OPEN-SET OBJECT RECOGNITION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Salman Mohammad
July 2022

Open-set object Recognition
By Salman Mohammad
July 2022

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Shervin R. Arashloo(Advisor)

Hamdi Dibeklioğlu

Gökçe Nur Yılmaz

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

OPEN-SET OBJECT RECOGNITION

Salman Mohammad
M.S. in Computer Engineering
Advisor: Shervin R. Arashloo
July 2022

Despite significant advances in object recognition and classification over the past couple of decades, there are various situations where collecting representative training samples from all classes in real-world scenarios is quite expensive, or the system may be exposed to unpredictable novel samples at the test time. The pattern classification problem is commonly referred to as an open-set recognition task in such cases where limited and incomplete knowledge of the entire data distribution is provided to the model during the training time. During test phase, unknown classes can be faced which requires the classifier to accurately classify the previously seen classes while effectively rejecting unseen ones. Among others, one-class classification serves as a plausible solution to the open-set recognition problem. Nevertheless, current one-class classifiers have their limitations. Classical kernel-based approaches require carefully designed features to obtain reasonable performance but rest on a solid basis in statistical learning theory, providing good robustness against training set impurities. More recent deep learning-based methods, on the other hand, focus on learning relevant features directly from the data but typically rely on ad hoc one-class loss functions, which very often do not generalize well and are not robust against the omnipresent noise and contamination in the training set. In this thesis, we introduce a novel approach which leverages the advantages of both kernel-based and deep-learning approaches by bringing the two learning formalisms under a common umbrella. In particular, the proposed method learns deep convolutional features to optimize a kernel Fisher null-space loss subject to a Tikhonov regularisation on the discriminant in the Hilbert space. As such, it can be trained in a deep end-to-end fashion while being robust against training set contamination.

Through extensive experiments conducted on different image datasets in various evaluation settings, the proposed approach is shown to be quite robust and more effective than the current state-of-the-art methods for anomaly detection

in the scenario where the training set is corrupted and contains noisy samples. At the same time, the proposed approaches can be effectively utilized in an unsupervised scenario to rank the data points based on their conformity with the majority of samples.

Keywords: Open-set classification, One class classification, deep end-to-end learning, reproducing kernel Hilbert space, regularization, contamination..

ÖZET

AÇK KÜME NESNESI TANMA

Salman Mohammad
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Shervin R. Arashloo
Temmuz 2022

Son birkaç on yılda nesne tanıma ve sınıflandırmadaki önemli ilerlemelere rağmen, gerçek dünya senaryolarında tüm sınıflardan temsili eğitim örnekleri toplamanın oldukça pahalı olduğu veya sistemin açıkta kalabileceği çeşitli durumlar vardır. test zamanında öngörülemeyen yeni örneklere. desen sınıflandırma problemi genellikle açık küme olarak adlandırılır. Tanıma görevi, eğitim süresi boyunca modele tüm veri dağıtımına ilişkin sınırlı ve eksik bilgi verildiği durumlarda. Test aşamasında, sınıflandırıcının daha önce görülen sınıfları doğru bir şekilde sınıflandırmasını ve görünmeyenleri etkili bir şekilde reddetmesini gerektiren bilinmeyen sınıflarla karşılaşılabilir. Diğerlerinin yanı sıra, tek sınıflı sınıflandırma, açık küme tanıma problemine makul bir çözüm olarak hizmet eder. Bununla birlikte, mevcut tek sınıflı sınıflandırıcıların sınırlamaları vardır. Klasik çekirdek tabanlı yaklaşımlar, makul performans elde etmek için dikkatlice tasarlanmış özellikler gerektirir, ancak eğitim kümesi safsızlıklarına karşı iyi bir sağlamlık sağlayarak istatistiksel öğrenme teorisinde sağlam bir temele dayanır. Diğer yandan, daha yeni derin öğrenme tabanlı yöntemler, ilgili özellikleri doğrudan verilerden öğrenmeye odaklanır, ancak genellikle genellikle iyi genellemeyen ve her yerde mevcut olan gürültüye karşı sağlam olmayan geçici tek sınıf kayıp işlevlerine dayanır. eğitim setinde kontaminasyon. Bu tezde, iki öğrenme formalizmini ortak bir çatı altında toplayarak hem çekirdek tabanlı hem de derin öğrenme yaklaşımlarının avantajlarından yararlanan yeni bir yaklaşım sunuyoruz. Özellikle, önerilen yöntem, Hilbert uzayındaki diskriminant üzerinde bir Tikhonov düzenlemesine tabi bir çekirdek Fisher boş-uzay kaybını optimize etmek için derin evrişimsel öznitelikleri öğrenir. Bu nedenle, eğitim seti kontaminasyonuna karşı sağlam iken derin bir uçtan uca şekilde eğitilebilir.

Çeşitli değerlendirme ayarlarında farklı görüntü veri setleri üzerinde yapılan kapsamlı deneyler sayesinde, önerilen yaklaşımın eğitim setinin bozulduğu ve içerdiği senaryoda anomali tespiti için mevcut en son teknoloji yöntemlerden

oldukça sağlam ve daha etkili olduğu gösterilmiştir. gürültülü örnekler Aynı zamanda, önerilen yaklaşımlar, denetimsiz bir senaryoda, veri noktalarını örneklerin çoğuna uygunluklarına göre sıralamak için etkin bir şekilde kullanılabilir.

Anahtar sözcükler: Açık küme sınıflandırma, Tek sınıf sınıflandırma, derin uçtan uca öğrenme, çekirdek Hilbert uzayını yeniden oluşturma, düzenlileştirme, kontaminasyon..

Acknowledgement

The completion of this thesis would never be possible without the exceptional tolerance, magnificent knowledge on the field, and of course, cordially professional behavior of my advisor, Prof. Shervin R. Arashloo, who was nothing less than a perfect advisor.

I would also like to express my deepest gratitude to my parents and my siblings for their amazing support, understanding and encouragement throughout my life. Without their support, it would be impossible for me to complete my study. Also, thank you to all my friends for making my life more enjoyable with their friendship and support through good times and hard times.

Mathematical Notations

Symbol	Meaning
$\mathcal{Z}$	Space for all data points
$\mathbf{z}$	Sample data point
$\bar{\mathbf{z}}$	Feature extracted for test sample
$\mathcal{X}$	Subspace for data points for a particular class
$\mathbf{X}$	Training set
$\mathbf{x}$	A sample from training set
$\mathbf{y}$	Label vector
y	Label of a data point
ψ	Basis of a subspace
$\mathbf{G}$	Criterion function for null-space
$\mathbf{S}_w$	Within class scatter
$\mathbf{S}_b$	Between class scatter
C	Number of Classes
ϕ	Projection function for One-class spectral regression
$\mathbf{K}$	RBF Kernel Matrix
$\boldsymbol{\alpha}$	Coefficients/weights of training samples
δ	Tikhonov regularization coefficient
γ	RBF kernel width
λ	L2 Regularization coefficient
Ψ	Neural Network
$\boldsymbol{\omega}$	Weights of the network
$\mathcal{L}$	Number of layers in a network
$\mathbf{I}$	Identity matrix
$\mathbf{O}$	Feature matrix of the training set
$\mathbb{R}$	The set of real numbers
bs	Number of samples in a batch

Acronyms

Acronym	Meaning
ALOCC	Adversarially Learned One-Class classifier
AUC	Area under the ROC curve
C-DRKSR	Closed form Deep Robust Kernel Spectral Regression
DCAE	Deep Convolutional Autoencoder
DPCP	Dual Principal Component Pursuit
DS-OCMPM	Dual slope Minimax Probability Machine
DSVDD	Deep Support Vector Data Description
FMS	Fast Median Subspace
GANs	Generative Adversarial Networks
G-DRKSR	Gradient based Deep Robust Kernel Spectral Regression
GODS	Generalized One-Class Discriminative Sub-space
HRN	Holistic approach to One-class Learning
ICS	Intra-class splitting
OCC	One-Class Classification
OCCNN	One-Class Convolutional Neural Network
OCMPM	One-class Minimax Probability Machine
OCSVM	One-class Support Vector Machine
OGN	Old is Gold
OP	Outlier Pursuit
OSR	Open-Set Recognition
PCA	Principal Component Analysis
RKHS	Reproducing kernel Hilbert Space
SRO	Self-representation based Outlier Detection
SSIM	Structural Similarity
SVM	Support Vector Machine
SVDD	Support Vector Data Description

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In a closed-set visual recognition task, it is assumed that the training and the test data points are drawn from the same label, and feature space *i.e.* a machine learning model aims to distinguish between several classes that remain the same during training and testing phase. Deep-learning-based algorithms have witnessed much success in recent years, performing on par with, if not better than, human performance in this area. The availability of large-scale annotated datasets that enable effective learning of distinctive features has contributed to the progress in this area to attain exceptional classification accuracy. However, these methods do not perform well in an open-set scenario, which is more realistic, where unseen class samples can emerge unexpectedly during the testing phase. In such cases, the model has to not only distinguish between the classes it encountered during training but also needs to reject samples from classes it has never seen before. The problem is called Open-Set Recognition (OSR) [1] which requires strong generalization and is a more difficult task than the conventional closed-set recognition problem. There have been various studies to adapt the current methods for OSR where the model has a reject option for unseen class samples [2] [3] [4] [5] [6] [7]. However, they still work under a closed-set assumption. One-class classifiers (OCC) [8] [9] [10] which are generally used for anomaly detection, are very suitable for the task of OSR. In OCC, the model tries to learn the distribution of the training data in such a way that it can be

separated from the space distant from known/seen samples. There are two ways that OCC can be used in the context of OSR: 1) treating all the known classes as a single class [11] [12]. 2) creating an OCC for each class known in the training set [4]. Although there are other methods proposed to solve OSR [13] [14] [15] [16], this work is focused on One-Class classifiers. OCC can be considered as an extreme case of OSR where only samples from one class are available during the training stage, and during the test stage, samples from unknown classes can be fed to the model.

1.1 Motivation

One-class classifiers differ from their binary/multiclass counterparts in that they mainly use samples corresponding to a single, very often the "normal" or "positive" class during training. OCC may be used to determine the unusual samples in the data [17], a problem also known as anomaly, novelty, abnormality, *etc.* detection [18]. In one-class classification, the objective is to learn a model which represents the "normal" class as closely as possible, while having the capability to keep the learned description away from any anomaly point. To be more precise, assume $\mathcal{Z}$ as the space of all possible data points and $\mathcal{X} \subseteq \mathcal{Z}$ as the collection of all samples that belong to a particular class. Using a given training set $\mathbf{X} \subset \mathcal{X}$, a one-class classifier must learn $y(\mathbf{z}) : \mathcal{Z} \rightarrow \{0, 1\}$, where $y(\mathbf{z}) = 1$ if $\mathbf{z} \in \mathcal{X}$, *i.e* $\mathbf{z}$ is a normal sample belonging to that specific class, and $y(\mathbf{z}) = 0$ otherwise, *i.e* $\mathbf{z}$ is an anomaly. Since in a one-class setting, typically, there exists no representative negative training set, learning effective feature representation or inferring a decision boundary is known to be more challenging than that of a binary/multi-class classification problem.

In spite of considerable advancements in one-class learning over the last couple of years [19, 20, 21, 22, 23, 24, 25, 26, 10, 27, 28, 29, 30, 31, 32], the majority of existing methods are not specifically designed to operate on contaminated data and implicitly presume the existence of a pure and uncontaminated positive training set. In real-life scenarios, however, the training data may be corrupted

and contaminated with noisy samples (labelled as positive) which may hinder learning an effective one-class description of the data, leading to a degradation in the performance. There are also specific applications where one is interested in inferring the degree of normality of a sample in a given dataset, *e.g.* for ranking, or fetching database items for a particular query [27]. Moreover, an effective ranking of observations in a dataset is quite desirable since it could be used to potentially enhance the performance of the classifier by either discarding the noisy samples from the training set or by using them as counter-examples to enhance the robustness of the classifier to noise. Although there are a number of non-deep approaches [27, 19, 20] that are well-suited for such an application scenario, they require substantial feature engineering. On the other hand, the recent state-of-the-art deep learning based approaches [32, 26] are not specifically equipped to provide robustness against noisy samples and may face a deterioration in the performance when training data is contaminated with noising objects. Among others, the work in [27] presents a non-deep method that operates in a reproducing kernel Hilbert space and provides robustness against noise through a Tikhonov regularisation on the discriminant in the Hilbert space. A comparison of this method against other approaches confirms its superior performance on contaminated datasets. Although the method is quite robust to contamination and can also be used to perform observation ranking effectively, it has certain limitations. In particular, as the method in [27] is unable to learn representations directly from the data, substantial feature engineering is required to extract useful features to effectively utilise this approach. Consequently, the results are highly dependent on the features being fed to the one-class learner. Moreover, at its core, the method in [27] operates on an alternating optimization approach that requires a matrix inversion operation which not only can be computationally expensive when the number of training observations is large but would also be very demanding in terms of the required system memory during the training stage which may impede its applicability to large-scale datasets.

1.2 Contributions

In this study, we present two deep one-class approaches namely - Closed-form Deep Robust Kernel Spectral Regression (C-DRKSR) and Gradient-based Deep Robust Kernel Regression (G-DRKSR) which are robust against contamination and noise in the training set by utilizing the concepts from kernel machines [27] and combining them with deep convolutional structures to enable end-to-end learning. We address the shortcomings of the current OCC methods by combining a regularised kernel-based learning formalism with a convolutional network learning approach in a way that it is trainable in an end-to-end fashion using a gradient-based method. Moreover, the proposed approaches can be used to rank samples based on how well they match the bulk of observations.

The main contributions of the proposed approaches may be summarised as:

- Combining a kernel-based learning formalism with a deep-learning-based methodology to enable learning a one-class deep kernel-based classifier in an end-to-end manner;
- A Tikhonov regularization of the RKHS discriminant to improve robustness of the proposed deep kernel-based model;
- Providing the functionality to infer soft labels for objects in a dataset reflecting their degree of normality, thus, enabling a ranking of data items through a robust deep one-class network.
- Learning the discriminant in the Hilbert space and the network parameters jointly via a batch gradient-based approach, considering the memory constraints;
- And the last but not the least, an experimental evaluation of the proposed approaches on different datasets and a comparison against existing baselines and state-of-the-art methods in different application scenarios.

1.3 Organization

In Chapter 2, we review the literature on one-class classification by categorizing these approaches into different groups. The differences between different groups their strengths and weaknesses are discussed, along with details of various methods in each category. Chapter 3 provides a brief overview of the null-space one-class Fisher analysis followed by an overview of the robust kernel method of [27] which forms the basis of our work. Chapter 4 introduces our proposed approach, including the neural networks used, the optimization process, details of the objective function, and handling the memory constraints. In Chapter 5, different flavours of the proposed approach in terms of the optimisation process are evaluated in three experimental setups: 1) with no contamination; 2) with contamination, and 3) unsupervised observation ranking and compared to the baseline and state-of-the-art methods on popular image datasets. Here contamination only refers to label contamination i.e noisy samples were included in the training set labelled as positive. Finally, In chapter 6, we summarize our work and propose possible directions for future research work.

Chapter 2

Related Work

2.1 Introduction

In this chapter, we briefly review some of the most well-known and state-of-the-art approaches for one-class classification. The methods are categorized into two groups - generative methods and discriminative approaches which are further categorized into deep vs. non-deep methods. As our work is focused on one-class classification, the emphasis in this section shall be on one-class approaches. However, for a more detailed review of generic open-set methods, the reader may refer to [33], and [34].

2.2 Categorization

Although other categorisations exist [10, 35], OCC methods may be broadly categorized into generative methods and discriminative methods.

2.2.1 Generative Methods

Generative methods focus on modelling the underlying generative process of the data, for example by learning the underlying distribution. Generative one-class classifiers may be learned following either a deep or non-deep learning methodology.

An example of the generative non-deep methods is the kernel PCA [36] approach which is a non-linear extension of PCA. It projects the training samples into an infinite dimensional feature space also known as kernel Hilbert space and extracts the principal components of the distribution of the data from this space. The novelty score is then calculated using the test sample's reconstruction error in the considered subspace. An alternative robust PCA was introduced in [37] that uses ℓ_1-norm which is less sensitive to outliers instead of the commonly used ℓ_2-norm to handle anomalies in the training set. This approach is rotational-invariant, straightforward to implement and has been proven to provide locally maximal solution. Another efficient convex optimization-based algorithm called Outlier Pursuit (OP) for the purpose of robust principal component analysis was introduced under a supposition on uncorrupted data points conform to the exact low-dimensional subspace while the corrupted data points do not, thereby identifying those points. It uses nuclear norm minimization by recovering the correct column space from the uncorrupted matrix.

A more recent approach that can potentially tolerate outliers by up to the square of the number of inliers is that of the Dual Principal Component Pursuit (DPCP) [19] which solves a non-convex ℓ_1-norm optimization problem on the sphere by recursively trying to learn a basis for the orthogonal complement of the subspace. This method provides an improvement in performance compared to some other robust PCA methods. A further improvement of DPCP was proposed recently called Noisy DPCP [38] that extended the convergence behavior and global optimality of DPCP for situations where data is corrupted by noisy samples.

Another fast and non-convex algorithm also referred to as fast median subspace

(FMS) was proposed in [39] that robustly determines the underlying subspace while being computationally less complex compared to other similar methods. It is designed such that it doesn't get affected by contamination in the training set and has been shown to converge near the correct vicinity in only a few iterations. The SRO [20], regarded to be one of the best unsupervised approach for novelty detection, is a weighted graph-based method that defines a Markov chain. It utilizes the property that data points can be represented as sparse linear combinations of one another to attain an asymmetric affinity matrix among data points. A weighted graph is obtained from the matrix and a connection between inliers/outliers and essential/inessential states is formed via an apt Markov chain. Outliers are then identified via random walks on the obtained graph.

Examples of the generative deep OCC methods include approaches based on deep autoencoders [40] which aim to learn salient features from the normal class by trying to minimize the reconstruction error. Deep autoencoders consist of an encoder network which comprises a number of convolutional layers followed by activation functions, followed by a decoder network which generally has a structure of an inverted encoder, comprising of transposed convolutional layers. Such methods are used either in conjunction with classical anomaly detection methods by providing the learned embedding as input [41, 42] to a one-class learner, or by directly utilizing the reconstruction error as the anomaly score [43, 44]. Denoising autoencoders [45] that focus on denoising a noisy version of their input, and sparse [46] and deep convolutional autoencoders [47] which encourage sparsity are some of the well known autoencoder-based anomaly detection methods.

Latent space Auto-regression (AND) [24] is a recently proposed approach that uses an auto-encoder structure to learn the probability distribution of the underlying latent representation of the data via an autoregressive procedure. It jointly optimizes a maximum likelihood objective, which behaves like a regularizer in this scenario, along with the reconstruction error of normal samples. The novelty score of a query image is obtained via a combined score of reconstruction error and the negative log likelihood of that image. A threshold is used to determine whether the query image is postive or not. ICS [26] which may be considered as one of the most successful generative method in this category trains a CNN

for one-class classification based on splitting the training data into two parts of typical and non-typical normal subsets. Initially, an auto-encoder is trained using positively labelled data followed by identifying the non-typical data points by using structural similarity (SSIM) as the similarity metric. After ICS, The three sub-networks namely - feature extraction sub-network, distance sub-network and classification sub-network are trained jointly. For a given sample, firstly, its latent representation is derived via the feature extraction sub-network, followed by classifying the sample as typical or atypical via the classification sub-network. Finally, the distance sub-network generates a score based on the latent representation of the data points. The probability of the test sample being a typical data point is used as at the time of inference to identify whether the given test sample is positive or not. One of the drawbacks of autoencoder-based approaches is the difficulty in selecting a suitable degree for compression as estimating the inherent dimensionality of the data is typically a major challenge [30].

Some other well-known generative deep OCC approaches are based on Generative Adversarial Networks (GANs) that have shown promise for anomaly detection. AnoGAN [21] is one of the first GAN-based methods used for OCC that learns a manifold of normal anatomical variability. For this purpose, initially, it trains a GAN for creating samples from the training set. For a given test sample, it then tries to map it to the latent space and find the most similar image by iteratively applying backpropagation. Adversarially Learned One-Class Classifier (ALOCC) [23] addressed the shortcoming of AnoGAN by using by utilizing a conditional GAN that takes an image and a noise vector as input unlike GAN. Old is Gold (OGN) [25] extended the work of ALOCC further by fine-tuning the ALOCC network using both bad quality images and pseudo anomaly images. For all three approaches, a pre-defined threshold is used to assign a test sample as positive or not.

Among others, OCGAN [22] is a novel hybrid approach that combines deep auto-encoders and a GAN style adversarial learning. They propose to constrain the latent space such that it only represents the given class. For that purpose, they utilize four sub-networks namely - a Denoising auto-encoder which is used as the generator during the training phase; a latent discriminator which learns

to distinguish between encoded real images and random samples drawn from the uniform distribution of the real images subspace; a visual discriminator which learns to discriminate between real and fake images; and a classifier which determines how well a provided image is similar to the content of the positive class. A common difficulty regarding the GAN-based methods is a lack of knowledge with respect to the regularisation of the generator to achieve compactness [30].

2.2.2 Discriminative Methods

Discriminative OCC methods aim to directly infer the optimal decision boundary that separates the positive class from anomalies. Similar to generative methods, discriminative one-class classifiers may be further categorized into deep or non-deep baesd approaches.

One of the most prominent discriminative non-deep methods is the OCSVM [8] that tries to learn a hyper-plane which optimizes the separation of the mapped data from the origin. It utilizes kernel SVM and treats the origin as the negative point. It tries to find a half-space where most data points lie and any data point outside of this half-space are considered anomalous. Another well-known and closely related method is that of the SVDD approach [10] where the objective is to infer the smallest hyper-sphere surrounding the training objects. The hyper-sphere is characterized by its centroid and radii. Any data point lying outside of the hyper-sphere is considered anomalous. Generalized One-Class Discriminative Sub-spaces (GODS) [28], which extends OCSVM formulation was introduced very recently. Additional to the first hyper-plane, it learns a second hyper-plane that aims for most of the data to lie in the negative space.

Just like OCSVM, the One Class Minimax Probability Machine (OCMPM) [48], tries to learn a hyper-plane with the aim of reaching a tight lower bound to the data and maximizing the distance between the origin and the hyper-plane. The only difference between OCMPM and OCSVM is that OCMPM utilizes second-order statistics during the training phase. An extension of OCMPM called

Dual slope Minimax Probability Machine (DS - OCMPM) [29] was recently introduced which had two major differences - 1) Another hyper-plane is learnt while training that considers the maximum data covariance; 2) The modeling of the training data distribution could be a collection of sub-distributions which is taken into account in this approach.

One of the more recent kernel-based approach is the one-class kernel spectral regression method [27] that tries to infer a robust one-class description in the presence of contamination in the training set by imposing a Tikhonov regularisation on the Fisher null discriminant in the Hilbert space. Determination of the optimal regularisation parameter is realised by posing the one-class learning problem as a sensitivity analysis task. Moreover, it proposed an alternating optimization which updates label confidences recursively and ranks the training data points based on their fit with the model. An evaluation of this method in the presence of noisy training data has verified its superior robustness as compared with other alternatives. All the above mentioned non-deep discriminative methods have the ability to integrate prior information about the percentage of outliers present in the training dataset into the model which is also known as υ property. Although the non-deep discriminative methods have their own merits, they require a careful feature engineering step to yield satisfactory performance in real-world scenarios.

Recently, different attempts have been made to compensate for this shortcoming of the discriminative methods. As an instance, deep SVDD (DSVDD) [30] may be considered as one of the earliest discriminative deep OCC methods which tries to combine deep learning formalism with an OCC-based objective function. For this purpose, similar to the conventional SVDD method, it tries to learn a minimum volume hypersphere enclosing the embeddings of the positive data. One-Class CNN (OCCNN) [31] serves as another deep discriminative approach that is influenced by the OCSVM method. It consists of two sub-networks serving as the classifier and the feature extractor. Inspired by the OCSVM method, the networks are trained so that the features learned are different compared to a zero-centered Gaussian distribution subject to a cross-entropy binary objective function.

ODIN [49] corresponds to another effective OCC approach that does not require any modifications to be made on a pre-trained neural network. It adds small perturbations to input and uses temperature scaling to detect anomalies which significantly improves the detection performance. Moreover, an analysis on semantic and non-semantic shifts in large scale image datasets was performed to show the circumestances when ODIN-like strategies result in improvements. One of the successful state-of-the-art OCC methods is that of HRN [32] which uses a classification network based on deep learning trained with log-likelihood loss in combination with a holistic regularization, known as H-regularization, imposed on the learned features. Furthermore, HRN proposed a new type of normalization for One-class classification called 2-norm instance level data normalization strategy to handle the various feature scales of images in the dataset.

Although successful OCC methods exist in the literature, the functionality of these methods may be seriously compromised when exposed to contamination and noise in the training set. Compared to the existing methods, the current work, building on ideas from both kernel machines and convolutional structures, presents a one-class learning approach that provides a high degree of robustness against training set impurities while enabling a direct end-to-end learning from the data.

Chapter 3

Background

3.1 Introduction

In this chapter, the Fisher null-space concept [12] [50] [51] is briefly introduced which is followed by a brief review on the Robust Kernel Spectral Regression method for One-class classification [27].

3.2 Null-Space Fisher Analysis

Fisher classification criterion is one of the most popular and widely used criterion in the area of statistical pattern classification where one attempts to find a projection function from the input space onto a subspace in such a manner that the between-class scatter is maximized while the within-class scatter is minimized of each classes provided in a given dataset. To be more precise, the criterion function $\mathbf{G}(\psi)$ can be formulated as:

$$\arg\max_{(\psi)} \mathbf{G}(\psi) = \arg\max_{(\psi)} \frac{\psi^T \mathbf{S}_b \psi}{\psi^T \mathbf{S}_w \psi} \tag{3.1}$$

where the within-class scatter and between-class scatter matrices are denoted as $\mathbf{S}_w$ and $\mathbf{S}_b$ respectively and ψ represents the basis defining the subspace. Since there are no negative instances available during the training phase of OCC, the origin could be utilized as the negative a negative example. Theoretically, a model that results in zero withing-class scatter and positive between-class scatter (also referred to as null projection) would be considered an optimal model *i.e.*

$$\begin{aligned} \psi^T \mathbf{S}_w \psi = 0, \\ \psi^T \mathbf{S}_b \psi > 0 \end{aligned} \tag{3.2}$$

The optimal solution of $\mathbf{G}(\psi)$ provided in Eq. 3.1 also referred to as a null projection function results in best separability with respect to the Fisher criterion. At max, $C-1$ null projection directions can be computed, where C represents the number of classes. For one-class classification, since there is only a single class in the training set, the eigenvector with the largest eigenvalue constitutes the optimizer which can be represented as:

$$\mathbf{S}_b \psi = a \mathbf{S}_w \psi \tag{3.3}$$

After determining the null projection direction, a sample x can be projected onto the null-space as:

$$\mathbf{y} = \psi^T \boldsymbol{x} \tag{3.4}$$

Since most data in real-life have a non-linear structure, extensions of this approach are proposed [12] [50] [51] based on kernels which are non-linear. Such methods try to make the data easily separable by projecting them into a high dimensional space which is also called reproducing kernel Hilbert space (RKHS) and require the eigen-decomposition of dense matrices.

3.3 Robust One-Class Kernel Spectral Regression

The one-class kernel spectral regression method in [27] operates by tailoring the Fisher null concept [52] to an OCC context through using the origin as a hypothetical object of the non-target class and only employing target objects for training. A maximisation of the Fisher classification criterion, in this case, entails mapping all normal/target observations to nearby positions in the optimal feature space to minimise the within-class scatter. By formulating the learning problem as regression in the Hilbert space, the work in [27] requires the response of all normal data items to be close to each other subject to a regularisation on the projection function:

$$\phi(\cdot)_{opt} = \arg\min_{\phi(\cdot)} \sum_{i=1}^{n} (\phi(\mathbf{x}_i) - y_i)^2 + \delta \|\phi(.)\|_{\mathcal{H}}^2 \ , \tag{3.5}$$

where $\mathbf{x}_i$ represents a sample from the training set and $\|.\|_{\mathcal{H}}^2$ represents the squared norm in the Hilbert space. The first term on the RHS of Eq. 3.5 measures the discrepancies of object responses w.r.t. their target values while the second term imposes a Tikhonov regularisation on function $\phi(.)$ and controls its flexibility. δ is a trade-off parameter. By operating in a reproducing kernel Hilbert space, it is shown that the projection function $\phi(.)$ may be written as [27]

$$\phi(\cdot) = \sum_{i=1}^{n} \alpha_i \kappa(., x_i), \tag{3.6}$$

where $\kappa(.,.)$ denotes the kernel function and α_i's are real coefficients to be learned. Using a matrix notation, Eq. 3.6 can be equivalently written as

$$L(\boldsymbol{\alpha}, \mathbf{y}) = \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \delta \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}, \tag{3.7}$$

where $\mathbf{K}$ denotes the kernel matrix, and $\mathbf{y}$ is a vector collection of the labels. In the equation above, a Tikhonov regularisation on $\boldsymbol{\alpha}$ serves to introduce additional restrictions that aid in uniquely determining the solution in the case where there are more variables than observations. More importantly, a Tikhonov regularisation constrains the model during the training phase and is instrumental

in enhancing the generalization ability of the method. By formulating one-class classification subject to a noisy training set as a sensitivity analysis task, the work in [27] derives the regularisation parameter δ analytically to maximise its robustness against noise. Learning the regularised one-class model above entails inferring the optimal parameter $\boldsymbol{\alpha}$ along with the labels $\mathbf{y}$ that minimise the loss function $L(\boldsymbol{\alpha}, \mathbf{y})$. Minimisation in $\boldsymbol{\alpha}$ is performed by requiring the derivative of the loss w.r.t $\boldsymbol{\alpha}$ to vanish, yielding

$$\boldsymbol{\alpha}_{opt} = (\mathbf{K} + \delta \mathbf{I_n})^{-1} \mathbf{y} \ , \tag{3.8}$$

where $\mathbf{I}_n$ represents an $n \times n$ identity matrix. For optimisation in $\mathbf{y}$, one may equate the derivative of the loss with respect to $\mathbf{y}$ to zero, which results in $\mathbf{y} = \mathbf{K}\boldsymbol{\alpha}$. The approach in [27] then iteratively updates $\boldsymbol{\alpha}$ and $\mathbf{y}$ via a block coordinate descent minimisation approach. Based on the available training dataset, for initialisation of the labels $\mathbf{y}$, two cases are considered in [27]: 1) if the training set is presumed to only contain positive data points, $\mathbf{y}$ is initialised as $\mathbf{y} = [\overbrace{1, 1, \ldots, 1}^{n}]$; and 2) if the training set is known to contain both positive and negative labelled data points, $\mathbf{y}$ is initialised as $\mathbf{y} = [\overbrace{1, \ldots, 1}^{n-n_0}, \overbrace{0, \ldots, 0)}^{n_0}]$ where n_0 represents the number of negative samples. By virtue of a Tikhonov regularisation, the method penalizes larger values of the $\boldsymbol{\alpha}$ coefficients resulting in a smaller variance in the coefficients, which has been shown to be beneficial for learning in the presence of a noisy data set. During the testing stage, Eq. 3.1 is used to project a test observation onto the subspace. The Euclidean distance between the positive class's mean and the test sample's projection is then used to calculate the anomaly score. The method in [27] has been shown to be quite robust against contamination in the dataset compared to several other alternatives.

