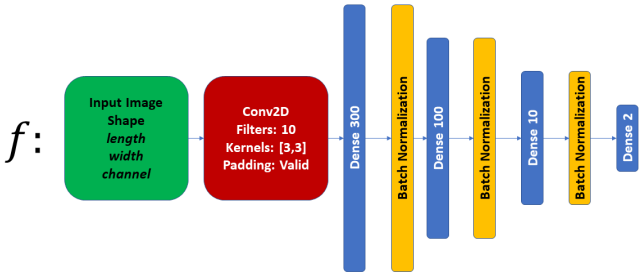

Figure 4. Architecture of the neural network f used to be trained on the MNIST, CIFAR10 and Concrete Cracks datasets. The neural network takes inputs any image and outputs a two dimensional space.

when the anomalous sample size is 1% and 10% of normal samples respectively to also fairly compare against the OC-NN results [16], shown in Tables 1 and 2. For the Concrete Cracks dataset we compare our results when the anomalous sample size is 50% of normal samples, shown in Table 3.

To demonstrate the effectiveness of the EIDN methodology we plot the mean AUC results for each dataset as anomalous sample size percentage drops from a completely balanced dataset to a heavily unbalanced dataset and benchmark them against other supervised (DenseNet-169, ResNet-152, Inception-V4), semi-supervised (GANomaly, AnoGAN, EGBAD) or unsupervised OC-NN) methods, shown in Fig.5. The results indicate that with a small proportion of labelled anomalous samples (1%, 5% and 10% respectively for the MNIST, CIFAR 10 and the Concrete Cracks datasets), performance is significantly improved in comparison to methods that ignore such labels and in most cases close to state-of-the-art supervised networks trained on balanced datasets.

The EIDN method is capable of coping with never-before-seen anomalies and still yield strong results. To demonstrate this ability, we trained an EIDN on the MNIST dataset with normal samples from class 0, and 5% anomalous samples from classes 1, 2, 3, 4 and 5. The EIDN was then tested on the test set's class 0, the normal class, and classes 6, 7, 8 and 9 to represent the never-before-seen anomalous samples. The result yielded an AUC score of 99.45%. A qualitative and quantitative (based on the average distance of anomalies from the Normal Sample Cluster Centre) comparison against PCA and T-SNE (Fig.6) demonstrates the effectiveness of our method to detect unseen types of anomalies.

To demonstrate the effectiveness of the EIDN's embedded space, we dimensionally reduced the MNIST and CIFAR10's training sets to be labelled as the normal samples using both EIDNs (which had been trained on an anomalous sample size which was 5% the normal samples size) and Linear Discriminant Analysis (LDA), another supervised learning dimensional reduction method. We then applied a variety of classical machine learning methods on the embedded normal samples and computed their AUC score. Fig.7 confirms the suitability of the EIDN embedded space for a wide range of methods.

