NON-PARAMETRIC PROBABILISTIC ANOMALY DETECTION IN EVOLVING DATA: APPLICATIONS TO TIME SERIES

by

Sreelekha Guggilam

February 10, 2022

A dissertation submitted to the
Faculty of the Graduate School of
The University at Buffalo, The State University of New
York in partial fulfillment of the requirements for the
degree of

Doctor of Philosophy

Institute for Computational Data Science

Acknowledgments

I would like to express my deepest sense of gratitude to my advisors, mentors, and guides **Dr. Abani Patra** and **Dr. Varun Chandola** for their continuous support, patience, and motivation. Their guidance and persistent help were instrumental in this dissertation. It was their work ethics, perseverance, and collaboration that have deeply influenced me and brought this thesis to completion. I could not imagine a better pair of advisors for my Ph.D. study.

I would also like to thank my committee members **Dr. Marianthi Markatou** and **Dr. Gino Biondini** for providing me with detailed suggestions and feedback that shaped my research.

I am grateful to all the hardworking former and present members of the Computational Sciences Department and University at Buffalo. I am fortunate to have either worked or known them during my Ph.D. studies.

During my stay at UB, I have interacted with several bright students while writing papers and proposals. I thank Arshad, Palak, Karthik, Anjali, Aafiya, Kusal, Praneeth, Paul, and others. Special thanks to Kaeleigh, Sonali, Monika, Divya, and other friends, who have been a great emotional support during the course of my Ph.D. Some of the research in this thesis has been funded by grants from NSF and I am grateful for their support.

Finally and most importantly, I would like to thank them for the immense support of my parents, my brothers, and other family members. This thesis could not have been completed without their unending motivation and encouragement which have been crucial.

Dedication

To mom and dad for all their unconditional sacrifice.

To Sreekar and Prabhanj.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis examines the problem of anomaly detection in evolving data. Data-driven anomaly detection methods typically build a model for the normal behavior of the target system and score each data instance with respect to this model. A threshold is invariably needed to identify data instances with high (or low) scores as anomalies. This presents a practical limitation on the applicability of such methods, since most methods are sensitive to the choice of the threshold, and it is challenging to set optimal thresholds. The issue is exacerbated in a streaming scenario, where the optimal thresholds vary with time.

Furthermore, the methods lack the ability to scale to high-dimensional settings or time-series databases. Anomaly detection for time series data is often aimed at identifying extreme behaviors within an individual time series. However, identifying extreme trends relative to a collection of other time series is of significant interest, like in the fields of public health policy, social justice and pandemic propagation.

This thesis has 3 parts, the first two parts are two novel anomaly detection algorithms for evolving data and the third part is an application of the second anomaly detection method to neural networks. In the first part, we present a probabilistic framework to explicitly model the normal and anomalous behaviors and probabilistically reason about the data. An extreme value theory based formulation is proposed to model anomalous behavior as the extremes of normal behavior. As a specific instantiation, a joint non-parametric clustering and anomaly detection algorithm (INCAD) is proposed that models the normal behavior as a Dirichlet Process Mixture Model. Results on a variety of data sets, including streaming data, show that the proposed method provides effective and simultaneous clustering and anomaly detection without requiring strong initialization and threshold parameters.

Since the INCAD model is unable to scale to time-series databases, we present a second algorithm in the second part of our thesis, *Large Deviations Anomaly Detection* (LAD), that can scale to large collections of time series data using the concepts from the theory of *large deviations*. Exploiting the ability of the algorithm to scale to high-dimensional data, we propose an online anomaly detection method to identify anomalies within individual time series and then to a collection of multivariate time series. We demonstrate the applicability of the proposed *Large Deviations Anomaly Detection* (LAD) algorithm in identifying counties in the United States with anomalous trends in terms of COVID-19 related cases and deaths. Several of the identified anomalous counties correlate with counties with documented poor responses to the COVID pandemic.

In the third part of this thesis, the computational efficiency of the LAD model is used to improve the training of artificial neural networks (ANNs) using our novel training algorithm. The aim of the novel training model, LAD Improved Iterative Training (LIIT), is to design a faster training approach using a smaller but better representative sample of the training data. We adopt the LAD anomaly scores to construct a series of Modified Training Samples (MTS) that are updated iteratively. The LIIT model incorporates ideas from Gradient Boosting methods to improve the learning process for the ANN. We present an extensive study on the performance of this training approach on simple clustering ANN as compared to the traditional training model and demonstrate the robustness of the LIIT approach to perturbations.

Chapter 1

Introduction

Anomaly detection refers to the identification or classification of uncommon and extreme behavior in data. Depending on the context, these behaviors are can be referred to as rare events, extreme events, novelty, attacks, fraud, noise, outliers, or anomalies. While one can reduce anomaly detection to a simple pre-processing step for small datasets, it is a significant challenge to identify them in most complex datasets. Particularly in evolving datasets or high-dimensional data, anomaly detection in a supervised or semi-supervised setting is not the most ideal approach (as the underlying assumptions about the data might be inaccurate). Thus, research on unsupervised anomaly detection algorithms is of much interest.

1.1 Defining and Studying Anomalies

Establishing an abstract mathematical definition of an anomaly has been challenging due to diversity among interpretations. The domain, dataset as well as the research problem of interest - all collectively dictate the way we define, identify and study anomalies. However, the shared theme amongst these variations is the scarcity of an anomaly.

The fundamental idea behind studying anomalies is to prepare for the unforeseen and exercise suitable measures to constrain loss. Hence, we need to define anomalies, identify them and understand their severity.

Traditionally, relative attributes such as extreme, unique, rare, novel, irregularity, low likelihood of occurrence and being extreme are used to define anomalies making it challenging to study them. Consequently, behaviors that diverge from the expected are anomalous. Thus, a metric for apt quantification of this divergence must be defined.

1.2 Contributions

This thesis makes the following key contributions:

1.2.1 Unsupervised Anomaly Detection in Evolving Data Streams

Anomaly detection heavily depends on the definitions of expected and anomalous behaviors [51, 35, 48]. In most real systems, observed system behavior typically forms natural clusters whereas anomalous behavior either forms a small cluster or is weakly associated with the natural clusters. Under such assumptions, clustering based anomaly detection methods form a natural choice [14, 27, 44] but have several limitations.

Firstly, clustering based methods usually require baseline assumptions that are often conjectures and generalizing them is not always trivial. This leads to inaccurate choices for model parameters such as the number of clusters or the thresholds that are required to classify anomalies. Score based models have thresholds that are often based on data/user preference. Such assumptions result in models that are susceptible to modeler's bias and possible over-fitting.

Secondly, setting the number of clusters has additional challenges when dealing with streaming data, where new behavior could emerge and form new clusters. Non-stationarity is inherent as data evolves over time. Moreover, the data distribution of a stream changes over time due to changes in the environment, trends or other unforeseen factors [34, 49]. This leads to a phenomenon called *concept drift*, due to which an anomaly detection algo-

Table 1.1: Comparison with other anomaly detection methods: The table illustrates the the gap that exists among existing anomaly detection methods and the research contributions of the INCAD model

	Neural Networks	LOF	KNN	Kmeans –	Kernel Function Based	Gaussian Model Based	INCAD
Clustering Based	✗	✗	✓	✓	✗	✗	✓
Multi-dimension	✓	✗	✓	✓	✓	✓	✓
Unsupervised	✗	✓	✓	✓	✓	✓	✓
Non-parametric	✗	✗	✗	✗	✓	✗	✓
Adaptable to streaming settings	✗	✗	✗	✗	✗	✗	✓
Adaptive thresholds	✗	✗	✗	✗	✗	✗	✓
Probabilistic scoring	✗	✗	✗	✗	✓	✗	✓

rithm cannot assume any fixed distribution for data streams. Thus, there arises a need for a definition of an anomaly that is dynamically adapted.

