Training Outlier Rejection for Unsupervised Anomaly Detection

Naji Najari
Orange Labs, LIRIS CNRS
Meylan, France
firstname.lastname@orange.com

Samuel Berlemont, Grégoire Lefebvre
Orange Labs
Meylan, France
firstname.lastname@orange.com

Stefan Duffner, Christophe Garcia
LIRIS CNRS, INSA Lyon
Villeurbanne, France
firstname.lastname@liris.cnrs.fr

Abstract

Anomaly detection consists in detecting non-conforming anomalous data points. Traditional one-class-based anomaly detectors assume that the training data are anomaly-free and generated from one single class, considered the norm. The performance of these approaches significantly degrades when a ratio of outliers contaminates the training dataset. In this paper, we propose GRAnD, a training strategy robust to outliers in training data and hyperparameter selection. Based on Variational Autoencorders and Normalizing Flows, our proposal filters training outliers by means of a robust rejection strategy derived from Extreme Value Theory. Then, it leverages filtered training anomalies to improve outlier detection performance and to generalize to unseen new anomalies. Extensive experiments and sensitivity analyses on image and network traffic benchmark datasets demonstrate the effectiveness of our approach in unsupervised anomaly detection.

1 Introduction

Anomaly Detection (AD), a.k.a., outlier and novelty detection, is the task of detecting anomalous data points that significantly deviate from expected normal samples Chandola et al. [2009]. Hawkins [1980] defines an anomaly as “an observation that deviates so significantly from other observations as to arouse suspicion that it was generated by a different mechanism”. Detecting abnormal data is of paramount importance to mitigate risks, prevent system failures, and have an in-depth understanding of the underlying data patterns. AD has received renewed attention in multiple research fields and has numerous applications in different domains, such as network security, fraud detection, healthcare, or drug discovery. Several recent surveys have been proposed to extensively review anomaly-based methods Chandola et al. [2009], Domingues et al. [2018], Thudumu et al. [2020].

AD is an arduous problem since outliers are scarce and anomalies cannot be thoroughly sampled. Therefore, supervised machine learning methods have limited success in solving such problem. The common and classical line of work for AD is based on One-Class Classification (OCC) Moya et al. [1993], Schölkopf et al. [2001]. The basic idea is to learn an accurate representation of the norm, relying only on nominal data points. Once the normal data are well-modeled, the algorithm assigns an abnormality score to each test sample. This score is thresholded to separate inliers and outliers. The choice of the model family is a determinant factor for the success of these anomaly detectors. Inlier models need to be flexible enough to accurately characterize the normality, as a poor fit can result in numerous false positives and negatives.

Preprint. Under review.

Nevertheless, OCC-based AD assumes that the training data are anomaly-free. Unfortunately, this strong assumption is not always guaranteed in real-world applications. Training samples may be corrupted with an unknown fraction of outliers. For example, in network traffic monitoring, collected network packets may comprise defective data sent by faulty sensors, damaged fiber connectors, or caused by network congestion Zhuo et al. [2017]. One-class-based anomaly detectors are sensitive to infiltrated training outliers, and their performance may degrade significantly when the one-class assumption is violated.

In this paper, we propose GRAnD, an algorithm for Generative Robust Anomaly Detection. This novel approach makes generative autoencoders more robust under the standard variational inference. We introduce a training strategy that alternates between filtering training outliers and learning a robust representation of the norm. Our training strategy involves little architectural changes and can be integrated with Variational Autoencoders (VAEs) and Normalizing Flows (NFs). Unlike recent robust generative methods Lai et al. [2020b], Eduardo et al. [2020], our approach makes no assumption about the anomaly distribution, or about the fraction of training outliers.

Our method comprises three contributions :

- a robust rejection strategy that filters corrupted training samples, based on Extreme Value Theory (EVT). Prior knowledge about the outlier ratio contaminating the training data is not required in advance.
- a training strategy that leverages filtered anomalies to learn a robust representation where inliers are well reconstructed and outliers are explicitly corrupted.
- an extensive empirical validation on public image and network traffic datasets, which demonstrates that our approach outperforms some state-of-the-art robust methods.

2 Related Work

Anomaly detection is an active research field that has always been a point of interest in different applications Chandola et al. [2009], Thudumu et al. [2020]. A myriad of approaches were proposed, ranging from probabilistic-based, to neighbor-based, domain-based, and reconstruction-based methods.

Statistical and probabilistic-based methods typically model the inlier distribution by learning the parameters of a parametric function. Samples that have low likelihood under this model are considered anomalies. This category includes Gaussian mixture models Yang et al. [2009], kernel density estimators Desforges et al. [1998], and probabilistic principal component analysis Tipping and Bishop [1999]. Neighbor-based methods, a.k.a. proximity-based methods, assume that outliers are far from their nearest neighbors, while inliers are close to each other. Well-known proximity-based method include Local Outlier Factor (LOF) Breunig et al. [2000], Angle-Based Outlier Detection (ABOD) Kriegel et al. [2008]. Domain-based methods estimate a boundary that separates the inlier domain from the rest. Anomalies are samples outside this inlier boundary. One-class SVM (OC-SVM) Schölkopf et al. and Support Vector Data Description (SVDD) Tax and Duin [2004] are two popular domain-based algorithms. Reconstruction-based anomaly detection assumes that, unlike outliers, inliers can be projected into a low-dimensional subspace. The reconstruction error represents a score of data abnormality, as the reconstruction errors of anomalies are higher than inliers. Particularly, AutoEncoders (AEs) have been trained to map nominal input data into a compact latent space, to learn a non-linear representation of the nominal class Chaurasia et al. [2020]. Besides, generative models have been profusely proposed for anomaly detection. An and Cho [2015] proposed a VAE-based outlier detector. The variational setting is used to estimate the log-likelihood of samples, and outliers are data points having low likelihoods. Ryzhikov et al. [2019] proposed a NF-based anomaly detection algorithm. This approach comprises two steps. Firstly, they train an NF, the Inverse Autoregressive Flow (IAF) Kingma et al. [2016], to model the normal data distribution. In the second step, they sample artificial anomalies from the tail of the nominal distribution, learned in the first step, and they train a binary classifier using normal samples and these artificial anomalies. Furthermore, numerous studies explored Generative Adversarial Networks (GANs) Radford et al. [2016], Goodfellow et al. [2014] for AD. Di Mattia et al. [2019] presents an extensive survey of GAN-based anomaly detectors.