Chapter 4

Proposed Approach

4.1 Introduction

In this section, we discuss our two proposed approaches namely Gradient based Deep Robust Kernel Spectral Regression (G-DRKSR) and Closed form Deep Robust Kernel Spectral Regression (C-DRKSR) that both extend the method of [27] to a deep end-to-end learning formalism while maintaining high robustness against noise. Moreover, we discuss the details of the objective function, the optimization and the memory constraints of both of these approaches.

4.2 Objective Function

The proposed objective function is designed to enable a joint learning of appropriate representations of the data in conjunction with optimizing a Fisher null-space one-class classification criterion subject to a Tikhonov regularisation on the discriminant in the Hilbert space. For this purpose, for some input space $\mathcal{Z} \subseteq \mathbb{R}^d$ and the real output space $\mathbb{R}$, let $\Psi(\cdot; \boldsymbol{\omega}; \boldsymbol{\alpha}, \mathbf{K}) : \mathcal{Z} \rightarrow \mathbb{R}$ denote a neural network

with $\mathcal{L}$ layers, the set of weights $\boldsymbol{\omega} = \{\boldsymbol{\omega}^1, \ldots, \boldsymbol{\omega}^{\mathcal{L}}\}$ where $\boldsymbol{\omega}^l$ are the weights associated with layer $l \in \{1, 2, \ldots, \mathcal{L}\}$. Moreover, let $\mathbf{K}$ be a positive semi-definite kernel matrix after $\mathcal{L}$ hidden layers and $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_n]^\top$ be the coefficients associated with the training samples in a given training data. Our goal in the proposed approach is to find an optimal feature space such that samples from the target/normal class are mapped onto nearby points to minimise the within-class scatter while imposing constraints on the network parameters so that it is able to handle noise and contamination. For this purpose, the proposed objective function is defined as

$$L(\boldsymbol{\alpha}, \mathbf{y}, \boldsymbol{\omega}) = \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \delta \boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha} + \lambda \|\boldsymbol{\omega}\|_2^2, \tag{4.1}$$

where δ is the Tikhonov regularization parameter for the RKHS discriminant while λ is the regularization coefficient for the weights of the network. In the proposed method, the network weights $\boldsymbol{\omega}$, the coefficients $\boldsymbol{\alpha}$ associated the training samples for the kernel-space classifier, and the soft labels $\mathbf{y}$ are learned jointly by optimizing the objective function defined in Eq. 4.1. In noisy datasets, the model may face an infinite number of decision boundaries, making it an ill-posed problem when the number of training observations are fewer than the model parameters. In such scenarios, a Tikhonov regularization is deployed to stabilize the decision boundary and create a unique solution. In our case, the Tikhonov regularization is used to maintain a balance between data fidelity and constraining the norm of the solution function *i.e* to control its smoothness. It penalises large magnitudes of the parameters to yield a small variance which is advantageous for one-class learning in the presence of a contaminated dataset. These merits of a Tikhonov regularization is instrumental in enhancing the generalization capabilities of the network as will be demonstrated through experiments in different application settings.

Note that a fundamental difference between the proposed approach in this work and that of [27] is that the kernel matrix in this study is built on the learned representations associated with the training samples which are iteratively updated during each training iteration whereas in [27], raw, or handcrafted features are use to build the kernel matrix. In Eq. 4.1, $\mathbf{y}$ denotes the vector of labels associated the

training samples which is initialized as $\mathbf{y} = [1, 1, \ldots, 1]^\top$ when no prior knowledge is available with regards to the existence of negative samples. Note that, during the training stage of the network, a soft label is inferred for each training sample by updating $\mathbf{y}$ at each iteration to reflect label confidences, discussed in the next section. A learning of the labels $\mathbf{y}$ allows the network to infer the degree of normality of a training sample in terms of a soft label, thus facilitating a ranking of training objects w.r.t. their conformity with the majority in the training set and improves the generalization capability of the network in the presence of a contaminated dataset.

4.3 G-DRKSR

In this section, we introduce our first approach called Gradient based Deep Robust Kernel Spectral Regression (G-DRKSR) where the updates are all based on gradient descent.

4.3.1 Optimization

For the optimisation of the proposed objective function w.r.t. $\boldsymbol{\alpha}$, we follow a gradient-based approach to infer the optimal coefficient vector $\boldsymbol{\alpha}$. The derivative of the loss function w.r.t. $\boldsymbol{\alpha}$ is

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = 2\mathbf{K}\left(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\right) + 2\delta\mathbf{K}\boldsymbol{\alpha}. \tag{4.2}$$

In order to update the network weights $\boldsymbol{\omega}$, since the kernel matrix $\mathbf{K}$ depends on the most recent representations of the objects during training, in addition to the regularization term $\|\boldsymbol{\omega}\|_2^2$, the derivative of the loss function w.r.t to the kernel matrix $\mathbf{K}$ needs to be computed too. As computing the gradient of $\|\boldsymbol{\omega}\|_2^2$ is straightforward, we shall focus on computing the gradient of $\mathbf{K}$. For this purpose, let $\mathbf{O}_{n\times m}$ denote the representations associated with the penultimate layer of the network where n denotes the number of training set samples and m stands for the the pen-ultimate layer dimensionality. Also, let $\mathbf{I}_{n\times n}$ stand for an identity

matrix and $\mathbf{1}_{n\times n}$ represent a constant $n \times n$ matrix of 1's. Let also $\circ$ stand for the element-wise product between two matrices, and "exp" represent an exponential function operable on a matrix in an element-wise fashion. The Gaussian (RBF) kernel function can then be represented as

$$\mathbf{K} = \exp\Big(\gamma((\mathbf{I} \circ \mathbf{O}\mathbf{O}^\top)\mathbf{1} + \mathbf{1}^\top(\mathbf{I} \circ \mathbf{O}\mathbf{O}^\top)^\top - 2\mathbf{O}\mathbf{O}^\top)\Big) \ , \tag{4.3}$$

and the derivative of the loss function with respect to $\mathbf{K}$ may be calculated as

$$\frac{\partial L}{\partial \mathbf{K}} = 2(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y})\boldsymbol{\alpha}^\top + \delta\boldsymbol{\alpha}\boldsymbol{\alpha}^\top. \tag{4.4}$$

Next, in order to update the weights, one needs to compute the gradient of the kernel matrix $\mathbf{K}$ w.r.t. $\mathbf{O}$ where $\mathbf{O}$ represents the representations obtained from the network's penultimate layer. To this end, let L be a composition function fed with the extracted features $\mathbf{O}$ as its arguments. $L(\mathbf{O})$ may be expressed as a composition function:

$$L(\mathbf{O}) = L_1(\mathbf{Q}) = L_2(\mathbf{W}) = L_3(\mathbf{K}) \ , \tag{4.5}$$

where $\mathbf{Q}$, $\mathbf{W}$, and $\mathbf{K}$ are defined as

$$\begin{aligned} \mathbf{Q} &= \mathbf{O}\mathbf{O}^\top, \\ \mathbf{W} &= ((\mathbf{I} \circ \mathbf{Q})1 + 1^\top(\mathbf{I} \circ \mathbf{Q})^\top - 2\mathbf{Q}), \\ \mathbf{K} &= \exp[-\gamma\mathbf{W}]. \end{aligned} \tag{4.6}$$

Following the rules of differentiation, one obtains

$$\begin{aligned} \partial\mathbf{Q} &= (\partial\mathbf{O})\mathbf{O}^\top + \mathbf{O}(\partial\mathbf{O}^\top), \\ \partial\mathbf{W} &= (\mathbf{I} \circ \partial\mathbf{Q}) + 1^\top(\mathbf{I} \circ \partial\mathbf{Q})^\top - 2\partial\mathbf{Q}, \\ \partial\mathbf{K} &= -\gamma\mathbf{K} \circ \partial\mathbf{W}. \end{aligned} \tag{4.7}$$

Furthermore, the differentiation of a scalar-valued matrix function yields

$$\begin{aligned} \partial L &= \mathrm{trace}\Big(\big(\frac{\partial L_3}{\partial \mathbf{K}}\big)^\top \partial\mathbf{K}\Big) \\ &= \mathrm{trace}\Big(\big(-\gamma\mathbf{K} \circ \frac{\partial L_3}{\partial \mathbf{K}}\big)^\top \partial\mathbf{W}\Big) \\ &= \mathrm{trace}\Big(\big(\frac{\partial L_2}{\partial \mathbf{W}}\big)^\top \partial\mathbf{W}\Big). \end{aligned} \tag{4.8}$$

Using the equation above, one obtains

$$\frac{\partial L_2}{\partial \mathbf{W}} = -\gamma \mathbf{K} \circ \frac{\partial L_3}{\partial \mathbf{K}} \ . \tag{4.9}$$

$\frac{\partial L_3}{\partial \mathbf{K}}$ is nothing but the differential of the loss function w.r.t the kernel matrix $\mathbf{K}$ which is derived in Eq. 4.4. In a similar fashion, one may derive:

$$\begin{aligned} \frac{\partial L_1}{\partial \mathbf{Q}} &= \mathbf{I} \circ \Big(\Big(\frac{\partial L_2}{\partial \mathbf{W}} + (\frac{\partial L_2}{\partial \mathbf{W}})^\top \Big) \mathbf{1}^\top \Big) - 2 \frac{\partial L_2}{\partial \mathbf{W}} \\ \frac{\partial L}{\partial \mathbf{O}} &= \Big(\frac{\partial L_1}{\partial \mathbf{Q}} + (\frac{\partial L_1}{\partial \mathbf{Q}})^\top \Big) \mathbf{O}. \end{aligned} \tag{4.10}$$

Once the derivatives are computed, one may apply a backpropagation through the entire network to update the weights. After updating the network parameters, similar to [27], optimisation w.r.t. $\mathbf{y}$ requires the gradient of the objective function with respect to $\mathbf{y}$ to vanish followed by an ℓ_2 normalisation, *i.e.*

$$\mathbf{y} = \frac{\mathbf{K}\boldsymbol{\alpha}}{\|\mathbf{K}\boldsymbol{\alpha}\|_2}. \tag{4.11}$$

4.3.2 Memory constraints

During the training phase, using a large number of training samples, manipulating a large kernel matrix may lead to large requirements in terms of system memory. In order to alleviate this problem, in this study, we follow a batch learning scheme and calculate the kernel matrix separately for each batch. Let nb be the total number of batches in the training set. In each batch $b = 1, \ldots, nb$, the size of the kernel matrix $\mathbf{K}_b$ is (bs, bs) where bs stands for the number of batch samples. We use a mini-batch gradient descent approach that allows one to benefit from the desirable properties of both stochastic and batch gradient descent updates to alleviate memory requirements for large training sets while minimising the chances of a local minima. In the proposed approach, the outputs $\mathbf{K}_b\boldsymbol{\alpha}_b$ are calculated for each batch by using the corresponding coefficients $\boldsymbol{\alpha}_b$ associated with the samples belonging to that batch. The weights of the network and the coefficients used are subsequently updated via a gradient descent method. After updating the coefficients $\boldsymbol{\alpha}$ and the weights $\boldsymbol{\omega}$ of the network, the features of all

Algorithm 1: Training of the G-DRKSR approach

Input: Training batches $\mathbf{X}_b|_{b=1}^{nb}$; labels of training batches $\mathbf{y}_b|_{b=1}^{nb}$
Output: Soft labels $\mathbf{y}_b|_{b=1}^{nb}$; penultimate layer features $\mathbf{O}$
Parameters: γ, δ, λ, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{nb}]$, $\boldsymbol{\omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_L]$
for $b = 1, \ldots, nb$ **do** `// initialization`
    $\boldsymbol{\alpha}_b = [1, \ldots, 1]^\top$
    $\mathbf{y}_b = [1, \ldots, 1]^\top$
for $t = 1, \ldots, T$ **do** `// #epochs`
    for $b = 1, \ldots, nb$ **do** `// iterate over batches`
        $\mathbf{O_b} = \Psi(\boldsymbol{\omega}_b, \mathbf{X}_b)$
        $\mathbf{K}_b = \exp\left[-\gamma((\mathbf{I} \circ \mathbf{O}_b\mathbf{O}_b^\top)\mathbf{1} + \mathbf{1}^\top(\mathbf{I} \circ \mathbf{O}_b\mathbf{O}_b^\top)^\top - 2\mathbf{O}_b\mathbf{O}_b^\top)\right]$
        $\boldsymbol{\alpha}_b = \boldsymbol{\alpha}_b - \eta(2\mathbf{K}_b(\mathbf{K}_b\boldsymbol{\alpha}_b - \mathbf{y}_b) + 2\delta\mathbf{K}_b\boldsymbol{\alpha}_b)$
        $\boldsymbol{\omega}_b = \boldsymbol{\omega}_b - \eta\partial L(\boldsymbol{\alpha}_b, \mathbf{y}_b, \boldsymbol{\omega})/\partial\boldsymbol{\omega}_b$ `// due to §4.3.1`
    $\mathbf{O} = [\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_{nb}]$
    $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{nb}]$
    for $b = 1, \ldots, nb$ **do** `// compute batch labels`
        $\mathbf{y}_b = \mathbf{K}(\mathbf{O}_b, \mathbf{O})\boldsymbol{\alpha}$
    $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{nb}]$
    $\mathbf{y} = \mathbf{y}/\|\mathbf{y}\|_2$

the training samples are extracted and stored in an array $\mathbf{O}_{n\times m}$. The coefficients $\boldsymbol{\alpha}$ are then utilized to derive the soft label corresponding to each batch as

$$\mathbf{y}_b = \mathbf{K}(\mathbf{O_b}, \mathbf{O})\boldsymbol{\alpha} \ , \tag{4.12}$$

which are subsequently normalized as

$$\mathbf{y} = \frac{\mathbf{y}}{\|\mathbf{y}\|_2} \ , \tag{4.13}$$

where $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{nb}]^\top$. The process is repeated for a number of epochs till convergence. The final feature matrix $\mathbf{O}_{n\times m}$ is utilized at the time of inference on the test set. Algorithm 1 summarizes the training stage of the proposed G-DRKSR approach.

4.4 C-DRKSR

In this section we propose an alternative approach to G-DRKSR which uses closed-form solution to optimize the coefficients $\boldsymbol{\alpha}$ instead of a gradient-descent based approach called Closed form Deep Robust Kernel Spectral Regression (C-DRKSR).

4.4.1 Optimization

For the optimization of the proposed objective function w.r.t $\boldsymbol{\alpha}$ using a closed-form solution, we can set the derivative of the loss function w.r.t $\boldsymbol{\alpha}$ to zero i.e:

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = 2\mathbf{K}\left(\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\right) + 2\delta\mathbf{K}\boldsymbol{\alpha} = 0,$$
$$\boldsymbol{\alpha} = (\mathbf{K} + \delta\mathbf{I})^{-1}\mathbf{y} \tag{4.14}$$

To update the weights of the network, the equations $4.4 - 4.10$ hold true for this approach as well. Once the optimal coefficients $\boldsymbol{\alpha}$ are obtained using the closed-form solution, subsequently normalized and the network parameters are updated via gradient descent, the response vector $\mathbf{y}$ can be normalized using equation 4.11. Similar to G-DRKSR, this approach is also based on computing the kernel matrix for each batch due to large requirements in system's memory. Hence, after the update of the coefficients $\boldsymbol{\alpha}$ and weights $\boldsymbol{\omega}$ of the network using all the batches, the features stored in the array $\mathbf{O}_{n\times m}$ are used to derive the soft-labels for each batch and subsequently normalized using Equations 4.12 and 4.13 respectively. At the time of inference on the test set, the feature matrix $\mathbf{O}_{n\times m}$ is used. The approach is summarized below.

Algorithm 2: Training of the C-DRKSR approach

Input: Training batches $\mathbf{X}_b|_{b=1}^{nb}$; labels of training batches $\mathbf{y}_b|_{b=1}^{nb}$
Output: Soft labels $\mathbf{y}_b|_{b=1}^{nb}$; penultimate layer features $\mathbf{O}$
Parameters: γ, δ, λ, $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{nb}]$, $\boldsymbol{\omega} = [\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_L]$
for $b = 1, \ldots, nb$ **do** // initialization
    $\boldsymbol{\alpha}_b = [1, \ldots, 1]^\top$
    $\mathbf{y}_b = [1, \ldots, 1]^\top$
for $t = 1, \ldots, T$ **do** // #epochs
    for $b = 1, \ldots, nb$ **do** // iterate over batches
        $\mathbf{O_b} = \Psi(\boldsymbol{\omega}_b, \mathbf{X}_b)$
        $\mathbf{K}_b = \exp\left[-\gamma((\mathbf{I} \circ \mathbf{O}_b\mathbf{O}_b^\top)\mathbf{1} + \mathbf{1}^\top(\mathbf{I} \circ \mathbf{O}_b\mathbf{O}_b^\top)^\top - 2\mathbf{O}_b\mathbf{O}_b^\top)\right]$
        $\boldsymbol{\alpha}_b = (\mathbf{K}_b + \delta\mathbf{I}_b)^{-1}\mathbf{y}_b$
        $\boldsymbol{\alpha}_b = \boldsymbol{\alpha}_b / \|\boldsymbol{\alpha}_b\|_2$
        $\boldsymbol{\omega}_b = \boldsymbol{\omega}_b - \eta\partial L(\boldsymbol{\alpha}_b, \mathbf{y}_b, \boldsymbol{\omega})/\partial\boldsymbol{\omega}_b$ // due to §4.4.1
    $\mathbf{O} = [\mathbf{O}_1, \mathbf{O}_2, \ldots, \mathbf{O}_{nb}]$
    $\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{nb}]$
    for $b = 1, \ldots, nb$ **do** // compute batch labels
        $\mathbf{y}_b = \mathbf{K}(\mathbf{O}_b, \mathbf{O})\boldsymbol{\alpha}$
    $\mathbf{y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{nb}]$
    $\mathbf{y} = \mathbf{y} / \|\mathbf{y}\|_2$

4.5 Decision strategy

After inferring the optimal parameter $\boldsymbol{\alpha}$ and weights $\boldsymbol{\omega}$ for both the approaches, the normality score of a test sample $\mathbf{z}$, *i.e.* $y(\mathbf{z})$, is computed as below where we use the final features obtained for each training sample, *i.e.* $\mathbf{o}_i$'s:

$$y(\mathbf{z}) = \sum_{i=1}^{n} \alpha_i \kappa(\bar{\mathbf{z}}, \mathbf{o}_i) \ , \tag{4.15}$$

where $\bar{\mathbf{z}}$ denotes the representation derived from the penultimate layer of the network for the test sample $\mathbf{z}$. The projections of the normal class samples are expected to be near 1 in the feature subspace, whereas the anomalous samples are expected to be projected further away closer to the origin. A threshold τ is ultimately applied on $y(\mathbf{z})$ to decide if the test sample $\mathbf{z}$ is normal, or anomalous.

Chapter 5

Experimental Evaluation

5.1 Introduction

This section presents the details of the experiments conducted including the databases used, the network architecture, implementation details along with convergence characteristics and the computatiaonal complexity of both the approaches. Furthermore, the proposed approaches are compared to the baseline and state-of-the-art methods on the databases chosen. We evaluate the proposed approaches in three different settings: (1) one-class classification with no contaminations; (2) one-class classification subject to training set contamination; and (3) unsupervised observation ranking where the dataset includes both normal and anomalous samples.

5.2 Databases

Databases are very important for the progress of scientific research. It allows the comparison of performance of various different methods objectively, which, otherwise would not be possible. For this reason, plenty of different databases

have been collected and made accessible to everyone which allows fair evaluation and comparison of several methodologies. In the sub-sections below, two most popular image datasets for one-class classification have been described which have been utilized for our experimental evaluations.

5.2.1 MNIST Dataset

The MNIST dataset [53] includes images of 28×28-pixel handwritten digits $0 - 9$. The database was created from NIST's Special Database (SD) 3 and Special Database (SD) 1 that contain binary images of handwritten datasets. SD-3 dataset was collected among Census Bureau employees while SD-1 was collected among high-school students. The digits have been size-normalized and centered. There are 60000 images in the training set and 10000 in the test set. In our analysis, one digit is considered as the normal class and all other digits represent anomalous samples. Therefore, ten one-class classification tasks are considered. There are approximately 6000 samples in the training set for each class. We use randomly selected 10% of the data to serve as the validation set so as to determine the (hyper)parameters of the proposed approach, leading to approximately 600 positive and 5400 anomalous samples per each class. The test set contains about 1000 samples of normal class and 9000 samples corresponding to anomalies. The original test-train split provided by the MNIST dataset is followed in our experiments. Some sample images are provided in the figure 5.1.

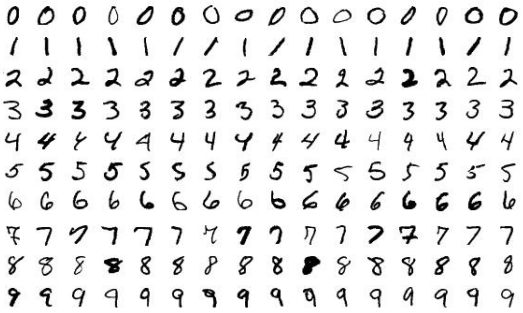

Figure 5.1: Samples from MNIST dataset

5.2.2 CIFAR10 Dataset

The CIFAR10 dataset [54] comprises 60000 color images of 32×32 pixels from 10 classes of animals, birds, and vehicles. The dataset is split into a training set comprising 50000 images in addition to a test set containing 10000 images. Each class includes 6000 images and is divided into 5000 images for training and 1000 for testing. Similar to the MNIST dataset, ten one-class classification problems are created for this dataset where in each round, one class is assumed as the positive/target class while the others serve as outliers. In order to determine the optimal parameters of the proposed approach, 10% of the training set is randomly chosen and used as validation data, accounting for 500 normal samples and 4500 anomalous samples. The test set contains 1000 images per each class. Figure 5.2 provides a few samples from each class present in Cifar10 dataset.

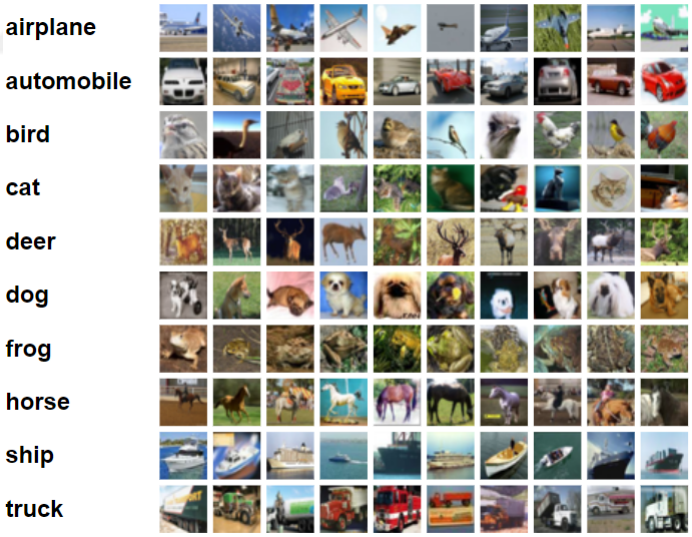

Figure 5.2: Samples from CIFAR10 dataset

5.3 Neural Network architecture

In order to facilitate a fair evaluation against the existing methods, we use LeNet-based architectures [53] as the convolutional backbone architecture for both Cifar10 and MNIST datasets as it serves as a widely used baseline and was used in

the similar studies [30]. For the MNIST dataset, a CNN architecture is employed with two modules - $8 \times (5 \times 5 \times 1)$ filters followed by $4 \times (5 \times 5 \times 1)$ filters. Finally a dense layer with 32 units is utilized at the end. After each convolutional layer, a batch-normalization layer, a leaky ReLU layer and a max-pooling layer is employed. The architecture used is exactly the same one used in [30]. The figure 5.3 shows the architecture used for the MNIST dataset.'

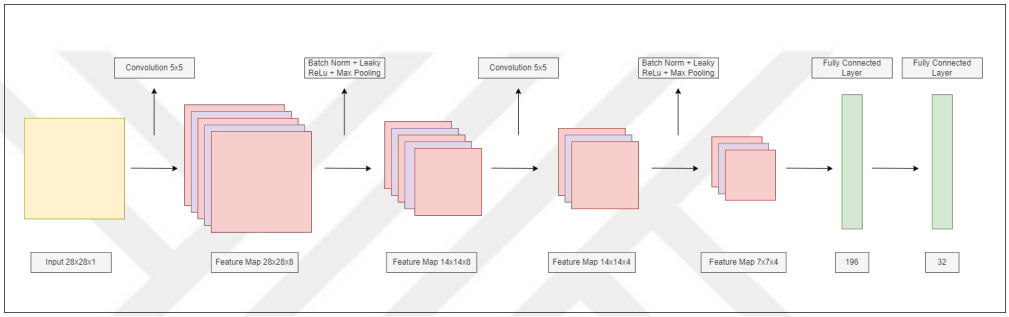

Figure 5.3: Neural Net architecture used for MNIST dataset

Similarly, for the Cifar10 dataset, a similar CNN architecture was used with 3 convolutional modules - $32 \times (5 \times 5 \times 1)$, $64 \times (5 \times 5 \times 1)$ and $128 \times (5 \times 5 \times 1)$ followed by a dense layer of 128 units. Just like the previous architecture, each convolutional layer is followed by a batch-norm layer, a leaky ReLU layer and a max-pooling layer. The figure 5.4 shows the architecture used for CIFAR10 dataset.

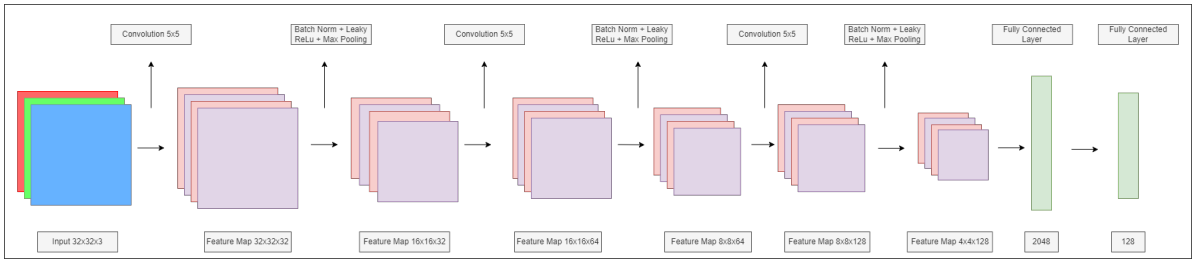

Figure 5.4: Neural Net architecture used for CIFAR10 dataset

Although our method used Le-Net based architecture, the proposed method can be used with any arbitrary CNN structure to represent and classify images.

5.4 Competing Methods

In the experiments to follow, we present a comparison between the proposed approach and state-of-the-art and baseline methods. The approaches considered in the comparative study correspond to both deep and non-deep approaches. For these methods, we use the implementations of the corresponding algorithms provided by the authors with the best parameters reported for all these methods in all experiments. The methods and the corresponding implementation details are summarised as follows.

- OCSVM [8]: we use a Gaussian kernel with the width parameter $\gamma \in \{2^{-10}, 2^{-9}, \ldots, 2^{-1}\}$, and run all experiments with v selected from $\{0.01, 0.1\}$ where v reflects one's assumptions with respect to the train set outlier fraction. The dimensionality of the data is initially reduced via PCA such that 95% of the variance is retained.

- For the baseline method of [27], designated as "Robust-kernel" in the subsequent sections, following the setting presented in [27], we use raw pixels as features and normalize them by dividing each image by its ℓ_2-norm, and use the average of the Euclidean distance between the features as the Gaussian width of the kernel.

- For Deep SVDD [30], and DCAE [30] methods, the data is pre-processed based on a normalization of the global contrast using an ℓ_1-norm and then re-scaled to $[0, 1]$ via a max-min scaling.

- For HRN [32], the data is pre-process using ℓ_2-norm instance-level data normalization and the method is implemented using the best parameter settings suggested in [32].

- For ICS [26], we use the min-max normalization for pre-processing and employ the best parameter settings reported in [26].

- Regarding SRO [20], the parameter controlling the balance between sparseness and connectivity, is set as $\lambda = 0.95$, and the coefficient of self-representation errors, is selected from $[0, 20]$ for each dataset separately.

5.5 Implementation Details

In the experiments to follow, a Gaussian kernel is used, *i.e.* the $(i,j)^{\text{th}}$ element of the kernel matrix is defined as $\mathbf{K}_{ij} = \exp(\gamma\|\mathbf{o}_i - \mathbf{o}_j\|^2)$ where γ controls the width of the Gaussian kernel and $\mathbf{o}_i$ and $\mathbf{o}_j$ stand for the features extracted from the last layer of the network associated with the training samples $\mathbf{x}_i$ and $\mathbf{x}_j$. The kernel width parameter γ is chosen from $\{0.5, 1, 2, 3\}$. The regularization parameter δ which controls the weight of the Tikhonov regularization applied to $\boldsymbol{\alpha}$ is selected from $\{0.01, 0.05, 0.1, 0.5, 1\}$. The regularization parameter λ that weights the regularization applied to network weights $\boldsymbol{\omega}$ is chosen from $\{10^{-4}, 10^{-3}, 10^{-2}\}$. We fix the batch size at 2500 samples for both datasets. The coefficients $\boldsymbol{\alpha}$ are initialised to 1. For learning network parameters the Adam optimizer operating on a learning rate of 5×10^{-4} is used. We also tune the Adam optimizer using one-cycle policy [55] with maximum learning rate set of 5×10^{-3}. Tables 5.1 - 5.4 show the best hyper-parameters that were found for each class in MNIST and CIFAR10 dataset and used in our experiments for evaluation on the test set.