Thirdly, when anomaly detection is performed post clustering [3, 32], the presence of anomalies gives a skewed (usually slight) definition of traditional/normal behavior. However, since the existence of anomalies impacts the clustering as well as the definition of the '*normal*'[1] behavior, it seems counter-intuitive to classify anomalies based on such definitions[2].To avoid this, simultaneous clustering and anomaly detection needs to be performed.

In addition to the above, extending these assumptions to the streaming context leads to a whole new set of challenges. Many supervised [16, 40] and unsupervised anomaly detection techniques [7, 16, 38, 44] are offline learning methods that require the full data set in advance for data mining which makes them unsuitable for real-time streaming data. Although supervised anomaly detection techniques may be effective in yielding good results, they are typically unsuitable for anomaly detection in streaming data [40]. We propose a method called Integrated Clustering and Anomaly Detection (INCAD), that couples Bayesian nonparametric modeling and extreme value theory to simultaneously perform clustering and anomaly detection. Table 1.1 summarizes the properties of INCAD vs other strategies for anomaly detection. The primary contributions of the algorithm are as follows:

1. **Generalized anomaly definition with adaptive interpretation**: The model's def-

[1] Non-anomalous behavior is described as “normal" behavior. Should not be confused with Gaussian/Normal distribution.

[2] Clustering and defining “normal/traditional" behavior in presence of anomalies develop in skewed and inconsistent results.

inition of an anomaly has dynamic interpretation allowing anomalous behaviors to evolve into normal behaviors and vice versa. This definition not only evolves the number of clusters with an incoming stream of data (using non-parametric mixture models) but also helps evolve the classification of anomalies.

2. **Combination of Bayesian non-parametric models and extreme value theory(EVT)**: The novelty of INCAD approach lies in blending extreme value theory and Bayesian non-parametric models. Non-parametric mixture models [45], such as *Dirichlet Process Mixture Models* (DPMM) [6, 63, 73], allow the number of components to vary and evolve during inference. While there has been limited work that has explored DPMM for the task of anomaly detection [68, 75], they have not been shown to operate in a streaming mode or ignore online updates to the DPMM model. On the other hand, EVT gives the probability of a point being anomalous which has a more universal interpretation, in contrast to the scoring schema with user-defined thresholds. Although EVT's definition of anomalies is more adaptable for streaming data sets [69, 8, 30], fitting an extreme value distribution (EVD) on a mixture of distributions or even multivariate distributions is challenging. This novel combination brings out the much-needed aspects in both the models.

3. **Extension to streaming settings**: The model is non-exchangeable which is well suited to capture the effect of the order of data input and utilize this dependency to develop streaming adaptation.

4. **Ability to handle complex data generative models**: The model can be generalized to multivariate distributions and complex mixture models.

1.2.2 Large Deviations Anomaly Detection for High Dimensional Data and Time Series Database

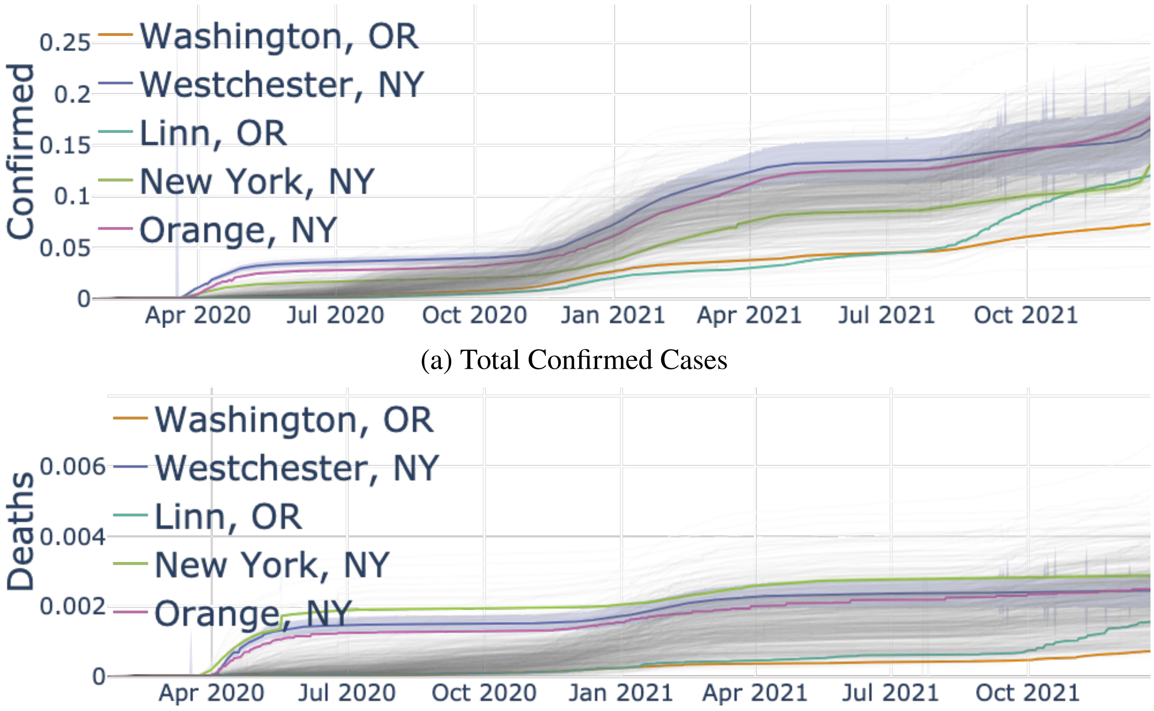
High dimensional data is often subjected to dimensionality reduction to address the challenges associated with curse of dimensionality. But anomalies is these less relevant features can often be missed by reducing our data. Additionally, in an evolving setting like a multivariate time series or a time series database, these methods tend to become computationally very expensive and inaccurate. Though challenging, these studies can be of significant importance to monitor multiple trends simultaneously like weather data across multiple geographical locations, pandemic trends, multiple sensor data from different sources, etc. For instance, consider the anomalous COVID-19 trends observed in US counties (See Figure 1.1). Identifying the most extreme trends can help us understand or study the mandates and public health policies implemented in these locations and how they have impacted the population. In such cases, studying them collectively is inevitable to have a comparative evaluation against rest of the geographical locations. Thus, the need for studying a collection of multivariate evolving trends is substantial.

Thus, we propose a novel, large deviations theory based anomaly detection algorithm that provides a computationally inexpensive solution to studying high dimensional evolving data without additional dimensionality reduction. This part of the thesis has following key contributions:

1. We propose the Large deviations Anomaly Detection (LAD) algorithm which is a data driven model that returns probabilistic evolving anomaly score for all the observations.

2. The model uses large deviations principle based methodology making it highly scalable, without any additional dimensionality reduction.

[3] In early November, these counties in North Dakota were exhibiting infection rates that were six times the national rate - https://www.washingtonpost.com/opinions/2020/11/06/north-dakota-covid-19-cases/



(a) Total Confirmed Cases

(b) Total Deaths

Figure 1.1: Top 5 anomalous counties identified by the proposed LAD algorithm based on the daily multivariate time-series, consisting of cumulative COVID-19 per-capita infections and deaths. At any time instance, the algorithm analyzes the bivariate time series for all the counties to identify anomalies. The time-series for the non-anomalous counties are plotted (light-gray) in the background for reference. For the counties in North Dakota (Burleigh and Grand Forks), the number of confirmed cases (*top*), and the sharp rise in November 2020, is the primary cause for anomaly[3]. On the other hand, Wayne County in Michigan was identified as anomalous primarily because of its abnormally high death rate, especially when compared to the relatively moderate confirmed infection rate.