The above methods are effective in the context of one-class classification, i.e., when the training data are anomaly-free. However, their performance drastically decreases in the presence of unlabeled training outliers. To our knowledge, limited studies have focused on training robust anomaly detection models when the training data may contain an unknown ratio of unlabeled outliers.

Arpit et al. [2017] empirically investigated the capacity of neural networks to fit data contaminated with random noise (i.e. both data and label noise). They found that Deep Neural Networks (DNNs) gradually learn more complex representations during training. They state that "the model first learns the simple and general patterns of the real data before fitting the noise".

Inspired by Robust PCA (RPCA) Candès et al. [2011], Zhou and Paffenroth [2017] proposed Robust Deep Autoencoders (RDAs) to filter sparse corrupted samples from the input data matrix. Robust Subspace Recovery (RSR) Lerman and Maunu [2018] is another line of work in robust anomaly detection. RSR assumes that inliers can be projected into a linear low-dimensional subspace, while outliers are not well modeled in this subspace. Consequently, RSR extracts a linear geometric structure from the input space, and outliers are samples that lie outside. Lai et al. [2020a] introduced Robust Subspace Recovery AutoEncoder (RSRAE), where they integrated an RSR-layer in a classical autoencoder. Regarding robust generative autoencoders, Akrami et al. [2019] proposed a Robust VAE (RVAE). Their approach uses the robust β-divergence Basu [1998] instead of the standard Kullback-Leibler (KL) divergence. Minimizing the β-divergence involves reweighting each sample likelihood gradient with its probability density. Therefore, outliers are down-weighted since their densities are much lower than inliers Futami et al. [2018]. In a similar vein, Eduardo et al. [2020] proposed an RVAE for mixed-type data.

All above methods involve an explicit regularization, defined by one or many critical hyperparameters. Prior knowledge about the outlier ratio and additional assumptions either on the inlier or the outlier class, or both, are required to select the optimal hyperparameters. Generally, such hyperparameters are empirically tuned with a dedicated validation subset containing ground-truth-labeled data. In the context of anomaly detection, labeled outliers are too scarce to form a balanced validation subset. Also, in most situations, the ratio of training outliers is not known. Therefore, hyperparameter selection is prone to misspecification. However, the methods above are all sensitive to their hyperparameters, since slightly changing them can drastically degrade their anomaly detection performances.

Another way to approach this problem with VAEs is to assume that the output of the encoder follows a mixture of two distributions, each component models one class (i.e., inlier or outlier). Lai et al. [2020b] model the latent embedding with a mixture of two Gaussians with two different means. However, the strong assumption that anomalies are generated from a single distribution, a.k.a., the cluster assumption Ruff et al. [2020], is not realistic. Unlike inliers, outliers can be heterogeneous and generated by different underlying processes.

In contrast, our approach does not make any assumptions about outlier distribution. We propose a robust training strategy that jointly performs two tasks. This strategy filters training outliers using EVT. Then, training outliers are leveraged to infer a better representation that can be generalized to unseen anomalies. This strategy can be incorporated with VAEs and NFs, and involves minimal architectural changes.

3 Background

Firstly, we will present the theoretical background of generative AEs, under the standard variational setting. Secondly, we explain the main results of EVT, and how they can be applied in AD to automatically compute a rejection threshold.

3.1 Generative AEs

We consider the task of unsupervised AD under the standard variational inference setting. Generative models aim to find the optimal parameters θ that maximize the likelihood $p_\theta(x) = \mathbb{E}_{p(z)}[p_\theta(x|z)]$, where z is the model latent variable and $p(z)$ is a predefined prior (e.g. the Gaussian distribution). They empirically select the optimal parameters that maximize the log-likelihood $\hat{\theta} = argmax_\theta \ logp_\theta(x)$. However, this log-likelihood is intractable because of the marginalization over the latent variable z. Variational inference aims to approximate the posterior probability $p(z|x)$ with a parametric distribution $q_\phi(z|x)$, parameterized by ϕ. Regardless of the choice of this distribution, we can reformulate the log-likelihood as follows:

$$\log p_\theta(x) \geq \mathbb{E}_q[\log p_\theta(x|z)] - \mathbb{D}_{KL}[q_\phi(z|x)||p(z)] = -\mathcal{F}(x), \tag{1}$$

where $q_\phi(z|x)$ is the approximate posterior distribution for the latent variables, and $\mathcal{F}$ is the negative free energy, a.k.a., the evidence lower bound (ELBO). This energy comprises two terms. The first term is the reconstruction error, and the second one represents the KL divergence between the approximate distribution and the prior distribution. A common choice of the approximate distribution family is the multivariate Gaussian distribution with diagonal covariance matrix. Recently, NFs have been used to provide a richer parametric family of approximate posterior to capture complex structures of the latent space. NFs transform an initial simple density function to a more sophisticated one, by applying a sequence of invertible transformations. Formally, let $z_0 \in \mathbb{R}^d$ be a random variable that follows a probability distribution $q_0(z_0)$ and $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ an invertible mapping. We can transform z_0 to z_k by applying K mappings:

$$z_k = f_K \circ ... \circ f_1(z_0) \quad \text{and} \quad q_k(z_k) = q_0(z_0) \prod_{k=1}^{K} \left| \det \frac{\partial f_k}{\partial z_{k-1}} \right|^{-1}. \tag{2}$$

$q_k(z_k)$ is the probability distribution of z_k. The free-energy function can be rewritten as :

$$\mathcal{F}(x) = \mathbb{E}_{q_0(z_0)}[\log q_0(z_0) - \sum_{k=1}^{K} \log \left| \det \frac{\partial f_k}{\partial z_{k-1}} \right| - \log p(z_K)] - \mathbb{E}_{q_0(z_0)}[\log p(x|z_K)]. \tag{3}$$

where $q_0(z_0)$ is the intitial distribution before applying the K mapping. Several architectures have been proposed in the literature, including planar and radial flows Rezende and Mohamed [2016], coupling flows Dinh et al. [2015], autoregressive flows Kingma et al. [2017].

3.2 EVT

The objective of EVT is to quantify the probability of occurrence of extreme values in a distribution function. Recently, EVT has been applied to detect anomalies in many applications including network traffic data streams Siffer et al. [2017]. The Peaks-Over-Threshold (POT) is a typical approach used to model the extreme values of samples that exceed a specific high threshold. This approach is a result of the Picakands-Balkema-de-Han theorem of EVT Balkema and Haan [1974].

