Continents

1 Introduction 2

2 Related Work 2

3 Deep One-Class Classification 3
   3.1 Learning the representation(mapping) 3
   3.2 Objective Functions 3
      3.2.1 Soft-Boundary Deep SVDD 3
      3.2.2 One-Class Deep SVDD 3
   3.3 Choosing the centre c 3

4 Proposed Improvements 3
   4.1 Robustness 4
   4.2 Supervision 4

5 Experiments and Results 4
   5.1 Dataset 4
   5.2 Basic Setup 4
   5.3 Modified Setup 5
   5.4 Robustness Setup 5
      5.4.1 Results 5
   5.5 Supervision Setup 7
      5.5.1 Results 7

6 Conclusion 8

7 Future Work 8
1 Introduction

Anomaly detection is defined as a task aimed at identifying anomalous samples that differ significantly from the supposedly normal data samples given in a dataset.

Typically this task of Anomaly detection is often modelled as an unsupervised learning task. Most of the approaches try to model the normal data distribution during the training phase and aim to detect the outlier/anomaly encountered during the test time by relying on its deviation from the learned model. Often we have only normal data with us for training, and hence most of the models are based on this assumption.

Our work attempts to investigate the case, which involves the presence of outliers in the given training data and their effect on selected Anomaly detection approaches. We also attempt to explore the ways to use any available supervision provided in training data.

2 Related Work

Majority of existing approaches for anomaly detection (AD) approaches can be broadly divided into two categories. One, the classical AD approaches involving various kernel-based methods. Other one, the deep learning-based methods involving a wide variety of architectures.

The classical AD approaches involve approaches like One-Class Support Vector Machine (OC-SVM), kernel density estimation, Support Vector Data Description (SVDD), among others. OC-SVM tries to find the largest margin hyperplane that separates the mapped data from the origin in the feature space. SVDD also employs a similar approach in which it tries to find a hypersphere with minimum volume that best encloses the mapped data points in feature space. Both these methods attempt to learn a compact representation of the normal data in some feature space in order to model the normal behaviour of the given data.

Though these methods have a clear objective and specifically Anomaly Detection targeted approach, at times they fail at the task of scalability. These methods often are susceptible to fail in tasks involving high dimensional data. Moreover, these kernel methods many time require extensive feature engineering for good performance, a common shortcoming that many other kernel approaches suffer.

There has been a significant development in the Deep learning based approaches. Some of the commonly used deep learning approaches involve the use of Autoencoders. The primary way to use Autoencoder for Anomaly detection involves the learning to reconstruct the training data samples and use the reconstruction error as the anomaly score. There are various types of autoencoders which can be used for AD anomaly detection tasks. Some of them include denoising autoencoders, variational autoencoders, Deep Convolutional Autoencoders (primarily used for image data). Another way to use deep learning in anomaly detection is to use autoencoders to learn a compressed representation and then plugging this representation as features to the classical approaches.

One interesting approach belonging to the deep learning based methods is AnoGAN\cite{4}. This approach involves learning the normal data distribution using a GAN, where the generator learns to generate normal data samples. During testing any data sample encountered is mapped back to the latent space of the generator. This latent space representation is further used to recreate the data sample. The anomaly score is the reconstruction error in this process. The AnoGAN employs an iterative method in finding the closest possible hidden representation, which is, to an extent slow and computationally expensive. In a recent development, the authors of AnoGAN were able to overcome this shortcoming by employing a better method of inverting to latent space, hence the name f-AnoGAN (fast AnoGAN).
One shortcoming for the deep learning approaches is that these approaches do not directly target Anomaly detection and instead try to use reconstruction error.

3 Deep One-Class Classification

This approach \(\Pi\) aims to introduce a deep learning approach for anomaly detection which is based on the lines of \textit{SVDD}. The said approach introduces a neural network based approach which maps a the normal data into a feature space. The objective is similar to \textit{SVDD} in which the learned representation from the neural network is enclosed in a sphere of minimum volume. The approach proposes a an objective function which targets directly the anomaly detection. Our further work is based on improving this approach.