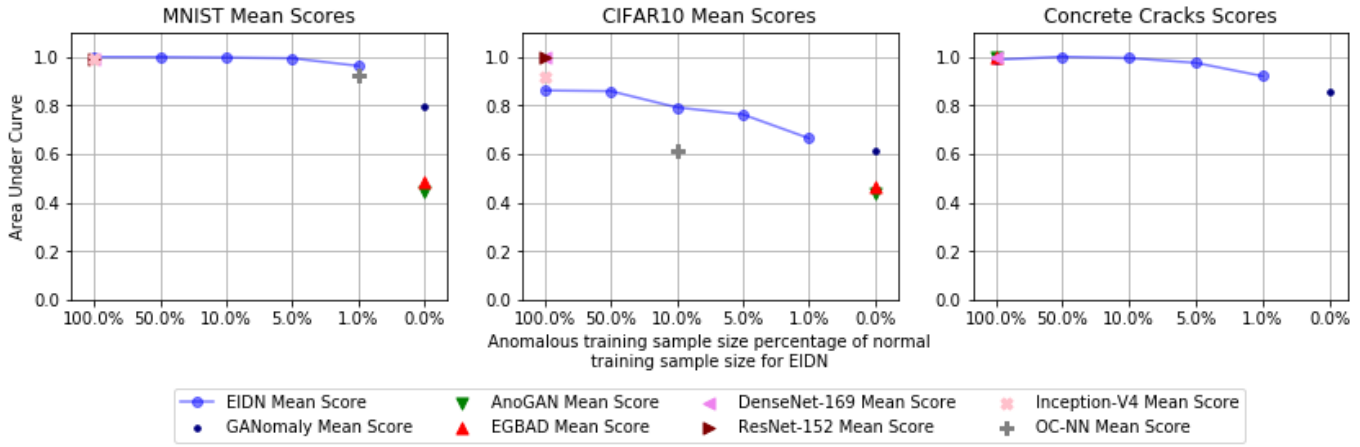

Figure 5. Mean AUC results for MNIST, CIFAR10 and Concrete Cracks datasets as anomalous sample sizes for drop from a completely balanced dataset to a heavily unbalanced dataset.

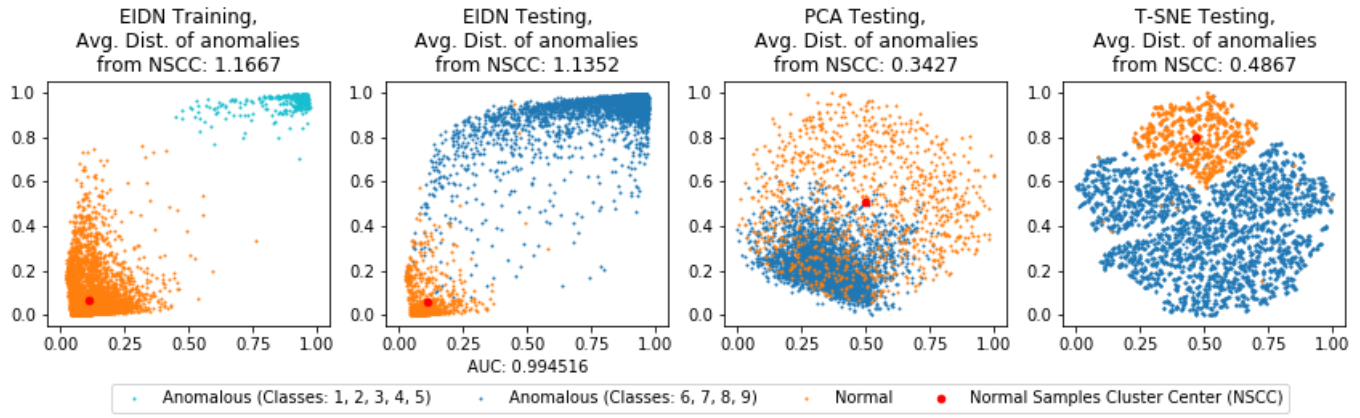

Figure 6. Scatter plots of the MNIST data on the embedded spaces derived by EIDN (both training and testing datasets), PCA and T-SNE (only testing dataset).

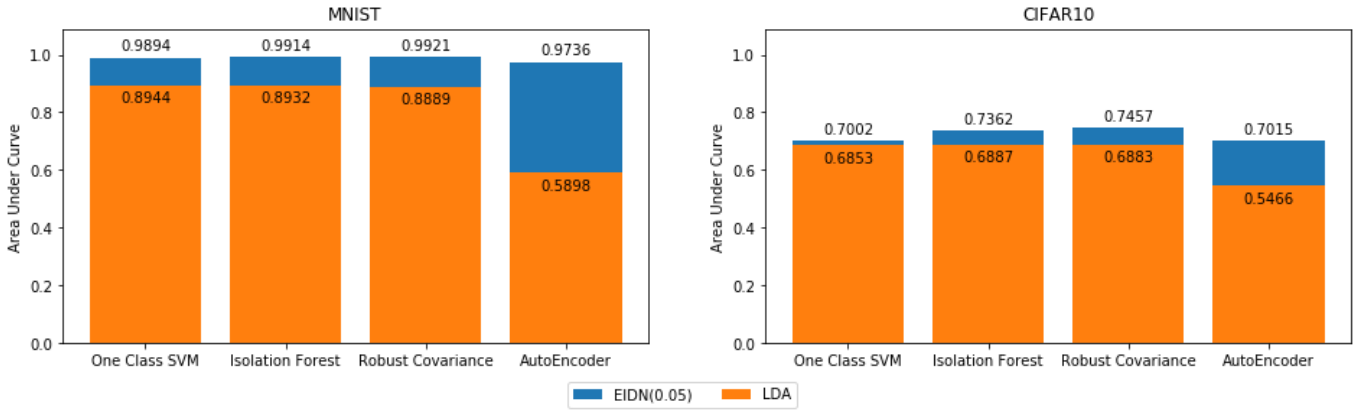

Figure 7. Mean AUC scores across all classes of MNIST and CIFAR10 for various classic anomaly detection methods after dimensionality reduction using EIDN or LDA.