3. The LAD model is easily generalized with online extension to multivariate time series as well as collections of multivariate time series.

4. Due to its scalability and broad applicability, we extend the work to study COVID-19 pandemic trends to identify geographical locations with extreme patterns.

1.2.3 Using Large Deviations for Training Stable Neural Networks

Artificial neural networks (ANNs) require tremendous amount of data to train on. However, in classification models, most data features are often similar which can lead to increase in training time without significant improvement in the performance. Thus, we hypothesize that there could be a more efficient way to train an ANN using a better representative sample. For this, we propose the LAD Improved Iterative Training (LIIT), a novel training approach for ANN using large deviations principle to generate and iteratively update training samples in a fast and efficient setting. This is exploratory work with extensive opportunities for future work. The thesis presents this ongoing research work with the following contributions from this study:

1. We propose a novel ANN training method, LIIT, based on the large deviations theory where additional dimensionality reduction is not needed to study high dimensional data.

2. The LIIT approach uses a Modified Training Sample (MTS) that is generated and iteratively updated using a LAD anomaly score based sampling strategy.

3. The MTS sample is designed to be well representative of the training data by including most anomalous of the observations in each class. This ensures distinct patterns and features are learnt with smaller samples.

4. We study the classification performance of the LIIT trained ANNs with traditional batch trained counterparts.

1.3 Thesis Outline

This thesis is organized in following three parts:

Part I deals with anomaly detection for evolving streams. Chapter 2 provides an overview of evolving data and background on Extreme Value theory (EVT). Chapter 3 describes Dirichlet Process Mixture Models (DPMM) and introduces a novel algorithm INCAD using EVT and DPMM. Chapter 4 provides a comparative evaluation of INCAD against state-of-the-art anomaly detection algorithms for non-streaming and streaming settings.

Part II deals with anomaly detection for high dimensional datasets. Chapters 5 provides an overview of relevant existing methods for anomaly detection in high dimensional settings and a short background on underlying large deviations theory and the approach to anomaly detection using the same. Chapter 6 details our LAD model for detecting unsupervised anomalies in multivariate time series. Chapter 7 describes the experiments and demonstrates the state-of-the-art performance of the LAD method. Chapter 8 discussed the anomalous COVID-19 trends captured by the LAD model.

Part III deals with devising improvised training samples for artificial neural networks. Chapter 9 provides an overview of relevant existing methods for reducing training data for ANNs. Chapter 10 details the training methodology for the LIIT approach and the sampling strategies used in conjunction with LIIT. Chapter 11 describes the experiments and demonstrates the state-of-the-art performance of our method.

Part I

Unsupervised Anomaly Detection in Streaming Data

Chapter 2

Anomaly Detection in Streaming Data

2.1 Introduction

Anomalies are unusual, unexpected, and surprising phenomena that need to be detected and explained. Identifying, understanding, and prediction of anomalies from data forms one of the key pillars of modern data mining, and has applications in almost every application domain. For instance, effective detection of anomalies can reveal critical information needed to stop malicious attacks, detect and repair faults, and, ultimately, understand the behavior of a complex system. In fact, one of the most practical applications of anomaly detection is for monitoring system behavior and detecting when the system exhibits anomalous behavior due to external or internal stress factors [33]. In this regard, two types of anomaly detection methods, viz., online anomaly detection [69, 2, 71] and clustering based anomaly detection [27, 44, 56], are highly relevant. Online methods, that can simultaneously identify clusters and the anomalies from streaming data, are especially beneficial, as complex system behavior typically falls into multiple regimes or clusters.

However, existing anomaly detection methods face two key challenges in this context. *First* challenge is the reliance of existing anomaly detection methods on an *a priori* user-defined threshold which makes them highly sensitive to the choice of the threshold. While a large literature on anomaly detection exists (Chandola, Banerjee, and Kumar, 2009), most of the existing methods follow a general two-phase strategy: i). learn a model, $\mathcal{N}$, for the normal behavior of the underlying system, and ii). score a data instance, x, with respect to $\mathcal{N}$ using a scoring function, $s_{\mathcal{N}}()$. Typically, the score is uncalibrated, though some methods produce a calibrated score (probability). However, to identify anomalies, every method requires a notion of a threshold, δ, such that the data instances whose score is above (or below) δ are anomalous. While unthresholded scores are sufficient for evaluation purposes, e.g., generating an ROC curve or comparing different methods on a validation data set, an optimal threshold is necessary in an operational setting. A very high threshold could potentially result in missing many anomalies while a low threshold would have a high false positive rate. The issue is exacerbated in a streaming setting, where both $\mathcal{N}$ and δ can evolve. While current streaming anomaly detection methods allow updates to $\mathcal{N}$, none of them allow for updating the threshold, δ. *Second* challenge is specific to clustering-based anomaly detection methods. Traditional methods learn the clustering structure from the observed data as a surrogate for the normal behavior, $\mathcal{N}$. Adapting such methods for streaming data requires the ability to allow the clustering to evolve, i.e., new clusters can form and old clusters can grow or split. Current clustering-based methods are not equipped to adapt to such evolving stream behavior.

One possible solution would be to explicitly learn a model, $\mathcal{A}$, for the anomalous behavior, and then compare the scores, $s_{\mathcal{N}}(x)$ and $s_{\mathcal{A}}(x)$, to declare if a data instance is normal or anomalous. By allowing both models to "evolve" in a streaming setting, a robust streaming anomaly detector can be developed. However, given the lack of sufficient (or any) anomalous data, learning $\mathcal{A}$ is not possible. We advocate the use of *Extreme value theory* (EVT) (Charras-Garrido and Lezaud, 2013) to learn a surrogate for $\mathcal{A}$. The core idea is to assume that the anomalous observations are the *extreme* values of $\mathcal{N}$. Using a key result in EVT, which states that the extreme values can be modeled as a parameterized distribution (referred to as an *Extreme value distribution* or EVD), one can learn $\mathcal{A}$ for a given $\mathcal{N}$.

In principle, this is a fundamental breakthrough in anomaly detection, and some initial work has been recently published in this direction (Siffer et al., 2017). However, current EVT supports a limited class of base distributions ($\mathcal{N}$); in fact, while dealing with extremes of a univariate and unimodal distribution is well understood in EVT, handling multivariate and/or richer distributions, e.g., mixture models, is a challenge. We propose an EVT driven strategy that can admit a richer class of n distributions. A generalization of EVT to multivariate and multimodal distributions (Clifton et al., 2014) is employed, which uses EVT on the likelihood of the observations, thus reducing the problem to univariate setting.