Table 5.1: Optimal hyper-parameters for G-DRKSR for MNIST Dataset

Digit	0	1	2	3	4	5	6	7	8	9
λ	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
δ	0.01	0.01	0.1	0.05	0.01	0.1	0.05	0.005	0.01	0.05
γ	3	2	2	2	3	3	3	3	3	3

Table 5.2: Optimal hyper-parameters for C-DRKSR for MNIST Dataset

Digit	0	1	2	3	4	5	6	7	8	9
λ	0.0001	0.0001	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.01
δ	0.01	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.01
γ	3	2	2	2	2	2	2	2	2	2

Table 5.3: Optimal hyper-parameters for G-DRKSR for CIFAR10 Dataset

Class	Aeroplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
λ	0.0001	0.0001	0.001	0.0001	0.0001	0.0001	0.0001	0.0001	0.001	0.001
δ	0.1	0.05	0.01	0.05	0.005	0.001	0.1	0.005	0.05	0.1
γ	3	0.5	1	2	1	1	1	1	1	2

Once the optimal parameters for each class are determined using the validation set, for each experiment, the network is trained using all the training samples for

Table 5.4: Optimal hyper-parameters for C-DRKSR for CIFAR10 Dataset

Class	Aeroplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
λ	0.0001	0.0001	0.001	0.0001	0.0001	0.0001	0.0001	0.0001	0.001	0.001
δ	0.1	0.1	0.5	0.05	0.05	0.1	0.1	0.1	0.05	0.1
γ	2	0.5	1	2	2	2	1	1	1	2

100 epochs and then evaluated on the test set of each dataset using the best configuration of parameters obtained during the training stage. We report the best results obtained for all the methods. For injecting noisy samples into the training set, we randomly select samples from the classes other than the target class. We repeat this process five times and report the results. The data in both datasets are pre-processed via dividing each image by its ℓ_2-norm followed by removing the mean.

5.6 Convergence Characterstics

This section studies the convergence characteristics of the proposed approaches. For this purpose, the loss function and the AUC on the validation set versus number of epochs are recorded for all classes from the MNIST and CIFAR10 datasets.

5.6.1 MNIST Convergence

Figure 5.5 depicts the behavior of G-DRKSR appraoch in terms of the training loss and AUC on the validation set for all the digits. As it can be observed, the loss starts at a very high magnitude and immediately drops significantly partly due to the fact that the labels are initialized further away from the optimal values.

As the optimization proceeds and the labels are updated, the loss function decreases. A difference in the magnitude of the losses corresponding to different digits can also be observed in the figure 5.5 which is due to the difference in

Gaussian widths of the kernel function chosen for different classes. It can also be observed that the AUC's of the validation set for each digit in the dataset typically converges within 40 epochs.

A similar behavior is observed for C-DRKSR approach where the loss function decreases as the number of epochs. AUC's of the validation set again typically converge within 40 epochs.

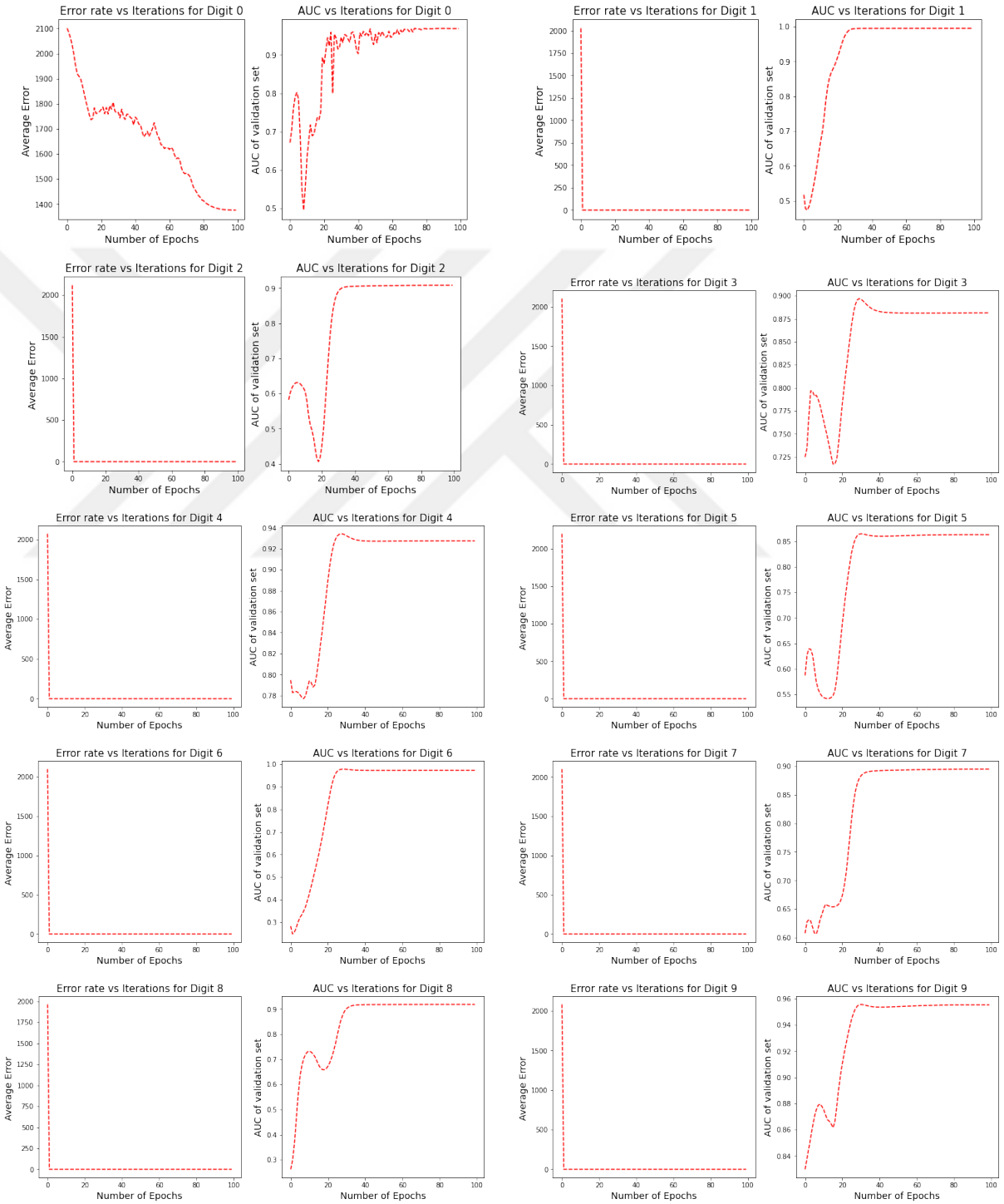

Figure 5.5: Convergence behavior of each digit in MNIST dataset for G-DRKSR

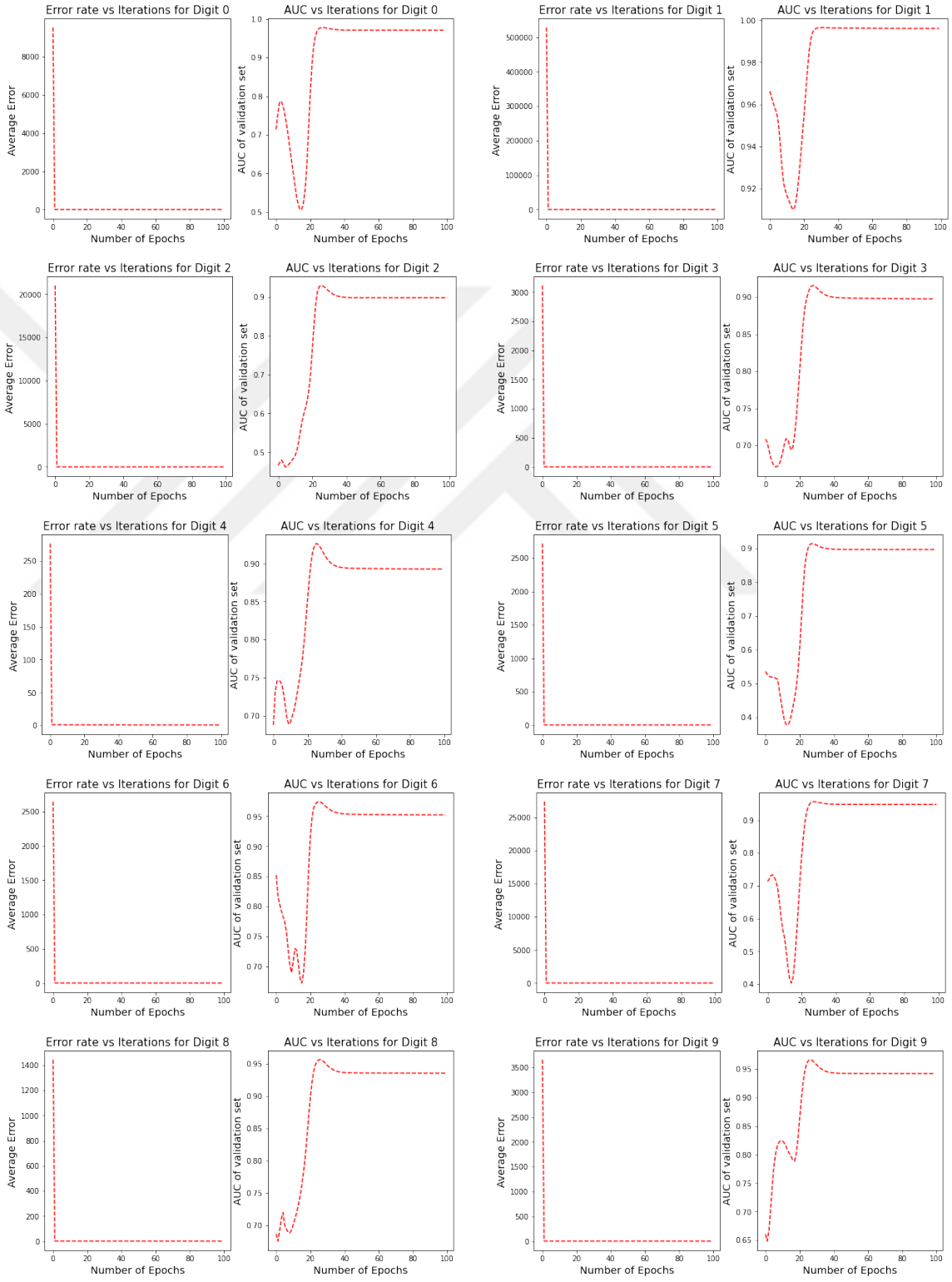

Figure 5.6: Convergence behavior of each digit in MNIST dataset for C-DRKSR

5.6.2 CIFAR10 Convergence

Figure 5.7 depicts the behavior of G-DRKSR appraoch in terms of the training loss and AUC on the validation set for all the class. Here the convergence behavior is a bit more complex than the MNIST dataset.

For half the classes, the loss starts at a high magnitude and drops significantly immediately partly due to the fact that labels are initialized far away from the optimal values. However, for the other half, the loss initially increases in the first epoch and then gradually falls. It can also be observed that the AUC starts increasing with decrease in the loss with some oscillations. However, for some classes such as Truck, after training for some epochs, the AUC starts decreasing which shows a sign of over-fitting. Typically, all the classes converge within 40 epochs.

For C-DRKSR approach, it can be observed in figure 5.8 that the loss drops significantly after the first epoch for all classes. AUC's of the validation set again typically converge within 40 epochs. However, for some classes like Automobile, the validation AUC starts decreasing showing the sign of over-fitting.

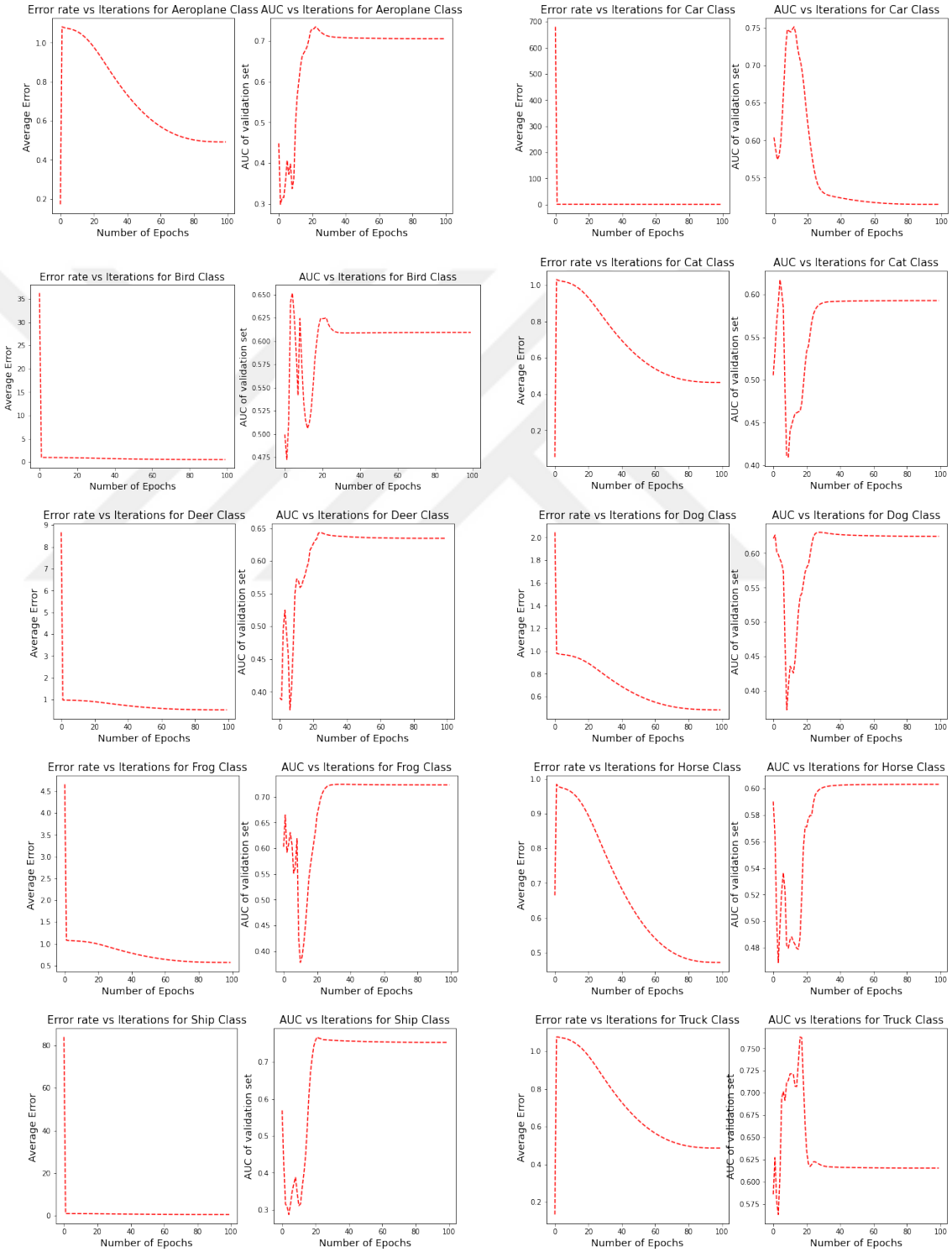

Figure 5.7: Convergence behavior of each class in CIFAR10 dataset for G-DRKSR

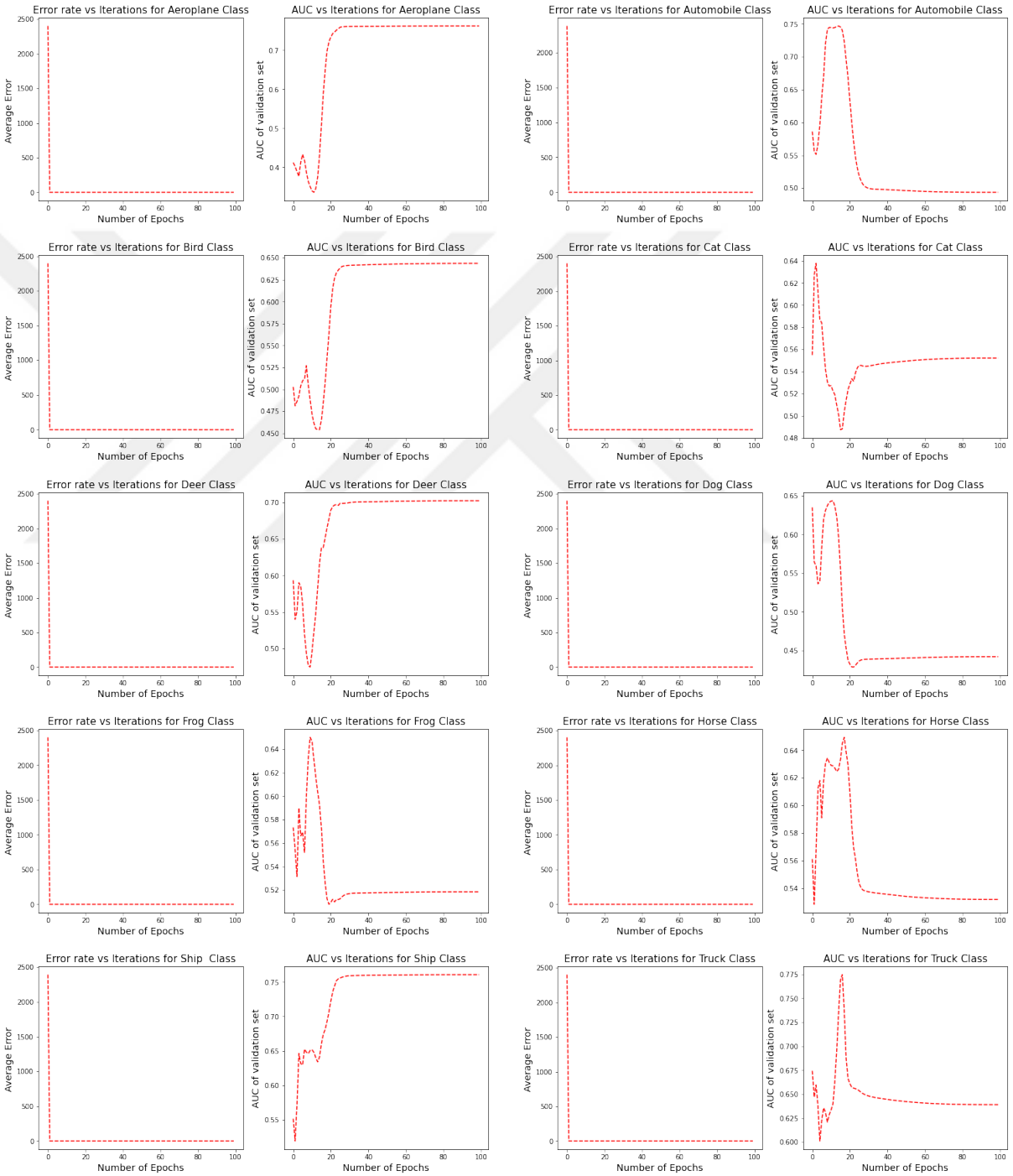

Figure 5.8: Convergence behavior of each class in CIFAR10 dataset for C-DRKSR

5.7 Computational Complexity

In addition to the runtime complexity of a convolutional neural network, there exists an additional computational overhead of $\mathcal{O}(bs^2)$ incurred for each batch during each iteration in the training phase, where bs is the number of batch samples for G-DRKSR. The additional computational complexity is incurred due to the calculation of kernel matrix during the training stage. For C-DRKSR, an additional overhead of $\mathcal{O}(bs^3)$ exists due to the calculation of the inverse of the kernel matrix during the training stage. During the testing stage of a sample, the computational complexity overhead on top of passing the test sample through the network for both approaches is due to the calculation of the kernel function in $\mathcal{O}(n)$ time, where n corresponds to the total number of samples in the training set.

5.8 Results

In this section, we discuss the results of our experiments for the three different settings - (1) No contamination in training set; (2) Contamination present in the training set; (3) Observation Ranking

5.8.1 OCC with no Contamination

Although the proposed approach is directed towards providing robustness against contamination and noise in the training set, in the first set of experiments, we investigate the performance of the proposed DRKSR method and other state-of-the-art approaches in a setting where the training set contains no contamination.

5.8.1.1 Evaluation on MNIST dataset

Table 5.5 report the results on MNIST dataset and compares our proposed approaches with other approaches from the literature. For performance reporting, the AUC (Area under the ROC curve) is used, which is the most commonly used evaluation metric in one-class classification and other anomaly detection tasks that provides a performance measure of the system independent from a specific operating threshold. Each row in the tables gives the average AUC over five runs for all the algorithms and their standard deviations for each digit in the dataset. The results of different methods in this set of experiments are reported from the corresponding papers as a similar experimental set-up setup is followed.

Table 5.5: AUC's (mean±std%) of different approaches for one-class classification on the MNIST dataset.

Digit	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
0	98.4 ± 0.0	98.0 ± 0.7	$\mathbf{99.5 \pm 0.0}$	96.2 ± 0.0	97.6 ± 0.7	98.9 ± 0.0	98.9 ± 0.2	98.8 ± 0.3
1	99.5 ± 0.0	99.7 ± 0.1	$\mathbf{99.9 \pm 0.0}$	97.7 ± 0.0	98.3 ± 0.6	99.8 ± 0.1	99.5 ± 0.0	98.4 ± 0.4
2	82.5 ± 0.1	91.7 ± 0.8	$\mathbf{96.5 \pm 0.1}$	88.0 ± 0.0	85.4 ± 2.4	91.7 ± 1.8	95.5 ± 0.4	93.1 ± 0.8
3	88.1 ± 0.0	91.9 ± 1.5	$\mathbf{97.4 \pm 0.1}$	91.2 ± 0.0	86.7 ± 0.9	96.6 ± 0.2	95.0 ± 0.6	91.8 ± 0.9
4	94.9 ± 0.0	94.9 ± 0.8	$\mathbf{97.2 \pm 0.0}$	88.4 ± 0.0	86.5 ± 2.0	86.5 ± 1.3	96.0 ± 0.7	94.1 ± 0.5
5	77.1 ± 0.0	88.5 ± 0.9	$\mathbf{97.2 \pm 0.2}$	75.3 ± 0.0	78.2 ± 2.7	88.9 ± 0.0	94.2 ± 0.4	91.8 ± 0.7
6	96.5 ± 0.0	98.3 ± 0.5	$\mathbf{99.2 \pm 0.0}$	91.8 ± 0.0	94.6 ± 0.5	98.8 ± 0.2	98.2 ± 0.1	97.1 ± 0.3
7	93.7 ± 0.0	94.6 ± 0.9	$\mathbf{97.6 \pm 0.0}$	90.6 ± 0.0	92.3 ± 1.0	96.1 ± 0.2	$\mathbf{97.6 \pm 0.3}$	93.2 ± 0.3
8	88.9 ± 0.0	93.9 ± 1.6	94.3 ± 0.1	90.2 ± 0.0	86.5 ± 1.6	$\mathbf{95.0 \pm 0.2}$	94.1 ± 0.7	93.6 ± 0.4
9	93.1 ± 0.0	96.5 ± 0.3	$\mathbf{97.1 \pm 0.0}$	89.7 ± 0.0	90.4 ± 1.8	90.0 ± 0.4	96.4 ± 0.3	92.7 ± 0.4
Average	91.27	94.80	**97.59**	89.91	89.65	94.23	96.54	94.46

Table 5.5 shows that the two methods with the highest average performance in terms of AUC are HRN and G-DRKSR where both the methods perform significantly better than other approaches. HRN performs slightly better than the proposed approach (by $\sim 1\%$ on average). It is interesting to note that on this method, C-DRKSR does not perform very well and is the fourth best method overall. Note that robust kernel-based method of [27] yields a zero variance performance on this dataset since the corresponding regularisation parameter is determined using the analytical relation given in [27] which is only dependent on the minimum eigenvalue of the kernel matrix which shuffling the data from each dataset has negligible impact on.

In general, it can be observed that the deep-approaches are performing better than the non-deep approaches. Moreover, the deep-approaches also have a higher standard deviation in performance compared to non-deep appraoches. This can be attributed to the random initialisation of weights and the limited size of the training set in these experiments. It can be seen that the DCAE and Deep SVDD have the highest standard deviation among the non-deep methods.

5.8.1.2 Evaluation on CIFAR10 dataset

Table 5.6 report the results on CIFAR10 dataset and compares our proposed approaches with other approaches from the literature. For performance reporting, again, the AUC (Area under the ROC curve) is used. Each row in the tables gives the average AUC over five runs for all the algorithms and their standard deviations for each class in the dataset. The results of different methods in this set of experiments are reported from the corresponding papers as a similar experimental set-up setup is followed.

Table 5.6: AUC's (mean±std%) of different approaches for one-class classification on the CIFAR-10 dataset.

Class	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
Airplane	61.6 ± 0.9	61.7 ± 4.1	77.3 ± 0.2	74.5 ± 0.0	59.1 ± 5.1	76.8 ± 3.2	78.9 ± 1.8	$\mathbf{80.7 \pm 1.0}$
Automobile	63.8 ± 0.6	65.9 ± 2.1	69.9 ± 1.3	38.7 ± 0.0	57.4 ± 2.9	71.3 ± 0.2	$\mathbf{78.6 \pm 1.6}$	72.1 ± 1.1
Bird	50.0 ± 0.5	50.8 ± 0.8	60.6 ± 0.3	63.6 ± 0.0	48.9 ± 2.4	63.0 ± 0.8	$\mathbf{67.5 \pm 1.2}$	66.9 ± 0.6
Cat	55.9 ± 1.3	59.1 ± 1.4	64.4 ± 1.3	45.9 ± 0.0	58.4 ± 1.2	60.1 ± 3.4	$\mathbf{64.5 \pm 0.9}$	63.2 ± 1.0
Deer	66.0 ± 0.7	60.9 ± 1.1	71.5 ± 1.0	65.4 ± 0.0	54.0 ± 1.3	$\mathbf{74.9 \pm 0.9}$	68.9 ± 2.7	71.3 ± 1.1
Dog	62.4 ± 0.8	65.7 ± 2.5	$\mathbf{67.4 \pm 0.5}$	49.7 ± 0.0	62.2 ± 1.8	66.0 ± 1.1	64.3 ± 0.8	65.1 ± 0.8
Frog	74.7 ± 0.3	67.7 ± 2.6	77.4 ± 0.2	58.4 ± 0.0	51.2 ± 5.2	71.6 ± 0.8	$\mathbf{77.5 \pm 1.6}$	72.7 ± 1.7
Horse	62.6 ± 0.5	67.3 ± 0.9	64.9 ± 1.1	47.7 ± 0.0	58.6 ± 2.9	64.1 ± 1.6	$\mathbf{67.4 \pm 2.9}$	65.7 ± 1.3
Ship	74.9 ± 0.4	75.9 ± 1.2	$\mathbf{82.5 \pm 0.2}$	70.9 ± 0.0	76.8 ± 1.4	78.9 ± 0.5	80.0 ± 1.7	77.7 ± 1.0
Truck	75.9 ± 0.3	73.1 ± 1.2	77.3 ± 0.9	54.2 ± 0.0	67.3 ± 3.0	66.0 ± 2.5	75.1 ± 0.9	$\mathbf{77.6 \pm 0.6}$
Average	64.78	64.81	71.32	56.90	59.39	69.27	**72.27**	71.35

Table 5.6 shows that the method with the highest average performance in terms of AUC on the CIFAR10 dataset is the proposed G-DRKSR and C-DRKSR approaches with an average AUC of 72.27% and 71.35% respectively, followed by the state-of-the-art method HRN with an average AUC of 71.32%. ICS is the fourth

best performing method and provides another strong baseline. It can also be observed that these best performing methods approximately perform $8-10\%$ better then other methods on average while providing a huge $15-20\%$ improvement for some classes when compared to other approaches. In terms of standard deviation, similar to the behavior on MNIST dataset, the non-deep approaches have a lower standard deviation than the deep-approaches with DCAE and DSVDD having the highest standard deviation. The robust kernel based method of [27] again yields a zero variance in performance due to the determination of the hyper-parameters using the analytical relations provided in [27] and shuffling having negligible impact on this approach.

5.8.1.3 Summary

Table 5.7 provides average ranking of all the approaches based on performance for both the datasets using Friedman's test. Based on the table, for the MNIST dataset, HRN is ranked the highest followed by G-DRKSR. However, for CIFAR10 dataset, G-DRKSR is ranked the best followed by HRN and C-DRKSR.

Table 5.7: Average ranking of different approaches based on Freidman's test for one-class classification on the CIFAR10 (p-value= $2.07e-07$ and MNIST (p-value= $6.58e-09$) datasets when no contamination is present.

Method	Mean Rank (Cifar10)	Mean Rank (MNIST)
OC-SVM	5.7	5.7
DSVDD	5.1	3.9
HRN	2.6	**1.15**
Robust-kernel	6.9	7.2
DCAE	7.0	7.05
ICS	3.8	3.55
G-DRKSR (proposed approach)	**2.2**	2.75
C-DRKSR (proposed approach)	2.7	4.7

Summarising the evaluation results in the no contamination setting, one concludes that both the proposed approaches and the HRN approach perform very well compared to other existing approaches. Nevertheless, as will be illustrated

in the subsequent sections, the merits of the proposed approaches in terms of robustness against contamination and noise when the training data includes noisy objects.

5.8.2 OCC with contamination

In this section, different OCC approaches are examined for one-class classification subject to training set contamination. Following a real-world realistic setting, in this set of experiments it is assumed that the fraction of contamination in the training set is not known in advance. For the purpose of this experiment, on both the MNIST and CIFAR10 datasets, each model is trained subject to different training set contamination levels ranging from 5% to 50% in steps of 5% and then assessed on the corresponding test set. In each experiment, the number of training samples for each class remains the same for both datasets. A contaminated dataset, in this case, is created by randomly removing some positive samples from the target training set and replacing them with random samples from other classes. The distribution of the contamination is uniform in the sense that the same number of samples are randomly selected from each negative class. For computational efficiency, we determine the optimal parameters where the training set did not contain any contamination and only optimize the Tikhonov regularization parameter δ. Nevertheless, some improvement in the performance is expected if all the parameters are tuned for each different level of contamination for each class separately. We compare the proposed approach to six other methods mentioned earlier and report the best results obtained for all the methods. For this experiment, we use sci-kit learn library to get results for OCSVM while for the other methods, we run the code provided by the authors.

5.8.2.1 Evaluation on MNIST dataset

Table 5.8 report the results on MNIST dataset and compares our proposed approaches with other approaches from the literature. For performance reporting,

again, the AUC (Area under the ROC curve) is used. Each row in the tables reports the average AUC of all the classes for a particular level of contamination in the training set along with the corresponding standard deviations.

