A Radon–Nikodým Perspective on Anomaly Detection: Theory and Implications

SHLOK MEHENDALE, CSIS, BITS Pilani KK Birla Goa Campus, India
ADITYA CHALLA, CSIS, BITS Pilani KK Birla Goa Campus, India
RAHUL YEDIDA, Lexis Nexis, USA
SRAVAN DANDA, CSIS, BITS Pilani KK Birla Goa Campus, India
SANTONU SARKAR, CSIS, BITS Pilani KK Birla Goa Campus, India
SNEHANSHU SAHA, CSIS, BITS Pilani KK Birla Goa Campus, India

Which principle underpins the design of an effective anomaly detection loss function? The answer lies in the concept of Radon–Nikodým theorem, a fundamental concept in measure theory. The key insight from this article is – Multiplying the vanilla loss function with the Radon–Nikodým derivative improves the performance across the board. We refer to this as RN-Loss. We prove this using the setting of PAC (Probably Approximately Correct) learnability.

Depending on the context a Radon–Nikodým derivative takes different forms. In the simplest case of supervised anomaly detection, Radon–Nikodým derivative takes the form of a simple weighted loss. In the case of unsupervised anomaly detection (with distributional assumptions), Radon–Nikodým derivative takes the form of the popular cluster based local outlier factor.

We evaluate our algorithm on 96 datasets, including univariate and multivariate data from diverse domains, including healthcare, cybersecurity, and finance. We show that RN-Derivative algorithms outperform state-of-the-art methods on 68% of Multivariate datasets (based on F-1 scores) and also achieves peak F1-scores on 72% of time series (Univariate) datasets.

1 INTRODUCTION

Anomaly detection is the process of identifying rare yet significant deviations from normal patterns. This has become essential in various domains such as finance, healthcare, and cyber-security, where undetected anomalies can lead to catastrophic consequences. Moreover, when detected, anomalies can provide significant value. Despite its practical importance, the diversity of real-world settings hinders a unified theoretical treatment. Few aspects which affect the good theoretical framework are

(a) Supervised vs Unsupervised Anomaly Detection: One may or may not have access to sample anomalies in practical settings. If a sample of *labeled anomalies* is provided, it is referred to as supervised anomaly detection. Else it is referred to as unsupervised anomaly detection. **Remark:** An important subtlety here is that – One might have anomalies in the sample but they are not labeled. This is also conventionally categorized as unsupervised anomaly detection. In some settings, a small number of labeled anomalies (or partially labeled data) are available — this is often referred to as semi-supervised or weakly supervised anomaly detection ([28],[30])

(b) Percentage of Anomalies in the train set: It is intuitively clear that if the percentage of anomalies in the train set is higher, it makes anomaly detection simpler. However, different algorithms are usually preferred at different thresholds of anomalies.

(c) Dimensionality of the dataset: As with most of machine learning algorithms, dimension plays an important role in the ability to find the right hypothesis function. Simple univariate anomaly detection can be performed using classical statistical measures such as standard deviations and/or quantiles. Multivariate anomaly detection is significantly harder.

(d) Time Series or not: Time series anomaly detection adds an additional layer of complexity. Specifically evaluation is impacted in this setting.

Motivation of the present work: To our knowledge, there is no unifying principle which spans all the above mentioned aspects in anomaly detection. The aim of this article is to provide a simple foundational principle which can assist in all the settings above. Specifically, using the PAC learning framework we propose a simple principle – **Multiplication with the Radon–Nikodým derivative improves the performance**. And since PAC learning is a broad framework, this insight spans all the above mentioned aspects.

1.1 Contributions

The key contribution of this paper is an overarching technique for anomaly detection irrespective of the type of supervision, frequency of anomalies, size of the dataset or dimensionality. The technique is based on an elegant yet simple mathematical framework which connects both supervised and unsupervised anomaly detection paradigms without requiring changes in model architecture or optimization procedures. It is important to observe that the elegance lies in correcting the distributional differences between the training and evaluation distributions (Details are provided in Section 3).
Theoretical contributions: We

(A) Introduce a weighted loss function based on the Radon–Nikodým derivative (termed "RN-Loss" in this paper), tailored for supervised anomaly detection. This implies, one can estimate/approximate the Radon–Nikodým derivative using class dependent weights, resulting in the weighted loss function.
(B) Show that, in the context of unsupervised anomaly detection, popular and time-tested algorithms such as cluster based local outlier factor (CBLOF) and its variants can be derived using Radon–Nikodým derivative based correction. This correction enables using the same framework in unsupervised anomaly detection.
(C) Introduce the problem of (PAC-)learnability of anomaly detection problem in Section 2.
(D) Show that anomaly detection is indeed PAC learnable in Section 3.

The theoretical contributions and derivation provide the foundational principles and significant practical insights. We state them below and leverage these crucial facts throughout the remainder of the article.

- Product of the original loss function and the Radon–Nikodým derivative improves the loss function. This is theoretically demonstrated in Section 3.1 and empirically validated in Section 4.
- Radon–Nikodým derivative offers a mathematically grounded abstraction for designing the loss functions for anomaly detection. (See Section 3)
- Depending on the context (supervised/unsupervised), the Radon–Nikodým derivative can take different forms. (Derivations in Section 3.2)

Empirical Contributions: The proposed RN-Loss framework has several practical advantages

1. RN-Loss maintains *computational efficiency* by building on base loss functions like Binary Cross-Entropy,
2. RN-Loss can be readily incorporated into existing training pipelines, requiring no changes to model architectures or optimization procedures.
3. Unsupervised methods like dBTAI [33] benefit from using a modified version of RN-Loss. Using RN-Loss makes it capable of identifying anomalies even when the model is trained solely on normal data.

4. The loss function also demonstrates *flexibility*, fitting varied data distributions such as Weibull and Log-normal without requiring structural changes.

In summary, these properties make RN-Loss a robust and adaptable solution for anomaly detection, offering improved performance metrics, computational efficiency, and versatility across a wide range of real-world applications.

Empirical Performance: The RN-Loss function delivers significant improvements in anomaly detection, offering both *enhanced performance* and *broad adaptability*. It surpasses prior state-of-the-art (SoTA) methods, improving F1 scores on 68% and Recall on 46% of the multivariate datasets, with similar trends observed in univariate time-series data (F1: 72%, Recall: 83%). These results highlight its *consistent effectiveness across diverse benchmarks*.

Our experiments further demonstrate that RN-Loss substantially enhances the performance of unsupervised anomaly detection methods—*specifically the vanilla implementations of CBLOF and ECBLOF [32] when integrated with clustering algorithms such as K-Means [9] and dBTAI [33]*. The enhanced KMeans-CBLOF configuration achieved superior results on 93% of univariate datasets (27 out of 29) and on 48% of multivariate datasets (32 out of 67), relative to the original version. Although dBTAI previously achieved SoTA performance, its evaluation metrics—particularly precision—were inflated. By incorporating RN-Loss, these metrics were better calibrated, and the modified dBTAI maintained or improved overall performance across 59 multivariate datasets, while showing increased recall on nearly all univariate datasets. Figure 1 provides a visual summary the results.

2 PROBLEM SETUP AND NOTATION

Let $\mathcal{X} \subset \mathbb{R}^d$ denote the feature space. We assume that the sample is obtained from a *mixture of inline and anomaly distributions*. Let D_I denote the inline distribution and D_A denote the anomaly distribution. If the *contamination ratio* or *anomalous ratio* is given by α, the sample is obtained from

$$D^\alpha = (1-\alpha)D_I + \alpha D_A \tag{1}$$

Note that support of both D_I and D_A distributions is assumed to be $\mathcal{X}$. As stated before, anomaly detection can either be supervised or unsupervised.

Supervised Anomaly Detection: In the supervised case, we assume to also have access to labels, $\mathcal{Y} = \{0, 1\}$ where 1 indicates that the sample belongs to the inline distribution D_I and 0 indicates that the sample belongs to the anomaly distribution D_A.

We denote the joint distribution of n i.i.d samples from a given distribution D using D^n. If $S := \{(\boldsymbol{x}^i, y^i)\}$ denotes the sample of size n drawn i.i.d from D^α, this is assumed to have the joint distribution $D^{\alpha,n}$.

Given a sample of points $S \sim D^{\alpha,n}$ the aim is to obtain a classifier f such that, for any sample $\boldsymbol{x}$ from D^α : (i) if $\boldsymbol{x}$ is sampled from D_A then identify it as *anomaly* and (ii) if $\boldsymbol{x}$ is sampled from D_I identify it as *non-anomaly* or *inline*. Note that this becomes a binary classification problem

Unsupervised Anomaly Detection: Given a sample of points $S := \{(\boldsymbol{x}^i)\}$ drawn i.i.d from D^α, the aim is to obtain a classifier f such that, for any sample $\boldsymbol{x}$: (i) if $\boldsymbol{x}$ is sampled from D_A then identify it as *anomaly* and (ii) if $\boldsymbol{x}$ is sampled from D_I identify it as *non-anomaly*. **Remark:** For the sake of generality, we assume that the sample is obtained from the contaminated distribution D^α instead of the inline distribution D_I.

Space of Distributions: If there is no restriction on the set of possible distributions D_{XY}, no-free-lunch theorem [47] suggests that anomaly detection is impossible. So, we restrict the set of distributions to a set $\mathscr{D}_X$.