Let $(X_1, X_2, \ldots, X_n)$ be a sequence of n independent and identically-distributed random variables. Let F_u be their conditional excess distribution function, i.e., $F_u(x) = P(X - u < x|X > u)$, where u is a high threshold.

Theorem 1 (Pickands-Balkema-De-Hans)**.** *For a large class of underlying distribution functions F, and large u, F_u is well approximated by the generalized Pareto distribution. That is:*

$$F_u(x) \to G_{\xi,\sigma}(x), \; \textit{as } u \to \infty \quad \textit{such that} \quad \begin{cases} G_{\xi,\sigma}(x) = 1 - (1 + \frac{\xi x}{\sigma})^{-\frac{1}{\xi}}, & if\ \xi \neq 0 \\ G_{\xi,\sigma}(x) = 1 - e^{-\frac{x}{\sigma}}, & if\ \xi = 0. \end{cases} \tag{4}$$

This theorem states that the extreme values that exceed the threshold u, can be approximated by the Generalized Pareto Distribution (GPD) parametrized by two parameters, ξ and σ.

$$\tilde{F}_u(x) = P(X - u > x|X > u) = 1 - P(X - u < x|X > u) = 1 - F_u(x), \tag{5}$$

$$\tilde{F}_u(x) \to 1 - G_{\xi,\sigma}(x), \text{ as } u \to \infty. \tag{6}$$

In practice, the two parameters of the GPD are empirically estimated by fitting the GPD to the data. The maximum likelihood estimation is typically used to find these optimal parameters $\tilde{\xi}$ and $\tilde{\sigma}$. Once the extreme values are modeled with the optimal GPD, $G_{\tilde{\xi},\tilde{\sigma}}$, one can precisely quantify the probability of occurence of each extreme values Siffer et al. [2017]. Particularly, we can identify rare extreme samples that have very low probability. Given a small probability q, we can compute the threshold t such that, $P(X > t) < q$.

$$P(X - u > t|X > u) = \tilde{F}_u(t) \sim 1 - G_{\tilde{\xi},\tilde{\sigma}}(t). \tag{7}$$

$$\text{If } \xi \neq 0, \quad t \simeq u + \frac{\tilde{\xi}}{\tilde{\sigma}}((\frac{nq}{N})^{\tilde{\xi}} - 1), \tag{8}$$

where n is the total number of observations, and N_t is the number of X_i exceeding the threshold u, $X_i > u$. Appendix A.1 presents more details about the computation of the threshold t.
A key question arises as to how to choose the threshold u. Siffer et al. [2017] state that u "value is not paramount except that it must be “high"enough." In practice, u is generally selected as a high empirical quantile of the data (e.g., 90% quantile).

4 Contributions

This paper focuses on unsupervised anomaly detection where the unlabeled training data may contain both inliers and outliers, with an imbalanced class distribution. We assume that the majority of the training instances are nominal, along with a small ratio of “contaminants” (i.e. outliers). The ratio of these contaminants, which we call γ_p, is not known in advance. In the following, we introduce GRAnD, an algorithm for Generative Robust Anomaly Detection.

We build upon the observation from Arpit et al. [2017] that even if neural networks with sufficient capacity are capable of modeling corrupted data, they first prioritize learning common patterns shared by the majority of training samples . That is, early in the training, the model learns a global representation that fits the majority of the training data, and overfitting outliers occurs later in a second stage. Therefore, it is more appropriate to filter out potential outliers at the beginning of the training. Given unlabeled data containing inliers and a minority of outliers, we train a generative model to simultaneously discover these contaminants and to learn a robust distribution of the norm. Even though outliers are not well-sampled from the whole positive class, we find that they provide global insights that can be generalized to unseen anomalies.

In the following, we will first explain the rejection strategy that enables a generative AE to isolate training contaminant samples. Then, we will detail the objective function to optimize.

4.1 Robust Rejection Strategy

The objective of this rejection strategy is to separate nominal training data points from anomalies. The main idea consists in setting a relevant threshold to segment the normality scores assigned to training samples, in order to reject outliers having extreme scores. Unlike traditional AD approaches, we propose an EVT-based rejection strategy, which is more flexible and does not require any assumption about the true distribution of the data. Indeed, the rejection criterion in traditional AD approaches is selected manually or with statistical heuristics (e.g., using quantiles, the 3-sigma rule). Specific assumptions about the underlying data distribution are required to compute the optimal threshold. Such rejection strategies may be suboptimal, if the data do not follow the assumed distribution.

We hypothesize that, early in the training phase, contaminants have larger negative-log-likelihoods, compared to inliers. Consequently, we propose to isolate these extreme values by thresholding the negative-log-likelihood with the POT approach, described in Section 3.2. However, the POT approach relies on an important parameter, the risk parameter q, which controls the number of false positives and negatives. A q too high results in filtering out nominal samples (i.e., false positives), while a low q will dismiss false negatives.

Instead of splitting the data into two subsets inliers and outliers, we propose to incorporate a third subset that contains critical samples. These critical instances contain potential outliers dismissed by the POT approach, due to a misspecified q. More precisely, we propose to split the training data into three subsets $\mathbb{X} = \mathbb{L} \cup \mathbb{S} \cup \mathbb{U}$. The subset $\mathbb{S}$ contains nominal training samples, having a log-likelihood lower than the initial threshold u of the POT method. $\mathbb{L}$ contains anomalous data points, with a log-likelihood higher than q. $\mathbb{U}$ comprises the remaining critical samples, with a log-likelihood higher than u and lower than q. These sample log-likelihoods are neither very low to be considered nominal, nor high enough to be rejected as anomalies.

4.2 Training Loss

The rejection strategy splits the training data into three subsets $\mathbb{L}$, $\mathbb{S}$, and $\mathbb{U}$. We train a model to jointly perform two tasks. For $\mathbb{L}$ samples, the model is trained to maximize their likelihood, and to project these inliers in high-density regions of the latent space. Conversely, we train the autoencoder to badly reconstruct $\mathbb{S}$ samples and to disperse their embedding, far from high probability regions.

Given the three subsets of data $\mathbb{L}$, $\mathbb{S}$, and $\mathbb{U}$, respectively generated by three distributions, D_L, D_S, and D_U, we formalize the training energy:

$$\mathcal{F}(x) = \mathbb{E}_{x \sim D_L}[\mathcal{F}_{\mathcal{L}}(x)] + \frac{N_L}{N_S}\mathbb{E}_{x \sim D_S}[\mathcal{F}_{\mathcal{S}}(x)]. \tag{9}$$

Here, N_L represent the number of samples in $\mathbb{L}$ and N_S the number of samples of $\mathbb{S}$. We add this coefficient to compensate the difference in class priors. The output of the probabilistic decoder depends on the type of the original data.