Table 5.8: AUC's (mean±std%) of different approaches over all classes for one-class classification on the MNIST dataset in the presence of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	87.8 ± 0.1	90.9 ± 2.0	$\mathbf{96.0 \pm 0.3}$	90.4 ± 0.0	85.4 ± 2.1	91.7 ± 1.6	95.7 ± 0.5	93.7 ± 0.6
10	85.7 ± 0.1	89.5 ± 2.3	94.9 ± 0.4	89.9 ± 0.0	83.8 ± 2.3	90.4 ± 1.8	$\mathbf{95.0 \pm 0.6}$	92.8 ± 0.6
15	83.6 ± 0.1	87.6 ± 2.7	93.8 ± 0.4	89.5 ± 0.0	82.4 ± 2.8	89.1 ± 2.0	$\mathbf{94.7 \pm 0.6}$	91.7 ± 0.77
20	81.4 ± 0.2	85.4 ± 3.1	92.5 ± 0.4	89.1 ± 0.0	80.9 ± 3.1	88.2 ± 2.3	$\mathbf{94.5 \pm 0.7}$	90.5 ± 0.8
25	79.4 ± 0.2	83.6 ± 3.6	90.9 ± 0.4	88.8 ± 0.0	79.3 ± 3.5	85.9 ± 2.0	$\mathbf{94.3 \pm 0.7}$	89.2 ± 0.8
30	77.3 ± 0.2	81.6 ± 4.0	89.7 ± 0.5	88.3 ± 0.0	77.6 ± 4.0	84.3 ± 2.1	$\mathbf{94.1 \pm 0.7}$	88.4 ± 0.8
35	75.6 ± 0.2	80.2 ± 4.2	88.1 ± 0.5	87.9 ± 0.0	76.1 ± 4.2	83.4 ± 2.4	$\mathbf{93.8 \pm 0.8}$	87.1 ± 0.9
40	74.3 ± 0.3	78.9 ± 4.4	86.3 ± 0.5	87.5 ± 0.0	74.5 ± 4.4	84.0 ± 2.6	$\mathbf{93.6 \pm 0.8}$	85.8 ± 1.0
45	72.6 ± 0.3	77.4 ± 4.7	84.6 ± 0.6	87.1 ± 0.0	72.5 ± 4.7	82.7 ± 2.9	$\mathbf{93.5 \pm 0.8}$	85.0 ± 1.1
50	70.8 ± 0.3	75.9 ± 5.0	83.1 ± 0.7	86.2 ± 0.0	71.0 ± 5.0	81.3 ± 3.2	$\mathbf{93.3 \pm 0.9}$	83.5 ± 1.2
Average	78.85	83.10	89.99	88.47	78.35	86.10	**94.25**	88.77

As it can be observed from Table 5.8, G-DRKSR, one of our proposed approach, is the best performing method on average, for all different levels of contamination with an average AUC of 94.25%. The second best performing method is that of HRN, which also seems to be robust to contamination up to a certain extent. However, with increase in the level of contamination, one observes quite drastic drop in performance of HRN approach whereas the performance of the proposed G-DRKSR method does not get affected as much. The only case where HRN performs better than G-DRKSR is when the contamination is set to 5%. Overall, G-DRKSR performs approximately 4% better than HRN on average on this dataset. C-DRKSR, however, does not seem to work well on this dataset with its performance also falling with increase in contamination and is the third best performing method. The robust kernel-based approach of [27] appears to be quite robust, with minimal decrease in the performance when the training set contamination is increased. The ICS, DCAE, and DSVDD approaches perform poorly on this dataset as their performances degrade rapidly with increased contamination. The standard deviations of the DSVDD as well as the DCAE methods are observed to be the highest for the MNIST dataset in this scenario as well.

Tables A.1 - A.10 in the appendix provide the details of the performance of all methods for each digit for each level of contamination. It can be observed that G-DRKSR outperforms all other methods for all the digits by a margin as high as 8% AUC on average except digit 1 where ICS performs better than G-DRKSR by a very small margin. HRN yields slightly better results for low levels of contamination. However, the performances of the HRN and ICS methods deteriorate quite drastically in the presence of high levels of contamination. C-DRKSR performance deteriorates quite drastically on this dataset as well. Robust kernel does seem to outperform even HRN on digits $0, 2, 3$ and gives consistently good baseline results that outperform most of the deep-approaches.

5.8.2.2 Evaluation on CIFAR10 dataset

Table 5.9 summarizes the results obtained on CIFAR10 dataset and compares our proposed approaches with other approaches from the literature. For performance reporting, again, the AUC (Area under the ROC curve) is used. Each row in the tables reports the average AUC of all the classes for a particular level of contamination in the training set along with the corresponding standard deviations.

Table 5.9: AUC's (mean±std%) of different approaches over all classes for one-class classification on the CIFAR-10 dataset in the presence of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	64.1 ± 0.7	61.8 ± 2.5	70.9 ± 0.6	56.5 ± 0.0	59.4 ± 3.2	66.1 ± 1.7	$\mathbf{72.1 \pm 1.8}$	71.3 ± 1.2
10	63.6 ± 0.7	60.7 ± 2.6	70.5 ± 0.6	56.2 ± 0.0	58.0 ± 3.3	65.5 ± 1.8	$\mathbf{71.6 \pm 1.9}$	71.0 ± 1.2
15	62.7 ± 0.8	59.2 ± 2.8	70.3 ± 0.7	56.0 ± 0.0	57.1 ± 3.5	64.8 ± 1.6	$\mathbf{71.2 \pm 1.9}$	70.7 ± 1.2
20	61.9 ± 0.9	57.5 ± 3.0	70.1 ± 0.7	55.7 ± 0.0	56.2 ± 3.6	64.4 ± 1.5	$\mathbf{70.9 \pm 1.9}$	70.5 ± 1.2
25	61.0 ± 0.9	56.9 ± 3.0	69.7 ± 0.7	55.4 ± 0.0	55.3 ± 3.7	63.8 ± 1.5	$\mathbf{70.5 \pm 2.0}$	$\mathbf{70.5 \pm 1.2}$
30	59.8 ± 1.0	55.8 ± 3.2	69.4 ± 0.8	55.2 ± 0.0	54.4 ± 3.9	63.1 ± 1.5	70.2 ± 2.0	$\mathbf{70.3 \pm 1.2}$
35	58.9 ± 1.0	54.4 ± 3.3	68.6 ± 0.9	55.0 ± 0.0	53.9 ± 3.9	62.8 ± 1.5	69.7 ± 2.1	$\mathbf{70.0 \pm 1.2}$
40	57.8 ± 1.1	53.5 ± 3.5	67.9 ± 1.0	54.8 ± 0.0	52.8 ± 4.0	62.5 ± 1.5	69.6 ± 2.2	$\mathbf{69.7 \pm 1.2}$
45	56.9 ± 1.2	52.7 ± 3.6	66.8 ± 1.3	54.6 ± 0.0	51.6 ± 4.2	62.1 ± 1.6	69.3 ± 2.2	$\mathbf{69.6 \pm 1.3}$
50	56.1 ± 1.2	52.3 ± 3.8	65.6 ± 1.7	54.5 ± 0.0	50.7 ± 4.3	61.7 ± 1.7	68.9 ± 2.3	$\mathbf{69.1 \pm 1.4}$
Average	60.28	56.48	68.98	55.39	54.94	63.68	**70.39**	70.27

As it can be observed from Table 5.9, both G-DRKSR and C-DRKSR, are the best performing method on average, for all different levels of contamination with

an average AUC of 70.39% and 70.27% respectively. The second best performing method is that of HRN, which also seems to be robust to contamination on this dataset. However, with very high level of contamination, one observes quite drastic drop in performance of HRN approach whereas the performance of the proposed approaches does not get affected as much. Overall, G-DRKSR is the best performing method followed very closesly by C-DRKSR. The robust kernel-based approach of [27] appears to be quite robust, with minimal decrease in the performance when the training set contamination is increased. ICS also seems to be robust and provides another strong baseline on this dataset. It is also interesting to note that OCSVM outperforms its deep conterpart DSVDD by about 4% on average on this dataset. DSVDD and DCAE approaches perform quite poorly on this dataset where their performances degrade rapidly with increased contamination. The standard deviations of the DSVDD as well as the DCAE methods are observed to be the highest for the MNIST dataset in this scenario as well.

Tables A.11 - A.20 in the appendix provide the details of the performance of all methods for each class for each level of contamination. HRN method and our proposed two approaches G-DRKSR and C-DRKSR outperform other methods consistently for all the classes except for the DOG class where ICS performs better. There is a small decrease in the performance for the HRN method for the majority of classes with the exception of the SHIP class where HRN's performance drops relatively fast for high levels of contamination. The proposed approaches outperform all other classes for the majority of classes by a margin as high as 10% AUC on average on the Cifar10 dataset. It is also worth noting that the Deep SVDD and DCAE methods yield relatively unstable results with their standard deviations generally increasing with an increase in the contamination level.

5.8.2.3 Summary

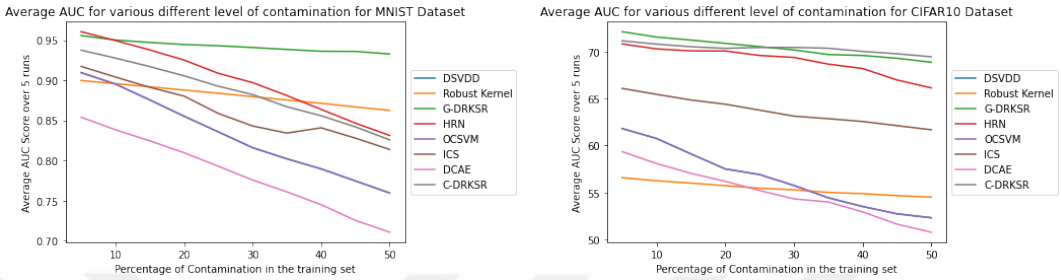

Figure 5.9: Average AUC's for all classes for the MNIST and CIFAR10 datasets for OCC.

Figure 5.9 provides a summary of the performance in terms of average AUC for all classes in both datasets, for all the methods used in our experiment. Moreover, the figure 5.10 and figure 5.11 show the detailed summary of performance of all methods used in our experiments for MNIST and CIFAR10 respectively.

Table 5.10: Average ranking of different approaches based on Freidman's test for one-class classification on the CIFAR10 (p-value= $6.47e-12$ and MNIST (p-value= $2.14e-11$) datasets.

Method	Mean Rank (Cifar10)	Mean Rank (MNIST)
OC-SVM	5.0	7.4
DSVDD	6.4	5.9
HRN	3.1	2.4
Robust-kernel	7.0	3.6
DCAE	7.6	7.6
ICS	3.9	4.8
G-DRKSR	1.55	**1.1**
C-DRKSR	**1.45**	3.2

Table 5.10 provides an average ranking of different methods based on their performances in this set of experiments using the Friedman's test. Inspecting Table 5.10, one can observe that the proposed G-DRKSR is the best performing method for MNIST dataset but is very slightly lower than C-DRKSR on Cifar10 dataset. C-DRKSR works very well on Cifar10 dataset but works poorly on MNIST dataset. The next best performing method corresponds to that of HRN [32]. It is interesting to note that OC-SVM outperforms DSVDD in the presence of contamination on MNIST dataset.

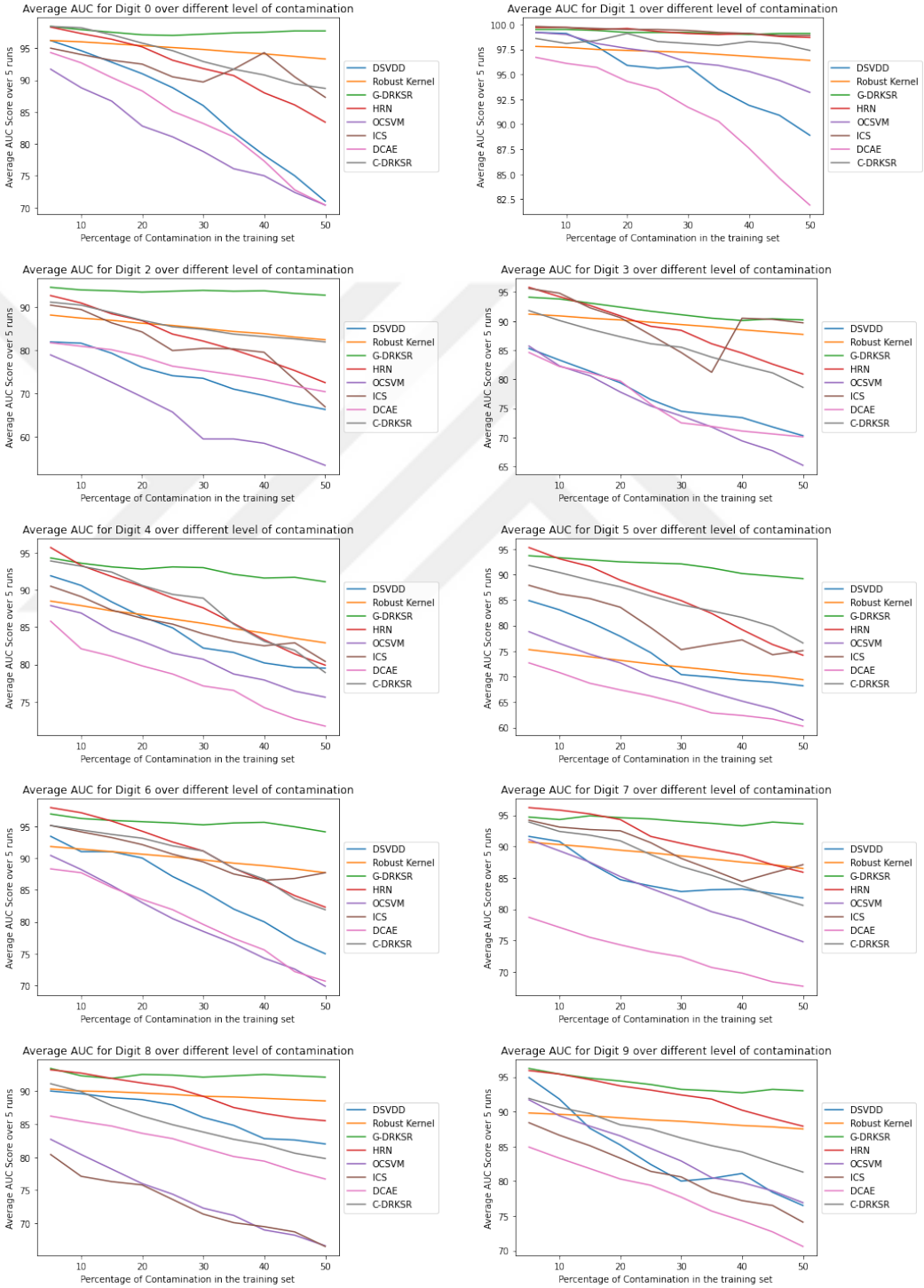

Figure 5.10: Average AUC's of different approaches for each class on the MNIST dataset for one-class classification.

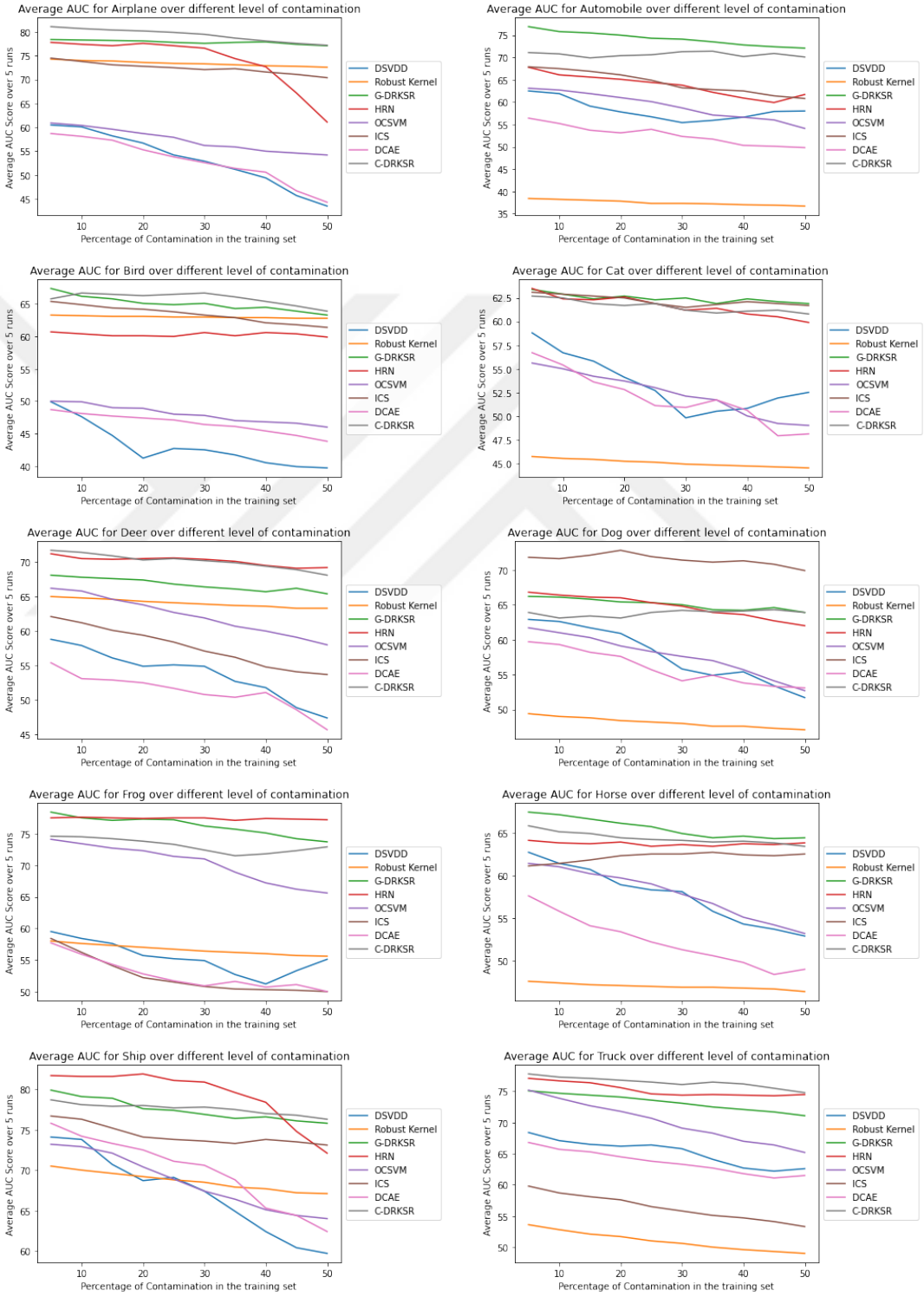

Figure 5.11: Average AUC's of different approaches for each class on the Cifar10 dataset for one-class classification.

5.8.3 Observation Ranking

In this set of experiments, different methods are evaluated in an unsupervised observation ranking scenario. For this purpose, the training set is contaminated with varied levels of contamination ranging from 5% to 50%, and the observations are ranked based on their conformity with the learned model. The techniques used in this setting implicitly assume that normal samples are in a majority and appear more frequently compared to anomalies. We follow the same protocol as in the previous experiments, *i.e.*, the number of samples in the training set remains the same and some positive samples are randomly replaced with some negative samples from the other classes which are considered as contamination/noise. In this experiment, in addition to the approaches included in the earlier experiments, we also include the SRO method [20] which is considered as an effective unsupervised approach for observation ranking. Similar to the previous experiments, the performance metric employed in this set of experiments is that of the AUC measure.

5.8.3.1 Evaluation on MNIST dataset

Table 5.11 report the results on MNIST dataset and compares our proposed approaches with other approaches from the literature. Each row in the tables reports the average AUC of all the classes for a particular level of contamination in the training set along with the corresponding standard deviations.

As it can be observed from Table 5.11, G-DRKSR, one of our proposed approach, is the best performing method on average, for all different levels of contamination with an average AUC of 92.65%. The second best performing method is that of HRN, which is followed very closely by C-DRKSR in terms of performance. However, with increase in the level of contamination, one observes quite drastic drop in performance of HRN approach whereas the performance of the proposed G-DRKSR method does not get affected as much. HRN does perform

Table 5.11: AUC's (mean±std%) of different approaches over all classes for observation ranking on the MNIST dataset in the presence of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	87.6 ± 0.1	90.2 ± 2.0	$\mathbf{95.0 \pm 0.3}$	89.3 ± 0.0	84.3 ± 1.9	90.6 ± 1.6	73.8 ± 0.0	94.8 ± 0.4	94.1 ± 0.8
10	85.7 ± 0.1	89.4 ± 2.3	$\mathbf{94.0 \pm 0.4}$	89.0 ± 0.0	83.3 ± 1.8	89.7 ± 1.8	72.8 ± 0.0	94.3 ± 0.4	93.4 ± 0.8
15	83.6 ± 0.1	87.7 ± 2.6	92.9 ± 0.4	88.6 ± 0.0	81.7 ± 2.1	88.2 ± 1.9	71.6 ± 0.0	$\mathbf{93.6 \pm 0.6}$	92.4 ± 0.9
20	81.5 ± 0.2	85.6 ± 2.9	91.6 ± 0.4	88.3 ± 0.0	80.4 ± 2.4	87.0 ± 2.1	70.1 ± 0.0	$\mathbf{93.1 \pm 0.7}$	91.1 ± 1.0
25	79.7 ± 0.2	83.9 ± 3.2	90.3 ± 0.4	87.8 ± 0.0	78.8 ± 2.7	85.7 ± 2.0	68.6 ± 0.0	$\mathbf{92.7 \pm 0.7}$	90.0 ± 1.1
30	77.6 ± 0.2	82.6 ± 3.5	88.9 ± 0.4	87.3 ± 0.0	77.6 ± 2.9	83.8 ± 2.1	66.9 ± 0.0	$\mathbf{92.3 \pm 0.7}$	88.8 ± 1.2
35	75.9 ± 0.2	80.8 ± 3.9	87.4 ± 0.5	87.1 ± 0.0	75.8 ± 3.3	83.4 ± 2.3	66.1 ± 0.0	$\mathbf{91.9 \pm 0.8}$	87.4 ± 1.3
40	74.5 ± 0.2	79.4 ± 4.1	85.6 ± 0.5	86.6 ± 0.0	74.2 ± 3.5	83.2 ± 2.5	65.4 ± 0.0	$\mathbf{91.5 \pm 0.8}$	85.9 ± 1.5
45	72.7 ± 0.2	77.9 ± 4.4	84.0 ± 0.6	86.1 ± 0.0	72.2 ± 3.7	82.3 ± 2.7	64.7 ± 0.0	$\mathbf{91.3 \pm 0.8}$	84.6 ± 1.5
50	71.0 ± 0.2	76.5 ± 4.7	82.1 ± 0.7	85.6 ± 0.0	70.7 ± 3.9	81.4 ± 3.0	63.9 ± 0.0	$\mathbf{90.8 \pm 0.7}$	82.9 ± 1.7
Average	78.98	83.39	89.18	87.55	77.90	85.53	68.48	**92.65**	89.10

better than G-DRKSR is when the contamination is set to 5% and 10%. Overall, G-DRKSR performs approximately 3% better than HRN on average on this dataset. C-DRKSR approach is the third best approach= but doesn't seem to be very robust to contamination on this dataset. The robust kernel-based approach of [27] appears to be quite robust providing a very strong baseline. It outperforms most of the deep-approaches with the exception of HRN. The ICS, DCAE, and DSVDD approaches perform poorly on this dataset as their performances degrade rapidly with increased contamination. The standard deviations of the DSVDD as well as the DCAE methods are observed to be the highest for the MNIST dataset in this scenario as well.

Tables A.21 - A.30 in the Appendix provide the details of the performance of all methods for each digit for each level of contamination. It can be observed that G-DRKSR outperforms all the other methods for almost all the digits in this scenario as well. However, for low levels of contamination, both the HRN and ICS approaches seem to perform better for some classes.

5.8.3.2 Evaluation on CIFAR10 dataset

Table 5.9 summarizes the results obtained on CIFAR10 dataset and compares our proposed approaches with other approaches from the literature. Each row in the tables reports the average AUC of all the classes for a particular level of contamination in the training set along with the corresponding standard deviations.

Table 5.12: AUC's (mean±std%) of different approaches over all classes for observation ranking on the CIFAR10 dataset in the presence of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	63.8 ± 0.8	61.5 ± 2.7	71.1 ± 0.5	57.0 ± 0.0	58.4 ± 3.3	65.2 ± 1.6	58.8 ± 0.0	$\mathbf{84.1 \pm 1.0}$	80.6 ± 1.4
10	63.3 ± 0.8	60.4 ± 2.8	70.8 ± 0.6	56.5 ± 0.0	57.4 ± 3.4	64.8 ± 1.7	58.3 ± 0.0	$\mathbf{83.7 \pm 1.1}$	79.9 ± 1.5
15	62.6 ± 0.9	59.1 ± 3.0	70.5 ± 0.6	56.2 ± 0.0	56.4 ± 3.5	64.5 ± 1.3	57.6 ± 0.0	$\mathbf{83.4 \pm 1.2}$	79.2 ± 1.5
20	61.7 ± 1.0	57.5 ± 3.3	70.2 ± 0.6	56.0 ± 0.0	55.5 ± 3.7	64.1 ± 1.4	57.0 ± 0.0	$\mathbf{83.0 \pm 1.2}$	78.7 ± 1.5
25	60.7 ± 1.0	56.5 ± 3.4	69.8 ± 0.7	55.8 ± 0.0	54.9 ± 3.8	63.7 ± 1.4	57.0 ± 0.0	$\mathbf{82.3 \pm 1.3}$	78.1 ± 1.6
30	59.6 ± 1.1	55.4 ± 3.6	69.4 ± 0.8	55.5 ± 0.0	53.8 ± 3.9	63.4 ± 1.5	57.0 ± 0.0	$\mathbf{81.7 \pm 1.3}$	77.4 ± 1.6
35	58.7 ± 1.1	54.4 ± 3.7	68.7 ± 0.8	55.2 ± 0.0	53.5 ± 4.0	63.2 ± 1.5	56.7 ± 0.0	$\mathbf{81.0 \pm 1.4}$	77.0 ± 1.7
40	57.7 ± 1.2	53.4 ± 3.9	68.1 ± 0.9	54.8 ± 0.0	52.2 ± 4.2	62.8 ± 1.6	56.4 ± 0.0	$\mathbf{80.3 \pm 1.5}$	76.3 ± 1.7
45	56.9 ± 1.2	52.7 ± 4.0	67.4 ± 1.2	54.7 ± 0.0	51.6 ± 4.4	62.6 ± 1.7	56.2 ± 0.0	$\mathbf{79.7 \pm 1.6}$	75.8 ± 1.8
50	56.2 ± 1.3	52.0 ± 4.3	66.6 ± 1.6	54.5 ± 0.0	50.3 ± 4.7	62.4 ± 1.7	56.0 ± 0.0	$\mathbf{79.1 \pm 1.8}$	75.2 ± 1.9
Average	60.14	56.27	69.29	55.65	54.40	63.70	57.14	**81.82**	80.25

As it can be observed from Table 5.12, both G-DRKSR and C-DRKSR, are the best performing method on average, for all different levels of contamination with an average AUC of 81.82% and 80.25% respectively. The second best performing method is that of HRN, which also seems to be robust to contamination on this dataset. However, with very high level of contamination, one observes quite drastic drop in performance of HRN approach whereas the performance of the proposed approaches does not get affected as much. Overall, G-DRKSR is the best performing method followed very closesly by C-DRKSR. The robust kernel-based approach of [27] appears to be quite robust, with minimal decrease in the performance when the training set contamination is increased. ICS also seems to be robust and provides another strong baseline on this dataset. It is also interesting to note that OCSVM outperforms its deep conterpart DSVDD by about 4% on average on this dataset. DSVDD and DCAE approaches perform quite poorly on this dataset where their performances degrade rapidly with increased contamination. The standard deviations of the DSVDD as well as the

DCAE methods are observed to be the highest for the MNIST dataset in this scenario as well.

Tables A.31 - A.40 in the Appendix provide the details of the performance of all methods for each class for each level of contamination. our proposed two approaches G-DRKSR and C-DRKSR outperform other methods consistently for all the classes. HRN is the third best performing method on majority of classes on average with the exception of a few classes of AUTOMOBILE, BIRD, CAT and DOG where ICS performs better. The proposed approaches outperform all other classes for the majority of classes by a margin as high as 15% AUC on average on the Cifar10 dataset. It is also worth noting that the Deep SVDD and DCAE methods yield relatively unstable results with their standard deviations generally increasing with an increase in the contamination level.

5.8.3.3 Summary

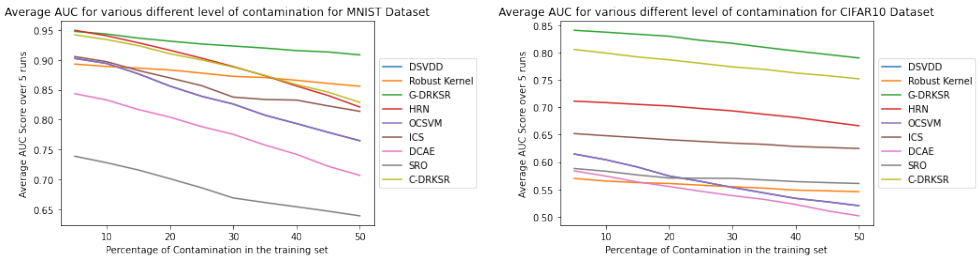

Figure 5.12: Average AUCs for all classes for the MNIST and CIFAR10 datasets for observation ranking.

Figure 5.12 provides a summary of the performance in terms of average AUC for all classes in both datasets, for all the methods used in our experiment. Moreover, the figure 5.14 and figure 5.13 show the detailed summary of the performance of all the methods for each class of MNIST and CIFAR10 used in our experiments respectively.

Table 5.13: Average ranking of different approaches based on Freidman's test for observation ranking on the CIFAR10 (p-value= $1.79e-13$) and MNIST (p-value= $4.14e-13$) datasets.

Method	Mean Rank (Cifar10)	Mean Rank (MNIST)
OC-SVM	5.0	7.0
DSVDD	7.1	5.8
HRN	3.0	2.5
Robust-kernel	7.8	3.8
SRO	6.4	9.0
DCAE	8.7	7.9
ICS	4.0	4.8
G-DRKSR	**1.0**	**1.1**
C-DRKSR	2.0	2.9

Table 5.13 provides an average ranking of different methods based on their performances in this set of experiments using the Friedman's test. Inspecting Table 5.10, one can observe that on both datasets, the proposed G-DRKSR is the best performing method that gives consistent results. The second best performing method corresponds to C-DRKSR.