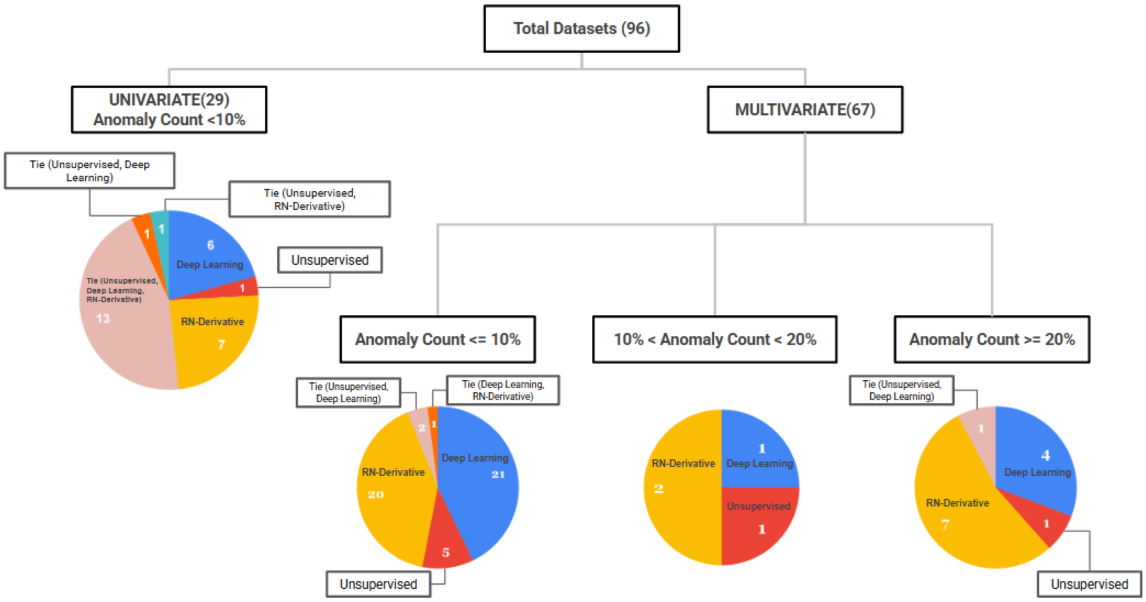


Fig. 1. Comparative Analysis of Anomaly Detection Algorithms. Performance evaluation prioritizes recall, with precision as the secondary metric for tied cases. The comparison spans three algorithm categories: (1) Deep Learning approaches (AutoEncoders, DAGMM, DevNet, GAN, DeepSAD, FTTransformer (current state-of-the-art), and PReNet), (2) Unsupervised methods (LOF, Elliptic Envelope, Isolation Forest, dBTAI, MGBTAI, and quantile-based approaches including q-LSTM variants and QReg), and (3) RN-Derivative algorithms (RN-Net and RN-LSTM). RN-Net outperforms state-of-the-art methods on 68% of Multivariate datasets (based on F-1 scores), while the RN-LSTM + RN-Net combination achieves peak F1-scores on 72% of time series (Univariate) datasets. For detailed numerical comparisons

Hypothesis Space, Loss function and Risks : Let $\mathcal{H} \subset \{h : \mathcal{X} \to \{0,1\}\}$ denote the set of functions from which we choose our classifier f. Any specific $h \in \mathcal{H}$ is referred to as hypothesis function or simply function if the context is clear. We consider 0-1 loss function – $\ell : \mathcal{Y} \times \mathcal{Y} \to \{0,1\}$ where $\ell(y_1, y_2) = 0$ if $y_1 = y_2$ and $\ell(y_1, y_2) = 1$ otherwise.

The risk associated with a hypothesis is defined as the expected loss when the hypothesis is used for identifying anomalies for a given distribution $D \in \mathscr{D}_X$

$$R_D(h) = E_D[\ell(h(\boldsymbol{x}), y)] \tag{2}$$

2.1 PAC Learning Framework

We use the following definition of PAC learning [27] in this article –

PAC Learning: A hypothesis class $\mathcal{H}$ is PAC learnable if there exists (i) an algorithm $\mathbf{A} : \cup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{H}$, (ii) a decreasing sequence $\epsilon(n) \to 0$, and (iii) for all $D \in \mathscr{D}_X$

$$E_{S \sim D^n}[R_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R_D(h)] \leq \epsilon(n) \tag{3}$$

Existing results [44] show that if $\mathcal{H}$ has finite VC dimension, then it is learnable for all possible distributions $\mathscr{D}_{XY}$. Specifically, neural networks are PAC learnable.

An important subtlety in the PAC learning framework is the implicit assumption that the train and evaluation distributions are the same. However, this is not true for anomaly detection.

PAC Learning for Anomaly Detection: Define $\alpha-$risk to quantify the expected loss when *anomaly contamination ratio* is $\alpha \in [0, 1)$. That is, $D^{\alpha} = (1 - \alpha)D_I + \alpha D_A$

$$R_D^{\alpha}(h) = (1 - \alpha)R_{D_I}(h) + \alpha R_{D_A}(h) \tag{4}$$

A hypothesis class $\mathcal{H}$ is PAC learnable for Anomaly Detection if there exists (i) an algorithm $\mathbf{A} : \cup_{n=1}^{\infty}(\mathcal{X}\times\mathcal{Y})^n \rightarrow \mathcal{H}$, (ii) a decreasing sequence $\epsilon(n) \rightarrow 0$, and (iii) for all $D_{XY} \in \mathscr{D}_{XY}$

$$E_{S\sim D^{\alpha,n}}[R_D^{0.5}(\mathbf{A}(S)) - \inf_{h\in\mathcal{H}} R_D^{0.5}(h)] \leq \epsilon(n) \tag{5}$$

Important Remark: In words, while learning is done using the samples are obtained from specific α, the evaluation is fixed at $\alpha = 0.5$. This is because, in practice one is interested in the metrics associated with anomaly class. Thus, convergence with respect to the risks R_D^{α} is not consistent with anomaly detection. Instead, one has to consider the risks $R_D^{0.5}$. As we shall shortly see, this observation allows us to correct the loss function easily using Radon–Nikodým derivative.

3 RN-LOSS : DERIVATION FROM FIRST PRINCIPLES

In this section, we

1. Answer the question – *When is a hypothesis class $\mathcal{H}$ PAC Learnable for Anomaly Detection?*
2. Propose RN-Loss, which is a generic way to design loss functions for anomaly detection. We illustrate this by applying the RN-Loss principle in both supervised and unsupervised context.

3.1 PAC Learnable for Anomaly Detection

Recall the Radon–Nikodým theorem from measure theory.

Theorem 3.1 (Radon–Nikodým [19]). *Let $(\Omega, \mathcal{A})$ be a measurable space with $\mathcal{A}$ as the σ algebra and μ, ν denote two $\sigma-$finite measures such that $\nu << \mu$ (ν is absolutely continuous with respect to μ). Then, there exists a function f such that,*

$$\nu(A) = \int_A f d\mu \quad (or) \quad d\nu = f d\mu \tag{6}$$

where $A \in \mathcal{A}$.

Assumption 1 (Absolute Continuity): For any fixed $\alpha \in (0, 1)$, let ν denote the measure induced by $D^{0.5}$ and μ denote the measure induced by D^{α}. We assume that *ν is absolutely continuous with respect to μ* ($\nu << \mu$). This is a reasonable practical assumption since both $D^{0.5}$ and D^{α} are a mixture of same distributions D_I, D_A.

Assumption 2 (Boundedness): From Assumption 1 and Theorem 3.1 we have that there exists a Radon–Nikodým derivative f relating ν, μ as – $d\nu = f d\mu$. We further assume that this is bounded, i.e $1/b < f < b$ for some b on the support of μ. This is a reasonable assumption as well since both μ, ν represent mixtures of the same underlying distributions.

We then have the following proposition:

Proposition 3.2. *Let ν denote the measure induced by $D^{0.5}_{XY}$ and μ denote the measure induced by D^{α}_{XY}. Also, let $d\nu = f d\mu$ (absolutely continuous) where $1/b < f < b$ for some $b < \infty$ on the support of μ. For all $h \in \mathcal{H}$, there exists a $\Delta_{\mu,\nu}$ such that,*

$$\frac{1}{\Delta_{\mu,\nu}} \leq \frac{R^{0.5}_D(h)}{R^{\alpha}_D(h)} \leq \Delta_{\mu,\nu} \tag{7}$$

Proof. Recall, that

$$R^{0.5}_D(h) = \int \ell(h(\boldsymbol{x}), y) d\nu = \int \ell(h(\boldsymbol{x}), y) f d\mu \quad \textit{and} \quad R^{\alpha}_D = \int \ell(h(\boldsymbol{x}), y) d\mu \tag{8}$$

So, we have

$$\frac{R^{0.5}_D(h)}{R^{\alpha}_D(h)} = \frac{\int \ell(h(\boldsymbol{x}), y) f d\mu}{\int \ell(h(\boldsymbol{x}), y) d\mu} \tag{9}$$

Now, observe that ℓ is taken to be 0-1 loss, f is bounded by $1/b$ and b on the support of μ, and $\int d\mu = 1$ (probability measure). Note that the bound b depends on the distributions μ, ν. Hence, we have

$$\frac{1}{\Delta_{\mu,\nu}} \leq \frac{R^{0.5}_D(h)}{R^{\alpha}_D(h)} \leq \Delta_{\mu,\nu} \tag{10}$$

for some constant $\Delta_{\mu,\nu}$ □

Theorem 3.3. *If $\mathcal{H}$ is PAC Learnable, then $\mathcal{H}$ is PAC Learnable for Anomaly Detection.*

Proof. Assume that $D^{\alpha} \in \mathscr{D}_X$ for all $\alpha \in (0, 1)$. If $\mathcal{H}$ is PAC learnable, we have that there exists

(i) an algorithm $\mathbf{A} : \cup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \longrightarrow \mathcal{H}$
(ii) a decreasing sequence $\epsilon(n) \longrightarrow 0$ such that,

for all $D \in \mathscr{D}_X$

$$E_{S \sim D^n} [R_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R_D(h)] \leq \epsilon(n) \tag{11}$$

Now, from Proposition 3.2, we have

$$R^{0.5}_D(\mathbf{A}(S)) \leq R^{\alpha}_D(\mathbf{A}(S)) \times \Delta_{\mu,\nu} \tag{12}$$

and,

$$\inf_{h \in \mathcal{H}} R^{0.5}_D(h) \geq \frac{1}{\Delta_{\mu,\nu}} \inf_{h \in \mathcal{H}} R^{\alpha}_D(h) \tag{13}$$