3.1 Learning the representation(mapping)

For input domain \(\chi \subseteq \mathbb{R}^d\) and output feature space \(\mathcal{F} \subseteq \mathbb{R}^p\) the neural network represented by \(\phi(\cdot; \mathcal{W})\) maps a data sample \(x_i \in \chi\) to the compressed representation \(\phi(x_i; \mathcal{W}) \in \mathcal{F}\). The DeepSVDD aims to learn this network in order to map the training data samples to hypersphere in output feature space with minimum volume, represented by centre \(c \in \mathcal{F}\) and radius \(R\).

3.2 Objective Functions

The authors describes 2 two objective functions.

3.2.1 Soft-Boundary Deep SVDD

\[
\min_{R, \mathcal{W}} R^2 + \frac{1}{\nu n} \sum_{i=1}^{n} \max\{0, ||\phi(x_i; \mathcal{W}) - c||^2 - R^2\} + \frac{\lambda}{2} ||\mathcal{W}||_F^2
\]

3.2.2 One-Class Deep SVDD

\[
\min_{\mathcal{W}} \frac{1}{n} \sum_{i=1}^{n} ||\phi(x_i; \mathcal{W}) - c||^2 + \frac{\lambda}{2} ||\mathcal{W}||_F^2
\]

For our later experiments we have used the one-class SVDD objective as it was shown by the authors to slightly out perform the former objective function.

3.3 Choosing the centre \(c\)

It is explained by author that selection of a non-zero centre \(c\) is essential for preventing the representation from trivially collapsing to a single point. Initially the network \(\phi(\cdot; \mathcal{W})\) is trained as a part of an autoencoder for network pretraining. This is done by learning to reconstruct the given data samples in training set.

The \(c\) is chosen as the mean of the compressed representation obtained by passing all the data samples through the network.

\[
c = \frac{1}{n} \sum_{i=1}^{n} \phi(x_i; \mathcal{W})
\]

4 Proposed Improvements

We propose improvement to the existing approach with two targets.
4.1 Robustness

We aim to make the existing approach robust to the outliers/anomalous examples that might be present in the assumed normal training dataset. Though the assumption of only normal samples being present in the available training data is true to some extent, it is highly possible that a portion of data is anomalous or corrupted. We aim analyse and mitigate the effects of such situation on the proposed DeepSVDD approach.

To reduce the effects of outlier from skewing the training objective’s loss term, we raised the distance squared between data sample mapping and the hypersphere centre \( \| \phi(x_i; W) - c \|^2 \) to power \( \frac{p}{2} \). Hence the loss term becomes:

\[
\frac{1}{n} \sum_{i=1}^{n} (\| \phi(x_i; W) - c \|^2 + \epsilon)^{\frac{p}{2}}
\]

The \( \epsilon \) to deal with the possible non-differentiability.

4.2 Supervision

The second task that we look to accomplish is to find ways to leverage any available supervision in the form of labelled outlier/anomalous data present in the training dataset.

To achieve this we adopt a simple approach for our current initial investigation. We simply subtract the loss term corresponding to the labelled anomalous samples. The additional terms become:

\[
\frac{-\gamma}{n} \sum_{i=1}^{n} (\| \phi(x_i; W) - c \|^2 + \epsilon)^{\frac{p}{2}}
\]

where \( \gamma \) is a hyper parameter.

5 Experiments and Results

5.1 Dataset

For all our current experiments we used MNIST handwritten digit dataset.

- The training set contains 60,000 images roughly divided equally across all 10 digits (0-9).
- Test set contains 10,000 images with 1000 images for all 10 digit classes.

5.2 Basic Setup

For the basic setup i.e. without any modification in the original DeepSVDD approach the setup is described in this section.

The normal class for the Anomaly Detection task is chosen to be one of the 10 digit classes. All other classes serve as the anomalous classes.

- The training set comprises of roughly 6000 images of the chosen normal class from the MNIST training set.
- The test set comprises of complete 10,000 images of MNIST test set.

The evaluation metric used is the AUC (Area under the curve) of the ROC (receiver operating characteristic) curve.
5.3 Modified Setup

Since MNIST dataset is a very simple dataset, their is very less scope of improvement in the baseline results.

To address this problem we chose a subset of 600 samples from the roughly 6000 large training dataset for each class. We chose these 600 samples in order to represent the maximum intra-class variation present in the data. To accomplish this we use the greedy approximate solution to **k-centre** problem\[^2\].

**K-centre problem**  Given a set P of n points, chose a set C of k\(|\leq\)n points such that the maximum distance of any point in P to its nearest point in set C is minimized.