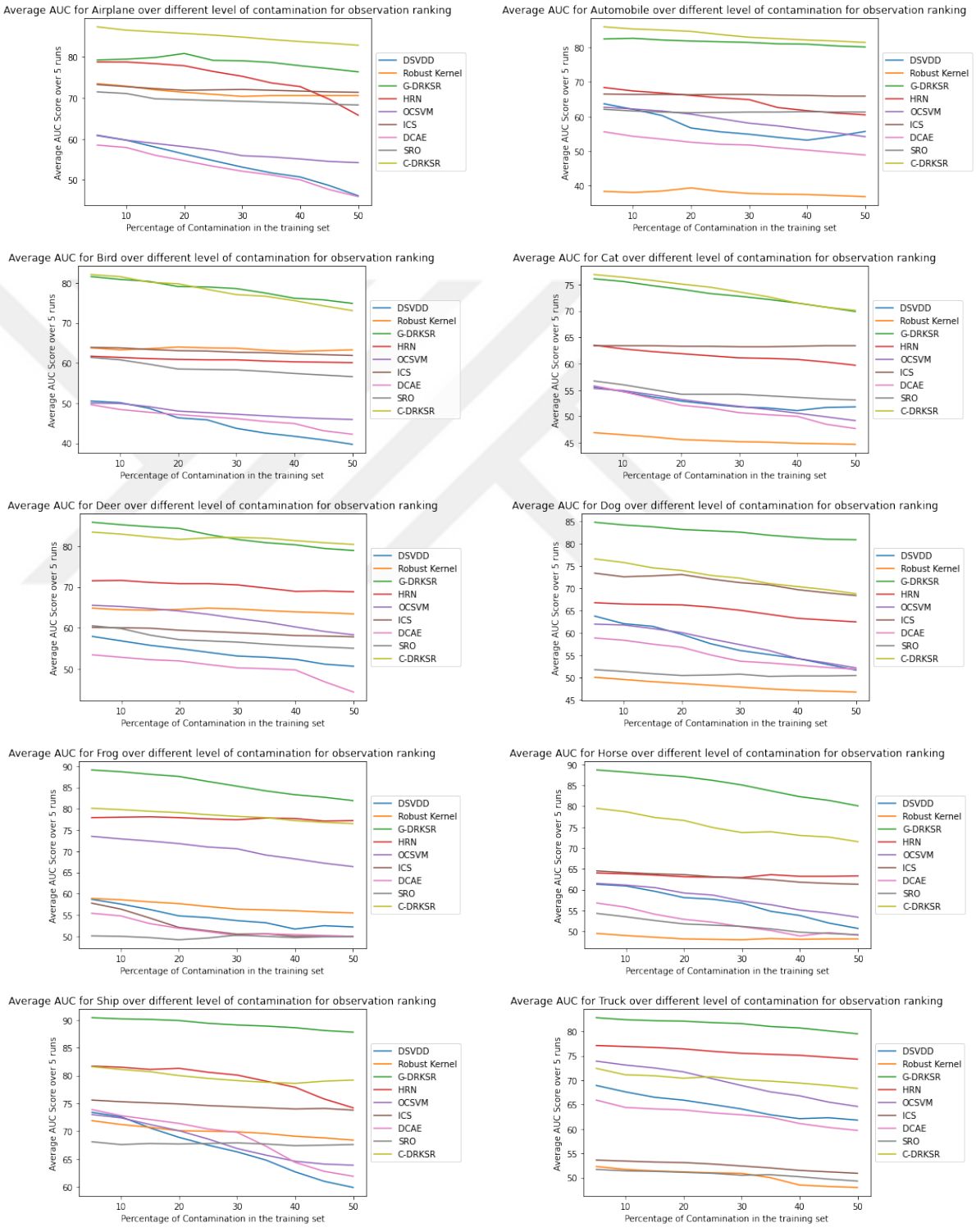

Figure 5.13: Average AUC of different approaches for each class on the Cifar10 dataset for observation ranking.

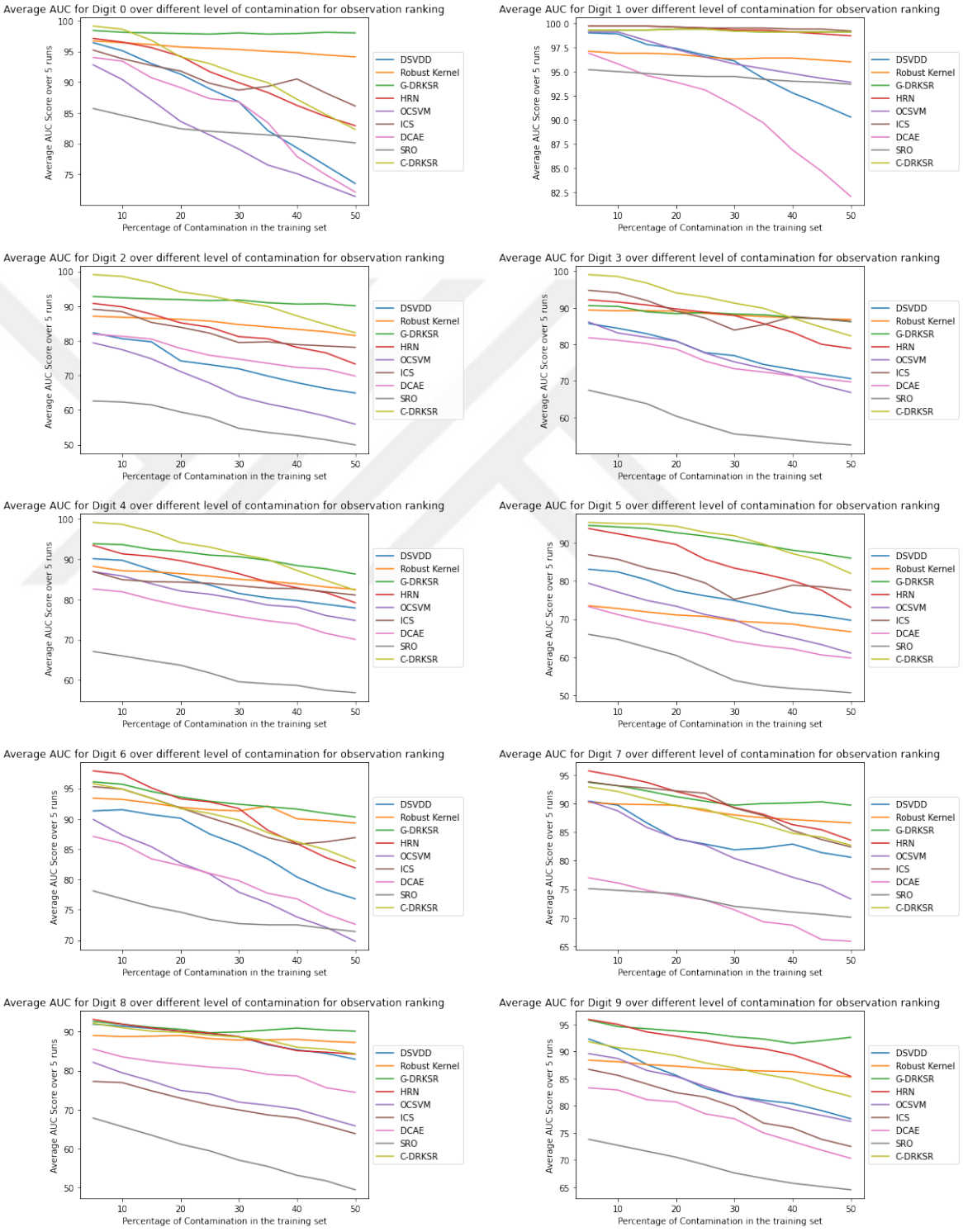

Figure 5.14: Average AUC's of different approaches for each class on the MNIST dataset for observation ranking.

Chapter 6

Conclusion

In this study, we addressed the open-set recognition task in terms of a novel one-class classification approach. For this purpose, we presented a novel one-class kernel-based deep learning approach called Deep Robust Kernel Spectral Regression (DRKSR) that is trainable in an end-to-end fashion. The proposed method utilizes the Fisher null concept for classification and benefits from regularisation theories in kernel-based algorithms to achieve robustness against contamination while at the same time enjoying deep end-to-end convolutional learning. In this context, the proposed approaches facilitated a concurred learning of a discriminative one-class representation of the data and a kernel Fisher null-space one-class classifier subject to a Tikhonov regularisation. For the learning stage, we presented two different alternating optimisation mechanisms: one based on gradient descent (G-DRKSR) and the other based on a closed-form solution (C-DRKSR). Through experiments on multiple datasets, it was illustrated that the proposed G-DRKSR approach provides close to state-of-the-art results when the datasets are not contaminated but it yields a substantially superior performance when the training set is contaminated with noisy observations. The C-DRKSR method, on the other hand, gave mixed results and may not work well for a given dataset but still yields results that are close to the state-of-the-art. Additionally, it was illustrated that the proposed methods can provide state-of-the-art results for unsupervised observation ranking where a performance improvement as high as 12%

in terms of average AUC may be obtained. As potential future directions for investigation, one may consider trying to learn the hyper-parameters of the network in an adaptive way while training. Moreover, the proposed approach can be extended into an open-set classification problem where multiple one-class classifiers are learned concurrently.

Bibliography

[1] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.

[2] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Transactions on information theory*, vol. 16, no. 1, pp. 41–46, 1970.

[3] P. L. Bartlett and M. H. Wegkamp, "Classification with a reject option using a hinge loss.," *Journal of Machine Learning Research*, vol. 9, no. 8, 2008.

[4] D. M. Tax and R. P. Duin, "Growing a multi-class classifier with a reject option," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1565–1570, 2008.

[5] L. Fischer, B. Hammer, and H. Wersing, "Optimal local rejection for classifiers," *Neurocomputing*, vol. 214, pp. 445–457, 2016.

[6] G. Fumera and F. Roli, "Support vector machines with embedded reject option," in *International Workshop on Support Vector Machines*, pp. 68–82, Springer, 2002.

[7] Y. Grandvalet, A. Rakotomamonjy, J. Keshet, and S. Canu, "Support vector machines with a reject option," *Advances in neural information processing systems*, vol. 21, 2008.

[8] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[9] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *Journal of machine Learning research*, vol. 2, no. Dec, pp. 139–154, 2001.

[10] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[11] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*, pp. 393–409, Springer, 2014.

[12] P. Bodesheim, A. Freytag, E. Rodner, M. Kemmler, and J. Denzler, "Kernel null space methods for novelty detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3374–3381, 2013.

[13] A. Garg and D. Roth, "Margin distribution and learning," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 210–217, 2003.

[14] L. Reyzin and R. E. Schapire, "How boosting the margin can also boost classifier complexity," in *Proceedings of the 23rd international conference on Machine learning*, pp. 753–760, 2006.

[15] F. Aiolli, G. D. San Martino, and A. Sperduti, "A kernel method for the optimization of the margin distribution," in *International Conference on Artificial Neural Networks*, pp. 305–314, Springer, 2008.

[16] K. Pelckmans, J. Suykens, and B. Moor, "A risk minimization principle for a class of parzen estimators," *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[17] P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," *arXiv preprint arXiv:2101.03064*, 2021.

[18] C. C. Aggarwal, "Outlier analysis second edition," 2016.

[19] Z. Zhu, Y. Wang, D. Robinson, D. Naiman, R. Vidal, and M. Tsakiris, "Dual principal component pursuit: Improved analysis and efficient algorithms," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," in *Proceedings of the ieee conference on computer vision and pattern recognition*, pp. 3395–3404, 2017.

[21] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*, pp. 146–157, Springer, 2017.

[22] P. Perera, R. Nallapati, and B. Xiang, "Ocgan: One-class novelty detection using gans with constrained latent representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2898–2906, 2019.

[23] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3379–3388, 2018.

[24] D. Abati, A. Porrello, S. Calderara, and R. Cucchiara, "Latent space autoregression for novelty detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 481–490, 2019.

[25] M. Z. Zaheer, J.-h. Lee, M. Astrid, and S.-I. Lee, "Old is gold: Redefining the adversarially learned one-class classifier training paradigm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14183–14193, 2020.

[26] P. Schlachter, Y. Liao, and B. Yang, "Deep one-class classification using intra-class splitting," in *2019 IEEE Data Science Workshop (DSW)*, pp. 100–104, IEEE, 2019.

[27] S. R. Arashloo and J. Kittler, "Robust one-class kernel spectral regression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 999–1013, 2020.

[28] J. Wang and A. Cherian, "Gods: Generalized one-class discriminative subspaces for anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8201–8211, 2019.

[29] P. Perera and V. M. Patel, "Dual-minimax probability machines for one-class mobile active authentication," in *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pp. 1–8, IEEE, 2018.

[30] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*, pp. 4393–4402, PMLR, 2018.

[31] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Processing Letters*, vol. 26, no. 2, pp. 277–281, 2018.

[32] W. Hu, M. Wang, Q. Qin, J. Ma, and B. Liu, "Hrn: A holistic approach to one class learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19111–19124, 2020.

[33] A. Mahdavi and M. Carvalho, "A survey on open set recognition," in *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, pp. 37–44, IEEE, 2021.

[34] C. Geng, S.-j. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.

[35] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014.

[36] H. Hoffmann, "Kernel pca for novelty detection," *Pattern recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[37] N. Kwak, "Principal component analysis based on l1-norm maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 9, pp. 1672–1680, 2008.

[38] T. Ding, Z. Zhu, T. Ding, Y. Yang, D. P. Robinson, M. C. Tsakiris, and R. Vidal, "Noisy dual principal component pursuit.," in *ICML*, pp. 1617–1625, 2019.

[39] G. Lerman and T. Maunu, "Fast, robust and non-convex subspace recovery," *Information and Inference: A Journal of the IMA*, vol. 7, no. 2, pp. 277–336, 2018.

[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] J. T. Andrews, E. J. Morton, and L. D. Griffin, "Detecting anomalous data using auto-encoders," *International Journal of Machine Learning and Computing*, vol. 6, no. 1, p. 21, 2016.

[42] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette, "Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes," *Computer Vision and Image Understanding*, vol. 172, pp. 88–97, 2018.

[43] S. Hawkins, H. He, G. Williams, and R. Baxter, "Outlier detection using replicator neural networks," in *International Conference on Data Warehousing and Knowledge Discovery*, pp. 170–180, Springer, 2002.

[44] J. Chen, S. Sathe, C. Aggarwal, and D. Turaga, "Outlier detection with autoencoder ensembles," in *Proceedings of the 2017 SIAM international conference on data mining*, pp. 90–98, SIAM, 2017.

[45] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion.," *Journal of machine learning research*, vol. 11, no. 12, 2010.

[46] A. Makhzani and B. Frey, "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013.

[47] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," *Advances in neural information processing systems*, vol. 28, 2015.

[48] J. T. Kwok, I. W.-H. Tsang, and J. M. Zurada, "A class of single-class minimax probability machines for novelty detection," *IEEE transactions on neural networks*, vol. 18, no. 3, pp. 778–785, 2007.

[49] S. Liang, Y. Li, and R. Srikant, "Enhancing the reliability of out-of-distribution image detection in neural networks," *arXiv preprint arXiv:1706.02690*, 2017.

[50] S. R. Arashloo and J. Kittler, "One-class kernel spectral regression," *arXiv preprint arXiv:1807.01085*, 2018.

[51] J. Liu, Z. Lian, Y. Wang, and J. Xiao, "Incremental kernel null space discriminant analysis for novelty detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 792–800, 2017.

[52] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.

[53] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[54] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[55] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006, p. 1100612, International Society for Optics and Photonics, 2019.

Appendices

Appendix A

Appendix

A.1 OCC Results for each class in MNIST dataset

In this section, the results of One-class classification in the presence of contamination are presented for each digit of MNIST dataset.

Table A.1: AUC's (mean±std%) of different approaches for Digit 0 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	91.7 ± 0.0	96.2 ± 1.4	98.3 ± 0.3	96.2 ± 0.0	94.3 ± 1.6	95.0 ± 1.6	$\mathbf{98.4 \pm 0.4}$	$\mathbf{98.4 \pm 0.2}$
10	88.8 ± 0.1	94.6 ± 1.8	97.3 ± 0.3	95.9 ± 0.0	92.7 ± 2.1	93.9 ± 1.2	97.9 ± 0.3	$\mathbf{98.2 \pm 0.2}$
15	86.7 ± 0.1	92.8 ± 2.1	96.4 ± 0.4	95.7 ± 0.0	90.4 ± 2.5	93.1 ± 1.0	$\mathbf{97.5 \pm 0.4}$	97.1 ± 0.2
20	82.8 ± 0.1	91.0 ± 2.4	95.2 ± 0.4	95.4 ± 0.0	88.3 ± 3.1	92.5 ± 1.1	$\mathbf{97.1 \pm 0.4}$	95.8 ± 0.3
25	81.1 ± 0.2	88.8 ± 3.0	93.1 ± 0.4	95.1 ± 0.0	85.1 ± 4.0	90.4 ± 2.0	$\mathbf{97.0 \pm 0.4}$	94.6 ± 0.3
30	78.8 ± 0.2	85.9 ± 3.6	91.8 ± 0.4	94.7 ± 0.0	83.2 ± 4.4	91.7 ± 1.9	$\mathbf{97.2 \pm 0.3}$	92.9 ± 0.4
35	76.1 ± 0.2	81.7 ± 4.1	90.7 ± 0.5	94.4 ± 0.0	81.1 ± 4.6	89.8 ± 1.8	$\mathbf{97.4 \pm 0.3}$	91.7 ± 0.4
40	74.9 ± 0.3	78.2 ± 4.8	88.0 ± 0.5	94.0 ± 0.0	77.3 ± 5.1	94.3 ± 1.7	$\mathbf{97.5 \pm 0.3}$	90.8 ± 0.4
45	72.5 ± 0.3	75.0 ± 5.3	86.1 ± 0.5	93.7 ± 0.0	72.8 ± 5.8	90.4 ± 1.5	$\mathbf{97.8 \pm 0.3}$	89.4 ± 0.5
50	70.4 ± 0.4	71.1 ± 5.8	83.4 ± 0.5	93.3 ± 0.0	70.4 ± 6.3	87.3 ± 1.6	$\mathbf{97.7 \pm 0.3}$	88.7 ± 0.5
Average	80.38	85.53	92.03	94.84	83.56	91.84	**97.55**	93.76

Table A.2: AUC's (mean±std%) of different approaches for Digit 1 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	99.2 ± 0.0	99.2 ± 0.2	99.7 ± 0.0	97.8 ± 0.0	96.7 ± 0.6	$\mathbf{99.8 \pm 0.1}$	99.5 ± 0.0	98.6 ± 0.1
10	99.0 ± 0.0	99.1 ± 0.2	99.7 ± 0.0	97.7 ± 0.0	96.1 ± 0.8	$\mathbf{99.7 \pm 0.2}$	99.5 ± 0.0	98.1 ± 0.1
15	98.1 ± 0.0	97.8 ± 0.6	99.5 ± 0.0	97.5 ± 0.0	95.7 ± 0.8	$\mathbf{99.6 \pm 0.2}$	99.4 ± 0.1	98.4 ± 0.1
20	97.6 ± 0.0	95.9 ± 0.9	99.6 ± 0.0	97.4 ± 0.0	94.3 ± 0.9	$\mathbf{99.5 \pm 0.2}$	99.2 ± 0.1	99.1 ± 0.1
25	97.2 ± 0.0	95.6 ± 1.0	99.3 ± 0.0	97.3 ± 0.0	93.5 ± 1.1	$\mathbf{99.5 \pm 0.2}$	99.2 ± 0.1	98.3 ± 0.1
30	96.2 ± 0.0	95.8 ± 1.0	99.1 ± 0.0	97.2 ± 0.0	91.7 ± 1.2	$\mathbf{99.4 \pm 0.2}$	99.2 ± 0.1	98.1 ± 0.1
35	95.9 ± 0.0	93.5 ± 1.7	99.0 ± 0.0	97.0 ± 0.0	90.3 ± 1.4	$\mathbf{99.2 \pm 0.2}$	99.1 ± 0.1	97.9 ± 0.1
40	95.3 ± 0.1	91.9 ± 2.1	99.1 ± 0.0	96.8 ± 0.0	87.6 ± 1.5	$\mathbf{99.1 \pm 0.2}$	99.0 ± 0.1	98.3 ± 0.1
45	94.4 ± 0.1	90.9 ± 2.4	98.8 ± 0.1	96.6 ± 0.0	84.6 ± 1.9	98.9 ± 0.2	$\mathbf{99.1 \pm 0.1}$	98.1 ± 0.2
50	93.2 ± 0.1	88.9 ± 2.8	98.7 ± 0.1	96.4 ± 0.0	81.9 ± 2.4	98.9 ± 0.2	$\mathbf{99.1 \pm 0.1}$	97.9 ± 0.2
Average	96.61	94.86	99.25	97.17	91.24	**99.36**	99.23	98.23

Table A.3: AUC's (mean±std%) of different approaches for Digit 2 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	78.9 ± 0.2	81.9 ± 3.4	92.6 ± 0.6	88.1 ± 0.0	81.7 ± 2.2	90.4 ± 1.7	$\mathbf{94.5 \pm 0.6}$	91.1 ± 0.9
10	75.9 ± 0.2	81.6 ± 3.5	90.9 ± 0.6	87.4 ± 0.0	80.9 ± 2.9	89.4 ± 1.9	$\mathbf{93.9 \pm 0.7}$	90.4 ± 0.9
15	72.6 ± 0.2	79.3 ± 3.8	88.4 ± 0.6	86.9 ± 0.0	80.1 ± 3.0	86.3 ± 1.8	$\mathbf{93.7 \pm 0.7}$	88.7 ± 1.0
20	69.2 ± 0.3	76.0 ± 4.2	86.8 ± 0.6	86.2 ± 0.0	78.5 ± 3.3	84.2 ± 1.8	$\mathbf{93.4 \pm 0.8}$	86.9 ± 1.1
25	65.7 ± 0.3	74.1 ± 4.5	83.7 ± 0.6	85.7 ± 0.0	76.3 ± 3.5	79.9 ± 1.5	$\mathbf{93.6 \pm 0.7}$	85.4 ± 1.1
30	59.5 ± 0.3	73.5 ± 4.6	82.1 ± 0.7	85.0 ± 0.0	75.3 ± 3.7	80.4 ± 1.2	$\mathbf{93.8 \pm 0.8}$	84.8 ± 1.2
35	59.5 ± 0.3	71.0 ± 4.9	80.1 ± 0.7	84.3 ± 0.0	74.3 ± 3.8	80.3 ± 1.8	$\mathbf{93.6 \pm 0.9}$	83.7 ± 1.2
40	58.5 ± 0.3	69.5 ± 5.1	77.8 ± 0.7	83.8 ± 0.0	73.2 ± 4.1	79.5 ± 1.9	$\mathbf{93.7 \pm 1.0}$	83.1 ± 1.3
45	56.1 ± 0.4	67.7 ± 5.6	75.3 ± 0.8	83.0 ± 0.0	71.7 ± 4.4	73.2 ± 2.4	$\mathbf{93.1 \pm 0.9}$	82.6 ± 1.3
50	53.4 ± 0.5	66.3 ± 5.7	72.5 ± 0.8	82.4 ± 0.0	70.4 ± 4.9	66.9 ± 2.9	$\mathbf{92.7 \pm 0.9}$	81.9 ± 1.3
Average	64.93	74.09	83.05	85.28	76.24	81.05	**93.60**	85.86

Table A.4: AUC's (mean±std%) of different approaches for Digit 3 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	85.7 ± 0.0	85.3 ± 2.8	$\mathbf{95.8 \pm 0.5}$	91.2 ± 0.0	84.6 ± 1.8	95.6 ± 2.7	94.1 ± 0.7	91.8 ± 0.6
10	82.3 ± 0.1	83.3 ± 3.1	94.2 ± 0.6	90.9 ± 0.0	82.2 ± 1.9	$\mathbf{94.8 \pm 2.9}$	93.8 ± 0.8	90.1 ± 0.7
15	80.6 ± 0.2	81.4 ± 3.5	92.7 ± 0.6	90.5 ± 0.0	81.1 ± 2.3	92.3 ± 2.6	$\mathbf{93.1 \pm 1.0}$	88.6 ± 0.9
20	77.8 ± 0.2	79.4 ± 3.8	90.9 ± 0.7	90.2 ± 0.0	79.1 ± 2.5	90.6 ± 2.2	$\mathbf{92.4 \pm 1.2}$	87.3 ± 1.0
25	75.4 ± 0.2	76.5 ± 4.3	89.1 ± 0.6	89.8 ± 0.0	75.7 ± 2.7	87.6 ± 2.4	$\mathbf{91.7 \pm 1.1}$	86.1 ± 1.0
30	73.7 ± 0.2	74.5 ± 4.5	88.4 ± 0.6	89.4 ± 0.0	72.5 ± 3.1	84.6 ± 2.7	$\mathbf{91.1 \pm 1.1}$	85.5 ± 1.1
35	71.8 ± 0.2	73.9 ± 4.6	86.1 ± 0.7	89.0 ± 0.0	71.9 ± 3.4	81.2 ± 2.9	$\mathbf{90.5 \pm 1.2}$	83.8 ± 1.1
40	69.4 ± 0.2	73.4 ± 4.6	84.5 ± 0.7	88.5 ± 0.0	71.1 ± 3.5	90.5 ± 3.1	$\mathbf{90.1 \pm 1.1}$	82.4 ± 1.2
45	67.7 ± 0.2	71.8 ± 5.0	82.6 ± 0.8	88.1 ± 0.0	70.6 ± 3.8	90.3 ± 3.3	$\mathbf{90.4 \pm 1.3}$	81.1 ± 1.3
50	65.2 ± 0.2	70.3 ± 5.3	80.9 ± 0.8	87.7 ± 0.0	70.1 ± 4.1	89.7 ± 4.0	$\mathbf{90.2 \pm 1.5}$	78.6 ± 1.5
Average	74.96	76.98	88.52	89.53	75.89	89.72	**91.74**	85.53

Table A.5: AUC's (mean±std%) of different approaches for Digit 4 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	87.9 ± 0.0	91.9 ± 1.4	$\mathbf{95.7 \pm 0.2}$	88.5 ± 0.0	85.8 ± 2.3	90.5 ± 2.4	94.3 ± 0.8	93.9 ± 0.7
10	86.9 ± 0.0	90.6 ± 1.5	93.3 ± 0.2	87.9 ± 0.0	82.1 ± 2.5	89.1 ± 2.9	$\mathbf{93.6 \pm 0.9}$	93.2 ± 0.7
15	84.5 ± 0.0	88.4 ± 1.9	91.8 ± 0.2	87.2 ± 0.0	81.1 ± 2.7	87.3 ± 3.0	$\mathbf{93.1 \pm 0.8}$	92.4 ± 0.7
20	83.1 ± 0.1	86.4 ± 2.2	90.5 ± 0.2	86.7 ± 0.0	79.8 ± 3.1	86.2 ± 3.1	$\mathbf{92.8 \pm 0.9}$	90.6 ± 0.9
25	81.5 ± 0.1	84.9 ± 2.7	88.9 ± 0.2	86.1 ± 0.0	78.7 ± 3.2	85.4 ± 2.4	$\mathbf{93.1 \pm 0.9}$	89.4 ± 1.0
30	80.7 ± 0.2	82.2 ± 3.0	87.6 ± 0.3	85.5 ± 0.0	77.1 ± 3.2	84.1 ± 2.6	$\mathbf{93.0 \pm 0.9}$	88.9 ± 1.0
35	78.7 ± 0.2	81.6 ± 3.6	85.5 ± 0.3	84.8 ± 0.0	76.5 ± 3.4	83.1 ± 3.4	$\mathbf{92.1 \pm 0.9}$	85.4 ± 1.2
40	77.9 ± 0.2	80.2 ± 3.8	83.3 ± 0.3	84.2 ± 0.0	74.2 ± 3.7	82.5 ± 4.0	$\mathbf{91.6 \pm 1.0}$	83.1 ± 1.3
45	76.4 ± 0.2	79.6 ± 4.0	81.4 ± 0.3	83.5 ± 0.0	72.7 ± 3.9	82.9 ± 4.4	$\mathbf{91.7 \pm 1.0}$	81.9 ± 1.5
50	75.6 ± 0.2	79.5 ± 4.1	79.9 ± 0.3	82.9 ± 0.0	71.7 ± 4.1	80.4 ± 5.0	$\mathbf{91.1 \pm 1.0}$	78.9 ± 1.5
Average	81.32	84.53	87.79	85.73	77.97	85.15	**92.64**	87.78

Table A.6: AUC's (mean±std%) of different approaches for Digit 5 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	78.8 ± 0.0	84.9 ± 1.9	$\mathbf{95.3 \pm 0.5}$	75.3 ± 0.0	72.7 ± 3.1	87.9 ± 2.8	93.7 ± 0.5	91.8 ± 0.8
10	76.5 ± 0.0	83.1 ± 2.1	93.1 ± 0.6	74.6 ± 0.0	70.8 ± 3.4	86.2 ± 3.3	$\mathbf{93.3 \pm 0.6}$	90.4 ± 0.9
15	74.4 ± 0.1	80.7 ± 2.9	91.6 ± 0.6	73.9 ± 0.0	68.7 ± 3.8	85.3 ± 3.7	$\mathbf{92.9 \pm 0.5}$	88.9 ± 0.9
20	72.7 ± 0.2	77.9 ± 3.5	88.9 ± 0.5	73.2 ± 0.0	67.4 ± 4.2	83.6 ± 4.1	$\mathbf{92.5 \pm 0.7}$	87.6 ± 1.0
25	70.1 ± 0.2	74.7 ± 4.2	86.8 ± 0.6	72.5 ± 0.0	66.2 ± 4.4	79.6 ± 2.9	$\mathbf{92.3 \pm 0.7}$	85.8 ± 1.1
30	68.7 ± 0.2	70.4 ± 5.1	84.9 ± 0.5	71.9 ± 0.0	64.7 ± 4.7	75.3 ± 3.0	$\mathbf{92.1 \pm 0.7}$	84.1 ± 1.3
35	66.9 ± 0.2	69.9 ± 5.3	82.4 ± 0.8	71.3 ± 0.0	62.9 ± 4.9	76.3 ± 2.8	$\mathbf{91.3 \pm 0.8}$	82.9 ± 1.3
40	65.2 ± 0.3	69.3 ± 5.5	79.2 ± 0.9	70.6 ± 0.0	62.4 ± 5.2	77.2 ± 2.6	$\mathbf{90.2 \pm 1.0}$	81.6 ± 1.4
45	63.7 ± 0.4	68.9 ± 5.6	76.3 ± 1.1	70.1 ± 0.0	61.7 ± 5.6	74.3 ± 2.9	$\mathbf{89.7 \pm 0.9}$	79.8 ± 1.4
50	61.5 ± 0.4	68.2 ± 5.6	74.2 ± 1.3	69.4 ± 0.0	60.3 ± 5.8	75.1 ± 3.1	$\mathbf{89.2 \pm 1.0}$	76.6 ± 1.6
Average	69.85	74.80	85.27	72.28	65.78	80.08	**91.72**	84.95