Hence,

$$\begin{aligned} R^{0.5}_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R^{0.5}_D(h) &\leq R^{\alpha}_D(\mathbf{A}(S)) \times \Delta_{\mu,\nu} - \frac{1}{\Delta_{\mu,\nu}} \inf_{h \in \mathcal{H}} R^{\alpha}_D(h) \\ &\leq \Delta_{\mu,\nu} \left[R^{\alpha}_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R^{\alpha}_D(h) \right] + \inf_{h \in \mathcal{H}} R^{\alpha}_D(h) \left[-\frac{1}{\Delta_{\mu,\nu}} + \Delta_{\mu,\nu} \right] \end{aligned} \tag{14}$$

Taking expectations on both sides,

$$\begin{aligned} E_{S \sim D^{\alpha,n}} [R^{0.5}_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R^{0.5}_D(h)] \leq \Delta_{\mu,\nu} E_{S \sim D^{\alpha,n}} [R^{\alpha}_D(\mathbf{A}(S)) - \inf_{h \in \mathcal{H}} R^{\alpha}_D(h)] \\ + \left[-\frac{1}{\Delta_{\mu,\nu}} + \Delta_{\mu,\nu} \right] E_{S \sim D^{\alpha,n}} \left[\inf_{h \in \mathcal{H}} R^{\alpha}_D(h) \right] \end{aligned} \tag{15}$$

Using the Realizability Assumption of the PAC learning framework, we have that

$$E_{S\sim D^{\alpha,n}}\left[\inf_{h\in\mathcal{H}} R_D^{\alpha}(h)\right] = 0. \tag{16}$$

Thus, we have that for the same algorithm $\mathbf{A}(S)$ and sequence $\epsilon(n)$,

$$E_{S\sim D^{\alpha,n}}[R_D^{0.5}(\mathbf{A}(S)) - \inf_{h\in\mathcal{H}} R_D^{0.5}(h)] \leq \epsilon(n) \tag{17}$$

Hence $\mathcal{H}$ is PAC learnable for anomaly detection as well. □

The above proof offers some significant insights into learning for Anomaly Detection:

1. The algorithm $\mathbf{A}$ does not have to change significantly. The only change required is to modify the algorithm to suit Equation (5) instead of Equation (3).
2. The modification of $\mathbf{A}$ depends on the Radon–Nikodým derivative. This can be observed from Equation (15), where the constant $\Delta_{\mu,\nu}$ plays a key role in ensuring convergence. Proposition 3.2 and Equation (8) shows that $\Delta_{\mu,\nu}$ is the constant depending on Radon–Nikodým derivative.
3. The hypothesis class $\mathcal{H}$ should be large enough to accommodate all possible distributions D^{α}. However, thanks to recent advances in machine learning, this is not a strong practical restriction. Standard classes such as neural networks and even boosted trees satisfy this assumption.
4. *Impact of Realizability Assumption:* In the standard PAC learning where the realizability assumption can be easily overcome by considering the bayes optimal classifier. However, for anomaly detection realizability assumption becomes crucial for ensuring learnability. This can be observed from Equation (15). Nevertheless, this is not a strong restriction in practice.

3.2 Estimating Radon–Nikodým derivatives in different contexts

We now dive deeper into how the algorithm $\mathbf{A}$ must be adapted for anomaly detection. The overarching principle is to estimate the Radon–Nikodým derivative $f \approx \hat{f}$, so that the loss function $\ell(h(x), y)$ is transformed to $\ell(h(x), y)\hat{f}$. We refer to this as **RN-Loss**.

3.2.1 Case 1: Supervised Anomaly Detection. Recall, D_I and D_A are two probability distributions on a measurable space $\mathcal{X}\times\mathcal{Y}$. Denote their respective densities by $p_I(\boldsymbol{x}, y)$ and $p_A(\boldsymbol{x}, y)$. Recall, for $\alpha \in [0, 1]$, D^{α} is given by $(1-\alpha)D_I + \alpha D_A$. Denote by μ the probability measure induced by D^{α}. Equivalently, μ has density

$$\mu(\boldsymbol{x}, y) = (1-\alpha)p_I(\boldsymbol{x}, y) + \alpha p_A(\boldsymbol{x}, y). \tag{18}$$

We also consider distribution $D^{0.5}$, whose induced measure is denoted ν, with density

$$\nu(\boldsymbol{x}, y) = 0.5 p_I(\boldsymbol{x}, y) + 0.5 p_A(\boldsymbol{x}, y) \tag{19}$$

ν is absolutely continuous with respect to μ $\nu << \mu$. By the Radon–Nikodým theorem, there is a function $f(\boldsymbol{x}, y)$ such that

$$d\nu = f d\mu \quad \text{or} \quad \frac{d\nu}{d\mu}(\boldsymbol{x}, y) = f(\boldsymbol{x}, y). \tag{20}$$

Under the usual condition that $\mu(\boldsymbol{x}, y) > 0$, we obtain

$$f(x, y) = \frac{\nu(\boldsymbol{x}, y)}{\mu(\boldsymbol{x}, y)} = \frac{0.5 p_I(\boldsymbol{x}, y) + 0.5 p_A(\boldsymbol{x}, y)}{(1-\alpha)p_I(\boldsymbol{x}, y) + \alpha p_A(\boldsymbol{x}, y)}. \tag{21}$$

The key idea is to use the empirical distribution function to estimate f. Given a sample $\{\boldsymbol{x}^i, y^i\}$ from D^α where $y^i = 1$ if it belongs to D_I and $y^i = 0$ if it belongs to D_A. From above we have,

$$f(\boldsymbol{x}, +1) = \frac{0.5}{1-\alpha} \quad \textit{and} \quad f(\boldsymbol{x}, 0) = \frac{0.5}{\alpha} \tag{22}$$

Important Remark: Interestingly, due to the setup we have that the weights $0.5/(1-\alpha), 0.5/\alpha$ does not depend on $\boldsymbol{x}$. This is because the Radon–Nikodým derivative includes the *same* distribution both in the numerator and denominator in each of the cases.

Since, constant multiples do not effect the optimization, we make another simplification as follows – Let the weight of the anomalous class D_A to be 1, samples from the inline class D_I are then reweighted by

$$\omega = \frac{\alpha}{1-\alpha} \approx \frac{\#\text{Anomalies}}{\#\text{Inline Samples}} \tag{23}$$

Thus, in the case of supervised anomaly detection one only needs to adjust the weights of the samples when learning the hypothesis h.

3.2.2 Case 2: Unsupervised Anomaly Detection. In the case of unsupervised anomaly detection, one does not have access to sample anomalies. In such cases, one makes additional assumptions such as gaussian distribution to obtain the hypothesis function. The most popular assumption is that the sample is obtained from *mixture of gaussian.* Here, we illustrate how the Radon–Nikodým derivative correction is applied.

Let $\mathcal{X} \subset \mathbb{R}^d$ be the data space. Given a sample $S = \{\boldsymbol{x}_i\}_{i=1}^n$ drawn from the distribution D^α, we aim to assign an anomaly score to each $x \in \mathcal{X}$. Moreover, since we do not have a explicit loss function, one uses a heuristic $h(x)$ to assign scores. In this case we have $\ell(h(\boldsymbol{x}), y) \approx h(x)$ – That is we replace the loss itself with the heuristic scores. Accordingly following our Radon–Nikodým derivative principle we correct the heuristic $h(x)$ with the Radon–Nikodým derivative $f(x)$ to obtain $h(x)f(x)$ as the final scoring rule.

Hypothesis Function h^*: Any point can be considered an anomaly if it lies far-away from the center. Thus, under the assumption of mixture-of-gaussians, we have the following anomaly score

$$h^*(x) = \arg\min_i ||x - \mu_i|| \tag{24}$$

where μ_i denotes the centroid of the ith Gaussian. In words, we consider the distance of the point from the nearest cluster.

Distribution assumption of h^*: Let $\{C_1, C_2, \cdots, C_m\}$ denote the clusters and $p_{C_i}(x)$ denote the distribution of each cluster C_i. We have the following implicit *distribution assumption*

$$D^\alpha = \frac{1}{m}\sum_{i=1}^m p_{C_i}(x) \tag{25}$$

That is, samples from each cluster are equally likely.

Evaluation Distribution of h^*: However, for h^* defined above to be effective, one needs *same scale of the nearest neighbors irrespective of the variance of gaussians.* Thus, to apply (and consequently to evaluate) the above hypothesis, we assume that the distribution is

$$D^{0.5} = \frac{1}{m}\sum_{i=1}^m \gamma_i p_{C_i}(x) \tag{26}$$

where γ_i denotes the proportion of the samples from C_i. For the scale to be equivalent, one must obtain a larger sample from clusters with larger variance.

Correcting this Discrepancy using Radon–Nikodým Derivative: Following the reasoning from above, we correct the discrepancy using the Radon–Nikodým derivative which is given by

$$f(x) = \sum_{i=1}^{m} \gamma_i I[x \in C_i] \tag{27}$$

If we estimate γ_i from the data, we get $\gamma_i \approx |C_i|$.

Final Algorithm for Anomaly Detection: We further also use a "filtering" to ignore small clusters. This leads to the following

1. Cluster the sample into clusters $\{C_1, C_2, \cdots, C_m\}$, obtain the m corresponding means μ_k
2. Consider only the largest $k < m$ clusters to estimate the inline density p_I. This filtering step allows us to estimate p_I.
3. Obtain the base-scores $h^*(x)$ using Equation (24) and the cluster centroids.
4. Adjust the scores using Radon–Nikodým derivative correction.

$$h^\dagger(x) = \begin{cases} |C_i| d(x, C_i), & \text{if } x \in C_i, C_i \text{ is large} \\ |C_i| d(x, C_L), & \text{if } x \in C_i, C_L \text{ is the nearest large cluster} \end{cases} \tag{28}$$

where C_L is the closest large cluster.