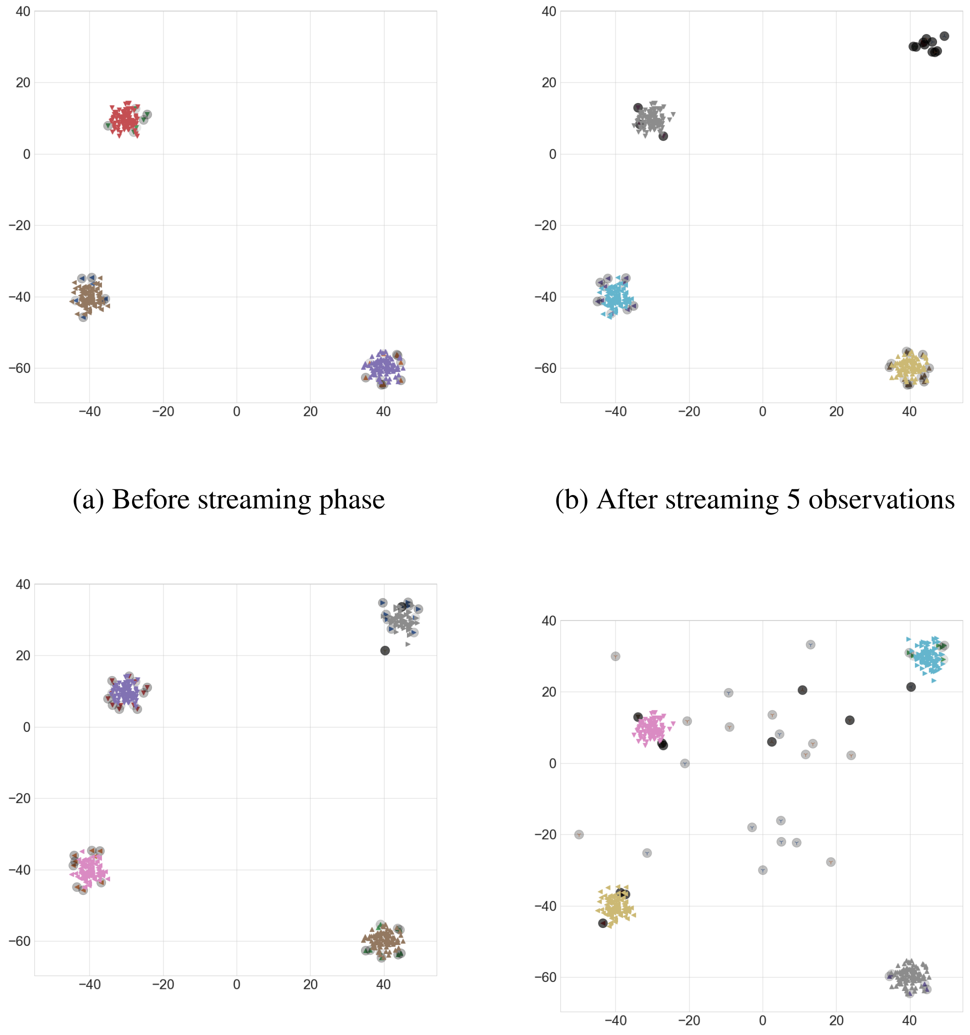
As an instantiation of the EVT driven strategy, we propose an anomaly detection method in which the normal behavior, $\mathcal{N}$, is modeled as a non-parametric mixture model—*Dirichlet Process Mixture Model* (Frigyik, Kapila, and Gupta, 2010), or DPMM—which allows clustering the data without pre-specifying the number of clusters. This, especially when adapted to the streaming setting, is an invaluable feature for anomaly detection, where the normal clustering pattern can evolve with the stream. This is an invaluable feature for anomaly detection in an online setting, where the normal clustering pattern can evolve with incremental data addition. The anomalous distribution, $\mathcal{A}$, is also a DPMM with a coupling with $\mathcal{N}$ which forces the parameters of $\mathcal{A}$ to be generated from the extremes of the prior distribution that generates the parameters for $\mathcal{N}$. The resulting method can perform joint clustering and anomaly detection and can be adapted to a streaming setting, with robustness to the choice of threshold for identifying anomalies. Experimental results on synthetic and publicly available data sets are provided to demonstrate the effectiveness of the proposed method over state of art methods.

2.1.1 INCAD Contributions

The model makes the following key contributions:

1. We propose a method called *Integrated Clustering and Anomaly Detection* (INCAD)[1],

[1] A preliminary version of INCAD was published here [42]



(a) Before streaming phase

(b) After streaming 5 observations

(c) After introducing all instances for fourth cluster

(d) End of streaming phase

Figure 2.1: Illustration of INCAD performance on a synthetic streaming data set. (a). After the initial batch phase, INCAD correctly and automatically identifies three clusters in the data, along with some peripheral data instances as anomalies (denoted by a ○, where the transparency intensity denotes the probability of the observation being anomalous). (b). As new instances arrive in the stream, INCAD first identifies them as anomalies, and then, (c). identifies a new cluster. (d). The truly anomalous instances in the stream are labeled as anomalies with a higher probability than the false positives (instances on the periphery of the clusters).

that couples Bayesian non-parametric modeling and extreme value theory to simultaneously perform clustering and anomaly detection. INCAD uses a dynamic definition of anomalous and non-anomalous behavior, which makes it well-suited for continuous monitoring applications. At the same time, by using a non-parametric clustering mechanism, i.e., *Dirichlet Process Mixture Models* (DPMM), the model permits the formation of new clusters at subsequent processing steps. This feature helps in addressing issues in open set classification[9, 36]. Moreover, by explicitly modeling the anomalous behavior, the model can directly produce an anomaly label, instead of relying on a user-defined threshold on a score.

2. We provide a key theoretical result that enables us to extend the EVT formulation to multi-dimensional data, via the *extended Generalized Pareto Distribution* modification.

3. We put forward a streaming extension to the INCAD model that captures *drift or evolution* in streams as illustrated in Figure 2.1.

4. We provide a comprehensive evaluation of the model on a variety of benchmark data sets to highlight its effectiveness and provide a comparison against existing models.

2.2 Extreme Value Theory

Extreme value theory (EVT) [18] is the study of extremes of data distributions. The foundations were laid by Fisher and Tippett (1928) and Gnedenko (1943) who demonstrated the closed forms of the distributions of the extreme values of i.i.d. samples. In this part, we follow the theory by De Haan and Ferreira (2007).

Broadly speaking, there are two principal approaches to studying extreme values. One of the approaches is to study the *block maxima* i.e. the largest observations in multiple large samples (or blocks) of identically distributed observations. For instance, consider a random variable, X, with G as the Cumulative Distribution Function (CDF)[2]. Given n realizations of this random variable, $\{X_1, X_2, \dots, X_n\}$, let , $M_n = max\{X_1, X_2, \dots, X_n\}$. If there exists a sequence of constants $a_n > 0, b_n \in \mathbb{R}$, such that $\frac{M_n - b_n}{a_n}$ has a non-degenerate distribution as $n \to \infty$.

$$P\left(\frac{M_n - b_n}{a_n} \leq x\right) \to G(x) \text{ as } n \to \infty \tag{2.1}$$

In other words, if Equation 2.1 holds for every continuity point x of the non-degenerate distribution G^{EV}, then G^{EV} is called an extreme value distribution and the class of distributions G satisfying (2.1) are said to be in the domain of attraction of G^{EV}.

For univariate data, the Generalized Extreme Value (GEV) distribution, $G^{EV}(x)$, takes the following form:

$$G^{EV}(x) = exp\left\{-\left[1 + \zeta\left(\frac{x - \nu}{\beta}\right)\right]^{-1/\zeta}\right\} \tag{2.2}$$

where ν, β and $\zeta \geq 0$ are the location, scale and shape parameters of the distribution. For $\zeta = 0$ the distribution takes the form

$$G^{EV}(x) = exp\left\{-exp\left[-\frac{x - \nu}{\beta}\right]\right\} \tag{2.3}$$

ζ is typically referred to as the *extreme value index* and depends on the shape of the tail of the data distribution, G. For instance, if G is a univariate Gaussian distribution, then $\zeta = 0$. Table 2.1 and Figure 2.2 shows the shapes of the tail for different distributions, and the corresponding value for ζ.

Given a distribution, G, and the corresponding EVD, one can calculate the cumulative probability of an observation x to be an extreme value with respect to G. This requires estimation of the shape parameter, ζ, which can be done directly from data. However, the

[2] We will use G_X to denote the CDF of the data X and G_X^{EV} to denote the corresponding tail distribution. Unless needed, the subscript is omitted for ease of notation.