Table A.7: AUC's (mean±std%) of different approaches for Digit 6 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	90.4 ± 0.0	93.4 ± 2.6	$\mathbf{97.9 \pm 0.1}$	91.8 ± 0.0	88.3 ± 2.1	95.1 ± 1.3	96.9 ± 0.4	95.1 ± 0.3
10	88.2 ± 0.0	91.0 ± 3.0	$\mathbf{97.1 \pm 0.1}$	91.4 ± 0.0	87.7 ± 2.2	94.1 ± 1.5	96.2 ± 0.5	94.4 ± 0.3
15	85.7 ± 0.0	91.0 ± 3.1	95.8 ± 0.2	91.0 ± 0.0	85.4 ± 2.4	93.2 ± 2.4	$\mathbf{95.9 \pm 0.5}$	93.7 ± 0.4
20	83.0 ± 0.0	90.0 ± 3.3	94.2 ± 0.3	90.6 ± 0.0	83.5 ± 2.7	92.1 ± 3.1	$\mathbf{95.7 \pm 0.6}$	93.1 ± 0.4
25	80.5 ± 0.1	87.1 ± 4.2	92.5 ± 0.2	90.2 ± 0.0	81.9 ± 3.0	90.6 ± 2.8	$\mathbf{95.5 \pm 0.6}$	91.9 ± 0.5
30	78.5 ± 0.2	84.8 ± 4.9	91.1 ± 0.3	89.7 ± 0.0	79.6 ± 3.2	89.4 ± 2.6	$\mathbf{95.2 \pm 0.7}$	91.1 ± 0.5
35	76.6 ± 0.2	82.0 ± 5.1	88.4 ± 0.3	89.2 ± 0.0	77.4 ± 3.5	87.5 ± 3.2	$\mathbf{95.5 \pm 0.6}$	88.4 ± 0.7
40	74.3 ± 0.2	80.0 ± 5.6	86.4 ± 0.3	88.8 ± 0.0	75.6 ± 3.9	86.5 ± 4.0	$\mathbf{95.6 \pm 0.7}$	86.7 ± 0.8
45	72.6 ± 0.2	77.1 ± 5.9	84.1 ± 0.4	88.3 ± 0.0	72.2 ± 4.1	86.8 ± 3.8	$\mathbf{94.9 \pm 0.7}$	83.6 ± 1.0
50	69.9 ± 0.2	75.0 ± 6.3	82.3 ± 0.4	87.7 ± 0.0	70.7 ± 4.2	87.7 ± 3.9	$\mathbf{94.1 \pm 0.8}$	81.9 ± 1.0
Average	79.97	85.14	90.98	89.87	80.23	90.30	**95.55**	89.99

Table A.8: AUC's (mean±std%) of different approaches for Digit 7 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	91.1 ± 0.0	91.6 ± 2.1	$\mathbf{96.2 \pm 0.1}$	90.7 ± 0.0	78.7 ± 3.4	94.2 ± 0.6	94.7 ± 0.6	93.9 ± 0.5
10	89.3 ± 0.0	90.8 ± 2.4	$\mathbf{95.8 \pm 0.2}$	90.3 ± 0.0	77.1 ± 3.8	93.1 ± 0.6	94.3 ± 0.7	92.4 ± 0.5
15	87.5 ± 0.0	87.4 ± 2.9	$\mathbf{95.2 \pm 0.2}$	89.9 ± 0.0	75.5 ± 3.9	92.7 ± 0.7	94.9 ± 0.7	91.8 ± 0.5
20	85.2 ± 0.0	84.7 ± 3.5	94.3 ± 0.2	89.4 ± 0.0	74.3 ± 4.1	92.5 ± 0.9	$\mathbf{94.6 \pm 0.6}$	90.9 ± 0.6
25	83.3 ± 0.0	83.7 ± 3.6	91.6 ± 0.2	89.0 ± 0.0	73.2 ± 4.3	90.6 ± 1.0	$\mathbf{94.4 \pm 0.5}$	88.7 ± 0.8
30	81.5 ± 0.0	82.8 ± 3.9	90.5 ± 0.2	88.5 ± 0.0	72.4 ± 4.2	88.1 ± 1.1	$\mathbf{94.0 \pm 0.6}$	86.8 ± 0.8
35	79.6 ± 0.0	83.1 ± 3.8	89.5 ± 0.4	88.0 ± 0.0	70.7 ± 4.7	86.3 ± 1.6	$\mathbf{93.7 \pm 0.7}$	85.4 ± 1.1
40	78.3 ± 0.1	83.2 ± 3.9	88.6 ± 0.4	87.5 ± 0.0	69.8 ± 5.1	84.4 ± 2.0	$\mathbf{93.3 \pm 0.8}$	83.7 ± 1.1
45	76.5 ± 0.1	82.5 ± 3.9	87.1 ± 0.4	87.1 ± 0.0	68.4 ± 5.2	85.8 ± 2.4	$\mathbf{93.9 \pm 0.8}$	82.1 ± 1.2
50	74.8 ± 0.1	81.8 ± 4.0	85.9 ± 0.5	86.5 ± 0.0	67.7 ± 5.5	87.1 ± 2.9	$\mathbf{93.6 \pm 0.7}$	80.6 ± 1.5
Average	82.71	85.16	91.47	88.69	72.78	89.48	**94.14**	87.63

Table A.9: AUC's (mean±std%) of different approaches for Digit 8 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	82.7 ± 0.2	90.0 ± 1.9	93.2 ± 0.2	90.3 ± 0.0	86.2 ± 1.8	80.4 ± 0.5	$\mathbf{93.4 \pm 0.7}$	91.1 ± 0.9
10	80.4 ± 0.2	89.6 ± 2.1	$\mathbf{92.7 \pm 0.2}$	90.0 ± 0.0	85.4 ± 1.9	77.1 ± 0.8	92.3 ± 0.8	89.9 ± 0.9
15	78.2 ± 0.2	89.0 ± 2.1	$\mathbf{91.9 \pm 0.2}$	89.9 ± 0.0	84.7 ± 1.9	76.3 ± 0.9	$\mathbf{91.9 \pm 0.7}$	87.8 ± 1.0
20	76.0 ± 0.3	88.7 ± 2.3	91.2 ± 0.2	89.7 ± 0.0	83.6 ± 2.0	75.8 ± 1.1	$\mathbf{92.5 \pm 0.8}$	86.2 ± 1.0
25	74.4 ± 0.3	87.9 ± 2.5	90.6 ± 0.2	89.5 ± 0.0	82.8 ± 2.2	73.6 ± 0.9	$\mathbf{92.4 \pm 0.7}$	84.9 ± 1.1
30	72.3 ± 0.4	86.0 ± 2.8	89.2 ± 0.2	89.2 ± 0.0	81.4 ± 2.3	71.4 ± 0.8	$\mathbf{92.1 \pm 0.8}$	83.8 ± 1.3
35	71.2 ± 0.4	84.8 ± 3.2	87.5 ± 0.3	89.1 ± 0.0	80.1 ± 2.2	70.1 ± 0.9	$\mathbf{92.3 \pm 1.0}$	82.7 ± 1.3
40	69.0 ± 0.5	82.8 ± 3.7	86.6 ± 0.4	88.9 ± 0.0	79.4 ± 2.5	69.5 ± 0.8	$\mathbf{92.5 \pm 0.9}$	81.9 ± 1.5
45	68.2 ± 0.5	82.6 ± 3.7	85.9 ± 0.3	88.7 ± 0.0	77.9 ± 2.7	68.7 ± 1.8	$\mathbf{92.3 \pm 1.0}$	80.6 ± 1.6
50	66.6 ± 0.5	82.0 ± 3.7	85.5 ± 0.4	88.5 ± 0.0	76.7 ± 2.9	66.5 ± 2.2	$\mathbf{92.1 \pm 1.0}$	79.8 ± 1.8
Average	73.90	86.34	89.43	89.38	81.82	72.94	**92.38**	84.87

Table A.10: AUC's (mean±std%) of different approaches for Digit 9 for one-class classification on the MNIST dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	91.7 ± 0.0	94.9 ± 1.0	95.9 ± 0.2	89.8 ± 0.0	84.9 ± 2.5	88.4 ± 0.9	$\mathbf{96.2 \pm 0.3}$	91.9 ± 0.6
10	89.4 ± 0.1	91.8 ± 2.1	95.4 ± 0.3	89.6 ± 0.0	83.3 ± 2.7	86.6 ± 1.2	$\mathbf{95.7 \pm 0.4}$	90.6 ± 0.6
15	87.9 ± 0.1	87.6 ± 3.2	94.6 ± 0.3	89.4 ± 0.0	81.8 ± 3.0	85.1 ± 1.7	$\mathbf{94.8 \pm 0.5}$	89.7 ± 0.6
20	86.5 ± 0.1	85.2 ± 3.5	93.7 ± 0.3	89.1 ± 0.0	80.3 ± 3.1	83.3 ± 2.1	$\mathbf{94.4 \pm 0.4}$	88.1 ± 0.7
25	84.7 ± 0.1	82.4 ± 4.2	93.1 ± 0.3	88.8 ± 0.0	79.4 ± 3.1	81.4 ± 2.3	$\mathbf{93.9 \pm 0.5}$	87.5 ± 0.9
30	82.9 ± 0.3	80.0 ± 4.5	92.4 ± 0.3	88.6 ± 0.0	77.7 ± 3.4	80.6 ± 2.9	$\mathbf{93.4 \pm 0.7}$	86.2 ± 1.0
35	80.5 ± 0.3	80.4 ± 4.4	91.8 ± 0.3	88.3 ± 0.0	75.7 ± 3.7	78.4 ± 2.7	$\mathbf{93.0 \pm 0.6}$	85.1 ± 1.0
40	79.8 ± 0.3	81.1 ± 4.2	90.2 ± 0.3	88.0 ± 0.0	74.3 ± 3.8	77.2 ± 3.0	$\mathbf{92.7 \pm 0.7}$	84.2 ± 1.1
45	78.6 ± 0.3	78.4 ± 4.9	89.0 ± 0.4	87.8 ± 0.0	72.7 ± 4.1	76.5 ± 3.4	$\mathbf{93.2 \pm 0.6}$	82.7 ± 1.3
50	76.9 ± 0.3	76.5 ± 5.3	87.9 ± 0.6	87.5 ± 0.0	70.6 ± 4.4	74.1 ± 3.8	$\mathbf{93.0 \pm 0.7}$	81.3 ± 1.4
Average	83.89	83.83	92.40	88.69	78.07	81.16	**94.03**	86.73

A.2 OCC Results for each class in CIFAR10 dataset

In this section, the results of One-class classification in the presence of contamination are presented for each class of Cifar-10 dataset.

Table A.11: AUC's (mean±std%) of different approaches for AIRPLANE class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	60.9 ± 0.9	60.5 ± 4.0	77.8 ± 0.2	74.3 ± 0.0	58.7 ± 5.3	74.5 ± 1.1	78.4 ± 1.0	$\mathbf{81.1 \pm 0.8}$
10	60.4 ± 1.0	60.1 ± 4.0	77.4 ± 0.2	74.0 ± 0.0	58.1 ± 5.4	73.8 ± 1.3	78.3 ± 1.1	$\mathbf{80.7 \pm 0.8}$
15	59.6 ± 1.0	58.2 ± 4.1	77.1 ± 0.2	73.9 ± 0.0	57.3 ± 5.4	73.1 ± 1.2	78.2 ± 1.1	$\mathbf{80.4 \pm 0.9}$
20	58.7 ± 1.2	56.7 ± 4.2	77.6 ± 0.2	73.6 ± 0.0	55.3 ± 5.6	72.8 ± 1.4	78.1 ± 1.0	$\mathbf{80.2 \pm 0.8}$
25	57.9 ± 1.1	54.2 ± 4.8	77.1 ± 0.2	73.4 ± 0.0	53.8 ± 5.9	72.5 ± 1.5	77.8 ± 1.2	$\mathbf{79.9 \pm 0.9}$
30	56.2 ± 1.2	52.9 ± 4.5	76.6 ± 0.4	73.3 ± 0.0	52.6 ± 6.0	72.1 ± 1.5	77.6 ± 1.1	$\mathbf{79.5 \pm 1.0}$
35	55.9 ± 1.1	51.2 ± 4.6	74.4 ± 0.7	73.1 ± 0.0	51.4 ± 6.0	72.3 ± 1.6	77.8 ± 1.2	$\mathbf{78.7 \pm 1.0}$
40	55.0 ± 1.2	49.4 ± 4.9	72.7 ± 0.9	72.9 ± 0.0	50.6 ± 6.2	71.6 ± 1.8	77.9 ± 1.1	$\mathbf{78.1 \pm 1.0}$
45	54.6 ± 1.3	45.7 ± 5.1	67.2 ± 2.1	72.8 ± 0.0	46.7 ± 6.8	71.1 ± 2.0	77.4 ± 1.1	$\mathbf{77.6 \pm 1.1}$
50	54.2 ± 1.2	43.5 ± 5.5	61.1 ± 3.7	72.6 ± 0.0	44.3 ± 7.1	70.4 ± 2.1	77.1 ± 1.1	$\mathbf{77.2 \pm 1.1}$
Average	57.34	53.24	73.90	73.39	52.88	72.42	77.86	**79.34**

Table A.12: AUC's (mean±std%) of different approaches for AUTOMOBILE class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	63.1 ± 0.7	62.5 ± 2.9	67.8 ± 1.3	38.4 ± 0.0	58.7 ± 3.1	67.9 ± 2.3	$\mathbf{76.9 \pm 1.9}$	71.1 ± 1.3
10	62.7 ± 0.7	61.9 ± 3.0	66.1 ± 1.2	38.2 ± 0.0	58.1 ± 3.0	67.5 ± 2.5	$\mathbf{75.8 \pm 2.0}$	70.8 ± 1.3
15	61.9 ± 0.9	59.1 ± 3.2	65.6 ± 1.3	38.0 ± 0.0	53.7 ± 4.2	66.9 ± 2.6	$\mathbf{75.5 \pm 2.0}$	69.9 ± 1.4
20	61.0 ± 1.0	57.8 ± 3.5	65.1 ± 1.3	37.8 ± 0.0	53.1 ± 4.1	66.1 ± 2.8	$\mathbf{75.0 \pm 2.1}$	70.4 ± 1.3
25	60.1 ± 1.1	56.7 ± 3.7	64.4 ± 1.4	37.3 ± 0.0	53.9 ± 4.0	64.9 ± 2.6	$\mathbf{74.3 \pm 2.3}$	70.6 ± 1.2
30	58.7 ± 1.1	55.4 ± 4.0	63.8 ± 1.7	37.3 ± 0.0	52.3 ± 4.2	63.2 ± 2.5	$\mathbf{74.1 \pm 2.3}$	71.3 ± 1.0
35	57.1 ± 1.3	55.9 ± 4.1	62.2 ± 1.8	37.2 ± 0.0	51.7 ± 4.0	62.8 ± 2.7	$\mathbf{73.5 \pm 2.5}$	71.4 ± 0.9
40	56.6 ± 1.2	56.6 ± 4.0	60.9 ± 2.0	37.0 ± 0.0	50.3 ± 4.3	62.5 ± 2.6	$\mathbf{72.8 \pm 2.7}$	70.9 ± 1.0
45	56.0 ± 1.3	57.9 ± 3.8	59.9 ± 2.3	36.9 ± 0.0	50.1 ± 4.2	61.4 ± 2.7	$\mathbf{72.4 \pm 2.9}$	70.2 ± 1.1
50	54.1 ± 1.3	58.0 ± 3.6	61.7 ± 2.0	36.7 ± 0.0	49.8 ± 4.3	60.8 ± 2.8	$\mathbf{72.1 \pm 3.0}$	71.1 ± 1.1
Average	59.13	58.18	63.75	37.48	52.65	64.40	**74.24**	70.67

Table A.13: AUC's (mean±std%) of different approaches for BIRD class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	50.0 ± 0.5	49.9 ± 1.0	60.7 ± 0.3	63.3 ± 0.0	48.7 ± 2.4	65.4 ± 0.7	$\mathbf{67.4 \pm 1.4}$	65.8 ± 1.2
10	49.9 ± 0.5	47.6 ± 1.2	60.4 ± 0.3	63.2 ± 0.0	48.1 ± 2.4	64.9 ± 0.7	66.2 ± 1.4	$\mathbf{66.7 \pm 1.2}$
15	49.0 ± 0.6	44.7 ± 1.5	60.1 ± 0.3	63.1 ± 0.0	47.7 ± 2.5	64.4 ± 0.8	65.8 ± 1.3	$\mathbf{66.5 \pm 1.1}$
20	48.9 ± 0.5	41.2 ± 1.7	60.1 ± 0.4	63.1 ± 0.0	47.4 ± 2.5	64.2 ± 0.8	65.1 ± 1.4	$\mathbf{66.3 \pm 1.2}$
25	48.0 ± 0.6	42.7 ± 1.6	60.0 ± 0.4	63.0 ± 0.0	47.1 ± 2.5	63.8 ± 0.8	64.9 ± 1.1	$\mathbf{66.5 \pm 1.3}$
30	47.8 ± 0.7	42.5 ± 1.7	60.6 ± 0.4	63.0 ± 0.0	46.4 ± 2.7	63.3 ± 0.9	65.1 ± 1.4	$\mathbf{66.7 \pm 1.0}$
35	47.0 ± 0.7	41.7 ± 2.1	60.1 ± 0.4	62.9 ± 0.0	46.1 ± 2.6	62.9 ± 1.0	64.3 ± 1.6	$\mathbf{66.1 \pm 1.1}$
40	46.8 ± 0.7	40.5 ± 2.4	60.6 ± 0.4	62.9 ± 0.0	45.4 ± 2.7	62.1 ± 1.3	64.5 ± 1.5	$\mathbf{65.4 \pm 1.0}$
45	46.6 ± 0.9	39.9 ± 2.7	60.4 ± 0.4	62.8 ± 0.0	44.7 ± 2.9	61.8 ± 1.2	63.9 ± 1.8	$\mathbf{64.7 \pm 1.2}$
50	46.0 ± 0.9	39.7 ± 2.6	59.9 ± 0.4	62.8 ± 0.0	43.8 ± 3.1	61.4 ± 1.2	63.3 ± 1.9	$\mathbf{63.9 \pm 1.4}$
Average	48.00	43.04	60.29	63.01	46.54	63.42	**65.05**	65.86

Table A.14: AUC's (mean±std%) of different approaches for CAT class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	55.6 ± 1.2	58.8 ± 1.4	$\mathbf{63.5 \pm 0.7}$	45.7 ± 0.0	56.7 ± 1.3	63.1 ± 0.8	63.4 ± 1.0	62.7 ± 0.8
10	55.0 ± 1.3	56.7 ± 1.9	62.4 ± 0.9	45.5 ± 0.0	55.4 ± 1.6	62.9 ± 0.8	$\mathbf{62.9 \pm 1.0}$	62.5 ± 0.8
15	54.2 ± 1.2	55.8 ± 2.1	62.3 ± 0.9	45.4 ± 0.0	53.6 ± 1.9	62.7 ± 0.7	$\mathbf{62.4 \pm 1.1}$	61.9 ± 0.8
20	53.7 ± 1.3	54.1 ± 2.3	62.6 ± 0.9	45.2 ± 0.0	52.8 ± 2.2	62.5 ± 0.8	$\mathbf{62.7 \pm 1.1}$	61.7 ± 0.8
25	53.0 ± 1.4	52.7 ± 2.6	61.9 ± 0.9	45.1 ± 0.0	51.1 ± 2.5	61.9 ± 0.8	$\mathbf{62.3 \pm 1.1}$	61.9 ± 0.8
30	52.1 ± 1.2	49.8 ± 2.9	61.2 ± 1.0	44.9 ± 0.0	50.9 ± 2.8	61.5 ± 0.9	$\mathbf{62.5 \pm 1.2}$	61.2 ± 0.8
35	51.7 ± 1.1	50.5 ± 2.9	61.4 ± 1.0	44.8 ± 0.0	51.7 ± 2.5	61.8 ± 0.9	$\mathbf{61.9 \pm 1.3}$	60.9 ± 0.9
40	50.0 ± 1.0	50.8 ± 3.0	60.8 ± 1.0	44.7 ± 0.0	50.6 ± 2.9	62.1 ± 0.9	$\mathbf{62.4 \pm 1.2}$	61.1 ± 0.9
45	49.2 ± 0.9	51.9 ± 2.9	60.5 ± 1.0	44.6 ± 0.0	47.9 ± 3.4	61.9 ± 1.0	$\mathbf{62.1 \pm 1.3}$	61.2 ± 0.9
50	49.0 ± 0.9	52.5 ± 2.8	59.9 ± 1.0	44.5 ± 0.0	48.1 ± 3.2	61.7 ± 1.3	$\mathbf{61.9 \pm 1.5}$	60.8 ± 0.9
Average	52.35	53.36	61.65	45.04	51.88	62.21	**62.45**	61.59

Table A.15: AUC's (mean±std%) of different approaches for DEER class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	66.2 ± 0.8	58.8 ± 1.2	71.2 ± 0.5	65.0 ± 0.0	55.4 ± 1.4	62.1 ± 0.9	68.1 ± 2.9	$\mathbf{71.7 \pm 1.4}$
10	65.8 ± 0.8	57.9 ± 1.4	70.5 ± 0.6	64.8 ± 0.0	53.1 ± 1.4	61.2 ± 1.1	67.8 ± 2.8	$\mathbf{71.4 \pm 1.4}$
15	64.6 ± 0.9	56.1 ± 1.7	70.4 ± 0.6	64.6 ± 0.0	52.9 ± 1.6	60.1 ± 1.4	67.6 ± 2.9	$\mathbf{70.9 \pm 1.5}$
20	63.8 ± 0.9	54.9 ± 2.0	$\mathbf{70.5 \pm 0.5}$	64.3 ± 0.0	52.5 ± 1.7	59.4 ± 1.7	67.4 ± 3.0	70.3 ± 1.5
25	62.7 ± 0.9	55.1 ± 1.9	$\mathbf{70.6 \pm 0.6}$	64.1 ± 0.0	51.7 ± 1.9	58.4 ± 1.5	66.8 ± 3.0	70.5 ± 1.5
30	61.9 ± 1.0	54.9 ± 1.9	$\mathbf{70.4 \pm 0.5}$	63.9 ± 0.0	50.8 ± 2.1	57.1 ± 1.6	66.4 ± 3.1	70.2 ± 1.5
35	60.7 ± 1.0	52.7 ± 2.2	$\mathbf{70.1 \pm 0.8}$	63.7 ± 0.0	50.4 ± 2.0	56.2 ± 1.4	66.1 ± 3.1	69.9 ± 1.5
40	60.0 ± 1.1	51.8 ± 2.5	$\mathbf{69.5 \pm 0.9}$	63.6 ± 0.0	51.1 ± 1.9	54.8 ± 1.5	65.7 ± 3.2	69.4 ± 1.6
45	59.1 ± 1.1	48.9 ± 2.9	$\mathbf{69.1 \pm 1.1}$	63.3 ± 0.0	48.6 ± 2.4	54.1 ± 1.6	66.2 ± 3.2	68.9 ± 1.7
50	58.0 ± 1.1	47.4 ± 3.3	$\mathbf{69.2 \pm 1.3}$	63.3 ± 0.0	45.7 ± 2.9	53.7 ± 1.8	65.4 ± 3.2	68.1 ± 1.9
Average	62.28	53.85	**70.15**	64.06	51.22	57.71	66.75	70.13

Table A.16: AUC's (mean±std%) of different approaches for DOG class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	61.7 ± 1.8	62.9 ± 2.6	66.8 ± 0.1	49.4 ± 0.0	59.7 ± 2.2	$\mathbf{71.8 \pm 1.0}$	66.2 ± 1.7	63.9 ± 0.8
10	61.0 ± 1.8	62.6 ± 2.6	66.4 ± 0.1	49.0 ± 0.0	59.3 ± 2.3	$\mathbf{71.6 \pm 1.2}$	66.1 ± 1.7	63.1 ± 0.8
15	60.3 ± 1.8	61.7 ± 2.8	66.1 ± 0.2	48.8 ± 0.0	58.2 ± 2.6	$\mathbf{72.1 \pm 1.2}$	65.8 ± 1.7	63.4 ± 0.7
20	59.1 ± 1.9	60.9 ± 2.9	66.0 ± 0.3	48.4 ± 0.0	57.6 ± 2.7	$\mathbf{72.8 \pm 1.4}$	65.4 ± 1.8	63.1 ± 0.8
25	58.3 ± 1.0	58.7 ± 3.2	65.3 ± 0.3	48.2 ± 0.0	55.7 ± 2.9	$\mathbf{71.9 \pm 1.3}$	65.3 ± 1.8	63.9 ± 0.8
30	57.6 ± 1.0	55.8 ± 3.5	64.8 ± 0.3	48.0 ± 0.0	54.1 ± 3.1	$\mathbf{71.4 \pm 1.4}$	65.0 ± 1.8	64.2 ± 0.6
35	57.0 ± 1.2	54.9 ± 3.5	63.9 ± 0.3	47.6 ± 0.0	54.9 ± 2.9	$\mathbf{71.1 \pm 1.4}$	64.3 ± 1.8	64.0 ± 0.5
40	55.7 ± 1.3	55.4 ± 3.4	63.6 ± 0.3	47.6 ± 0.0	53.8 ± 3.1	$\mathbf{71.3 \pm 1.5}$	64.2 ± 1.8	64.1 ± 0.5
45	54.1 ± 1.5	53.4 ± 3.8	62.7 ± 0.3	47.3 ± 0.0	53.3 ± 3.3	$\mathbf{70.8 \pm 1.7}$	64.6 ± 1.8	64.3 ± 0.5
50	52.7 ± 1.7	51.7 ± 4.3	62.0 ± 0.3	47.1 ± 0.0	53.1 ± 3.2	$\mathbf{69.9 \pm 1.9}$	63.9 ± 1.9	63.9 ± 0.8
Average	57.75	57.80	64.76	48.14	55.97	**71.47**	65.08	63.79

Table A.17: AUC's (mean±std%) of different approaches for FROG class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	74.1 ± 0.4	59.5 ± 3.8	77.5 ± 0.2	58.0 ± 0.0	57.7 ± 5.4	58.4 ± 4.1	$\mathbf{78.4 \pm 1.9}$	74.6 ± 1.6
10	73.4 ± 0.4	58.4 ± 3.9	$\mathbf{77.6 \pm 0.2}$	57.6 ± 0.0	55.9 ± 5.7	56.2 ± 3.9	77.5 ± 2.0	74.5 ± 1.6
15	72.7 ± 0.5	57.6 ± 3.9	$\mathbf{77.5 \pm 0.2}$	57.3 ± 0.0	54.3 ± 5.7	54.1 ± 2.1	77.1 ± 2.1	74.2 ± 1.6
20	72.3 ± 0.5	55.7 ± 4.2	$\mathbf{77.4 \pm 0.2}$	57.0 ± 0.0	52.8 ± 5.9	52.2 ± 1.8	77.3 ± 2.0	73.8 ± 1.6
25	71.4 ± 0.5	55.2 ± 4.1	$\mathbf{77.5 \pm 0.2}$	56.7 ± 0.0	51.7 ± 6.1	51.5 ± 1.5	77.2 ± 2.0	73.3 ± 1.7
30	71.0 ± 0.6	54.9 ± 4.3	$\mathbf{77.5 \pm 0.2}$	56.4 ± 0.0	50.9 ± 6.2	50.8 ± 0.8	76.2 ± 2.2	72.4 ± 1.7
35	68.9 ± 0.9	52.7 ± 4.2	$\mathbf{77.1 \pm 0.2}$	56.2 ± 0.0	51.6 ± 6.2	50.4 ± 0.5	75.7 ± 2.3	71.5 ± 1.7
40	67.2 ± 1.0	51.2 ± 4.5	$\mathbf{77.4 \pm 0.3}$	56.0 ± 0.0	50.7 ± 5.9	50.3 ± 0.3	75.1 ± 2.5	71.8 ± 1.7
45	66.2 ± 1.1	53.3 ± 4.1	$\mathbf{77.3 \pm 0.3}$	55.7 ± 0.0	51.1 ± 5.8	50.2 ± 0.3	74.2 ± 2.5	72.3 ± 1.6
50	65.6 ± 1.3	55.1 ± 3.7	$\mathbf{77.2 \pm 0.3}$	55.6 ± 0.0	50.0 ± 5.9	50.0 ± 0.0	73.7 ± 2.7	72.9 ± 1.6
Average	70.28	55.36	**77.40**	56.65	52.67	52.41	76.24	73.13

Table A.18: AUC's (mean±std%) of different approaches for HORSE class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	61.4 ± 0.5	62.7 ± 2.6	64.1 ± 0.2	47.6 ± 0.0	57.6 ± 2.9	61.1 ± 1.1	$\mathbf{67.4 \pm 3.0}$	65.8 ± 1.8
10	61.0 ± 0.5	61.4 ± 2.8	63.8 ± 0.2	47.4 ± 0.0	55.8 ± 2.9	61.4 ± 1.2	$\mathbf{67.1 \pm 3.2}$	65.1 ± 1.8
15	60.2 ± 0.5	60.7 ± 3.1	63.7 ± 0.2	47.2 ± 0.0	54.1 ± 3.1	61.8 ± 1.1	$\mathbf{66.6 \pm 3.2}$	64.9 ± 1.9
20	59.7 ± 0.7	58.9 ± 3.2	63.9 ± 0.2	47.1 ± 0.0	53.4 ± 3.1	62.3 ± 1.3	$\mathbf{66.1 \pm 3.1}$	64.4 ± 1.9
25	59.0 ± 0.7	58.3 ± 3.5	63.4 ± 0.2	47.0 ± 0.0	52.2 ± 3.3	62.5 ± 1.2	$\mathbf{65.7 \pm 3.2}$	64.2 ± 1.9
30	57.8 ± 0.9	58.1 ± 3.5	63.6 ± 0.2	46.9 ± 0.0	51.3 ± 3.4	62.5 ± 1.5	$\mathbf{64.9 \pm 3.1}$	64.1 ± 1.9
35	56.7 ± 1.0	55.8 ± 3.9	63.4 ± 0.2	46.9 ± 0.0	50.6 ± 3.7	62.7 ± 1.4	$\mathbf{64.4 \pm 3.3}$	63.9 ± 2.0
40	55.1 ± 1.1	54.3 ± 4.1	63.7 ± 0.3	46.8 ± 0.0	49.8 ± 4.0	62.4 ± 1.2	$\mathbf{64.6 \pm 3.3}$	64.0 ± 2.0
45	54.2 ± 1.2	53.7 ± 4.2	63.6 ± 0.3	46.7 ± 0.0	48.4 ± 4.2	62.3 ± 1.4	$\mathbf{64.3 \pm 3.2}$	63.8 ± 2.0
50	53.2 ± 1.4	52.9 ± 4.4	63.8 ± 0.3	46.4 ± 0.0	49.0 ± 4.0	62.5 ± 1.3	$\mathbf{64.4 \pm 3.1}$	63.4 ± 2.0
Average	57.81	57.68	63.70	47.00	52.22	62.12	**65.55**	64.36

Table A.19: AUC's (mean±std%) of different approaches for SHIP class for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	73.2 ± 0.4	74.1 ± 1.2	$\mathbf{81.7 \pm 0.2}$	70.5 ± 0.0	75.8 ± 1.5	76.7 ± 1.3	79.9 ± 1.7	78.7 ± 1.8
10	72.9 ± 0.4	73.8 ± 1.2	$\mathbf{81.6 \pm 0.2}$	70.0 ± 0.0	74.2 ± 1.6	76.3 ± 1.5	79.1 ± 1.8	78.1 ± 1.9
15	72.1 ± 0.4	70.7 ± 1.4	$\mathbf{81.6 \pm 0.2}$	69.6 ± 0.0	73.3 ± 1.8	75.2 ± 1.2	78.9 ± 1.8	77.9 ± 1.9
20	70.4 ± 0.5	68.7 ± 1.7	$\mathbf{81.9 \pm 0.2}$	69.2 ± 0.0	72.5 ± 1.9	74.1 ± 1.3	77.6 ± 1.9	78.0 ± 1.9
25	68.9 ± 0.5	69.1 ± 1.5	$\mathbf{81.1 \pm 0.3}$	68.8 ± 0.0	71.1 ± 1.9	73.8 ± 1.1	77.4 ± 1.9	77.8 ± 1.9
30	67.4 ± 0.8	67.4 ± 1.9	$\mathbf{80.9 \pm 0.3}$	68.5 ± 0.0	70.6 ± 2.0	73.6 ± 1.2	76.9 ± 2.0	77.7 ± 1.8
35	66.4 ± 0.9	64.9 ± 2.4	$\mathbf{79.6 \pm 0.7}$	67.9 ± 0.0	68.8 ± 2.3	73.3 ± 1.3	76.4 ± 2.1	77.5 ± 1.9
40	65.1 ± 1.0	62.4 ± 2.9	$\mathbf{78.4 \pm 1.0}$	67.7 ± 0.0	65.3 ± 2.7	73.8 ± 1.2	76.6 ± 2.1	77.0 ± 2.0
45	64.4 ± 1.1	60.4 ± 3.1	74.8 ± 1.8	67.2 ± 0.0	64.4 ± 3.0	73.5 ± 1.4	76.1 ± 2.0	$\mathbf{76.8 \pm 2.0}$
50	64.0 ± 1.1	59.7 ± 3.2	72.1 ± 2.6	67.1 ± 0.0	62.4 ± 3.1	73.1 ± 1.5	75.8 ± 2.1	$\mathbf{76.3 \pm 2.0}$
Average	68.48	67.12	**79.37**	68.64	69.84	74.34	77.47	77.58

Table A.20: AUC's (mean±std%) of different approaches for TRUCK for one-class classification on the CIFAR10 dataset in the presence of various levels of contamination.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	G-DRKSR	C-DRKSR
5	75.2 ± 0.4	68.4 ± 1.9	77.1 ± 0.9	53.6 ± 0.0	66.8 ± 3.0	59.8 ± 1.1	75.1 ± 0.9	$\mathbf{77.8 \pm 1.1}$
10	73.9 ± 0.5	67.1 ± 2.0	76.7 ± 0.9	52.8 ± 0.0	65.7 ± 3.1	58.7 ± 1.3	74.7 ± 0.9	$\mathbf{77.3 \pm 1.1}$
15	72.7 ± 0.5	66.5 ± 2.1	76.4 ± 0.9	52.1 ± 0.0	65.3 ± 3.1	58.1 ± 1.5	74.4 ± 0.9	$\mathbf{77.1 \pm 1.1}$
20	71.8 ± 0.5	66.2 ± 2.1	75.6 ± 0.9	51.7 ± 0.0	64.5 ± 3.3	57.6 ± 1.6	74.1 ± 1.0	$\mathbf{76.8 \pm 1.3}$
25	70.7 ± 0.7	66.4 ± 2.2	74.6 ± 1.0	51.0 ± 0.0	63.8 ± 3.5	56.5 ± 1.5	73.6 ± 1.0	$\mathbf{76.5 \pm 1.2}$
30	69.1 ± 0.9	65.8 ± 2.1	74.4 ± 1.0	50.6 ± 0.0	63.3 ± 3.4	55.8 ± 1.7	73.1 ± 1.1	$\mathbf{76.1 \pm 1.2}$
35	68.3 ± 1.0	64.1 ± 2.4	74.5 ± 1.0	50.0 ± 0.0	62.7 ± 3.7	55.1 ± 1.6	72.5 ± 1.3	$\mathbf{76.5 \pm 1.3}$
40	67.0 ± 1.1	62.7 ± 2.8	74.4 ± 1.0	49.6 ± 0.0	61.8 ± 4.0	54.7 ± 1.8	72.1 ± 1.4	$\mathbf{76.2 \pm 1.2}$
45	66.4 ± 1.2	62.2 ± 2.9	74.3 ± 1.0	49.3 ± 0.0	61.1 ± 4.1	54.1 ± 1.7	71.7 ± 1.7	$\mathbf{75.5 \pm 1.4}$
50	65.2 ± 1.1	62.6 ± 2.9	74.5 ± 1.1	49.0 ± 0.0	61.5 ± 4.0	53.3 ± 1.9	71.1 ± 1.8	$\mathbf{74.8 \pm 1.6}$
Average	70.03	65.20	75.25	50.97	63.65	56.37	73.24	**76.46**

A.3 Unsupervised Observation Ranking Results for each class in MNIST dataset

In this section, the results of One-class classification for unsupervised observation ranking are presented for each class of MNIST dataset.