This algorithm is the widely adopted cluster based local outlier factor (CBLOF) [9].

We also remark that there exists other modified versions of CBLOF.

(A) ECBLOF (Enhanced Cluster-Based Local Outlier Factor)[32], which assumes

$$D^{0.5} = \frac{1}{m} \sum_{i=1}^{m} p_{C_i}(x) \tag{29}$$

In this case there is no Radon–Nikodým derivative correction required.

(B) Observe that, we estimated $\gamma_i \approx |C_i|$. However, one can use techniques such as kernel density estimate (KDE) to explicitly estimate the Radon–Nikodým derivative.

Key Takeaway: The most important aspect to note here is that – Radon–Nikodým derivative can be applied in wide variety of contexts to obtain practically useful algorithms. Further it also has a flexibility of adapting to various application scenarios depending on the information available.

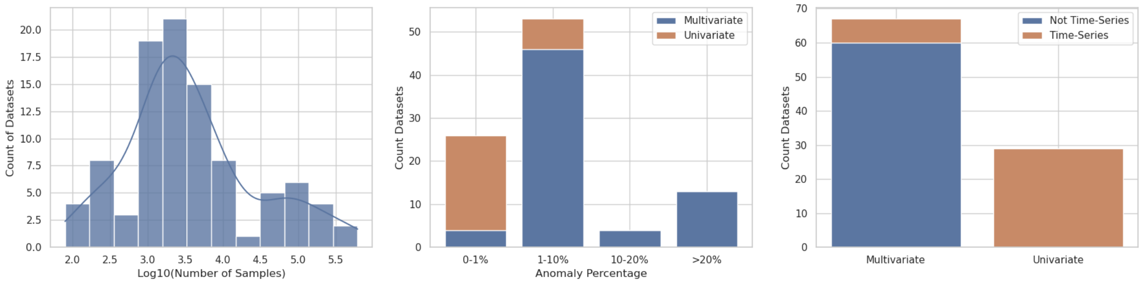
4 EMPIRICAL EVALUATIONS

In this section, we aim to show that **The Radon–Nikodým derivative correction proposed above demonstrates performance that is on par with, and in many cases surpasses, that of several state-of-the-art techniques.**.

We first discuss the experimental settings in detail in Section 4.1, and discuss the observations in Section 4.2.

4.1 Experimental Settings

4.1.1 Datasets for Empirical Validation: As discussed earlier, there exists several factors which effect the practical aspects of anomaly detection. Hence, empirical validation must be conducted on a wide variety of datasets. In this



(a) (b) (c)

Fig. 2. Overview of Datasets: Observe that the datasets considered have wide range of total sizes, anomaly percentages, and as well as diversity with respect to other characteristics such as Univariate vs Multivariate, Time-Series vs Non-Time-Series. (a) shows the number of datasets with different number of samples. Observe that the sizes vary from 80 to 619, 329. (b) Shows the number of datasets with respect to percentage anomalies. We consider 4 ranges corresponding to "very less", "less", "medium" and "large" number of anomalies. (c) indicates number of datasets which has a time-series characteristic.

article, we consider 96 datasets to cover the range of aspects of anomaly detection. Figure 2 shows provides an overview of the datasets considered.

(i) Firstly, we consider both univariate and multivariate datasets. In total we consider 29 univariate datasets and 67 multivariate datasets.
(ii) We consider datasets of different sizes ranging from 80 to 619, 329. Figure 2a shows the distribution of the sizes within the 96 datasets we consider.
(iii) We also consider datasets with wide-ranging anomaly percentages, from 0.03% to 43.51%. However, for simplicity we consider 4 ranges - $0 - 1\%$, $1 - 10\%$, $10 - 20\%$ and $> 20\%$. There are 22 univariate and 4 multivariate datasets in the range $0 - 1\%$, 40 multivariate and 7 univariate datasets in the range $1 - 10\%$, 3 multivariate datasets in the range $10 - 20\%$ and 16 in $> 20\%$ range. Figure 2b illustrates this.
(iv) We also consider the time-series aspect. However, since non time-series univariate datasets are not complex this aspect is not considered. In total, we consider 29 univariate time-dependent and 7 multivariate time-dependent datasets. Figure 2c illustrates this.
(v) Further, datasets cover multiple domains such as finance, healthcare, e-commerce, industrial systems, telecommunications, astronautics, computer vision, forensics, botany, sociology, linguistics, etc.

We source these datasets from a combination of ESA-ADB dataset [20] along with six other SWaT datasets [45] and a BATADAL dataset[43]

4.1.2 Evaluation of Anomaly Detection: We split the datasets into train/test as follows – We use 70% of the inline data for training and 30% of the inline data for testing. However, we only use 15% of the anomalous data for training and 85% of the anomalous data for testing. This procedure is follows to ensure robustness of the results. Further we made sure that the *anomaly contamination*(15%) in the train set is less than or equal to the other baseline methods.

Since anomalies are rare and the datasets are imbalanced we use – *Precision, Recall, AUROC and F1 score* – for all the evaluations.

4.1.3 Baselines and SoTA:. There exists several algorithms for anomaly detection. These algorithms include Local Outlier Factor(LOF), Isolation Forest (IForest), One-class SVM (OCSVM), Autoencoders, Deep Autoencoding Gaussian Mixture Model (DAGMM)[50], Quantile LSTM(q-LSTM)[31], Deep Quantile Regression [42], GNN [6], GAN [35], DevNet [8], MGBTAI[34] and d-BTAI[33] as covered in Table 2 in Appendix B. Above is a mix of supervised and unsupervised methods, forming our baselines for comparison on anomalous datasets. Other important algorithms that have been tested along with the above are ECOD [22], COPOD [21], KNN, LUNAR [2], PCA, DSVDD, NeuTraL-AD [26], ICL [39], SLAD [48].

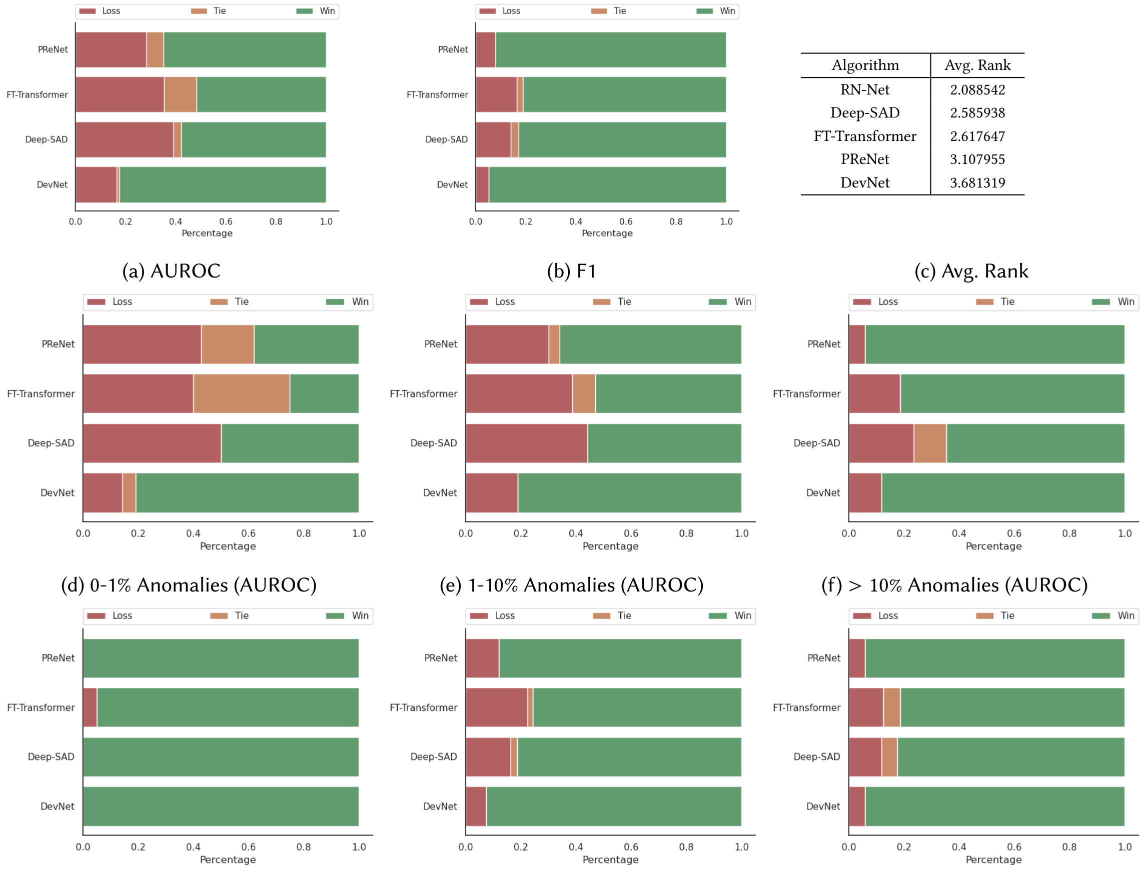
4.1.4 Tuning Hyperparameters: In the domain of anomaly detection, determining optimal threshold values is crucial due to the inherently rare and imbalanced nature of anomalies. Setting the threshold too low may lead to a high number of false positives, reducing the model's significance, and setting it too high may cause the model to miss critical anomalies. Therefore, aiming at the most optimal model performance, we set the following thresholds in accordance with the suggestions from the literature:

- For Autoencoders, the lower threshold is set at the 0.75th percentile, and the upper threshold is at the 99.25th percentile of Mean Squared Error(MSE) values.
- All the data points with discriminator scores less than 10th percentile were considered anomalies for GANs.
- DAGMM had a dual threshold setting with high and low thresholds, with two standard deviations above and below the mean
- For the tree-based approaches, MGBTAI was set to a minimum clustering threshold of 20% of the dataset size and leaf level threshold of 4, while for d-BTAI, the minimum clustering threshold was set to 10% of the dataset size
- For deep quantile regression, the lower threshold was at 0.9th percentile and the upper at 99.1st percentile of the predicted values

As can be observed this process becomes extremely meticulous and takes up most of the experimental time. For RN-Loss, we automate our threshold calculation process by maximizing the difference in True positive and False positive rates, which are very important metrics obtained from the AUC-ROC curve. This helps us get the best optimal threshold, and SoTA results across datasets and overall algorithms. The thresholds range from 0.001 to 0.999 and can be found for all the respective datasets in Appendix B, Table 4.