Table 2.1: Relation between G and ζ : The table presents types of extreme value tail distributions associated with different data distributions.

Tail Behavior	**Tail distribution**	**Examples**
Exponential tail	Gumbel ($\zeta = 0$)	Gaussian, Exponential, Gumbel, Lognormal
Heavy tail	Fréchet ($\zeta > 0$)	Pareto, Fréchet
Bounded tail	Reversed Weibull ($\zeta < 0$)	Uniform, Beta, Reversed Weibull

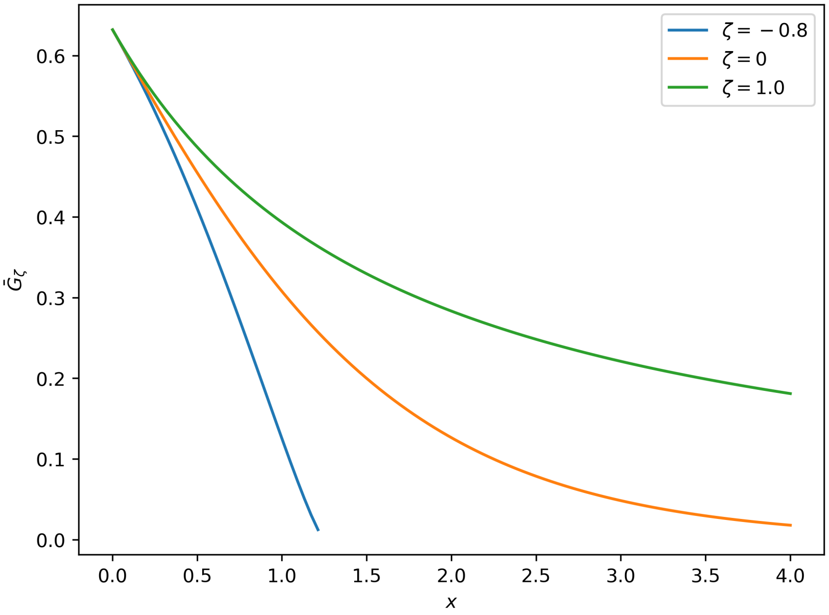


Figure 2.2: Tail distribution for different F for different values of ζ: In this figure we see the tail distributions associated with different shape parameters ζ of the extreme value distribution. Fatter tails are associated with larger value of ζ.

above approach only utilizes maximal value in each block, and is, thus, inefficient. A more economical approach to study extremes, called *Peaks-Over-Threshold* (POT) [61], studies all large observations which exceed a high threshold. In POT, the excesses over a user-specified threshold, t, i.e., $Z = X - t$ can be modeled as a *Generalized Pareto Distribution* (GPD), given by the following CDF:

$$G_Z^{EV}(z) = \begin{cases} 1 - \left(1 + \zeta\left(\frac{z-\mu}{\sigma}\right)\right)^{-\frac{1}{\zeta}} & \text{if } \zeta \neq 0 \\ 1 - \exp\left(-\frac{z-\mu}{\sigma}\right) & \text{if } \zeta = 0 \end{cases} \tag{2.4}$$

with μ, σ, and ζ as the location, scale, and shape parameters, respectively. The choice of the threshold, t, is often regarded as a bias-variance problem as very large or extreme thresholds lead to fewer observations and over-fitting whereas thresholds resulting in many tail observations result in bias. In this part, we favor the POT approach due to simplicity in implementation and explanation.

Of course, given a data distribution, G, there is no guarantee that a corresponding EVD exists. A simple theorem from De Haan and Ferreira (2007) on domains of attraction for univariate data is used to establish the necessary conditions for the existence of the EVD for G[3].

Theorem 1. *Let G be a distribution of X with u as the right upper limit on the realizations of X. Assume that second order derivatives G'' exists and the first order derivative G' is positive for all x in the left neighborhood of u. If*

$$\lim_{x \to u} \left(\frac{1 - G}{G'} \right)' (x) = \zeta \tag{2.5}$$

or alternately,

$$\lim_{x \to u} \frac{(1 - G(x))(G''(x)}{(G'(x))^2} = -\zeta - 1 \tag{2.6}$$

then G is in the maximum domain of attraction (MDA)[4] *of GEV family of distributions $\mathbf{G}_{\zeta}^{EV}$ with shape parameter ζ.*

2.2.1 Extreme Value Theory for Multivariate Data

In the previous section, we posed the different approaches in extreme value theory in the univariate space. However, most datasets are often multivariate rendering the above approach inapplicable. In this section, we develop the multivariate approach to extreme values.

[3] The detailed mathematical proofs for the above theorems is given in De Haan et al. [21].

[4] The maximum domain of attraction can be seen as a family of distributions with tail distributions that are unique up to location and scale parameters.

For the sake of notational simplicity we will discuss a 2-D case, where the random variable, X, is denoted as a tuple (X_1, X_2).

Definition 1. *Let $\{(X_{1,i}, X_{2,i})\}_{i=1}^{n}$ be a sequence of independent and identically distributed random tuples with distribution* $\mathbf{G}$*. Suppose that there exist sequences of constants $a_i, c_i > 0$ and $b_i, d_i \in \mathbb{R}$ and a distribution $\mathbf{G}^{EV}$ with non-degenerate marginals for all continuity points of (x_1, x_2). Then any limit function of $\mathbf{G}^{EV}$ given below with non-degenerate marginals is called a multivariate extreme value distribution,*

$$\lim_{i\to\infty} P\left(\frac{M_{X_{1,i}} - b_i}{a_i} \leq x, \frac{M_{X_{2,i}} - d_i}{c_i} \leq y\right) = \mathbf{G}^{EV}(x, y) \tag{2.7}$$

where $M_{X_{1,i}} = max(X_{1,1}, X_{1,2}, ...X_{1,i})$ and $M_{X_{2,i}} = max(X_{2,1}, X_{2,2}, ...X_{2,i})$.

Extending the univariate results to multivariate settings is often arduous and computationally complex. However, as most data is often multivariate, we study using an alternative approach where the probability image space is used to identify anomalies.

2.2.2 Using Probability Image Space for Handling Multi-modal and Multivariate Distributions

Estimation of parameters for extreme value distributions is often infeasible if the distribution is multi-modal and/or if the random variable is multivariate [61, 21]. To address this challenge, recent work by Clifton et al. (2014) shows that it is possible to construct, and examine, an equivalent univariate distribution by considering the probability image space. The result states that for a probability distribution function, $g_X : X \to Y$, where $Y \in \mathbb{R}^+$ is the probability image space, let random variable Y be defined as a distribution G_Y, with following CDF:

$$G_Y(y) = \int_{g_Y^{-1}([0,y])} g_X(x)dx \tag{2.8}$$

where $g_Y^{-1}([0, y])$ denotes all the values of the random variable X, whose probability density is between 0 and y. Using the POT result (Pickands, 1975), as discussed earlier, it can be shown that for a small positive value, u, the tail of G_Y can be modeled as a GPD for $y \in [0, u]$, as $u \to 0$, such that if an observation x is extreme with respect to the original distribution, G_X, if $g_X(x) < u$, then $y = g_X(x)$ will be extreme with respect to G_Y. The corresponding GPD for $(u - y)$, denoted as G_Y^{EV}, can be used to calculate the probability of x to be extreme, with respect to G_X.

A simulated example is shown in Figure 2.3, where 2000 observations from two univariate Gaussian distributions are studied. Unlike the traditional EVT approach that can only study tail distributions for unimodal data, the Ext-GPD approach is able to include rare observations between the two modes as seen in the shaded red zone in Figure 2.3a. The probability image space of the mixture distribution is used to study the observations with low probabilities i.e. the rare tail observations. The resulting image space is considered as the one-dimensional projections of the original data and the anomalies are identified by studying the left tail in Figure 2.3b. The Ext-GPD approach is discussed in detail in Section 2.2.3. The theory behind the extended GPD approach has not been presented earlier[20]. Hence, we present the necessary conditions one-dimensional data in Section 2.2.3. The proof for multi-dimensional case is similar and has been included in the supplementary.