- The training set now comprises of roughly 600 images of the set C for the k-centre problem solved above.
- The test set comprises of complete 10,000 images of MNIST test set.

5.4 Robustness Setup

To observe the effect of outliers/anomalous samples in training data on the AD approach, we modify the training data by adding different amounts of anomalous and corrupted data.

- We add 10% (60 samples) and 20% (120 samples) of outliers in the training data.
- Out of the the above 60/120 samples 80% (48/96) are samples from other classes
- The remaining 20% (12/24) are random noise with similar mean intensity to the original samples

5.4.1 Results

We plotted the AUC score for various experiments.

- Figure 1a shows the mean AUC score values v/s p-values for different levels of impurity for anomalous class 8. The highest mean AUC score for impurities 10% and 20% occurs at p=0.6.
- Figure 1b shows the mean AUC score values v/s impurity for different levels of p for anomalous class 8. The least reduction in AUC score after impurity addition is observed for p=0.6.
- Figure 2a shows the mean AUC score values v/s p-values for different levels of impurity for all 9 anomalous classes. The highest mean AUC score for impurities 10% and 20% occurs at p=0.8 and p=0.4 respectively.
- Figure 2b shows the mean AUC score values v/s impurity for different levels of p for all 9 anomalous classes. The least reduction in AUC score after impurity addition is observed for p=0.4.
- Figure 3a shows the mean AUC score values v/s p-values for different levels of impurity for anomalous classes 4,7,9. This result is not very conclusive but general trend shows that AUC score for 10% and 20% impurity is closer to that of pure dataset for lower values of p.
- Figure 3b shows the mean AUC score values v/s impurity for different levels of p for anomalous classes 4,7,9. The least reduction in AUC score after impurity addition is observed for p=0.8.
Figure 1: Normal Class 3 Anomalous Class 8

Figure 2: Normal Class 3 Anomalous all 9 Classes

Figure 3: Normal Class 3 Anomalous Classes 4,7,9
5.5 Supervision Setup

To observe the effect of labelled outliers/anomalous samples provided in training data on the AD approach, we modify the training data by adding different amounts of labelled anomalous and corrupted data. We then trained the models with the additional term for labelled outlier to observe the effects of supervision.

- We add 10% (60 samples) and 20% (120 samples) of outliers in the training data.
- Out of the above 60/120 samples 80% (48/96) are samples from other classes
- The remaining 20% (12/24) are random noise with similar mean intensity to the original samples

5.5.1 Results

Results for mean AUC score v/s supervision percentage for various amounts of impurities added. The value of \( p \) used for these experiment is \( p=0.6 \).

- Figure 4 Shows mean AUC score v/s Supervision percentage for different values of impurity added for normal class 3 anomalous class 8 supervision class 8.
- Figure 5 Shows mean AUC score v/s Supervision percentage for different values of impurity added for normal class 3 anomalous all 9 classes supervision all 9 classes.
- Figure 6 Shows mean AUC score v/s Supervision percentage for different values of impurity added for normal class 3 anomalous class 4,7,9 supervision classes 2,8.

The general trend for mean AUC score for these result is an increase as the supervision is increased while decrease as the impurity percentage is decreased.

Figure 4: Normal Class 3 Anomalous Class 8 Supervision class 8
6 Conclusion

The robustness modification approach seems to be working fine for some of the normal-anomalous class(es) combination. It can be concluded that for lower p the results are better. Though it would not be correct to rely heavily on these results since the variability in result is very high due to small training set.

Similarly the supervision approach seems to be working good if we consider the increase in the mean AUC score. Though the variability in results make these conclusions less reliable here as well.

7 Future Work

Need more analysis for the variability of trends across various combinations of normal-anomalous classes. Need to observe the performance for various type of classes when the added impurity belongs to a particular class i.e. skewed in proportion. Similar analysis to be performed for supervision as well.

One more line of work would be to try out the similar experiments for more sophisticated real world datasets. One good option is MVTec anomaly detection dataset.

These ideas of robustness and supervision can be extended for other Anomaly detection frameworks as well. Namely AnoGAN/f-AnoGAN
Figure 6: Normal Class 3 Anomalous Classes 4,7,9, Supervision classes 2,8

References

[1] Deep One-Class Classification, Ruff et al. ICML 2018