Bernoulli case In the first case, the data are binary, and $p_\theta(x|z)$ is a multivariate Bernouilli.

$$p_\theta(x|z) = \sum_{i=1}^{D} x_i \log y_i + (1 - x_i)\log(1 - y_i) = -l_{BCE}(x, y), \tag{10}$$

where y is reconstructed sample generated by the decoder, and D is the dimension of the data.

The energy function of $\mathbb{L}$ samples, $\mathcal{F}_{\mathcal{L}}(x)$, is exactly the classical energy function, defined as in equation 1 for the VAE-based approach, and in equation 3 for the NF-based approach. For $\mathbb{S}$ samples, we train the decoder to explicitly minimize their likelihood, and to map their latent representation far from the prior distribution. That is, for the VAE-based method,

$$\mathcal{F}_{\mathcal{L}}(x) = l_{BCE}(x, y) - \mathbb{D}_{KL}[q_\phi(z|x)||p(z)], \tag{11}$$

$$\mathcal{F}_{\mathcal{S}}(x) = l_{BCE}(x, 1 - y) - |m - \mathbb{D}_{KL}[q_\phi(z|x)||p(z)]|\,, \tag{12}$$

where $|.|$ is the absolute distance and $m \in \mathbb{R}^+$ is a margin value.

The first term of $\mathcal{F}_{\mathcal{S}}(x)$ aims to minimize the posterior $p_\theta(x|z)$ and to corrupt the reconstructed data. The second term maximizes the Kullback-Leiber divergence D_{KL} between $\mathbb{S}$ latent representations and the prior distribution. However, since D_{KL} is positive and unbounded, we propose to fix an upper bound m, to prevent D_{KL} from diverging in the training. In all our experiments, we fix $m = 10$.

Similarly, for NF, we have

$$\mathcal{F}_{\mathcal{L}}(x) = l_{BCE}(x, y) + \mathbb{E}_{q_0(z_0)}[\log q_K(z_K) - \log p(z_K)], \tag{13}$$

$$\mathcal{F}_{\mathcal{S}}(x) = l_{BCE}(x, 1 - y) + \left|m - \mathbb{E}_{q_0(z_0)}[\log q_K(z_K) - \log p(z_K)]\right|. \tag{14}$$

Continuous case When the data are continuous, we follow the common assumption that $p_\theta(x|z)$ is a Gaussian distribution, and its parameters are inferred at the output of the decoder. The only difference with the Bernouilli case is how to compute the first term of the energy, $\log p_\theta(x|z)$. Let x be the original instance, and μ and σ be the parameters inferred by the decoder. We assume that,

$$\mathbb{E}_{x \sim D_L}[\log p_\theta(x|z)] = \log \mathcal{N}(x; \mu, \sigma^2 I) \quad \text{and} \quad \mathbb{E}_{x \sim D_S}[\log p_\theta(x|z)] = -\log \mathcal{N}(x; \mu, \sigma^2 I). \tag{15}$$

The pseudo-code of the algorithm 1 is presented in Appendix A.2.

5 Experiments

In this section, we present an empirical evaluation of our approach against many competing methods, on different image and non-image benchmark datasets.

5.1 Dataset Description

MNIST The MNIST dataset LeCun and Cortes [2010] is a computer vision dataset containing 28 x 28 grayscale images of handwritten digits. It contains ten classes, representing digits from 0 to 9. The training dataset contains 60 000 images, and the test dataset has 10 000 images.

EMNIST The Extended MNIST (EMNIST) Cohen et al. [2017] consists of a series of six datasets derived from the NIST Special Database. EMNIST contains 28 x 28 grayscale images of handwritten digits, uppercase, and lowercase letters. In our experiments, we select the subset of the EMNIST that contains the handwritten letters. We call this subset Letter-EMNIST. In the preprocessing of both MNIST and Letter-EMNIST, we scale the pixel to the range $[0, 1]$. Similar to Akrami et al. [2019], we binarize the images, by thresholding the pixels according to 0.5.

NSL-KDD Dataset To further validate our approach on another domain using a non-image dataset, we conduct experiments using the NSL-KDD dataset. The NSL-KDD is a benchmark dataset used to evaluate network intrusion detectors. It was derived from the prevalent KDD Cup'99 dataset. The training subset contains 125 973 records and the test subset has 22 544 records. Each data point is represented by 41 features extracted from the network traffic, and labeled as normal or anomalous. Regarding data preprocessing, we scale numerical features using the min-max normalization method. Categorical features are one-hot encoded, resulting in an input dimension equal to 122.

5.2 Experimental Scenarios

We investigate the following three scenarios. Scenario 1 serves as a first validation on a well-known dataset. Scenario 2 explores the impact of the training anomaly ratio. Scenario 3 addresses the problem of network traffic analysis, further proving the performance of our proposition.

Scenario 1 MNIST: We conduct ten experimental setups following the common one-vs-all AD testing protocol. In each setup, one MNIST digit is considered nominal, and the rest nine categories are outliers. In each setup, we fix the ratio of outliers γ_o contaminating the training data to $\gamma_o = 0.1$. These outliers are selected randomly from all nine anomaly classes in each setup. To assess our approach capacity to generalize to unseen anomalous classes, we run further experiments where we vary the number of anomaly classes k_l contaminating the training dataset. These classes are randomly selected from the nine anomalous classes of MNIST.

Scenario 2 MNIST-EMNIST: In the previous scenario, the nominal class contains only one category. To validate our approach with more diverse nominal data, we consider in this setting all MNIST classes as inliers and EMNIST letters as outliers. To study the impact of training anomaly ratio γ_p on the performances, we vary $\gamma_p \in \{0.01, 0.05, 0.1, 0.2\}$.

Scenario 3 NSL-KDD: To validate our approach flexibility with different type of data, we conduct further experiments on the benchmark dataset NSL-KDD. This dataset is lower-dimensional than previous datasets, and contains a mixed type of data, with both categorical and numerical features.

5.3 Competing Methods

We compare our approach against several unsupervised anomaly detectors. Particularly, we consider common conventional AD methods frequently used in the literature: OCSVM with a Gaussian kernel, which is identical to SVDD, and the well-known Isolation Forest. Regarding baselines, we compare our approach against vanilla VAE and vanilla Planar Flow, Deep Autoencoding Gaussian Mixture Model (DAGMM) Zong et al. [2018], and RVAE Akrami et al. [2019]. We fine-tune the hyperparameters of these benchmarks by selecting the optimal AUROC on the validation subset.