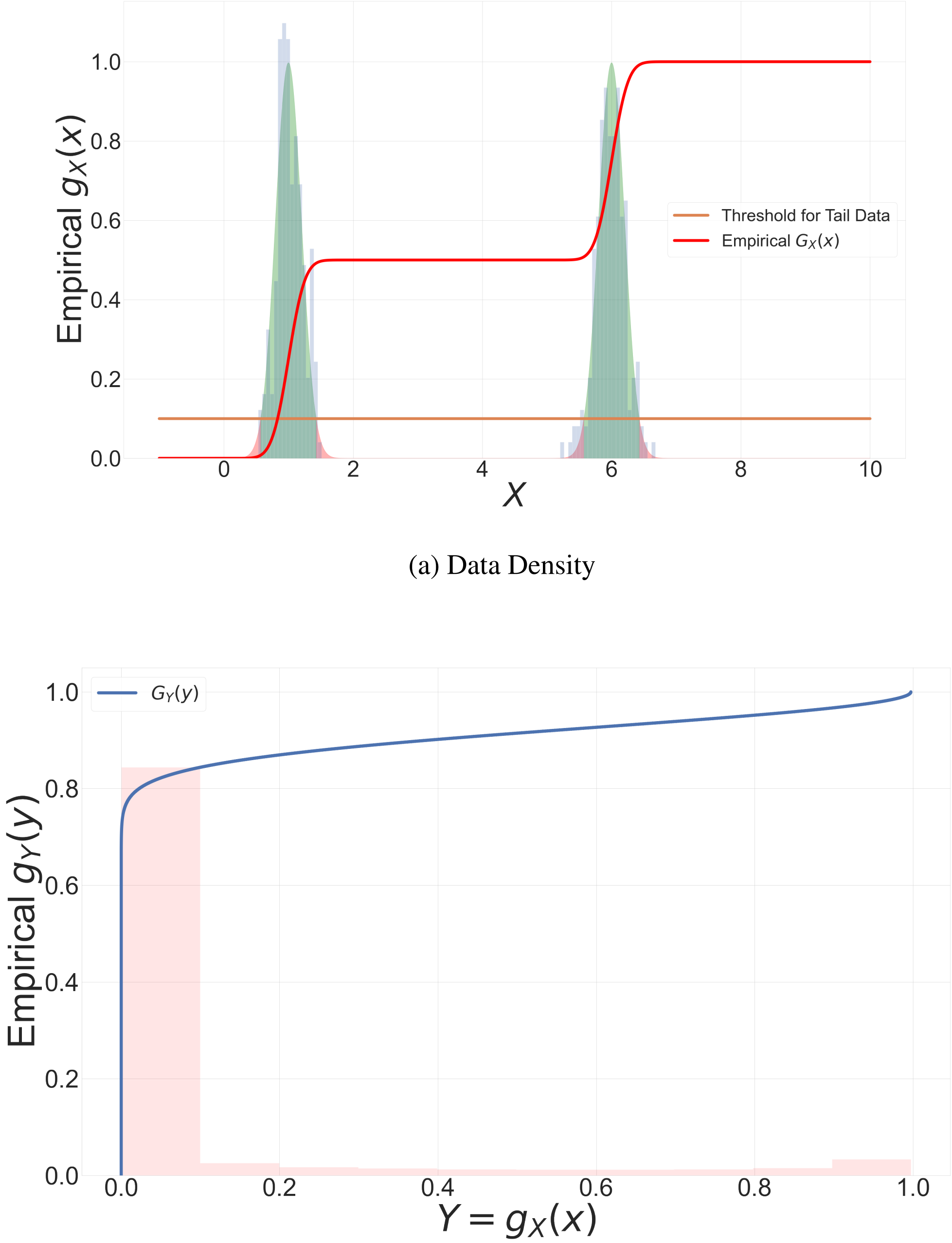
2.2.3 Ext-GPD approach

In this section, we derive the necessary conditions required for the Extended GPD approach. For this, consider the following setting in the univariate space[5].

Let $X \in \mathbb{R}$ be the data space with pdf[6] $g_X : \mathbb{R} \to \mathbb{R}^+$. Let $Y \in \mathbb{R}^+$ be the corresponding image space, i.e., $Y = g_X(X)$ and $Y_m = sup(g_X(X))$. As the limit distribution of the minima of Y is of interest, we wish to study the limit distribution of maxima of $Z = Y_m - Y$. Let

[5] The proof for the higher dimensional space is presented in the supplementary section

[6] Note: g_X^{-1} represents an image set as the function g_X is a many-to-one (non-injective) function.



(a) Data Density

(b) Extended GPD

Figure 2.3: Extended GPD distribution using Probability Image Space for Bi-modal Univariate Data. 2000 observations from two random normal distributions with mean and variance (0,2) and (6,2) respectively is shown. (a). Empirical density of the data is shown in green. The observations with probability density less than 0.1 are considered tail observations (shown in red shaded region). The empirical c.d.f. G_X is shown red. , (b). The empirical density of the probability image space is shown in red. The cumulative distribution used in the extended GPD approach is shown in blue.

the cdf of Z is given by G_Z. Then, we show that the Theorem 2 holds.

Theorem 2. *G_Z is in the maximum domain of attraction of a generalized extreme value (GEV) distribution $\mathbf{G}_\zeta^{EV}$, iff $\frac{dg_X(x)}{dz}$ and $\frac{d^2 g_X(x)}{dz^2}$ exists $\forall x \in g_X^{-1}(Y_m - z)$ in some neighborhood of Y_m.*

Proof. To derive the necessary conditions for the Ext-GDP approach, we make the following claims.

Claim 1. *G_Z is a cumulative distribution function.*

Proof. As the limit distribution of the minima of Y is of interest, we wish to study the limit distribution of maxima of $Z = Y_m - Y$. Then the cdf of Z is given by G_Z is

$$\begin{aligned} G_Z(z) &= P(Z \leq z) \\ &= P(Y_m - Y \leq z) \\ &= P(Y \geq Y_m - z) \\ &= 1 - G_Y(Y_m - z) \\ &= \int_{g_X^{-1}([Y_m - z, Y_m])} g_X(x) dx \end{aligned} \tag{2.9}$$

$\forall z \in [0, Y_m]$. □

For, G_Z, the corresponding maximum value, $x^* = Y_m$.

Claim 2. *$G_Z^{'}$ exists and is positive in some neighborhood of Y_m.*

Proof. If F be a distribution in 1D, $\exists \left\{x_1 = -\infty, x_1, x_2 \ldots, x_{2N} = \infty\right\}$ and intervals $I_1, I_2, \ldots, I_N$

such that $I_n = [x_{2n-1}, x_{2n}]\ \forall\ n = 1, 2, \ldots, N$ and $g_X^{-1}([0, Y_m - z]) = \cup_{n=1}^{N} I_n$

$$\begin{aligned} \int_{g_X^{-1}([0,Y_m-z])} g_X(x)dx &= \int_{\cup_{n=1}^{N}\ I_n} g_X(x)dx \\ &= \sum_{n=1}^{N} \int_{I_n} g_X(x)dx \\ &= \sum_{n=1}^{N} G_n(z) \end{aligned} \tag{2.10}$$

where $G_n(z) = \int_{I_n} g_X(x)dx$ and $\left\{x_1, x_2 \ldots, x_{N-1}\right\}$ are the solutions to $g_X^{-1}(Y_m - z)$.