Table A.21: AUC's (mean±std%) of different approaches for Digit 0 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	92.8 ± 0.0	96.4 ± 1.5	97.1 ± 0.3	96.7 ± 0.0	94.0 ± 1.3	95.2 ± 1.0	85.7 ± 0.0	98.4 ± 0.4	$\mathbf{99.1 \pm 0.2}$
10	90.4 ± 0.0	95.1 ± 1.6	96.5 ± 0.3	96.4 ± 0.0	93.4 ± 1.5	93.8 ± 1.3	84.6 ± 0.0	98.1 ± 0.3	$\mathbf{98.6 \pm 0.2}$
15	87.1 ± 0.1	93.0 ± 2.1	95.6 ± 0.4	96.1 ± 0.0	90.7 ± 1.5	92.7 ± 1.4	83.5 ± 0.0	$\mathbf{98.0 \pm 0.4}$	96.8 ± 0.4
20	83.6 ± 0.1	91.3 ± 2.3	94.2 ± 0.4	95.7 ± 0.0	89.1 ± 1.9	91.8 ± 1.3	82.4 ± 0.0	$\mathbf{97.9 \pm 0.4}$	94.1 ± 0.5
25	81.4 ± 0.2	88.9 ± 2.7	91.7 ± 0.4	95.5 ± 0.0	87.3 ± 2.3	89.8 ± 1.9	82.0 ± 0.0	$\mathbf{97.8 \pm 0.4}$	93.0 ± 0.5
30	79.1 ± 0.2	86.8 ± 3.3	89.9 ± 0.4	95.3 ± 0.0	86.8 ± 2.9	88.7 ± 2.1	81.7 ± 0.0	$\mathbf{98.0 \pm 0.3}$	91.3 ± 0.7
35	76.5 ± 0.2	82.1 ± 3.9	88.3 ± 0.4	95.0 ± 0.0	83.4 ± 3.7	89.3 ± 1.7	81.4 ± 0.0	$\mathbf{97.8 \pm 0.3}$	89.9 ± 0.8
40	75.1 ± 0.3	79.3 ± 4.1	86.2 ± 0.5	94.8 ± 0.0	77.9 ± 3.9	90.5 ± 1.5	81.1 ± 0.0	$\mathbf{97.9 \pm 0.3}$	87.2 ± 1.0
45	73.2 ± 0.3	76.4 ± 4.4	84.4 ± 0.7	94.4 ± 0.0	74.9 ± 4.2	88.2 ± 1.5	80.6 ± 0.0	$\mathbf{98.1 \pm 0.3}$	84.7 ± 1.0
50	71.4 ± 0.4	73.5 ± 4.8	82.9 ± 0.8	94.1 ± 0.0	72.1 ± 4.7	86.1 ± 1.6	80.1 ± 0.0	$\mathbf{98.0 \pm 0.3}$	82.3 ± 1.2
Average	81.06	86.28	90.68	95.40	84.96	90.61	82.31	**98.00**	91.70

Table A.22: AUC's (mean±std%) of different approaches for Digit 1 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	99.1 ± 0.0	99.0 ± 0.1	$\mathbf{99.7 \pm 0.0}$	97.2 ± 0.0	96.9 ± 0.6	$\mathbf{99.7 \pm 0.1}$	95.2 ± 0.0	99.3 ± 0.0	99.3 ± 0.0
10	99.1 ± 0.0	98.9 ± 0.2	$\mathbf{99.7 \pm 0.0}$	96.9 ± 0.0	95.8 ± 0.8	$\mathbf{99.7 \pm 0.2}$	95.0 ± 0.0	99.3 ± 0.0	99.3 ± 0.0
15	98.2 ± 0.0	97.8 ± 0.7	$\mathbf{99.7 \pm 0.0}$	96.9 ± 0.0	94.6 ± 0.8	$\mathbf{99.7 \pm 0.2}$	94.8 ± 0.0	99.3 ± 0.1	99.3 ± 0.0
20	97.3 ± 0.0	97.4 ± 0.9	$\mathbf{99.6 \pm 0.0}$	96.8 ± 0.0	93.9 ± 0.9	$\mathbf{99.6 \pm 0.2}$	94.6 ± 0.0	99.4 ± 0.1	99.4 ± 0.0
25	96.5 ± 0.0	96.7 ± 0.9	$\mathbf{99.5 \pm 0.0}$	96.5 ± 0.0	93.1 ± 1.1	$\mathbf{99.5 \pm 0.2}$	94.5 ± 0.0	99.4 ± 0.1	99.4 ± 0.0
30	95.8 ± 0.0	96.1 ± 0.9	99.3 ± 0.0	96.3 ± 0.0	91.5 ± 1.2	$\mathbf{99.5 \pm 0.2}$	94.5 ± 0.0	99.2 ± 0.1	99.2 ± 0.1
35	95.3 ± 0.0	94.3 ± 1.4	99.3 ± 0.0	96.4 ± 0.0	89.7 ± 1.4	$\mathbf{99.5 \pm 0.2}$	94.2 ± 0.0	99.2 ± 0.1	99.1 ± 0.1
40	94.8 ± 0.1	92.8 ± 1.7	99.1 ± 0.0	96.4 ± 0.0	86.9 ± 1.5	$\mathbf{99.4 \pm 0.2}$	94.0 ± 0.0	99.1 ± 0.1	99.1 ± 0.1
45	94.3 ± 0.1	91.6 ± 1.9	98.9 ± 0.1	96.2 ± 0.0	84.7 ± 1.9	$\mathbf{99.4 \pm 0.1}$	93.9 ± 0.0	99.1 ± 0.1	99.1 ± 0.1
50	93.9 ± 0.1	90.3 ± 2.3	98.7 ± 0.1	96.0 ± 0.0	82.1 ± 2.4	$\mathbf{99.2 \pm 0.2}$	93.7 ± 0.0	99.1 ± 0.1	99.1 ± 0.1
Average	96.43	95.49	99.35	96.56	90.92	**99.52**	94.44	99.24	99.23

Table A.23: AUC's (mean±std%) of different approaches for Digit 2 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	79.4 ± 0.1	82.3 ± 3.0	90.8 ± 0.6	87.1 ± 0.0	81.9 ± 2.2	89.1 ± 1.8	62.6 ± 0.0	$\mathbf{92.8 \pm 0.6}$	91.4 ± 0.9
10	77.4 ± 0.1	80.6 ± 3.2	89.8 ± 0.6	86.8 ± 0.0	81.3 ± 2.9	88.4 ± 1.9	62.3 ± 0.0	$\mathbf{92.4 \pm 0.7}$	90.2 ± 1.0
15	74.8 ± 0.1	79.7 ± 3.4	87.7 ± 0.6	86.5 ± 0.0	80.5 ± 3.0	85.3 ± 1.7	61.5 ± 0.0	$\mathbf{92.1 \pm 0.7}$	88.9 ± 1.0
20	71.1 ± 0.2	74.2 ± 3.9	85.2 ± 0.6	86.2 ± 0.0	77.8 ± 3.3	83.9 ± 1.6	59.4 ± 0.0	$\mathbf{91.9 \pm 0.8}$	87.0 ± 1.3
25	67.8 ± 0.2	73.1 ± 4.2	83.9 ± 0.6	85.7 ± 0.0	75.8 ± 3.5	82.2 ± 1.5	57.8 ± 0.0	$\mathbf{91.6 \pm 0.7}$	85.7 ± 1.4
30	63.9 ± 0.3	71.9 ± 4.4	81.2 ± 0.7	84.8 ± 0.0	74.7 ± 3.7	79.5 ± 1.2	54.7 ± 0.0	$\mathbf{91.8 \pm 0.8}$	84.0 ± 1.6
35	61.8 ± 0.3	69.8 ± 4.7	80.6 ± 0.7	84.0 ± 0.0	73.5 ± 3.8	79.7 ± 1.9	53.5 ± 0.0	$\mathbf{91.0 \pm 0.9}$	82.1 ± 1.9
40	60.1 ± 0.3	67.9 ± 5.0	78.1 ± 0.9	83.3 ± 0.0	72.3 ± 4.1	78.9 ± 2.0	52.6 ± 0.0	$\mathbf{90.6 \pm 1.0}$	80.3 ± 1.9
45	58.2 ± 0.4	66.2 ± 5.4	76.5 ± 1.0	82.6 ± 0.0	71.8 ± 4.4	78.5 ± 2.4	51.4 ± 0.0	$\mathbf{90.7 \pm 0.9}$	78.5 ± 2.1
50	55.9 ± 0.5	64.9 ± 5.7	73.3 ± 1.1	81.5 ± 0.0	69.8 ± 4.9	78.1 ± 2.7	49.9 ± 0.0	$\mathbf{90.1 \pm 0.9}$	77.9 ± 2.2
Average	67.04	73.06	82.71	84.88	75.94	82.36	56.57	**91.50**	84.30

Table A.24: AUC's (mean±std%) of different approaches for Digit 3 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	86.1 ± 0.0	85.7 ± 2.8	92.2 ± 0.5	89.4 ± 0.0	81.8 ± 1.8	$\mathbf{94.8 \pm 2.6}$	67.4 ± 0.0	90.6 ± 0.7	90.3 ± 0.8
10	83.1 ± 0.0	84.4 ± 3.2	91.6 ± 0.6	89.2 ± 0.0	81.1 ± 1.9	$\mathbf{94.1 \pm 2.5}$	65.5 ± 0.0	90.4 ± 0.8	89.1 ± 0.8
15	81.9 ± 0.0	82.9 ± 3.4	90.8 ± 0.6	89.2 ± 0.0	80.2 ± 2.3	$\mathbf{92.0 \pm 2.6}$	63.7 ± 0.0	88.9 ± 1.0	87.9 ± 0.9
20	80.9 ± 0.1	80.9 ± 3.6	$\mathbf{89.7 \pm 0.6}$	89.1 ± 0.0	78.7 ± 2.5	89.1 ± 2.3	60.3 ± 0.0	88.4 ± 1.2	85.6 ± 1.1
25	77.6 ± 0.2	77.7 ± 3.9	$\mathbf{88.8 \pm 0.6}$	88.5 ± 0.0	75.4 ± 2.7	87.2 ± 2.4	57.8 ± 0.0	88.7 ± 1.1	84.3 ± 1.2
30	75.2 ± 0.2	76.9 ± 4.3	87.9 ± 0.6	88.0 ± 0.0	73.3 ± 3.1	83.9 ± 2.5	55.4 ± 0.0	$\mathbf{88.3 \pm 1.1}$	82.5 ± 1.4
35	73.4 ± 0.2	74.5 ± 4.5	85.7 ± 0.6	87.6 ± 0.0	72.4 ± 3.4	85.4 ± 2.9	54.7 ± 0.0	$\mathbf{88.1 \pm 1.2}$	81.0 ± 1.5
40	71.6 ± 0.2	73.1 ± 4.6	83.3 ± 0.7	87.3 ± 0.0	71.4 ± 3.5	87.6 ± 3.0	53.8 ± 0.0	$\mathbf{87.4 \pm 1.1}$	79.9 ± 1.7
45	68.8 ± 0.2	71.8 ± 4.9	80.0 ± 0.9	$\mathbf{87.0 \pm 0.0}$	70.6 ± 3.8	$\mathbf{87.0 \pm 3.1}$	53.0 ± 0.0	$\mathbf{87.0 \pm 1.3}$	78.5 ± 1.8
50	66.8 ± 0.2	70.6 ± 5.3	78.9 ± 1.0	$\mathbf{86.8 \pm 0.0}$	69.7 ± 4.1	86.3 ± 3.8	52.4 ± 0.0	86.2 ± 1.5	77.9 ± 2.0
Average	76.54	77.85	83.89	88.21	75.46	**88.74**	58.40	88.40	83.67

Table A.25: AUC's (mean±std%) of different approaches for Digit 4 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	86.9 ± 0.0	90.1 ± 1.6	93.4 ± 0.2	88.2 ± 0.0	82.6 ± 2.3	86.9 ± 2.3	67.1 ± 0.0	$\mathbf{93.8 \pm 0.8}$	$\mathbf{93.8 \pm 0.9}$
10	85.8 ± 0.0	89.7 ± 1.7	91.3 ± 0.2	88.0 ± 0.0	81.9 ± 2.5	84.9 ± 2.8	66.0 ± 0.0	$\mathbf{93.6 \pm 0.9}$	93.1 ± 0.9
15	83.9 ± 0.0	87.4 ± 2.0	90.7 ± 0.2	86.9 ± 0.0	80.0 ± 2.7	84.4 ± 2.8	64.8 ± 0.0	$\mathbf{92.4 \pm 0.8}$	91.7 ± 1.0
20	82.1 ± 0.1	85.4 ± 2.3	89.6 ± 0.2	86.4 ± 0.0	78.4 ± 3.1	84.3 ± 2.9	63.7 ± 0.0	$\mathbf{91.9 \pm 0.9}$	89.6 ± 1.2
25	81.3 ± 0.1	83.6 ± 2.6	88.1 ± 0.2	85.8 ± 0.0	77.1 ± 3.2	84.0 ± 2.9	61.8 ± 0.0	$\mathbf{91.0 \pm 0.9}$	87.9 ± 1.3
30	80.1 ± 0.2	81.5 ± 3.0	86.4 ± 0.3	85.0 ± 0.0	75.8 ± 3.2	83.4 ± 2.6	59.6 ± 0.0	$\mathbf{90.6 \pm 0.9}$	86.2 ± 1.3
35	78.6 ± 0.2	80.4 ± 3.4	84.3 ± 0.3	84.5 ± 0.0	74.7 ± 3.4	82.8 ± 3.1	59.1 ± 0.0	$\mathbf{89.7 \pm 0.9}$	84.8 ± 1.5
40	78.1 ± 0.2	79.7 ± 3.9	82.9 ± 0.3	83.9 ± 0.0	73.9 ± 3.7	82.7 ± 3.9	58.7 ± 0.0	$\mathbf{88.4 \pm 1.0}$	83.1 ± 1.6
45	76.0 ± 0.2	78.8 ± 4.2	81.7 ± 0.3	83.1 ± 0.0	71.6 ± 3.9	81.9 ± 4.3	57.5 ± 0.0	$\mathbf{87.6 \pm 1.0}$	82.3 ± 1.8
50	74.8 ± 0.3	77.9 ± 4.3	79.2 ± 0.3	82.5 ± 0.0	70.1 ± 4.1	81.1 ± 4.8	56.9 ± 0.0	$\mathbf{86.3 \pm 1.0}$	81.1 ± 1.8
Average	80.76	83.45	86.76	85.43	76.61	83.64	61.52	**90.53**	87.36

Table A.26: AUC's (mean±std%) of different approaches for Digit 5 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	79.4 ± 0.0	83.1 ± 2.1	93.8 ± 0.5	73.5 ± 0.0	73.3 ± 3.1	86.9 ± 2.6	66.0 ± 0.0	94.6 ± 0.5	$\mathbf{95.4 \pm 0.7}$
10	77.1 ± 0.0	82.4 ± 2.2	92.4 ± 0.6	72.8 ± 0.0	71.2 ± 3.4	85.7 ± 2.8	64.7 ± 0.0	94.2 ± 0.6	$\mathbf{95.1 \pm 0.8}$
15	74.9 ± 0.1	80.3 ± 2.8	91.0 ± 0.6	71.9 ± 0.0	69.4 ± 3.8	83.4 ± 3.3	62.6 ± 0.0	93.8 ± 0.5	$\mathbf{95.0 \pm 0.9}$
20	73.4 ± 0.2	77.5 ± 3.3	89.6 ± 0.5	71.1 ± 0.0	67.9 ± 4.2	81.9 ± 3.8	60.5 ± 0.0	92.7 ± 0.7	$\mathbf{94.4 \pm 0.9}$
25	71.2 ± 0.2	76.1 ± 3.8	85.7 ± 0.6	70.7 ± 0.0	66.2 ± 4.4	79.5 ± 3.0	57.2 ± 0.0	91.8 ± 0.7	$\mathbf{92.8 \pm 0.9}$
30	69.8 ± 0.2	74.9 ± 4.1	83.4 ± 0.5	69.5 ± 0.0	64.2 ± 4.7	75.2 ± 3.2	53.9 ± 0.0	90.6 ± 0.7	$\mathbf{91.9 \pm 0.9}$
35	66.8 ± 0.2	73.3 ± 4.4	81.9 ± 0.8	69.1 ± 0.0	63.0 ± 4.9	76.9 ± 2.8	52.5 ± 0.0	89.4 ± 0.8	$\mathbf{89.7 \pm 0.9}$
40	65.1 ± 0.3	71.7 ± 4.8	80.1 ± 0.9	68.7 ± 0.0	62.2 ± 5.2	78.9 ± 2.6	51.8 ± 0.0	$\mathbf{88.1 \pm 1.0}$	87.2 ± 1.0
45	63.3 ± 0.4	70.9 ± 5.3	77.6 ± 1.1	67.6 ± 0.0	60.6 ± 5.6	78.5 ± 2.7	51.3 ± 0.0	$\mathbf{87.2 \pm 0.9}$	85.4 ± 1.0
50	61.1 ± 0.4	69.7 ± 5.6	73.1 ± 1.3	66.7 ± 0.0	59.8 ± 5.8	77.6 ± 2.9	50.7 ± 0.0	$\mathbf{86.0 \pm 1.0}$	82.0 ± 1.0
Average	70.21	75.99	84.86	70.16	65.78	80.45	57.12	90.84	**90.89**

Table A.27: AUC's (mean±std%) of different approaches for Digit 6 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	89.9 ± 0.0	91.3 ± 2.3	$\mathbf{97.9 \pm 0.1}$	93.4 ± 0.0	87.1 ± 2.1	95.3 ± 1.3	78.1 ± 0.0	96.1 ± 0.4	95.8 ± 0.7
10	87.3 ± 0.0	91.5 ± 2.6	$\mathbf{97.4 \pm 0.1}$	93.2 ± 0.0	85.9 ± 2.2	94.9 ± 1.5	76.8 ± 0.0	95.7 ± 0.5	94.9 ± 0.7
15	85.4 ± 0.0	90.7 ± 2.8	$\mathbf{95.1 \pm 0.2}$	92.6 ± 0.0	83.4 ± 2.4	93.4 ± 2.4	75.5 ± 0.0	94.5 ± 0.6	93.4 ± 0.8
20	82.7 ± 0.0	90.1 ± 3.1	93.3 ± 0.3	91.9 ± 0.0	82.3 ± 2.7	91.8 ± 3.1	74.6 ± 0.0	$\mathbf{93.6 \pm 0.6}$	91.7 ± 0.9
25	80.9 ± 0.1	87.5 ± 3.3	92.8 ± 0.2	91.5 ± 0.0	81.0 ± 3.0	90.2 ± 2.8	73.4 ± 0.0	$\mathbf{92.9 \pm 0.6}$	90.9 ± 1.0
30	77.9 ± 0.2	85.7 ± 3.8	91.7 ± 0.3	91.3 ± 0.0	79.8 ± 3.2	88.7 ± 2.6	72.7 ± 0.0	$\mathbf{92.4 \pm 0.7}$	89.8 ± 1.0
35	76.1 ± 0.2	83.4 ± 4.5	88.1 ± 0.3	92.1 ± 0.0	77.7 ± 3.5	86.9 ± 3.2	72.5 ± 0.0	$\mathbf{92.0 \pm 0.6}$	87.7 ± 1.1
40	73.8 ± 0.2	80.4 ± 4.9	85.9 ± 0.3	90.0 ± 0.0	76.8 ± 3.9	85.8 ± 4.0	72.5 ± 0.0	$\mathbf{91.6 \pm 0.7}$	86.2 ± 1.3
45	72.1 ± 0.2	78.3 ± 5.3	83.6 ± 0.4	89.7 ± 0.0	74.3 ± 4.1	86.2 ± 3.8	71.9 ± 0.0	$\mathbf{90.9 \pm 0.7}$	84.9 ± 1.4
50	69.8 ± 0.2	76.8 ± 5.7	81.9 ± 0.4	89.3 ± 0.0	72.6 ± 4.2	86.9 ± 3.9	71.4 ± 0.0	$\mathbf{90.3 \pm 0.8}$	83.0 ± 1.5
Average	79.59	85.57	90.77	91.50	80.09	90.01	73.94	**93.00**	89.83

Table A.28: AUC's (mean±std%) of different approaches for Digit 7 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	90.4 ± 0.0	90.4 ± 2.1	$\mathbf{95.7 \pm 0.1}$	90.2 ± 0.0	77.0 ± 3.4	93.8 ± 0.6	75.1 ± 0.0	93.7 ± 0.6	92.9 ± 0.8
10	88.7 ± 0.0	89.7 ± 2.3	$\mathbf{94.8 \pm 0.2}$	89.9 ± 0.0	76.1 ± 3.8	93.1 ± 0.6	74.8 ± 0.0	93.1 ± 0.7	92.1 ± 0.8
15	85.8 ± 0.0	86.6 ± 3.0	$\mathbf{93.7 \pm 0.2}$	89.8 ± 0.0	74.8 ± 3.9	92.7 ± 0.7	74.5 ± 0.0	92.2 ± 0.7	90.8 ± 1.0
20	83.9 ± 0.0	83.8 ± 3.7	92.1 ± 0.2	89.7 ± 0.0	73.9 ± 4.1	$\mathbf{92.2 \pm 0.9}$	74.2 ± 0.0	91.2 ± 0.6	89.6 ± 1.1
25	82.7 ± 0.0	82.9 ± 3.9	90.9 ± 0.2	88.7 ± 0.0	73.1 ± 4.3	$\mathbf{91.8 \pm 1.0}$	73.1 ± 0.0	90.4 ± 0.5	89.0 ± 1.1
30	80.4 ± 0.0	81.9 ± 4.1	89.3 ± 0.2	88.0 ± 0.0	71.4 ± 4.2	89.2 ± 1.1	72.0 ± 0.0	$\mathbf{89.7 \pm 0.6}$	87.5 ± 1.3
35	78.8 ± 0.0	82.2 ± 4.0	88.1 ± 0.4	87.5 ± 0.0	69.3 ± 4.7	87.9 ± 1.6	71.5 ± 0.0	$\mathbf{90.0 \pm 0.7}$	86.3 ± 1.4
40	77.1 ± 0.1	82.9 ± 3.8	86.3 ± 0.4	87.2 ± 0.0	68.7 ± 5.1	85.3 ± 2.0	71.0 ± 0.0	$\mathbf{90.1 \pm 0.8}$	84.8 ± 1.5
45	75.7 ± 0.1	81.4 ± 4.1	85.4 ± 0.4	86.9 ± 0.0	66.2 ± 5.2	83.7 ± 2.4	70.6 ± 0.0	$\mathbf{90.3 \pm 0.8}$	84.1 ± 1.5
50	73.3 ± 0.1	80.6 ± 4.3	83.6 ± 0.5	86.6 ± 0.0	65.9 ± 5.5	82.4 ± 2.9	70.1 ± 0.0	$\mathbf{89.7 \pm 0.7}$	82.7 ± 1.8
Average	81.68	84.24	89.99	88.45	71.64	89.21	72.69	**91.04**	87.98

Table A.29: AUC's (mean±std%) of different approaches for Digit 8 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	82.1 ± 0.2	91.9 ± 2.1	$\mathbf{93.1 \pm 0.2}$	89.0 ± 0.0	85.5 ± 1.8	77.2 ± 0.5	67.8 ± 0.0	92.6 ± 0.7	92.1 ± 1.0
10	79.4 ± 0.2	91.4 ± 2.3	$\mathbf{91.9 \pm 0.2}$	88.7 ± 0.0	83.5 ± 1.9	76.9 ± 0.8	65.6 ± 0.0	$\mathbf{91.9 \pm 0.8}$	91.0 ± 1.1
15	77.3 ± 0.2	90.8 ± 2.5	90.9 ± 0.2	88.8 ± 0.0	82.4 ± 1.9	74.8 ± 0.9	63.4 ± 0.0	$\mathbf{91.1 \pm 0.7}$	90.1 ± 1.2
20	74.9 ± 0.3	90.1 ± 2.5	90.1 ± 0.2	89.0 ± 0.0	81.6 ± 2.0	72.9 ± 1.1	61.1 ± 0.0	$\mathbf{90.6 \pm 0.8}$	89.8 ± 1.2
25	74.0 ± 0.3	89.5 ± 2.7	89.6 ± 0.2	88.2 ± 0.0	80.9 ± 2.2	71.2 ± 0.9	59.4 ± 0.0	$\mathbf{89.7 \pm 0.7}$	89.1 ± 1.2
30	71.9 ± 0.4	88.7 ± 3.2	88.7 ± 0.2	87.8 ± 0.0	80.4 ± 2.3	69.9 ± 0.8	57.0 ± 0.0	$\mathbf{89.9 \pm 0.8}$	88.6 ± 1.3
35	71.1 ± 0.4	86.6 ± 3.6	86.8 ± 0.3	87.9 ± 0.0	79.0 ± 2.2	68.6 ± 0.9	55.4 ± 0.0	$\mathbf{90.4 \pm 1.0}$	87.8 ± 1.4
40	70.1 ± 0.5	85.3 ± 3.8	85.1 ± 0.4	88.0 ± 0.0	78.6 ± 2.5	67.8 ± 0.8	53.1 ± 0.0	$\mathbf{90.9 \pm 0.9}$	86.0 ± 1.6
45	67.9 ± 0.5	84.4 ± 3.9	84.7 ± 0.3	87.5 ± 0.0	75.6 ± 2.7	65.9 ± 1.5	51.7 ± 0.0	$\mathbf{90.4 \pm 1.0}$	85.5 ± 1.6
50	65.8 ± 0.5	82.9 ± 4.2	84.2 ± 0.4	87.2 ± 0.0	74.4 ± 2.9	63.7 ± 2.0	49.4 ± 0.0	$\mathbf{90.1 \pm 1.0}$	84.3 ± 1.8
Average	73.45	88.16	88.51	88.21	80.19	70.89	58.39	**90.76**	88.43

Table A.30: AUC's (mean±std%) of different approaches for Digit 9 for one-class classification on the MNIST dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	89.6 ± 0.0	92.3 ± 1.4	**95.9 ± 0.2**	88.4 ± 0.0	83.3 ± 2.5	86.7 ± 0.9	73.8 ± 0.0	95.8 ± 0.3	91.8 ± 0.9
10	88.7 ± 0.1	90.4 ± 1.9	**95.0 ± 0.3**	88.1 ± 0.0	82.9 ± 2.7	85.6 ± 1.0	72.7 ± 0.0	94.6 ± 0.4	90.7 ± 0.9
15	86.5 ± 0.1	87.6 ± 2.6	93.6 ± 0.3	87.6 ± 0.0	81.1 ± 3.0	84.0 ± 1.2	71.6 ± 0.0	**94.1 ± 0.5**	90.1 ± 1.0
20	85.4 ± 0.1	85.6 ± 2.8	92.8 ± 0.3	87.3 ± 0.0	80.7 ± 3.1	82.4 ± 1.9	70.5 ± 0.0	**93.8 ± 0.4**	89.2 ± 1.0
25	83.6 ± 0.1	83.2 ± 3.2	92.0 ± 0.3	86.9 ± 0.0	78.5 ± 3.1	81.6 ± 2.1	69.1 ± 0.0	**93.4 ± 0.5**	87.9 ± 1.3
30	81.8 ± 0.3	81.8 ± 3.5	91.1 ± 0.3	86.6 ± 0.0	77.6 ± 3.4	79.8 ± 2.8	67.6 ± 0.0	**92.7 ± 0.7**	87.0 ± 1.3
35	80.6 ± 0.3	81.0 ± 3.7	90.5 ± 0.3	86.4 ± 0.0	75.0 ± 3.7	76.8 ± 2.8	66.6 ± 0.0	**92.3 ± 0.6**	85.8 ± 1.5
40	79.3 ± 0.3	80.4 ± 3.8	89.4 ± 0.3	86.3 ± 0.0	73.4 ± 3.8	75.9 ± 3.0	65.7 ± 0.0	**91.5 ± 0.7**	84.9 ± 1.6
45	78.2 ± 0.3	79.1 ± 4.1	87.6 ± 0.4	85.7 ± 0.0	71.8 ± 4.1	73.8 ± 3.1	65.1 ± 0.0	**92.0 ± 0.6**	83.1 ± 1.8
50	77.1 ± 0.3	77.6 ± 4.5	85.4 ± 0.6	85.3 ± 0.0	70.3 ± 4.4	72.5 ± 3.5	64.5 ± 0.0	**92.6 ± 0.7**	81.7 ± 2.1
Average	83.08	83.90	91.33	86.86	77.46	79.91	68.72	**93.28**	87.22

A.4 Unsupervised Observation Ranking Results for each class in CIFAR10 dataset

In this section, the results of One-class classification for unsupervisesd observation ranking are presented for each class of CIFAR-10 dataset.