5.4 Training Parameter Settings

In all experiments, we use symmetric autoencoders, with the standard Multilayer Perceptron (MLP) feed-forward architectures. In scenarios 1 and 2, the encoder is composed of a 6-layer MLP with 784-400-200-100-50-2 units. In Section 5.6, we investigate our approach sensitivity with respect to the bottleneck dimension. For non-image datasets, the encoder is a 3-layer MLP with 122-32-16-2 units. We use an adaptive learning rate: initially, we use a learning rate of 0.001, which is divided by two if the training loss does not decrease after 20 consecutive epochs. We stop the training when the learning rate is lower than 10^{-6} or the number of epochs becomes higher than 500 epochs. We use a batch size of 256 in the first scenario and of 512 for the second and the third scenarios. We randomly split the training data into 80% training subset and 20% validation subset. The validation subset is used to tune competing method hyperparameters. Our approach comprises two hyperparameters: the initial threshold u, and the risk parameter q. In all experiments, u is fixed to the 80% quantile of the data, and q to 0.01. We conduct a separate sensitivity analysis experiment in Section 5.6, to assess our approach robustness regarding the hyperparameter q. We initialize model parameters randomly with random seeds. To limit the impact of random parameter initialization, we repeat each experiment five times and average the results over these five runs. The experiments were run on a laptop equipped with a 12-core Intel i7-9850H CPU clocked at 2.6GHz and with NVIDIA Quadro P2000 GPU.

5.5 Experimental Results and Discussion

5.5.1 Scenario 1 Experimental Results

The average results over the AD setups of scenario 1 are presented in Table 1. Detailed results for each AD setup are reported in Appendix A.3. We observe that our approach globally outperforms competing methods, for both AUROC and AUPRC metrics. It is noteworthy that VAE-based and NF-based approaches report close results, with only a slight difference. Without loss of generality, we analyze our approach performance for one AD setup, where the digit seven is nominal and other digits are deemed anomalous. To analyze the robustness of our approach compared to the baseline PF, we illustrate in Figure 1 the test AUROC evolution at the end of each training epoch. We find that vanilla PF AUROC gradually increases to reach its maximum at 40 epochs. At this stage, the model focuses on learning a global representation of the common patterns in the training data, dominated by the majority of inlier samples. Here, the model does not overfit the minority of training outliers, and it can successfully discriminate between nominal and anomalous data. Then, AUROC starts decreasing, and the model overfits on outliers. This result is conform to previous studies Arpit et al. [2017], wherein they state that neural networks prioritize learning common simple patterns first. Unlike vanilla PF, our approach AUROC stabilizes around 93%, and overfitting to outliers is limited thanks to our rejection strategy.

Table 1: The experimental results of scenario 1. Results are in percentages and averaged over five runs. The optimal performance in each experiment is in bold.

Metric	OSVM	IF	VAE	NF	RVAE	DAGMM	GRAnD-VAE	GRAnD-NF
AUROC	75.8	75.4	77.2 ± 0.3	76.1 ± 0.3	82.9 ± 0.3	79.3 ± 1.1	87.5 ± 0.5	**88.7 ± 0.5**
AUPRC	96.9	97.0	96.7 ± 0.1	96.4 ± 0.1	97.4 ± 0.1	95.5 ± 0.1	97.5 ± 0.2	**98.1 ± 0.1**

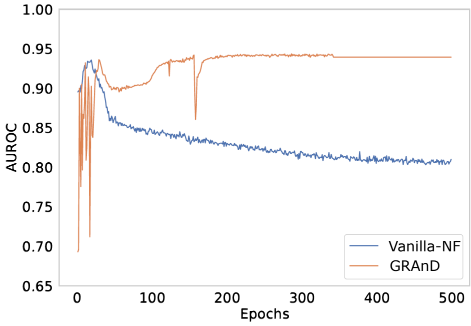

Figure 1: Variation of AUROC for our approach, GRAnD, compared to vanilla NF at the end of each epoch of training.

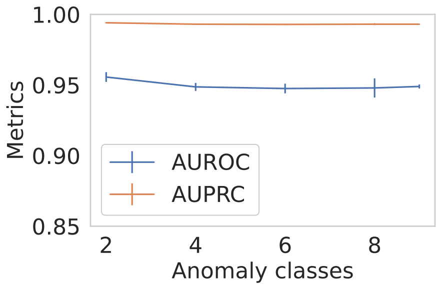

Figure 2: Performance of our NF-based approach with respect to the number of training anomaly classes k_l.

Figure 2 presents the variation of our NF-based approach, with different numbers of anomaly classes k_l corrupting the training data. We note that our approach can generalize to unseen anomalous classes, with a limited subset of training outlier classes. The model can take advantage of the limited training outliers to learn robust patterns that can be generalized to other unseen anomalies.

5.5.2 Scenarios 2 and 3 Experimental Results

Results of scenario 2 are presented in Table 2. While the performance of competing methods decreases with higher pollution ratios γ_p, our approach is more stable. These results mainly highlight the benefit of the robust rejection strategy, where no prior knowledge about the outlier ratio is required in advance. Table 3 depicts scenario three experimental results. We observe that our approach yields competitive results with the low-dimensional network traffic dataset. On average, Our approach mean AUROC is 94%, which is around 3% points better than RVAE.

Table 2: Scenario 2 experimental results. Results are in percentages and averaged over five runs. The optimal performance in each experiment is in bold. γ_p is the ratio of outliers in the training data.