Then,

$$\begin{aligned} G_Z^{'}(z) &= \frac{d}{dz} \int_{g_X^{-1}([Y_m-z,Y_m])} g_X(x)dx \\ &= \frac{d}{dz}\left(1 - \int_{g_X^{-1}([0,Y_m-z])} g_X(x)dx\right) \\ &= -\frac{d}{dz}\sum_{n=1}^{N} G_n(z) \end{aligned} \tag{2.11}$$

Since $G_n(z) = \int_{I_n} g_X(x)dx = \int_{x_{2n-1}}^{x_{2n}} g_X(x)dx$, by Leibniz integral rule, we get,

$$\begin{aligned} \frac{d}{dz}G_n(z) &= \frac{d}{dz}\int_{x_{2n-1}}^{x_{2n}} g_X(x)dx \\ &= g_X(x_{2n})\frac{dx_{2n}}{dz} - g_X(x_{2n-1})\frac{dx_{2n-1}}{dz} + \int_{x_{2n-1}}^{x_{2n}} \frac{d}{dz}g_X(x)dx \\ &= (Y_m - z)\left(\frac{dx_{2n}}{dz} - \frac{dx_{2n-1}}{dz}\right) \\ &= (Y_m - z)\left(\left|\frac{dx_{2n}}{dz}\right| + \left|\frac{dx_{2n-1}}{dz}\right|\right) \end{aligned} \tag{2.12}$$

Then,

$$G_Z^{'}(z) = \sum_{n=1}^{2N} (Y_m - z)\left|\frac{dx_n}{dz}\right| \tag{2.13}$$

□

Claim 3. *G_Z'' exists iff $\frac{dg_X(x)}{dz}$ and $\frac{d^2g_X(x)}{dz^2}$ exists $\forall x \in g_X^{-1}(Y_m - z)$.*

Proof.

$$\begin{aligned} G_Z''(z) &= \frac{d}{dz} G_Z'(z) \\ &= \frac{d}{dz}\left[(Y_m - z)\sum_{n=1}^{2N}\left|\frac{dx_n}{dz}\right|\right] \end{aligned} \tag{2.14}$$

It can be seen that G_Z'' exists iff $\frac{dg_X(x)}{dz}$ and $\frac{d^2g_X(x)}{dz^2}$ exists $\forall x \in \partial g_X^{-1}(Y_m - z)$ where ∂U are boundary points in U. This is true for all distributions in the exponential family. □

Claim 4. *G_Z is in the maximum domain of attraction of a generalized extreme value (GEV) distribution $\mathbf{G}_\zeta^{EV}$, where $\zeta \in \mathbb{R}$ is the rate parameter of the GEV distribution.*

Proof. By von Mises' Condition[7], and Claims 2 and 3, we can see that the G_Z' is positive and G_Z'' exists in some neighborhood of Y_m. Hence, G_Z is in domain of attraction of $\mathbf{G}_\zeta^{EV}$. □

Using Claims 1-4, we get the necessary conditions for the above claim. □

2.2.4 Ext-GPD for n-D case

The extension to the multivariate case is shown in Theorem 3.

Theorem 3. *Let $\vec{X} \in \mathbb{R}^n$ be the data space with pdf $g_{\vec{X}} : \mathbb{R}^n \to \mathbb{R}^+$. Let $Y \in \mathbb{R}^+$ be the corresponding image space. Let $\vec{X} \in \mathbf{R}^n$ and $g_{\vec{X}}^{-1}([0, Y_m - z]) = D(Y_m - z)$ be a n-manifold with a boundary $\partial D(Y_m - z)$. G_Z is in the maximum domain of attraction of a generalized extreme value (GEV) distribution iff :*

1. *$D(Y_m - z)$ is an n-manifold with a boundary $\partial D(Y_m - z)$,*

[7] **von Mises' Condition:** Let F be a distribution function and x^* is its right end point. Suppose F'' exists and F' is positive for all x in some neighborhood of x^*. If $\lim_{t\to x^*}\left(\frac{1-F}{F'}\right)'(t) = \zeta$ then, F is in the MDA of $\mathbf{G}_\zeta^{EV}$.

2. *The Eulerian velocity of the boundary* $\vec{\boldsymbol{v}}_b = \frac{dD(Y_m-z)}{dz}$ *exists,*

3. $d_x\left[g_{\vec{\mathbf{X}}}(\vec{\boldsymbol{x}})\vec{\boldsymbol{v}}_b \cdot d\boldsymbol{\Sigma}\right]$ *exists and*

4. $i_{\vec{v}}\left(d_x\left[g_{\vec{\mathbf{X}}}(\vec{\boldsymbol{x}})\vec{\boldsymbol{v}}_b \cdot d\boldsymbol{\Sigma}\right]\right)$ *exists.*

.

Let $X \in \mathbb{R}^n$ be the data space with pdf $g_{\vec{\mathbf{X}}} : \mathbb{R}^n \to \mathbb{R}^+$. Let $Y \in \mathbb{R}^+$ be the corresponding image space.

Definition 2. *$\forall y \in Y$, G_Y is defined as*

$$G_Y(y) = \int_{g_{\vec{\mathbf{X}}}^{-1}([0,y])} g_{\vec{\mathbf{X}}}(x)dx \tag{2.15}$$

Claim 5. *G_Y is a cumulative distribution function.*

As the limit distribution of the minima of Y is of interest, we wish to study the limit distribution of maxima of $Z = Y_m - Y$. Then the cdf of Z is given by G_Z is

$$\begin{aligned} G_Z(z) &= P(Z \leq z) \\ &= P(Y_m - Y \leq z) \\ &= P(Y \geq Y_m - z) \\ &= 1 - G_Y(Y_m - z) \\ &= \int_{g_{\vec{\mathbf{X}}}^{-1}([Y_m-z,Y_m])} g_{\vec{\mathbf{X}}}(x)dx \end{aligned} \tag{2.16}$$

$\forall z \in [0, Y_m]$.

For, G_Z, the corresponding maximum value, $x^* = Y_m$.

We need the necessary conditions for the above distribution to be in the domain of attraction of a GEV distribution. By von Mises' Condition, if we can prove that G_Z' is positive and G_Z'' exists in some neighborhood of Y_m, then G_Z is in domain of attraction of G_γ.

Proof. Let $\vec{\mathbf{X}} \in \mathbf{R}^n$ and $g_{\vec{\mathbf{X}}}^{-1}([0, Y_m - z]) = D(Y_m - z)$ be a n-manifold with a boundary $\partial D(Y_m - z)$. Then,

$$\begin{aligned} G_Z^{'}(z) &= \frac{d}{dz} \int_{g_{\vec{\mathbf{X}}}^{-1}([Y_m - z, Y_m])} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) d\vec{\mathbf{x}} \\ &= \frac{d}{dz} \left[1 - \int_{g_{\vec{\mathbf{X}}}^{-1}([0, Y_m - z])} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) d\vec{\mathbf{x}} \right] \\ &= -\frac{d}{dz} \int_{D(Y_m - z)} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) d\vec{\mathbf{x}} \end{aligned} \tag{2.17}$$

where, $d\vec{\mathbf{x}} = dx_1 \wedge dx_2 \wedge ... \wedge dx_n$.

Then, using Reynolds transport theorem, we get,

$$\begin{aligned} \frac{d}{dz} G(z) &= \frac{d}{dz} \int_D g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) d\vec{\mathbf{x}} \\ &= \int_{D(Y_m - z)} \frac{\partial}{\partial z} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}})\, dV + \int_{\partial D(Y_m - z)} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \end{aligned} \tag{2.18}$$

where $g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}})$, $D(Y_m - z)$ and $\partial D(Y_m - z)$ are as defined above, $\vec{\mathbf{v}}_b = \frac{dD(Y_m - z)}{dz}$ is the Eulerian velocity of the boundary, $\mathbf{n}$ is the outward unit normal, dS is the surface element in $\mathbf{R}^d$ and $d\mathbf{\Sigma} = \mathbf{n} dS$.