4.1.5 Network Architecture: In the supervised case – We use two architectures in our study. First, **RN-Net** is a ReLU feedforward network with RN-Loss, comprising 64 hidden units in a binary classification setting. We train for 50 epochs using the Adam optimizer. Additionally, we use batch normalization [14], dropout [40], and early stopping with a threshold of 10 epochs. We reduce our learning rate by half every 5 epochs until it reaches 10^{-6}. Parallel to this, we integrated L2 regularisation with RN-Net and noticed a further step-up in performance across datasets. Similarly, to demonstrate the flexibility and adaptability of RN-Loss, we create **RN-LSTM**: A LSTM with 32 hidden units coupled with the RN-Loss function.

In the unsupervised case, we use 3 different algorithms which are modified using Radon–Nikodým derivative – (i) **Kmeans(CBLOF)** uses K-Means to estimate the clusters and applies CBLOF (ii) **Kmeans(CBLOF, Mod.)** estimates the Radon–Nikodým using kernel density estimates, and (iii) **dBTAI(Mod.)** modies the dBTAI algorithm in [33] using the ECBLOF loss.



(a) AUROC

(b) F1

Algorithm	Avg. Rank
RN-Net	2.088542
Deep-SAD	2.585938
FT-Transformer	2.617647
PReNet	3.107955
DevNet	3.681319

(c) Avg. Rank

(d) 0-1% Anomalies (AUROC)

(e) 1-10% Anomalies (AUROC)

(f) > 10% Anomalies (AUROC)

(g) 0-1% Anomalies (F1)

(h) 1-10% Anomalies (F1)

(i) > 10% Anomalies (F1)

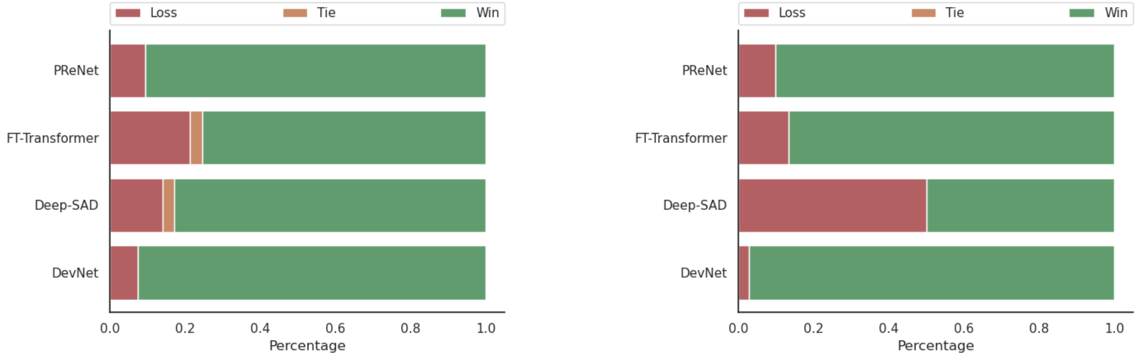
Fig. 3. Performance of RN-Loss for supervised anomaly detection. 🟥 indicates the percentage of the datasets in which the specific algorithm performs better than RN-Net. 🟧 indicates the percentage of the datasets in which the specific algorithm tied with RN-Net. 🟩 indicates the percentage of the datasets in which the specific algorithm performs worse than RN-Net. (a) shows the performance of RN-Loss with respect to AUROC. (b) shows the performance of the RN-Loss with respect to F1-score. (d)-(f) shows the performance of RN-Loss when considering various ranges of anamoly percentages.

Remark: The code and other resources are provided in the the anonymous repo - https://anonymous.4open.science/r/RN_Derivative_Official/. Note that we perform our experiments on 96 different datasets and 23 different algorithms. To maintain clarity and focus, we report only the most salient aspects of the results. Please refer to the repo for the extensive list of all the results.

4.2 Empirical Findings

4.2.1 Performance of RN-Net under supervised settings. Figure 3 summarizes the results of RN-Loss when compared with recent state-of-the-art approaches – DevNet[8], FT-Transformer[10], Deep-SAD[30], PReNet[28]. The key findings are :

(1) Across datasets, RN-Net performs better than the recent SoTA approaches.



(a) Multivariate (F1)

(b) Time-Series (F1)

Fig. 4. Performance of RN-Net on Multivariate and Time-Series Datasets. (a) illustrates the results on multivariate datasets. (b) shows the results on time-series datasets.

(2) Performance with respect to F1 score of RN-Loss is better than AUROC. This is thanks to the optimal and simple threshold finding for RN-Loss described above.

(3) Across all the datasets, the average rank (lower is better) for RN-Net is 2.08, Deep-SAD is 2.58, FT-Transformer is 2.61, PReNet is 3.10, DevNet is 3.68. These ranks are with respect to AUROC.

(4) Performance of RN-Net is also superior across different anomaly ratios. In the region of $0 - 1\%$ anomalies, with respect to AUROC, RN-Loss is comparable to the SoTA approaches, with respect to F1 scores RN-Net outperforms all competing approaches. For $0 - 10\%$ anomalies and for $> 10\%$ anomalies, RN-Net outperforms all competing approaches with respect to both AUROC and F1 scores.

4.2.2 Performance of RN-Net – Multivariate Datasets: RN-Net performs the best on 44 out of 67 datasets. The rest of the algorithms demonstrate varying performance across multiple datasets. FT-Transformer achieves the best results on 8 datasets in terms of precision, 2 datasets for recall, 9 datasets for F1-score, and 7 datasets for AUCROC. DeepSAD shows superior performance on 3 datasets for precision, 10 for recall, 9 for F1-score, and 8 for AUC-ROC. Meanwhile, PReNet achieves top performance on 2 datasets in terms of precision, 6 for recall, none for F1score, and 7 for AUCROC. This is summarized in Figure 4a.

4.2.3 Performance of RN-Net – Time-Series Datasets (Univariate and Multivariate): Across 29 univariate time-series benchmark datasets RN-Net consistently demonstrates superior performance. Specifically, it achieves the highest precision and recall on 20 datasets, the best F1 score on 21, and the top AUC-ROC on 17. Recent models, including FT-Transformer, DeepSAD, and PReNet, yield only sporadic wins. RN-LSTM – designed primarily to illustrate the adaptability of the proposed RN-loss—also performs competitively, though its effectiveness is somewhat limited by fixed timestep constraints and the extreme sparsity of anomalies. In multivariate settings (evaluated on SWaT, BATADAL, and ESA-ADB), RN-Net again exhibits optimal performance, achieving the highest F1 scores across *all* datasets and maintaining consistently strong recall and AUC-ROC. These results on all the time-series datasets are summarized in Figure 4b.

Table 1. Average Rank (w.r.t AUROC) of Unsupervised Methods for Anomaly Detection. Observe that the Radon–Nikodým correction methods take 3 out of the top 4 positions.

Algorithm	Avg. Rank
KMeans-CBLOF	2.656627
KMeans-CBLOF-Mod	2.861446
q-LSTM (sigmoid)	3.086207
dbTAI-Mod	3.446429
DAGMM	3.859375
GAN	4.861702
SLAD	5.231481
NeuTraL-AD	5.456897
ICL	5.527778

4.2.4 Performance of Radon–Nikodým derivative correction in unsupervised setting: Recall that the Radon–Nikodým correction is leads to 3 algorithms KMeans-CBLOF, KMeans-CBLOF-Mod, dbTAI-Mod (See Section 3.2). As baselines, we consider both *unsupervised methods* - GAN, q-LSTM (sigmoid), DAGMM as well as the recent state-of-the-art *self-supervised methods* - SLAD, NeuTraL-AD, ICL, GAN.

Across 96 diverse benchmarks spanning univariate and multivariate, time-series and non-time-series domains, the KMeans-CBLOF variant consistently outperformed the other Radon–Nikodým-corrected methods, achieving the highest AUROC on 37 datasets. Collectively, the trio – KMeans-CBLOF, KMeans-CBLOF-Mod, and dbTAI – attained top AUROC performance in 56 out of 96 datasets (~58%), demonstrating broad competitiveness. Notably, KMeans-CBLOF was particularly effective under sparse anomaly conditions (0–1% rate), leading in 16 of 26 such cases. While dbTAI achieved fewer AUROC wins, it attained the highest mean F1 score (0.197) and recall (0.703), suggesting a recall-focused trade-off and also displayed improved stability relative to KMeans-CBLOF-Mod. Among non-Radon–Nikodým baselines, q-LSTM (sigmoid) yielded the highest average AUROC (0.833), but was outperformed by the best of the Radon–Nikodým-corrected methods in 39 datasets by a margin of at least 0.05 AUROC. The largest observed AUROC gap, approximately 0.43, occurred on the `vowels` dataset. Table 1 reports the average rank of all algorithms, with Radon–Nikodým-corrected variants occupying 3 of the top 4 positions. Pairwise AUROC comparisons in Figure 5 further validate the efficacy of simple derivative correction in enhancing classical unsupervised methods.

5 CONCLUSION

Anomaly detection is a fundamental problem across multiple domains. Formally, an anomaly is any sample that does not belong to the underlying data distribution. However, identifying anomalies is challenging, particularly when the data distribution exhibits high variability. Despite its importance, the theoretical foundations of anomaly detection remain underexplored.