Table A.31: AUC's (mean±std%) of different approaches for AIRPLANE class for one-class classification on the CIFAR10 dataset for Observation Ranking

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	60.8 ± 1.0	60.9 ± 4.0	78.8 ± 0.2	73.5 ± 0.0	58.5 ± 5.2	73.3 ± 1.1	71.5 ± 0.0	79.3 ± 1.0	**87.4 ± 1.2**
10	59.7 ± 1.1	59.7 ± 4.0	78.7 ± 0.2	72.9 ± 0.0	57.9 ± 5.4	72.8 ± 1.1	71.1 ± 0.0	79.5 ± 1.1	**86.6 ± 1.3**
15	58.9 ± 1.2	58.0 ± 4.2	78.4 ± 0.2	72.0 ± 0.0	56.0 ± 5.5	72.3 ± 1.2	69.8 ± 0.0	79.1 ± 1.1	**86.2 ± 1.3**
20	58.1 ± 1.3	56.3 ± 4.4	77.9 ± 0.2	71.4 ± 0.0	54.7 ± 5.7	71.9 ± 1.4	69.6 ± 0.0	80.9 ± 1.0	**85.8 ± 1.3**
25	57.2 ± 1.3	54.7 ± 4.7	76.5 ± 0.2	70.9 ± 0.0	53.3 ± 5.6	72.0 ± 1.4	69.4 ± 0.0	79.2 ± 1.2	**85.4 ± 1.3**
30	55.9 ± 1.2	53.1 ± 4.9	75.3 ± 0.4	70.4 ± 0.0	52.1 ± 5.8	72.1 ± 1.4	69.2 ± 0.0	79.1 ± 1.1	**84.9 ± 1.4**
35	55.6 ± 1.3	51.7 ± 4.8	73.7 ± 0.6	70.6 ± 0.0	51.2 ± 6.0	71.9 ± 1.5	69.0 ± 0.0	78.7 ± 1.2	**84.3 ± 1.4**
40	55.1 ± 1.3	50.7 ± 4.8	72.8 ± 0.7	70.6 ± 0.0	50.0 ± 6.1	71.7 ± 1.8	68.8 ± 0.0	77.9 ± 1.1	**83.8 ± 1.4**
45	54.5 ± 1.4	48.6 ± 5.2	69.7 ± 2.0	70.6 ± 0.0	47.6 ± 6.6	71.5 ± 2.0	68.5 ± 0.0	77.2 ± 1.1	**83.4 ± 1.5**
50	54.2 ± 1.4	46.1 ± 5.5	65.8 ± 3.4	70.6 ± 0.0	45.9 ± 6.9	71.4 ± 2.0	68.3 ± 0.0	76.4 ± 1.1	**82.9 ± 1.5**
Average	57.00	53.98	74.77	71.35	52.72	72.09	69.52	78.81	**80.59**

Table A.32: AUC's (mean±std%) of different approaches for AUTOMOBILE class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	62.7 ± 0.8	63.7 ± 2.3	68.4 ± 1.0	38.4 ± 0.0	55.6 ± 3.0	66.5 ± 2.0	62.1 ± 0.0	82.4 ± 1.0	$\mathbf{85.9 \pm 1.0}$
10	62.2 ± 0.9	62.1 ± 2.5	67.4 ± 1.0	38.1 ± 0.0	54.3 ± 3.0	66.4 ± 2.1	61.6 ± 0.0	82.6 ± 1.0	$\mathbf{85.3 \pm 1.1}$
15	61.6 ± 0.9	60.3 ± 2.9	66.8 ± 1.2	38.5 ± 0.0	53.5 ± 3.3	66.4 ± 2.1	61.3 ± 0.0	82.1 ± 1.0	$\mathbf{85.0 \pm 1.1}$
20	60.7 ± 1.0	56.7 ± 3.4	66.1 ± 1.4	39.4 ± 0.0	52.6 ± 3.5	66.3 ± 2.2	61.1 ± 0.0	81.8 ± 1.1	$\mathbf{84.6 \pm 1.2}$
25	59.4 ± 1.0	55.6 ± 3.5	65.4 ± 1.4	38.4 ± 0.0	52.0 ± 3.4	66.4 ± 2.5	61.2 ± 0.0	81.6 ± 1.1	$\mathbf{83.7 \pm 1.3}$
30	58.1 ± 1.1	54.9 ± 3.7	64.9 ± 1.6	37.8 ± 0.0	51.8 ± 3.7	66.4 ± 2.5	61.3 ± 0.0	81.4 ± 1.2	$\mathbf{82.9 \pm 1.3}$
35	57.3 ± 1.3	54.0 ± 3.6	62.6 ± 1.7	37.6 ± 0.0	51.0 ± 3.7	66.2 ± 2.5	61.3 ± 0.0	81.0 ± 1.3	$\mathbf{82.5 \pm 1.3}$
40	56.2 ± 1.3	53.2 ± 3.8	61.7 ± 1.9	37.5 ± 0.0	50.3 ± 4.0	66.1 ± 2.5	61.4 ± 0.0	80.9 ± 1.3	$\mathbf{82.1 \pm 1.4}$
45	55.3 ± 1.4	54.3 ± 3.7	61.0 ± 2.1	37.2 ± 0.0	49.6 ± 4.0	65.9 ± 2.4	61.3 ± 0.0	80.4 ± 1.5	$\mathbf{81.8 \pm 1.4}$
50	54.2 ± 1.5	55.7 ± 3.5	60.5 ± 2.2	36.9 ± 0.0	48.9 ± 4.2	65.9 ± 2.5	61.3 ± 0.0	80.1 ± 1.5	$\mathbf{81.4 \pm 1.4}$
Average	58.77	57.05	64.48	37.98	51.96	66.25	61.30	81.43	**83.52**

Table A.33: AUC's (mean±std%) of different approaches for BIRD class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	50.0 ± 0.5	50.5 ± 0.9	61.7 ± 0.3	63.8 ± 0.0	49.6 ± 2.3	63.9 ± 0.7	61.4 ± 0.0	81.6 ± 1.6	$\mathbf{82.1 \pm 1.3}$
10	49.9 ± 0.5	50.1 ± 0.9	61.4 ± 0.3	63.3 ± 0.0	48.4 ± 2.2	63.8 ± 0.7	60.8 ± 0.0	80.9 ± 1.8	$\mathbf{81.6 \pm 1.4}$
15	49.1 ± 0.6	48.7 ± 1.3	61.1 ± 0.3	63.6 ± 0.0	47.7 ± 2.5	63.4 ± 0.7	59.7 ± 0.0	$\mathbf{80.4 \pm 1.9}$	80.2 ± 1.5
20	48.0 ± 0.6	46.3 ± 1.8	60.9 ± 0.4	64.0 ± 0.0	47.1 ± 2.4	63.1 ± 0.8	58.5 ± 0.0	79.1 ± 2.0	$\mathbf{79.8 \pm 1.5}$
25	47.6 ± 0.6	45.8 ± 2.0	60.8 ± 0.4	63.8 ± 0.0	46.6 ± 2.6	63.0 ± 0.8	58.4 ± 0.0	$\mathbf{79.0 \pm 2.0}$	78.4 ± 1.7
30	47.2 ± 0.7	43.7 ± 2.4	60.8 ± 0.4	63.7 ± 0.0	46.1 ± 2.6	62.7 ± 0.8	58.3 ± 0.0	$\mathbf{78.6 \pm 2.0}$	77.1 ± 1.8
35	46.8 ± 0.8	42.5 ± 2.3	60.5 ± 0.4	63.2 ± 0.0	45.4 ± 2.9	62.6 ± 0.9	57.9 ± 0.0	$\mathbf{77.5 \pm 2.0}$	76.7 ± 1.8
40	46.4 ± 0.8	41.7 ± 2.7	60.3 ± 0.4	62.9 ± 0.0	44.9 ± 3.0	62.3 ± 1.1	57.4 ± 0.0	$\mathbf{76.2 \pm 2.1}$	75.6 ± 1.9
45	46.1 ± 0.9	40.8 ± 3.4	60.2 ± 0.4	63.1 ± 0.0	43.1 ± 3.1	62.1 ± 1.1	57.0 ± 0.0	$\mathbf{75.8 \pm 2.3}$	74.3 ± 2.1
50	45.9 ± 0.9	39.7 ± 3.9	60.1 ± 0.4	63.3 ± 0.0	42.2 ± 3.4	61.9 ± 1.3	56.6 ± 0.0	$\mathbf{74.9 \pm 2.5}$	73.1 ± 2.3
Average	47.70	44.98	60.78	63.47	46.11	62.88	58.60	**78.40**	77.89

Table A.34: AUC's (mean±std%) of different approaches for CAT class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	55.3 ± 1.4	55.6 ± 1.9	63.5 ± 0.6	46.9 ± 0.0	55.8 ± 1.2	63.4 ± 0.8	56.7 ± 0.0	76.1 ± 1.0	$\mathbf{76.9 \pm 1.3}$
10	54.9 ± 1.4	54.7 ± 2.2	62.8 ± 0.8	46.5 ± 0.0	54.7 ± 1.6	63.4 ± 0.8	56.0 ± 0.0	75.6 ± 1.0	$\mathbf{76.4 \pm 1.3}$
15	54.1 ± 1.3	53.7 ± 2.1	62.3 ± 0.8	46.1 ± 0.0	53.4 ± 1.9	63.4 ± 0.8	55.1 ± 0.0	74.8 ± 1.1	$\mathbf{75.8 \pm 1.3}$
20	53.2 ± 1.5	52.9 ± 2.4	61.9 ± 0.9	45.6 ± 0.0	52.1 ± 1.8	63.3 ± 0.8	54.2 ± 0.0	74.1 ± 1.1	$\mathbf{75.1 \pm 1.3}$
25	52.5 ± 1.5	52.3 ± 2.3	61.5 ± 0.9	45.4 ± 0.0	51.6 ± 2.1	63.3 ± 0.9	54.2 ± 0.0	73.3 ± 1.1	$\mathbf{74.5 \pm 1.4}$
30	51.9 ± 1.4	51.8 ± 2.6	61.1 ± 0.9	45.2 ± 0.0	50.7 ± 2.4	63.2 ± 0.9	54.2 ± 0.0	72.8 ± 1.2	$\mathbf{73.6 \pm 1.4}$
35	51.3 ± 1.3	51.6 ± 2.7	61.0 ± 0.9	45.1 ± 0.0	50.3 ± 2.7	63.2 ± 0.9	53.9 ± 0.0	72.2 ± 1.2	$\mathbf{72.7 \pm 1.6}$
40	50.6 ± 1.3	51.1 ± 2.7	60.8 ± 0.9	44.9 ± 0.0	50.0 ± 2.9	62.3 ± 0.9	53.6 ± 0.0	$\mathbf{71.5 \pm 1.2}$	$\mathbf{71.5 \pm 1.7}$
45	49.9 ± 1.2	51.7 ± 2.8	60.3 ± 1.0	44.8 ± 0.0	48.5 ± 3.2	63.4 ± 1.0	53.3 ± 0.0	$\mathbf{70.7 \pm 1.3}$	$\mathbf{70.7 \pm 1.7}$
50	49.2 ± 1.1	51.8 ± 2.6	59.7 ± 1.0	44.7 ± 0.0	47.7 ± 3.5	63.4 ± 1.0	53.1 ± 0.0	69.9 ± 1.3	$\mathbf{70.1 \pm 1.7}$
Average	52.29	52.72	61.49	45.52	51.48	63.33	54.43	73.10	**73.73**

Table A.35: AUC's (mean±std%) of different approaches for DEER class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	65.5 ± 0.9	57.9 ± 1.5	71.5 ± 0.5	64.8 ± 0.0	53.4 ± 1.4	60.1 ± 0.7	60.5 ± 0.0	$\mathbf{85.8 \pm 1.0}$	83.4 ± 1.4
10	65.2 ± 1.0	56.8 ± 1.7	70.6 ± 0.8	64.4 ± 0.0	52.8 ± 1.6	60.0 ± 0.8	59.8 ± 0.0	$\mathbf{85.2 \pm 1.1}$	82.9 ± 1.4
15	64.7 ± 1.0	55.7 ± 1.8	71.1 ± 0.7	64.3 ± 0.0	52.2 ± 1.9	59.9 ± 0.8	58.2 ± 0.0	$\mathbf{84.7 \pm 1.1}$	82.2 ± 1.4
20	64.1 ± 1.0	54.9 ± 2.1	70.8 ± 0.5	64.5 ± 0.0	51.9 ± 1.9	59.4 ± 1.1	57.1 ± 0.0	$\mathbf{84.3 \pm 1.1}$	81.6 ± 1.5
25	63.3 ± 1.0	54.0 ± 2.2	70.8 ± 0.6	64.8 ± 0.0	51.0 ± 2.1	59.1 ± 1.2	56.8 ± 0.0	$\mathbf{82.8 \pm 1.3}$	82.0 ± 1.5
30	62.3 ± 1.1	53.1 ± 2.4	70.5 ± 0.5	64.6 ± 0.0	50.2 ± 2.5	58.8 ± 1.2	56.5 ± 0.0	81.6 ± 1.3	$\mathbf{82.1 \pm 1.5}$
35	61.4 ± 1.2	52.8 ± 2.5	69.7 ± 0.8	64.2 ± 0.0	50.0 ± 2.4	58.5 ± 1.5	56.0 ± 0.0	80.8 ± 1.4	$\mathbf{81.9 \pm 1.5}$
40	60.2 ± 1.1	52.3 ± 2.4	68.9 ± 1.0	63.9 ± 0.0	49.7 ± 1.4	58.1 ± 1.5	55.6 ± 0.0	80.3 ± 1.5	$\mathbf{81.3 \pm 1.5}$
45	59.1 ± 1.2	51.1 ± 2.8	69.0 ± 1.1	63.7 ± 0.0	46.8 ± 2.7	58.0 ± 1.6	55.3 ± 0.0	79.4 ± 1.5	$\mathbf{80.8 \pm 1.6}$
50	58.3 ± 1.3	50.6 ± 3.2	68.8 ± 1.3	63.4 ± 0.0	44.3 ± 3.3	57.8 ± 1.8	55.0 ± 0.0	78.9 ± 1.5	$\mathbf{80.4 \pm 1.6}$
Average	62.41	53.92	70.27	64.26	50.23	58.97	57.08	**82.38**	81.86

Table A.36: AUC's (mean±std%) of different approaches for DOG class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	62.0 ± 0.8	63.8 ± 2.8	66.8 ± 0.1	50.1 ± 0.0	58.9 ± 2.4	73.4 ± 1.3	51.8 ± 0.0	$\mathbf{84.8 \pm 1.0}$	76.6 ± 1.8
10	61.8 ± 0.8	62.1 ± 2.9	66.5 ± 0.1	49.6 ± 0.0	58.4 ± 2.6	72.6 ± 1.3	51.4 ± 0.0	$\mathbf{84.2 \pm 1.0}$	75.8 ± 1.8
15	61.0 ± 0.8	61.5 ± 3.1	66.4 ± 0.2	49.1 ± 0.0	57.5 ± 2.7	72.8 ± 1.2	50.9 ± 0.0	$\mathbf{83.8 \pm 1.0}$	74.6 ± 1.9
20	60.1 ± 0.9	59.7 ± 3.0	66.3 ± 0.3	48.7 ± 0.0	56.8 ± 2.9	73.1 ± 1.2	50.5 ± 0.0	$\mathbf{83.2 \pm 1.1}$	74.0 ± 1.9
25	58.7 ± 0.9	57.6 ± 3.3	65.8 ± 0.3	48.3 ± 0.0	55.1 ± 3.2	71.0 ± 1.6	50.6 ± 0.0	$\mathbf{82.9 \pm 1.1}$	72.9 ± 2.1
30	57.4 ± 1.1	56.1 ± 3.7	65.1 ± 0.3	47.9 ± 0.0	53.7 ± 3.1	71.3 ± 1.6	50.8 ± 0.0	$\mathbf{82.6 \pm 1.1}$	72.3 ± 2.1
35	56.1 ± 1.2	55.2 ± 3.8	64.2 ± 0.4	47.5 ± 0.0	53.3 ± 3.2	70.8 ± 1.7	50.3 ± 0.0	$\mathbf{81.9 \pm 1.2}$	71.1 ± 2.2
40	54.3 ± 1.3	54.3 ± 3.8	63.2 ± 0.4	47.2 ± 0.0	52.8 ± 3.0	69.7 ± 1.8	50.4 ± 0.0	$\mathbf{81.4 \pm 1.2}$	70.4 ± 2.2
45	53.3 ± 1.5	53.0 ± 4.1	62.9 ± 0.4	47.0 ± 0.0	52.3 ± 3.3	69.0 ± 1.7	50.4 ± 0.0	$\mathbf{81.0 \pm 1.4}$	69.7 ± 2.3
50	52.2 ± 1.6	51.7 ± 4.6	62.5 ± 0.4	46.8 ± 0.0	51.9 ± 3.9	68.4 ± 2.0	50.5 ± 0.0	$\mathbf{80.9 \pm 1.4}$	68.8 ± 2.4
Average	57.69	57.50	64.98	48.22	55.07	71.32	50.76	**82.67**	72.62

Table A.37: AUC's (mean±std%) of different approaches for FROG class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	73.5 ± 0.4	58.7 ± 4.1	77.9 ± 0.2	58.9 ± 0.0	55.5 ± 5.9	57.8 ± 3.9	50.1 ± 0.0	$\mathbf{89.1 \pm 0.8}$	80.1 ± 1.1
10	72.9 ± 0.4	57.6 ± 4.4	78.0 ± 0.2	58.6 ± 0.0	54.8 ± 6.0	56.4 ± 4.0	50.0 ± 0.0	$\mathbf{88.7 \pm 0.9}$	79.8 ± 1.1
15	72.4 ± 0.5	56.3 ± 4.6	78.1 ± 0.2	58.1 ± 0.0	53.0 ± 5.9	54.3 ± 1.9	49.7 ± 0.0	$\mathbf{88.1 \pm 0.9}$	79.4 ± 1.1
20	71.8 ± 0.6	54.8 ± 4.9	77.9 ± 0.2	57.7 ± 0.0	51.9 ± 6.3	52.1 ± 1.6	49.2 ± 0.0	$\mathbf{87.6 \pm 1.0}$	79.1 ± 1.1
25	71.0 ± 0.7	54.4 ± 5.0	77.5 ± 0.2	57.0 ± 0.0	51.1 ± 6.4	51.3 ± 1.2	49.6 ± 0.0	$\mathbf{86.4 \pm 1.1}$	78.6 ± 1.2
30	70.6 ± 0.8	53.7 ± 4.9	77.4 ± 0.2	56.4 ± 0.0	50.2 ± 6.4	50.5 ± 0.5	50.3 ± 0.0	$\mathbf{85.3 \pm 1.3}$	78.2 ± 1.2
35	69.1 ± 0.9	53.2 ± 5.0	77.8 ± 0.2	56.2 ± 0.0	50.6 ± 6.3	50.6 ± 0.3	50.0 ± 0.0	$\mathbf{84.2 \pm 1.3}$	77.9 ± 1.3
40	68.2 ± 1.1	51.7 ± 5.4	77.7 ± 0.2	56.0 ± 0.0	50.4 ± 6.2	50.0 ± 0.3	49.7 ± 0.0	$\mathbf{83.3 \pm 1.5}$	77.2 ± 1.3
45	67.2 ± 1.1	52.5 ± 5.2	77.1 ± 0.2	55.7 ± 0.0	50.2 ± 6.3	50.0 ± 0.2	49.9 ± 0.0	$\mathbf{82.7 \pm 1.6}$	76.8 ± 1.4
50	66.4 ± 1.2	52.2 ± 5.3	77.2 ± 0.2	55.5 ± 0.0	50.0 ± 6.2	50.0 ± 0.2	49.9 ± 0.0	$\mathbf{81.9 \pm 1.8}$	76.5 ± 1.4
Average	70.31	54.51	77.67	57.01	51.76	52.30	49.84	**85.73**	78.36

Table A.38: AUC's (mean±std%) of different approaches for HORSE class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	61.5 ± 0.5	61.3 ± 3.1	64.0 ± 0.2	49.5 ± 0.0	56.8 ± 3.1	64.5 ± 1.0	54.3 ± 0.0	$\mathbf{88.7 \pm 1.2}$	79.5 ± 1.8
10	61.1 ± 0.5	60.9 ± 3.4	63.8 ± 0.2	49.0 ± 0.0	55.8 ± 3.4	64.1 ± 1.1	53.5 ± 0.0	$\mathbf{88.2 \pm 1.2}$	78.7 ± 1.8
15	60.5 ± 0.6	59.6 ± 3.3	63.5 ± 0.2	48.6 ± 0.0	54.1 ± 3.6	63.8 ± 1.1	52.6 ± 0.0	$\mathbf{87.6 \pm 1.3}$	77.3 ± 1.8
20	59.2 ± 0.7	58.1 ± 3.5	63.1 ± 0.2	48.2 ± 0.0	52.9 ± 3.9	63.6 ± 1.2	51.8 ± 0.0	$\mathbf{87.1 \pm 1.3}$	76.6 ± 1.9
25	58.7 ± 0.7	57.7 ± 3.6	63.0 ± 0.2	48.1 ± 0.0	52.2 ± 3.8	63.1 ± 1.3	51.5 ± 0.0	$\mathbf{86.2 \pm 1.5}$	74.9 ± 2.1
30	57.3 ± 0.9	56.8 ± 3.9	62.9 ± 0.2	48.0 ± 0.0	51.1 ± 4.0	62.8 ± 1.5	51.2 ± 0.0	$\mathbf{85.1 \pm 1.7}$	73.7 ± 2.2
35	56.4 ± 1.0	54.8 ± 4.3	63.6 ± 0.2	48.3 ± 0.0	50.2 ± 4.1	62.4 ± 1.4	50.6 ± 0.0	$\mathbf{83.7 \pm 2.0}$	73.9 ± 2.1
40	55.1 ± 1.0	53.8 ± 4.4	63.2 ± 0.2	48.1 ± 0.0	48.9 ± 4.6	61.8 ± 1.4	49.8 ± 0.0	$\mathbf{82.3 \pm 2.1}$	73.0 ± 2.1
45	54.4 ± 1.3	52.0 ± 4.6	63.2 ± 0.3	48.2 ± 0.0	49.7 ± 4.4	61.5 ± 1.4	49.5 ± 0.0	$\mathbf{81.4 \pm 2.4}$	72.6 ± 2.2
50	53.4 ± 1.4	50.7 ± 5.0	63.3 ± 0.3	48.2 ± 0.0	49.1 ± 4.6	61.3 ± 1.5	49.2 ± 0.0	$\mathbf{80.1 \pm 2.6}$	71.5 ± 2.4
Average	57.76	56.57	63.36	48.42	52.08	62.89	51.40	**85.04**	75.17

Table A.39: AUC's (mean±std%) of different approaches for SHIP class for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	73.0 ± 0.4	73.4 ± 1.2	81.7 ± 0.2	71.9 ± 0.0	73.9 ± 1.9	75.6 ± 1.1	68.1 ± 0.0	$\mathbf{90.4 \pm 0.9}$	81.6 ± 1.9
10	72.4 ± 0.4	72.6 ± 1.2	81.5 ± 0.2	71.2 ± 0.0	72.8 ± 2.0	75.3 ± 1.1	67.6 ± 0.0	$\mathbf{90.2 \pm 0.9}$	81.1 ± 1.9
15	71.2 ± 0.6	70.6 ± 2.3	81.1 ± 0.2	70.7 ± 0.0	72.1 ± 2.0	75.1 ± 1.2	67.8 ± 0.0	$\mathbf{90.1 \pm 0.9}$	80.7 ± 1.9
20	70.1 ± 0.6	68.9 ± 2.9	81.3 ± 0.2	70.1 ± 0.0	71.4 ± 2.3	74.9 ± 1.3	67.7 ± 0.0	$\mathbf{89.9 \pm 0.9}$	80.0 ± 1.9
25	68.6 ± 0.7	67.5 ± 3.1	80.6 ± 0.3	70.0 ± 0.0	70.4 ± 2.5	74.6 ± 1.3	67.8 ± 0.0	$\mathbf{89.4 \pm 1.0}$	79.5 ± 2.0
30	66.9 ± 0.8	66.3 ± 3.3	80.1 ± 0.3	69.9 ± 0.0	69.8 ± 2.6	74.4 ± 1.2	67.9 ± 0.0	$\mathbf{89.1 \pm 1.0}$	79.1 ± 2.0
35	65.7 ± 0.9	64.8 ± 3.7	79.0 ± 0.7	69.6 ± 0.0	67.3 ± 2.9	74.2 ± 1.2	67.7 ± 0.0	$\mathbf{88.9 \pm 1.1}$	78.8 ± 2.0
40	64.6 ± 0.9	62.7 ± 4.1	77.9 ± 1.0	69.1 ± 0.0	64.4 ± 3.6	74.0 ± 1.2	67.4 ± 0.0	$\mathbf{88.6 \pm 1.1}$	78.6 ± 2.1
45	64.1 ± 1.1	61.0 ± 4.3	75.8 ± 1.9	68.8 ± 0.0	62.8 ± 4.1	74.1 ± 1.2	67.5 ± 0.0	$\mathbf{88.1 \pm 1.3}$	79.0 ± 1.9
50	63.9 ± 1.3	59.9 ± 4.6	74.2 ± 2.4	68.4 ± 0.0	61.9 ± 4.2	73.8 ± 1.3	67.6 ± 0.0	$\mathbf{87.8 \pm 1.5}$	79.2 ± 1.9
Average	68.05	66.77	79.32	69.97	68.68	74.60	67.71	**89.25**	79.76

Table A.40: AUC's (mean±std%) of different approaches for TRUCK for one-class classification on the CIFAR10 dataset for Observation Ranking.

Contamination (%)	OC-SVM	DSVDD	HRN	Robust-kernel	DCAE	ICS	SRO	G-DRKSR	C-DRKSR
5	73.9 ± 0.4	68.9 ± 2.7	77.1 ± 0.9	52.3 ± 0.0	65.9 ± 3.3	59.8 ± 1.0	51.7 ± 0.0	$\mathbf{82.8 \pm 1.0}$	72.4 ± 1.3
10	73.1 ± 0.5	67.6 ± 2.7	76.9 ± 0.9	51.7 ± 0.0	64.4 ± 3.4	58.7 ± 1.2	51.4 ± 0.0	$\mathbf{82.4 \pm 1.2}$	71.1 ± 1.4
15	72.5 ± 0.5	66.5 ± 2.9	76.7 ± 0.9	51.4 ± 0.0	64.1 ± 3.3	58.1 ± 1.4	51.3 ± 0.0	$\mathbf{82.2 \pm 1.2}$	70.9 ± 1.4
20	71.7 ± 0.6	65.9 ± 3.0	76.4 ± 0.9	51.2 ± 0.0	63.9 ± 3.5	57.6 ± 1.5	51.1 ± 0.0	$\mathbf{82.1 \pm 1.2}$	70.4 ± 1.4
25	70.3 ± 0.6	65.0 ± 3.1	75.9 ± 1.0	51.0 ± 0.0	63.3 ± 3.6	56.5 ± 1.5	50.9 ± 0.0	$\mathbf{81.8 \pm 1.2}$	70.7 ± 1.3
30	68.9 ± 0.9	64.1 ± 3.3	75.5 ± 1.0	50.9 ± 0.0	62.9 ± 3.8	55.8 ± 1.7	50.5 ± 0.0	$\mathbf{81.6 \pm 1.4}$	70.1 ± 1.3
35	67.6 ± 1.2	62.9 ± 3.8	75.3 ± 1.0	50.0 ± 0.0	62.4 ± 4.0	55.1 ± 1.9	50.6 ± 0.0	$\mathbf{81.0 \pm 1.4}$	69.8 ± 1.4
40	66.8 ± 1.1	62.1 ± 3.7	75.1 ± 1.0	48.5 ± 0.0	61.1 ± 4.2	54.7 ± 2.0	50.2 ± 0.0	$\mathbf{80.7 \pm 1.6}$	69.4 ± 1.4
45	65.5 ± 1.2	62.3 ± 3.6	74.7 ± 1.0	48.2 ± 0.0	60.3 ± 4.5	54.1 ± 1.9	49.7 ± 0.0	$\mathbf{80.1 \pm 1.6}$	68.9 ± 1.5
50	64.6 ± 1.3	61.8 ± 3.9	74.3 ± 1.0	48.0 ± 0.0	59.7 ± 4.8	53.3 ± 2.3	49.3 ± 0.0	$\mathbf{79.5 \pm 1.8}$	68.3 ± 1.5
Average	69.49	64.71	**75.79**	50.32	62.80	52.41	50.67	**81.42**	70.2