5 Conclusion

In this work we proposed EIDN, a novel Anomaly Detection approach, which utilises a refined Triplet Loss function that enforces normal and anomalous samples to be isolated within an embedded space. We introduced Random Triplet Sampling to properly handle unbalanced training datasets. We evaluated the suggested methodology on three publicly available datasets, across various quantities of unbalanced data sample sizes, and compared our method to the current state of the art. Our method, despite its simplistic network architecture, achieved results comparable with state-of-the-art supervised methods, even in cases where it was trained with heavily unbalanced datasets. When comparing to semi-supervised, approaches, our method can deliver significant improvement by including relatively few labeled anomalous samples in the training dataset. Further improvements could encompass a more complex neural network, which maps to a higher dimensional embedded space, and future work could apply the model to more convoluted and challenging datasets, such as videos.

References

[1] R. Chalapathy and S. Chawla, "Deep Learning for Anomaly Detection: A Survey," pp. 1–50, 2019. arxiv: 1901.03407.

[2] F. Tony Liu, K. Ming Ting, and Z.-H. Zhou, "Isolation Forest ICDM08," *Icdm*.

[3] M. Alshawabkeh, B. Jang, and D. Kaeli, "Accelerating the local outlier factor algorithm on a GPU for intrusion detection systems," *International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS*, pp. 104–110, 2010.

[4] L. Xu, Y. R. Yeh, Y. J. Lee, and J. Li, "A hierarchical framework using approximated local outlier factor for efficient anomaly detection," *Procedia Computer Science*, vol. 19, pp. 1174–1181, 2013.

[5] D. Tran, W. Ma, and D. Sharma, "Network Anomaly Detection using Fuzzy Gaussian Mixture Models," *International Journal of Future Generation . . .* , pp. 37–42, 2006.

[6] R. Laxhammar, "Anomaly detection for sea surveillance," *Proceedings of the 11th International Conference on Information Fusion, FUSION 2008*, no. January 2008, 2008.

[7] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[8] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10262 LNCS, pp. 189–196, 2017.

[9] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," *ACM International Conference Proceeding Series*, vol. 02-Decembe, pp. 4–11, 2014.

[10] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. Part F1296, pp. 665–674, 2017.

[11] Y. Lu and P. Xu, "Anomaly Detection for Skin Disease Images Using Variational Autoencoder," 2018. arxiv: 1807.01349.

[12] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "GANomaly: Semi-supervised Anomaly Detection via Adversarial Training," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11363 LNCS, pp. 622–637, 2019.

[13] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient GAN-Based Anomaly Detection," 2018. arxiv: 1802.06222.

[14] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10265 LNCS, pp. 146–147, 2017.

[15] M. S. Minhas and J. Zelek, "Anomaly Detection in Images," 2019. arxiv: 1905.13147.

[16] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly Detection using One-Class Neural Networks," 2018. arxiv: 1802.06360.

[17] A. Zhou, Z. Li, and Y. Shen, "Anomaly detection of CAN bus messages using a deep neural network for autonomous vehicles," *Applied Sciences (Switzerland)*, vol. 9, no. 15, 2019.

[18] Y. B. Y. LeCun, L. Bottou and P. Haffner, "Gradient-based learning applied to document recognition.," 1998. Proceedings of the IEEE, 86(11):2278-2324.

[19] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." http://www.cs.toronto.edu/ kriz/cifar.html.

[20] Y. D. Z. Lei Zhang, Fan Yang and L. Y. F. Z. Y. D. . Z. Y. J. Y. J. Z., Zhang, "Road crack detection using deep convolutional neural network.," 2016. In 2016 IEEE International Conference on Image Processing (ICIP).