What is the right principle to design loss function for anomaly detection? We show that the right principle should correct the discrepancies between the distributions. This is easily achieved by weighing the generic loss function with Radon–Nikodým derivative. We prove this by establishing the PAC learnability of anomaly detection. We refer to this approach as RN-Loss. Notably, we show that (supervised) weight-adjusted loss functions and unsupervised

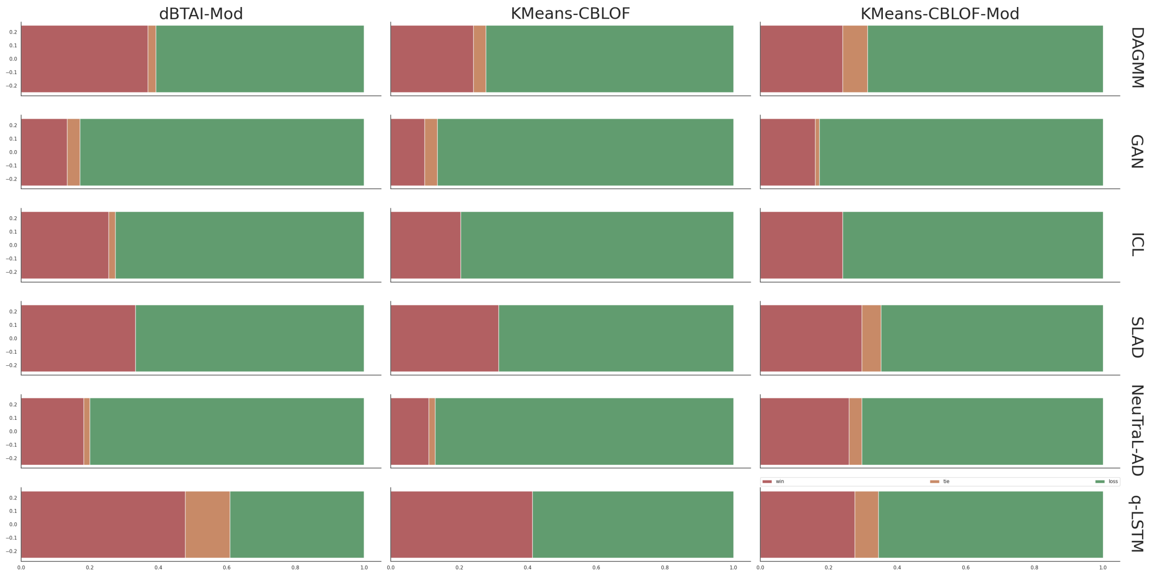


Fig. 5. Performance of Radon–Nikodým derivative correction of unsupervised algorithms with respect to AUROC. Observe that the Radon–Nikodým derivative corrected unsupervised algorithms – dbTAI(Mod.), KMeans-CBLOF and KMeans-CBLOF(Mod.) perform better than recent state-of-the-art algorithms such as ICL, NeuTraL, SLAD etc. 🟥 indicates the percentage of datasets where the row algorithm outperforms the column algorithm. 🟧 indicates the percentage where both algorithms perform equally. 🟩 indicates the percentage where the row algorithm underperforms relative to the column algorithm.

Cluster-Based Local Outlier Factor (CBLOF) naturally emerge as performant and conceptual instances of this correction mechanism.

Empirical evaluations across 96 datasets demonstrate that weighting a standard loss function by the Radon–Nikodým derivative enhances performance, making RN-Loss a robust, efficient, and adaptable solution that outperforms state-of-the-art methods under varying anomaly contamination levels.

The integral representation of the weighted loss function via Radon–Nikodým derivative can be used for imbalanced class boundaries. An interesting regularization interpretation can also be handy in future. The sparsity of the derivative implies a sparsity-inducing regularization; conversely, if it is dense, it might be interpreted as a smoothing regularizer.

REFERENCES

[1] Lirim Ashiku and Cihan Dagli. Network intrusion detection system using deep learning. *Procedia Computer Science*, 185:239–247, 2021. ISSN 1877-0509. doi: https://doi.org/10.1016/j.procs.2021.05.025. URL https://www.sciencedirect.com/science/article/pii/S1877050921011078. Big Data, IoT, and AI for a Smarter Future.

[2] Liron Bergman and Yedid Hoshen. Classification-based anomaly detection for general data. *ArXiv*, abs/2005.02359, 2020. URL https://api.semanticscholar.org/CorpusID:211549689.

[3] Nitesh Chawla, Natalie Japkowicz, and Alesia Kolcz, editors. *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets*. August 2003. URL http://www.site.uottawa.ca/~nat/Workshop2003/workshop2003.html. Workshop held in August 2003.

[4] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explor. Newsl.*, 6(1):1–6, jun 2004. ISSN 1931-0145. doi: 10.1145/1007730.1007733. URL https://doi.org/10.1145/1007730.1007733.

[5] P. Chhabra, C. Scott, E. D. Kolaczyk, and M. Crovella. Distributed spatial anomaly detection. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pages 1705–1713, 2008. doi: 10.1109/INFOCOM.2008.232.

[6] Ailin Deng and Bryan Hooi. Graph neural network-based anomaly detection in multivariate time series. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4027–4035, May 2021. doi: 10.1609/aaai.v35i5.16523. URL https://ojs.aaai.org/index.php/AAAI/article/view/16523.

[7] Mostafa Farshchi, Ingo Weber, Raffaele Della Corte, Antonio Pecchia, Marcello Cinque, Jean-Guy Schneider, and John Grundy. Contextual anomaly detection for a critical industrial system based on logs and metrics. In *2018 14th European Dependable Computing Conference (EDCC)*, pages 140–143, 2018. doi: 10.1109/EDCC.2018.00033.

[8] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alexander G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2568–2577, 2015. doi: 10.1109/CVPR.2015.7298872.

[9] Zengan Gao. Application of cluster-based local outlier factor algorithm in anti-money laundering. In *2009 International Conference on Management and Service Science*, pages 1–4, 2009. doi: 10.1109/ICMSS.2009.5302396.

[10] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=i_Q1yrOegLY.

[11] M. Gupta, J. Gao, Charu Aggarwal, and Jiawei Han. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5:1–129, 01 2014. doi: 10.2200/S00573ED1V01Y201403DMK008.

[12] Songqiao Han, Xiyang Hu, Hailiang Huang, Mingqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark, 2022. URL https://arxiv.org/abs/2206.09426.

[13] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. Local evaluation of time series anomaly detection algorithms. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 635–645, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539339. URL https://doi.org/10.1145/3534678.3539339.

[14] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 02 2015.

[15] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. Exathlon: a benchmark for explainable anomaly detection over time series. *Proc. VLDB Endow.*, 14(11):2613–2626, jul 2021. ISSN 2150-8097. doi: 10.14778/3476249.3476307. URL https://doi.org/10.14778/3476249.3476307.

[16] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6:429–449, 2002. URL https://api.semanticscholar.org/CorpusID:39321012.

[17] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7194–7201, Jun. 2022. doi: 10.1609/aaai.v36i7.20680. URL https://ojs.aaai.org/index.php/AAAI/article/view/20680.

[18] B. Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2), 2018. ISSN 2313-433X. doi: 10.3390/jimaging4020036. URL https://www.mdpi.com/2313-433X/4/2/36.

[19] Takis Konstantopoulos, Zurab Zerakidze, and Grigol Sokhadze. *Radon–Nikodým Theorem*, pages 1161–1164. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_468. URL https://doi.org/10.1007/978-3-642-04898-2_468.

[20] Krzysztof Kotowski, Christoph Haskamp, Jacek Andrzejewski, Bogdan Ruszczak, Jakub Nalepa, Daniel Lakey, Peter Collins, Aybike Kolmas, Mauro Bartesaghi, Jose Martinez-Heras, and Gabriele De Canio. European space agency benchmark for anomaly detection in satellite telemetry, 2024. URL https://arxiv.org/abs/2406.17826.

[21] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu. Copod: Copula-based outlier detection. In *2020 IEEE International Conference on Data Mining (ICDM)*, pages 1118–1123, Los Alamitos, CA, USA, nov 2020. IEEE Computer Society. doi: 10.1109/ICDM50108.2020.00135. URL https://doi.ieeecomputersociety.org/10.1109/ICDM50108.2020.00135.

[22] Zheng Li, Yue Zhao, Xiyang Hu, Nicola Botta, Cezar Ionescu, and George H. Chen. Ecod: Unsupervised outlier detection using empirical cumulative distribution functions. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12181–12193, 2023. doi: 10.1109/TKDE.2022.3159580.

[23] Jiaqi Liu, Guoyang Xie, Jinbao Wang, Shangnian Li, Chengjie Wang, Feng Zheng, and Yaochu Jin. Deep Industrial Image Anomaly Detection: A Survey. ArXiV-2301, 2023. doi: 10.48550/ARXIV.2301.11514. URL https://arxiv.org/abs/2301.11514.

[24] Chihiro Maru and Ichiro Kobayashi. Collective anomaly detection for multivariate data using generative adversarial networks. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 598–604, 2020. doi: 10.1109/CSCI51800.2020.00106.

[25] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. Tsb-uad: an end-to-end benchmark suite for univariate time-series anomaly detection. *Proc. VLDB Endow.*, 15(8):1697–1711, apr 2022. ISSN 2150-8097. doi: 10.14778/3529337.3529354. URL https://doi.org/10.14778/3529337.3529354.

[26] Chen Qiu, Timo Pfrommer, Marius Kloft, Stephan Mandt, and Maja Rudolph. Neural transformation learning for deep anomaly detection beyond images. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8703–8714. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/qiu21a.html.

[27] Joel Ratsaby. *PAC Learning*, pages 622–624. Springer US, Boston, MA, 2008. ISBN 978-0-387-30162-4. doi: 10.1007/978-0-387-30162-4_276. URL https://doi.org/10.1007/978-0-387-30162-4_276.