γ_p	Metric	OSVM	IF	VAE	NF	RVAE	DAGMM	GRAnD-VAE	GRAnD-NF
0.05	AUROC	87.4	88.1	88.6 ± 0.6	86.2 ± 0.2	91.4 ± 0.7	83.5 ± 0.4	**96.9 ± 0.2**	94.6 ± 0.7
	AUPRC	89.9	90.4	93.1 ± 0.3	91.7 ± 0.3	93.9 ± 0.5	88.2 ± 0.4	95.0 ± 0.9	**95.8 ± 0.8**
0.1	AUROC	89.2	88.3	85.9 ± 0.6	86.1 ± 0.3	91.3 ± 0.1	79.5 ± 0.8	95.6 ± 0.3	**97.0 ± 0.5**
	AUPRC	91.1	90.5	89.5 ± 0.4	89.9 ± 0.3	93.7 ± 0.1	84.8 ± 0.6	**98.3 ± 0.7**	98.2 ± 0.5
0.15	AUROC	88.5	92.3	85.0 ± 0.2	84.9 ± 0.2	88.2 ± 0.2	78.2 ± 0.3	94.0 ± 0.5	**96.3 ± 0.8**
	AUPRC	90.3	94.1	88.7 ± 0.1	90.9 ± 0.3	92.6 ± 0.1	82.0 ± 0.5	96.4 ± 0.7	**97.4 ± 1.0**
0.2	AUROC	87.6	89.1	83.9 ± 0.2	83.5 ± 0.4	87.1 ± 0.4	75.8 ± 0.2	92.8 ± 1.1	**94.1 ± 3.2**
	AUPRC	89.7	91.2	90.1 ± 0.1	89.9 ± 0.4	91.9 ± 0.3	80.9 ± 0.7	96.0 ± 0.6	**96.8 ± 2.2**

Table 3: Scenario 3 experimental results. Results are in percentages and averaged over five runs. The optimal performance in each experiment is in bold. γ_p is the ratio of outliers in the training data.

γ_p	Metric	OSVM	IF	VAE	NF	RVAE	DAGMM	GRAnD-VAE	GRAnD-NF
0.05	AUROC	84.3	84.5	85.9 ± 1.8	87.5 ± 2.9	92.5 ± 0.1	91.8 ± 1.1	93.4 ± 1.3	**93.7 ± 0.3**
	AUPRC	90.3	90.4	89.2 ± 1.0	89.5 ± 3.4	94.1 ± 0.1	91.1 ± 1.9	94.9 ± 0.1	**95.0 ± 0.5**
0.1	AUROC	83.2	77.9	85.1 ± 1.2	84.8 ± 2.3	91.0 ± 0.1	92.2 ± 1.6	**94.3 ± 0.3**	94.0 ± 0.1
	AUPRC	89.7	88.2	87.1 ± 0.5	86.5 ± 2.7	93.0 ± 0.1	90.1 ± 2.6	**95.5 ± 0.1**	95.4 ± 0.1
0.15	AUROC	82.9	85.4	82.2 ± 1.4	81.9 ± 1.6	91.0 ± 0.4	87.2 ± 2.3	**94.4 ± 0.1**	93.8 ± 0.2
	AUPRC	89.1	91.2	84.4 ± 0.8	83.9 ± 2.0	93.1 ± 0.3	85.5 ± 1.8	**95.6 ± 0.2**	95.2 ± 0.1
0.2	AUROC	80.3	84.5	82.1 ± 1.3	81.9 ± 1.6	90.9 ± 0.2	85.8 ± 2.8	**94.4 ± 0.2**	94.1 ± 0.2
	AUPRC	86.5	90.4	84.3 ± 1.5	83.9 ± 2.0	93.0 ± 0.1	81.9 ± 2.5	**95.6 ± 0.2**	95.5 ± 0.2

5.6 Sensitivity Analysis

We conduct further experiments to assess the sensitivity of our approach regarding two parameters: (i) q the value at risk, (ii) the dimension d of the bottleneck (embedding). We consider the same settings as in scenario one, where 7 images are nominal and all other digits are anomalies. Here, the ratio of training outliers is fixed to $\gamma_p = 0.1$. We train different models with distinct parameters and we evaluate their performance on the same test subset. In Figure 3, we analyse our approach sensitivity analysis with respect to two parameters: the parameter of risk q and the dimension of the latent space d. These results demonstrate that our approach is robust against variations of these two hyperprameters. The AUROC slightly decreases with 1.5% when the parameter of risk q varies from 10^{-4} to 10^{-2}. For all the embedding dimensions d, the AUROC is at least as high as 89%.

6 Conclusion and Future Work

In this paper, we proposed GRAnD, a robust generative method for unsupervised anomaly detection. Our approach uses Extreme Value Theory to filter out outliers contaminating the training data. These contaminants are explicitly taken into account in our model to learn a robust representation, where inliers can be accurately reconstructed, while outlier reconstructions are corrupted. Extensive experiments were conducted on benchmark public datasets, and showed that our approach outperforms classical anomaly detection methods. Moreover, we showed that our approach can generalize to unseen anomalies, even with a small fraction of training outliers. Our contribution opens new horizons for future research. Future research should further develop the rejection strategy to enhance the robustness of our method. It will be important to investigate the stability of the unlabeled samples, $\mathbb{U}$, during the training. In addition, since our contribution involves minimal change to he underlying model architecture, future studies could fruitfully explore other generative models, such as adversarial autoencoders and GANs.

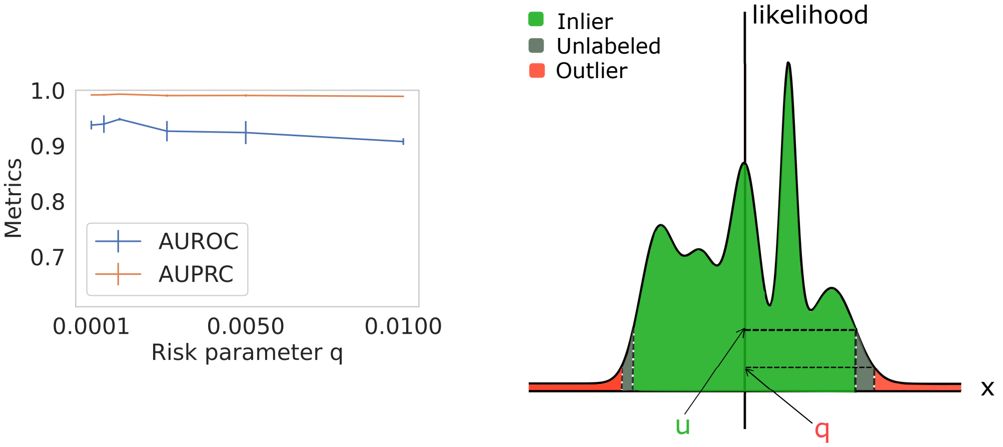

Figure 3: Our approach sensitivity with respect to the parameter of risk q (left) and the latent space dimension d (right).

References

Haleh Akrami, Anand A. Joshi, Jian Li, Sergul Aydore, and Richard M. Leahy. Robust Variational Autoencoder. *arXiv:1905.09961*, 2019.