[28] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. pages 3932–3941, 06 2019. doi: 10.1109/CVPR.2019.00406.

[29] Moumita Roy, Sukanta Majumder, Anindya Halder, and Utpal Biswas. Ecg-net: A deep lstm autoencoder for detecting anomalous ecg. *Engineering Applications of Artificial Intelligence*, 124:106484, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.106484. URL https://www.

sciencedirect.com/science/article/pii/S0952197623006681.

[30] Lukas Ruff, Robert A. Vandermeulen, Nico Görnitz, Alexander Binder, Emmanuel Müller, Klaus-Robert Müller, and Marius Kloft. Deep semi-supervised anomaly detection. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HkgH0TEYwH.

[31] Snehanshu Saha, Jyotirmoy Sarkar, Soma Dhavala, Santonu Sarkar, and Preyank Mota. Quantile LSTM: A Robust LSTM for Anomaly Detection In Time Series Data, February 2023. URL http://arxiv.org/abs/2302.08712. arXiv:2302.08712 [cs].

[32] Jyotirmoy Sarkar, Kartik Bhatia, Snehanshu Saha, Margarita Safonova, and Santonu Sarkar. Postulating exoplanetary habitability via a novel anomaly detection method. *Monthly Notices of the Royal Astronomical Society*, 510(4):6022–6032, 12 2021. ISSN 0035-8711. doi: 10.1093/mnras/stab3556. URL https://doi.org/10.1093/mnras/stab3556.

[33] Jyotirmoy Sarkar, Santonu Sarkar, Snehanshu Saha, and Swagatam Das. d-btai: The dynamic-binary tree based anomaly identification algorithm for industrial systems. In Hamido Fujita, Ali Selamat, Jerry Chun-Wei Lin, and Moonis Ali, editors, *Advances and Trends in Artificial Intelligence. From Theory to Practice*, pages 519–532, Cham, 2021. Springer International Publishing. ISBN 978-3-030-79463-7.

[34] Jyotirmoy Sarkar, Snehanshu Saha, and Santonu Sarkar. Efficient anomaly identification in temporal and non-temporal industrial data using tree based approaches. *Applied Intelligence*, 53(8):8562–8595, April 2023. ISSN 0924-669X, 1573-7497. doi: 10.1007/s10489-022-03940-3. URL https://link.springer.com/10.1007/s10489-022-03940-3.

[35] Thomas Schlegl, Philipp Seeböck, Sebastian Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. pages 146–157, 03 2017. ISBN 978-3-319-59049-3. doi: 10.1007/978-3-319-59050-9_12.

[36] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15:1779–1797, 05 2022. doi: 10.14778/3538598.3538602.

[37] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proc. VLDB Endow.*, 15 (9):1779–1797, may 2022. ISSN 2150-8097. doi: 10.14778/3538598.3538602. URL https://doi.org/10.14778/3538598.3538602.

[38] Lifeng Shen, Zhuocong Li, and James Kwok. Timeseries anomaly detection using temporal hierarchical one-class network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13016–13026. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/97e401a02082021fd24957f852e0e475-Paper.pdf.

[39] Tom Shenkar and Lior Wolf. Anomaly detection for tabular data with internal contrastive learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_hszZbt46bT.

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

[41] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018. doi: 10.1109/CVPR.2018.00678.

[42] Ahmad Tambuwal and Daniel Neagu. Deep quantile regression for unsupervised anomaly detection in time-series. *SN Computer Science*, 2, 11 2021. doi: 10.1007/s42979-021-00866-4.

[43] Riccardo Taormina, Stefano Galelli, Nils Ole Tippenhauer, Elad Salomons, Avi Ostfeld, Demetrios G. Eliades, Mohsen Aghashahi, Raanju Sundararajan, Mohsen Pourahmadi, M. Katherine Banks, B. M. Brentan, M. Herrera, Amin Rasekh, Enrique Campbell, I. Montalvo, G. Lima, J. Izquierdo, Kelsey Haddad, Nikolaos Gatsis, Ahmad Taha, Saravanakumar Lakshmanan Somasundaram, D. Ayala-Cabrera, Sarin E. Chandy, Bruce Campbell, Pratim Biswas, Cynthia S. Lo, D. Manzi, E. Luvizotto, Jr, Zachary A. Barker, Marcio Giacomoni, M. Fayzul K. Pasha, M. Ehsan Shafiee, Ahmed A. Abokifa, Mashor Housh, Bijay Kc, and Ziv Ohar. The battle of the attack detection algorithms: Disclosing cyber attacks on water distribution networks. *Journal of Water Resources Planning and Management*, 2018.

[44] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025.

[45] Yinping Wang, Rengui Jiang, Jiancang Xie, Yong Zhao, Dongfei Yan, and Siyu Yang. Soil and water assessment tool (swat) model: A systemic review. *Journal of Coastal Research*, 93:22, 09 2019. doi: 10.2112/SI93-004.1.

[46] Phillip Wenig, Sebastian Schmidl, and Thorsten Papenbrock. Timeeval: a benchmarking toolkit for time series anomaly detection algorithms. *Proc. VLDB Endow.*, 15(12):3678–3681, aug 2022. ISSN 2150-8097. doi: 10.14778/3554821.3554873. URL https://doi.org/10.14778/3554821.3554873.

[47] David H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):1341–1390, 1996. doi: 10.1162/neco.1996.8.7.1341.

[48] Hongzuo Xu, Yijie Wang, Juhui Wei, Songlei Jian, Yizhou Li, and Ning Liu. Fascinating supervisory signals and where to find them: Deep anomaly detection with scale learning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 38655–38673. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/xu23p.html.

[49] Yong-Ho Yoo, Ue-Hwan Kim, and Jong-Hwan Kim. Recurrent reconstructive network for sequential anomaly detection. *IEEE Transactions on Cybernetics*, PP:1–12, 08 2019. doi: 10.1109/TCYB.2019.2933548.

[50] 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. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJJLHbb0-.

A LITERATURE REVIEW

Throughout the years, much research has been conducted in anomaly detection with a multitude of explored methods such as in [5, 7, 23, 24, 36, 38, 41, 49]. Since then, this interest has increased significantly as various domains such as cybersecurity [1, 46], fraud detection, and healthcare [11, 29] became more relevant. The work on learning from imbalanced datasets was proposed in AAAI 2000 workshop and highlighted research on two major problems, types of imbalance that hinder the performance of standard classifiers and the suitable approaches for the same. [16] also showed that the class imbalance problem affects not only standard classifiers like decision trees but also Neural Networks and SVMs. The work by [3, 4] gave a further boost to research in imbalanced data classification. Extensive research was done on unsupervised learning methods to address issues such as relying on static thresholds, in turn struggling to adapt to dynamic data, resulting in high false positives and missed anomalies. However, the work by [18] observed that unsupervised anomaly detection can be computationally intensive, especially in high-dimensional datasets. Jacob and Tatbul [15] delved into explainable anomaly detection in time series using real-world data, yet deep learning-based time-series anomaly detection models were not thoroughly explored well enough. With significant growth in applying various ML algorithms to detect anomalies, there has been an avalanche of anomaly benchmarking data [12], [46], [25], as well as empirical studies of the performances of existing algorithms [13], [37] on different benchmark data. Due to the importance of the problem, there have also been efforts to produce benchmarks such as AD-Bench [12] and ESA-ADB [20]. Researchers have critically examined the suitability of evaluation metrics for machine learning methods in anomaly detection. Kim et al. [17] exposed the limitations of the F1-score with point adjustment, both theoretically and experimentally.

To conclude, recent benchmarking studies, concentrated on deep-learning-based anomaly detection techniques mostly, have not examined the performance across varying types of anomalies, such as singleton/point, small, and significantly large numbers, nor across different data types, including univariate, multivariate, temporal, and non-temporal data. Additionally, there has been a lack of exploration into how anomalies should be identified when their frequency is high. These observations prompt several critical questions. Is the current SoTA algorithm the most effective? Are we reaching the peak in anomaly detection using Deep Learning approaches? Are unsupervised learning algorithms truly better than supervised learning algorithms?

B BASELINES AND ADDITIONAL TABLES

Table 2. Descriptions and hyperparameter settings of SOTA algorithms benchmarked in this study

Algorithm	**Anomaly Detection Approach**
LOF	Uses data point densities to identify an anomaly by measuring how isolated a point is relative to its nearest neighbors in the feature space. Implemented using the Python Outlier Detection (PyOD) library with default parameters. Trained on 70% of data and tested on the entire dataset.
Iforest	Ensemble-based algorithm that isolates anomalies by constructing decision trees. Reported to perform well in high-dimensional data. The algorithm efficiently separates outliers by requiring fewer splits in the decision tree compared to normal data points. Implemented using scikit-learn with default parameters. Trained on 70% of data and tested on the entire dataset.
OCSVM	Constructs a hyperplane in a high-dimensional space to separate normal data from anomalies. Implemented using scikit-learn with default parameters. Trained on 70% normal data, or whatever normal data was available.
AutoEncoder	Anomalies are identified based on the reconstruction errors generated during the encoding-decoding process. *Requires training on normal data.* Implemented using Keras. Lower threshold set at the 0.75th percentile, and upper threshold at the 99.25th percentile of the Mean Squared Error (MSE) values. Trained on 70% normal data, or whatever normal data was available. Anomalies are detected by comparing the reconstruction error with the predefined thresholds.
DAGMM	Combines .autoencoder and Gaussian mixture models to model the data distribution and identify anomalies. It has a compression network to process low-dimensional representations, and the Gaussian mixture model helps capture data complexity. *This algorithm requires at least 2 anomalies to be effective.* It is trained on 70% normal data, or whatever normal data was available. Anomalies are detected by calculating anomaly scores, with thresholds set at two standard deviations above and below the mean anomaly score.
LSTM	Trained on a normal time-series data sequence. Acts as a predictor, and the prediction error, drawn from a multivariate Gaussian distribution, detects the likelihood of anomalous behavior. Implemented using Keras. Lower threshold set at the 5th percentile, and upper threshold at the 95th percentile of the Mean Squared Error (MSE) values. Trained on 70% normal data, or whatever normal data was available. Anomalies are detected when the prediction error lies outside the defined thresholds.
qLSTM	Augments LSTM with quantile thresholds to define the range of normal behavior within the data. Implementation follows the methodology described in the authors' paper, which applies quantile thresholds to LSTM predictions. Anomalies are detected when the prediction error falls outside the defined quantile range.
QREG	A multilayered LSTM-based RNN forecasts quantiles of the target distribution to detect anomalies. The core mathematical principle involves modeling the target variable's distribution using multiple quantile functions. Lower threshold set at the 0.9th percentile, and upper threshold at the 99.1st percentile of the predicted values. Anomalies are detected when the predicted value lies outside these quantile thresholds. Trained on 70% data and tested on the entire dataset.
Elliptic Envelope	Fits an ellipse around the central multivariate data points, isolating outliers. It needs a contamination parameter of 0.1 by default, with a support fraction of 0.75, and uses Mahalanobis distance for multivariate outlier detection. Implemented using default parameters from the sklearn package. Trained on 70% of data and tested on the entire dataset. Anomalies are detected when data points fall outside the fitted ellipse.
DevNet	A Deep Learning-based model designed specifically for anomaly detection tasks. Implemented using the Deep Learning-based Outlier Detection (DeepOD) library. Anomalies are detected based on the deviation score, with a threshold defined according to the model's performance and expected anomaly rate. Trained on 70% data and tested on the entire dataset; *it requires atleast 2 anomalies in its training set to function, and for optimal performance, it is recommended to include at least 2% anomalies in the training data.*
GAN	Creates data distributions and detects anomalies by identifying data points that deviate from the generated distribution. It consists of generator and discriminator networks trained adversarially. Implemented using Keras. All data points whose discriminator score lies in the lowest 10th percentile are considered anomalies. Trained on 70% normal data.
GNN	GDN, which is based on graph neural networks, learns a graph of relationships between parameters and detects deviations from the patterns. Implementation follows the methodology described in the authors' paper.
MGBTAI	An unsupervised approach that leverages a multi-generational binary tree structure to identify anomalies in data. Minimum clustering threshold set to 20% of the dataset size and leaf level threshold set to 4. Used k-means clustering function. No training data required.
dBTAI	Like MGBTAI, it does not rely on training data. It adapts dynamically as data environments change.The small cluster threshold is set to 2% of the data size. The leaf level threshold is set to 3. The minimum cluster threshold is set to 10% of the data size and the number of clusters are 2 (for KMeans clustering at each split). The split threshold is 0.9 (used in the binary tree function). The anomaly threshold is determined dynamically using the knee/elbow method on the cumulative sum of sorted anomaly scores. The kernel density uses a gaussian kernel with default bandwidth and uses imbalance ratio to weight the density ratios. Used k-means clustering function. No training data required.
FTTransformer	It is a sample adaptation of the original transformer architecture for tabular data. The model transforms all features (categorical and numerical) to embeddings and applies a stack of Transformer layers to the embeddings. However, as stated in the original paper's [10] limitations: FTTransformer requires more resources (both hardware and time) for training than simple models such as ResNet and may not be easily scaled to datasets when the number of features is "too large".
DeepSAD	It is a generalization of the unsupervised Deep SVDD method to the semi-supervised anomaly detection setting and thus needs labeled data for training. It is also considered as an information-theoretic framework for deep anomaly detection.
PReNet	It has a basic ResNet with input and output convolution layers, several residual blocks (ResBlocks) and a recurrent layer implemented using a LSTM. It is particularly created for the task of image deraining as mentioned in [28].

Dataset	Size	Dimension	# Anomalies	% Anomalies	Domain
ALOI	49534	27	1508	3.04	Image
annthyroid	7200	6	534	7.42	Healthcare
backdoor	95329	196	2329	2.44	Network
breastw	683	9	239	34.99	Healthcare
campaign	41188	62	4640	11.27	Finance
cardio	1831	21	176	9.61	Healthcare
Cardiotocography	2114	21	466	22.04	Healthcare
celeba	202599	39	4547	2.24	Image
cover	286048	10	2747	0.96	Botany
donors	619326	10	36710	5.93	Sociology
fault	1941	27	673	34.67	Physical
fraud	284807	29	492	0.17	Finance
glass	214	7	9	4.21	Forensic
Hepatitis	80	19	13	16.25	Healthcare
http	567498	3	2211	0.39	Web
InternetAds	1966	1555	368	18.72	Image
Ionosphere	351	33	126	35.9	Mineralogy
landsat	6435	36	1333	20.71	Astronautics
letter	1600	32	100	6.25	Image
Lymphography	148	18	6	4.05	Healthcare
magic.gamma	19020	6	260	1.37	Physical
mammography	11183	6	260	2.32	Healthcare
mnist	7603	100	700	9.21	Image
musk	3062	166	97	3.17	Chemistry
optdigits	5216	64	150	2.88	Image
PageBlocks	5393	10	510	9.46	Document
pendigits	6870	16	156	2.27	Image
Pima	768	8	268	34.9	Healthcare
satellite	6435	36	2036	31.64	Astronautics
satimage-2	5803	36	71	1.22	Astronautics
shuttle	49097	9	3511	7.15	Astronautics
skin	245057	3	50859	20.75	Image
smtp	95156	3	30	0.03	Web
SpamBase	4207	57	1679	39.91	Document
speech	3686	400	61	1.65	Linguistics
Stamps	340	9	31	9.12	Document
thyroid	3772	6	93	2.47	Healthcare
vertebral	240	6	30	12.5	Biology
vowels	1456	12	50	3.43	Linguistics
Waveform	3443	21	100	2.9	Physics
WBC	223	9	10	4.48	Healthcare
WDBC	367	30	10	2.72	Healthcare
Wilt	4819	5	257	5.33	Botany
wine	129	13	10	7.75	Chemistry
WPBC	198	33	47	23.74	Healthcare
yeast	1484	8	507	34.16	Biology
CIFAR10	5263	512	263	5	Image
FashionMNIST	6315	512	315	5	Image
MNIST-C	10000	512	500	5	Image
MVTec-AD	292	512	63	21.5	Image
SVHN	5208	512	260	5	Image
Agnews	10000	768	500	5	NLP
Amazon	10000	768	500	5	NLP
Imdb	10000	768	500	5	NLP
Yelp	10000	768	500	5	NLP
20newsgroups	3090	768	155	5	NLP
BATADAL 04	4177	43	219	5.24	Industrial
SWaT 1	50400	51	4466	8.86	Industrial
SWaT 2	86400	51	4216	4.88	Industrial
SWaT 3	86400	51	3075	3.56	Industrial
SWaT 4	86319	51	37559	43.51	Industrial
SWaT 5	86400	51	2167	2.51	Industrial
SWaT 6	54000	51	3138	5.81	Industrial
ecoli	336	7	9	2.68	Healthcare
cmc	1473	9	17	1.15	Healthcare
lympho h	148	18	6	4.05	Healthcare
wbc h	378	30	21	5.56	Healthcare

Table 3. Multivariate Datasets Characterisation

Dataset	Optimal Threshold	Dataset	Optimal Threshold
ALOI	0.007	yahoo1	0.130
annthyroid	0.318	yahoo2	0.594
backdoor	0.490	yahoo3	0.415
breastw	0.751	yahoo5	0.233
campaign	0.009	yahoo6	0.127
cardio	0.445	yahoo7	0.278
Cardiotocography	0.438	yahoo8	0.243
celeba	0.006	yahoo9	0.251
cover	0.0002	Speed_6005	0.483
donors	0.221	Speed_7578	0.463
fault	0.410	Speed_t4013	0.319
fraud	0.197	TravelTime_387	0.266
glass	0.502	TravelTime_451	0.151
Hepatitis	0.463	Occupancy_6005	0.217
http	0.928	Occupancy_t4013	0.363
InternetAds	0.684	yahoo_syn1	0.464
Ionosphere	0.393	yahoo_syn2	0.396
landsat	0.253	yahoo_syn3	0.534
letter	0.695	yahoo_syn5	0.449
Lymphography	0.474	yahoo_syn6	0.421
magic.gamma	0.074	yahoo_syn7	0.437
mammography	0.220	yahoo_syn8	0.335
mnist	0.319	yahoo_syn9	0.457
musk	0.999	aws1	0.125
optdigits	0.022	aws2	0.444
PageBlocks	0.398	aws3	0.238
pendigits	0.274	aws_syn1	0.387
Pima	0.455	aws_syn2	0.639
satellite	0.114	aws_syn3	0.398
satimage-2	0.222		
shuttle	0.861		
skin	0.005		
smtp	0.001		
SpamBase	0.409		
speech	0.361		
Stamps	0.487		
thyroid	0.341		
vertebral	0.446		
vowels	0.414		
Waveform	0.311		
WBC	0.547		
WDBC	0.837		
Wilt	0.321		
wine	0.436		
WPBC	0.417		
yeast	0.406		
CIFAR10	0.325		
FashionMNIST	0.409		
MNIST-C	0.004		
MVTec-AD	0.549		
SVHN	0.504		
Agnews	0.041		
Amazon	0.103		
Imdb	0.061		
Yelp	0.170		
20newsgroups	0.139		
BATADAL_04	0.413		
SWaT 1	0.006		
SWaT 2	0.002		
SWaT 3	0.003		
SWaT 4	0.005		
SWaT 5	0.004		
SWaT 6	0.010		
ecoli	0.505		
cmc	0.458		
lympho h	0.381		
wbc h	0.348		

Table 4. Optimal Thresholds for Various Datasets; Main Text(Contribution) : Automated Hyperparameter Tuning