Jinwon An and S. Cho. Variational autoencoder based anomaly detection using reconstruction probability. 2015.

Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A Closer Look at Memorization in Deep Networks. *ICML*, 2017.

A. A. Balkema and L. de Haan. Residual Life Time at Great Age. *The Annals of Probability*, 2(5): 792 – 804, 1974.

A Basu. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85: 549–559, 1998.

M. Breunig, H. Kriegel, R. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. 2000.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41:1–58, 2009.

Siddharth Chaurasia, Sagar Goyal, and Manish Rajput. Outlier Detection Using Autoencoder Ensembles: A Robust Unsupervised Approach. In *2020 International Conference on Contemporary Computing and Applications (IC3A)*, pages 76–80, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EMNIST: an extension of MNIST to handwritten letters. 2017.

M. J. Desforges, P. J. Jacob, and J. E. Cooper. Applications of probability density estimation to the detection of abnormal conditions in engineering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 212(8):687–703, 1998.

Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. A Survey on GANs for Anomaly Detection. *arXiv:1906.11632*, 2019.

Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *CoRR*, 2015.

Rémi Domingues, Maurizio Filippone, Pietro Michiardi, and Jihane Zouaoui. A comparative evaluation of outlier detection algorithms: Experiments and analyses. *Pattern Recognition*, 74: 406–421, 2018.

Simao Eduardo, Alfredo Nazabal, Christopher K. I. Williams, and Charles Sutton. Robust variational autoencoders for outlier detection and repair of mixed-type data. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 4056–4066, 2020.

Futoshi Futami, Issei Sato, and Masashi Sugiyama. Variational Inference based on Robust Divergences. *AISTATS*, 2018.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, 2014.

D.M. Hawkins. *Identification of Outliers*. Monographs on applied probability and statistics. 1980.

Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving Variational Inference with Inverse Autoregressive Flow. *29th Conference on Neural Information Processing Systems ,*, 2016.

Diederik P. Kingma, Tim Salimans, and M. Welling. Improved variational inference with inverse autoregressive flow. *ArXiv*, 2017.

Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 444–452, 2008.

Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Robust subspace recovery layer for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2020a.

Chieh-Hsin Lai, Dongmian Zou, and Gilad Lerman. Novelty Detection via Robust Variational Autoencoding. *arXiv:2006.05534 [cs, stat]*, 2020b.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery, 2018.

M. M. Moya, M. W. Koch, and L. D. Hostetler. One-class classifier networks for target recognition applications. *NASA STI/Recon Technical Report N*, 93, 1993.

NSL-KDD. NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB.

A. Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. *arXiv:1505.05770 [cs, stat]*, 2016.

Lukas Ruff, Robert A Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. DEEP SEMI-SUPERVISED ANOMALY DETECTION. page 23, 2020.

Artem Ryzhikov, Maxim Borisyak, Andrey Ustyuzhanin, and Denis Derkach. Normalizing flows for deep anomaly detection. *arXiv:1912.09323 [cs, stat]*, 2019.

Bernhard Schölkopf, Robert C Williamson, Alex J Smola, John Shawe-Taylor, and John C Platt. Support Vector Method for Novelty Detection. page 7.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the Support of a High-Dimensional Distribution. *Neural Computation*, 13(7):1443–1471, 2001.

Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly Detection in Streams with Extreme Value Theory. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1067–1075, Halifax NS Canada, 2017. ACM. ISBN 978-1-4503-4887-4.

David M.J. Tax and Robert P.W. Duin. Support Vector Data Description. *Machine Learning*, 54: 45–66, 2004.

Srikanth Thudumu, Philip Branch, Jiong Jin, and Jugdutt (Jack) Singh. A comprehensive survey of anomaly detection techniques for high dimensional big data. *Journal of Big Data*, 7, 2020.

Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Xingwei Yang, Longin Jan Latecki, and Dragoljub Pokrajac. Outlier Detection with Globally Optimal Exemplar-Based GMM. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pages 145–154, 2009.

Chong Zhou and Randy C. Paffenroth. Anomaly Detection with Robust Deep Autoencoders. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '17*, 2017.

Danyang Zhuo, Monia Ghobadi, Ratul Mahajan, Klaus-Tycho Förster, Arvind Krishnamurthy, and Thomas Anderson. Understanding and Mitigating Packet Corruption in Data Center Networks. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, pages 362–375, 2017.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep Autoencoding Gaussian Mixture Model For Unsupervised Anomaly Detection. *ICLR*, page 19, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] A sensitivity analysis regarding the choice of the hyperparameters is presented in Section 5.6
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section3
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] All the datasets used in our experiments are public and accessible online.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Implementation details and parameter settings are detailed in Section5.4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Each experiment si repeated five times (i.e., with five different seeds) and the results are averaged over these five repetitions. Error bars are reported in Tables 1,2,3
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 5.4

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We build our implementation with the open-source library, Pytorch v1.8.1. See Section 5.4
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We conducted experiments with public benchmark datasets, commonly used in AD.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use crowdsourcing or human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

A.1 Details on the threshold computation with EVT

Let $(X_1, X_2, \dots, X_n)$ be a sequence of n independent and identically-distributed random variables. Let F_u be their conditional excess distribution function, i.e., $F_u(x) = P(X - u < x | X > u)$, where u is a high threshold. We aim to find a threshold t such that

$$P(X - u > t, X > u) = q. \tag{16}$$

Let n be the total number of observations, and N_t be the number of X_i exceeding the threshold u, $X_i > u$. We have,

$$P(X - u > t | X > u) = \frac{P(X - u > t, X > u)}{P(X > u)} = \frac{q}{\frac{N_t}{n}} = \frac{qn}{N_t}. \tag{17}$$

According to Pickands-Balkema-De-Hans theorem, we have,

$$P(X - u > x | X > u) \to 1 - G_{\tilde{\xi},\tilde{\sigma}}(x), \text{ as } u \to \infty, \tag{18}$$

where $G_{\tilde{\xi},\tilde{\sigma}}$ is the Generalized Pareto Distribution (GPD) parametrized by two parameters, $\tilde{\xi}$ and $\tilde{\sigma}$. Particularily, when $\xi \neq 0$,

$$G_{\tilde{\xi},\tilde{\sigma}}(x) = 1 - (1 + \frac{\tilde{\xi}x}{\tilde{\sigma}})^{-\frac{1}{\tilde{\xi}}}. \tag{19}$$

By combining equations 17 and 18, we substitute x by $t - u$ and we get,

$$\frac{qn}{N_t} \sim 1 - G_{\tilde{\xi},\tilde{\sigma}}(t - u) = (1 + \frac{\tilde{\xi}(t-u)}{\tilde{\sigma}})^{-\frac{1}{\tilde{\xi}}}. \tag{20}$$

That is,

$$(\frac{qn}{N_t} - 1)^{-\tilde{\xi}} \sim \frac{\tilde{\xi}(t-u)}{\tilde{\sigma}}. \tag{21}$$

Consequently, we get,

$$t \sim u + \frac{\tilde{\sigma}}{\tilde{\xi}}(\frac{qn}{N_t} - 1)^{-\tilde{\xi}}. \tag{22}$$

A.2 Algorithm pseudo-code

Let X be the training data, $Epochs$ be the total number of training epochs, L_R be the learning rate, and ϵ be the stop criterion.

Algorithm 1: GRAnD training algorithm

```
Input :
    $X \in \{x_1, ..., x_n\}$, Epochs, $\epsilon$, $L_R$, $q$
Intialization:
    $Epochs = 500$, $L_R = 0.001$, $\epsilon = 10^{-6}$, $q = 0.01$, $step = 0$
    Random initialization of GRAnD parameters
    $\mathbb{L} = X$, and $\mathbb{S} = \mathbb{U} = empty$
while (step $\leq$ Epochs) and ( $L_R \geq \epsilon$ ) do
    if the data are binary then
        Optimize equations 11 and 12 for GRAnD-VAE or 13 and 14 for GRAnD-NF.
    else
        Consider the continuous case 4.2.
    end
    Compute the thresholds $u$ as the 80% quantile of the data log-likelihoods and $t$ using
      equation 8.
    Update $\mathbb{L}$, $\mathbb{S}$ and $\mathbb{U}$.
    Update the learning rate:
    if the training loss does not decrease after 20 consecutive epochs then
        $L_R \leftarrow \frac{L_R}{2}$
    end
    step $\leftarrow$ step + 1
end
```

A.3 Complete table of scenario 1 experimental results

Table 4: Complete table of scenario 1 experimental results. Results are in percentages and averaged over five runs. The optimal performance in each experiment is in bold. γ_p is the ratio of outliers in the training data.

C_i	Metric	OSVM	IF	VAE	NF	RVAE	DAGMM	GRAnD-VAE	GRAnD-NF
0	AUROC	79.1	81.2	74.0 ± 0.5	73.3 ± 0.1	91.1 ± 1.6	80.7 ± 2.2	92.5 ± 0.9	**96.3 ± 2.8**
	AUPRC	97.6	98.0	96.3 ± 0.1	96.0 ± 0.2	98.3 ± 0.7	96.0 ± 0.4	98.2 ± 1.7	**98.9 ± 1.0**
1	AUROC	92.2	95.9	98.1 ± 0.2	98.3 ± 0.1	**99.6 ± 0.1**	96.1 ± 0.2	99.2 ± 0.6	98.6 ± 1.2
	AUPRC	99.1	99.5	99.6 ± 0.1	99.5 ± 0.2	99.2 ± 0.1	**99.9 ± 0.0**	99.8 ± 0.2	99.4 ± 0.1
2	AUROC	65.4	65.9	66.6 ± 1.7	63.2 ± 0.6	73.5 ± 0.9	75.6 ± 1.3	79.2 ± 0.7	**80.7 ± 0.5**
	AUPRC	95.3	95.4	94.8 ± 0.4	93.8 ± 0.2	95.6 ± 0.2	96.1 ± 0.2	**96.9 ± 1.1**	96.8 ± 0.1
3	AUROC	70.8	70.4	68.9 ± 0.6	71.3 ± 1.4	79.6 ± 0.1	79.5 ± 1.3	82.6 ± 2.7	**83.1 ± 2.5**
	AUPRC	96.3	96.2	95.4 ± 0.1	95.8 ± 0.3	96.9 ± 0.1	96.1 ± 0.2	97.3 ± 0.7	**97.4 ± 0.5**
4	AUROC	79.5	75.0	79.6 ± 0.6	76.8 ± 0.7	81.5 ± 0.1	72.2 ± 5.2	85.3 ± 2.6	**89.9 ± 0.7**
	AUPRC	97.0	96.9	97.0 ± 0.1	96.5 ± 0.1	97.1 ± 0.1	85.4 ± 0.8	97.4 ± 1.0	**98.7 ± 0.1**
5	AUROC	66.2	67.8	72.6 ± 0.5	70.5 ± 1.6	75.6 ± 2.3	81.6 ± 1.0	81.4 ± 2.2	**82.2 ± 1.9**
	AUPRC	96.1	96.3	96.1 ± 0.1	96.0 ± 0.5	96.8 ± 0.4	97.0 ± 0.1	96.9 ± 0.4	**97.1 ± 0.4**
6	AUROC	78.0	75.8	79.8 ± 1.2	75.8 ± 0.8	84.7 ± 0.6	75.7 ± 2.0	98.4 ± 2.3	**91.8 ± 1.0**
	AUPRC	97.5	97.2	97.5 ± 0.2	96.5 ± 0.2	96.9 ± 0.1	96.1 ± 0.5	97.0 ± 0.4	**97.3 ± 0.1**
7	AUROC	81.0	78.8	82.2 ± 0.4	82.1 ± 1.5	88.1 ± 1.0	77.0 ± 0.3	90.6 ± 1.2	**92.3 ± 0.2**
	AUPRC	97.7	97.5	97.5 ± 0.1	97.3 ± 0.3	98.0 ± 0.2	96.2 ± 0.1	98.8 ± 0.2	**99.0 ± 0.2**
8	AUROC	66.3	66.4	70.3 ± 2.0	68.9 ± 0.9	70.1 ± 0.1	72.4 ± 9.4	74.6 ± 0.9	**79.0 ± 0.4**
	AUPRC	95.7	95.7	95.0 ± 0.3	95.1 ± 0.2	95.3 ± 0.1	95.7 ± 1.7	95.5 ± 0.2	**97.0 ± 0.6**
9	AUROC	79.8	77.5	80.3 ± 0.4	80.9 ± 0.8	85.5 ± 1.0	82.7 ± 1.0	**91.1 ± 0.2**	92.7 ± 1.2
	AUPRC	97.6	97.3	97.7 ± 0.1	97.3 ± 0.1	97.6 ± 0.1	97.0 ± 0.1	98.0 ± 0.1	**99.0 ± 0.2**