Since, $\frac{\partial}{\partial z} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) = 0$,

$$G_Z^{'}(z) = \int_{\partial D(Y_m - z)} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \tag{2.19}$$

Claim 2: $G_Z^{''}$ exists.

$$\begin{aligned} G_Z^{''}(z) &= \frac{d}{dz} G_Z^{'}(z) \\ &= \frac{d}{dz} \int_{\partial D(Y_m - z)} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \end{aligned} \tag{2.20}$$

Since, $\partial D(Y_m - z)$ an (n-1)-closed manifold, i.e. (n-1)-manifold without a boundary, we use the general statement of the Leibniz integral rule to compute the second order derivative,

$$\begin{aligned} G_Z''(z) &= \frac{d}{dz} G_Z'(z) \\ &= \frac{d}{dz} \int_{\partial D(Y_m - z)} g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \\ &= \int_{\partial D(Y_m - z)} i_{\vec{\mathbf{v}}} \left(d_x \left[g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \right] \right) \end{aligned} \tag{2.21}$$

where, $d_x\, f$ is the exterior derivative of f w.r.t space variables only, $\vec{\mathbf{v}} = \frac{\partial \vec{\mathbf{x}}}{\partial z}$ is the vector field of the velocity and $i_{\vec{\mathbf{v}}}$ denotes the interior product with $\vec{\mathbf{v}}$.

□

Thus, it can be seen that G_Z is in the maximum domain of attraction of a generalized extreme value (GEV) distribution iff :

1. $D(Y_m - z)$ is an n-manifold with a boundary $\partial D(Y_m - z)$,

2. The Eulerian velocity of the boundary $\vec{\mathbf{v}}_b = \frac{dD(Y_m - z)}{dz}$ exists,

3. $d_x \left[g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \right]$ exists and

4. $i_{\vec{\mathbf{v}}} \left(d_x \left[g_{\vec{\mathbf{X}}}(\vec{\mathbf{x}}) \vec{\mathbf{v}}_b \cdot d\mathbf{\Sigma} \right] \right)$ exits.

.

Chapter 3

Integrated Clustering and Anomaly Detection (INCAD) for Streaming Data

3.1 Anomaly Detection using EVT for unimodal data

EVT plays a significant role in studying rare events and so, several methods have been proposed that incorporate these features in anomaly detection. Here, we present a novel methodology which involves both EVT and non-parametric modeling for anomaly detection. The core principles that lead to the development of the integrated algorithm are discussed here. We start with a basic case of one cluster data with anomalies.

Based on the EVT concepts discussed above, we first propose a simple anomaly detec-

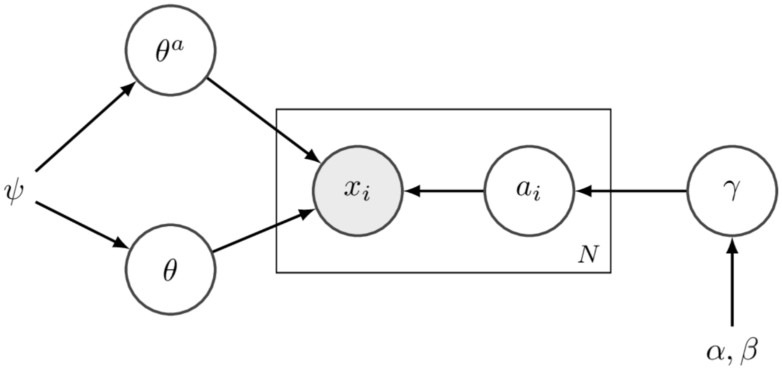


Figure 3.1: Graphical representation of the proposed probabilistic model: The figure illustrates the generation of non-anomalous and anomalous observations from two distinct prior distributions for a unimodal case.

tion model (See Figure 3.1), which is equivalent to the following generative distributions:

$$\theta|\psi \sim \mathtt{G}_0(\psi) \tag{3.1}$$

$$\theta^a|\psi \sim \mathtt{G}_0^{\mathtt{EV}}(\psi) \tag{3.2}$$

$$\gamma|\alpha, \beta \sim \mathtt{Beta}(\alpha, \beta) \tag{3.3}$$

$$a_i|\gamma \sim \mathtt{Bernoulli}(\gamma) \tag{3.4}$$

$$x_i|a_i, \theta, \theta^a \sim \begin{cases} G(\theta) & \text{if } a_i = 1 \\ G(\theta^a) & \text{if } a_i = -1 \end{cases} \tag{3.5}$$

The model is a mixture of two components, $\mathcal{N}$ and $\mathcal{A}$, parameterized by θ and θ^a, respectively. a_i is an indicator latent variable denoting if x_i is normal or anomalous, and γ is the mixture weight with a Beta distribution prior.

The mixture of models representation allows us to sketch a Gibbs sampling based inference scheme, similar to a mixture model [29], using the following conditional posteriors:

$$p(\gamma|\mathbf{a}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) = \mathtt{Beta}(\alpha + n^a, \beta + n - n^a) \tag{3.6}$$

where **x** denotes the vector of n observed data instances, **a** is a binary indicator vector, i.e., $a_i = -1 \Rightarrow x_i$ is anomalous, and n^a is the number of anomalous instances. The posteriors for the indicators can be computed as:

$$p(a_i = -1|\mathbf{a}_{-i}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto \gamma p_G(x_i|\theta^a) \tag{3.7}$$

$$p(a_i = 1|\mathbf{a}_{-i}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto (1-\gamma) p_G(x_i|\theta) \tag{3.8}$$

Finally, the posteriors for the mixture parameters, θ and θ^a, can be computed as:

$$p(\theta|\mathbf{a}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto p_{\mathtt{G}_0}(\theta|\psi) \prod_{i:a_i=1} p_G(x_i|\theta) \tag{3.9}$$

$$p(\theta^a|\mathbf{a}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto p_{\mathtt{G}_0^{\mathtt{EV}}}(\theta^a|\psi) \prod_{i:a_i=-1} p_G(x_i|\theta^a) \tag{3.10}$$

Starting from an initial estimate of the latent variables, γ, **a**, θ, and θ^a, the inference can be done via Gibbs update, in which new estimates for the latent variables are sampled from the conditional posteriors given in (3.6), (3.8) and (3.10), respectively.

3.1.1 Modified posterior expressions

Let y_i denote the pdf of an observation x_i according the to the normal distribution, i.e., $y_i = p_G(x_i|\theta)$. Using a threshold u[1], we define the "tail" of the distribution $\mathtt{G}_Y$ using samples $\{y_i\}_{i:y_i \le u}$. A GPD, $\mathtt{G}_Y^{EV}$, is fitted on the samples $\{u - y_i\}_{i:y_i \le u}$. The conditional posteriors for a_i for tail instances can be written as:

$$p(a_i = -1|\mathbf{a}_{-i}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto \gamma(1 - P_Y^{EV}(u - y_i)) \tag{3.11}$$

$$p(a_i = 1|\mathbf{a}_{-i}, \mathbf{x}, \theta, \theta^a, \alpha, \beta, \psi) \quad \propto (1-\gamma) P_Y^{EV}(u - y_i) \tag{3.12}$$

[1] Note that u is not a threshold for determining if an observation is anomalous or not; instead, it defines the "tail" of the original distribution, which are then used to determine the parameters of the corresponding GPD.

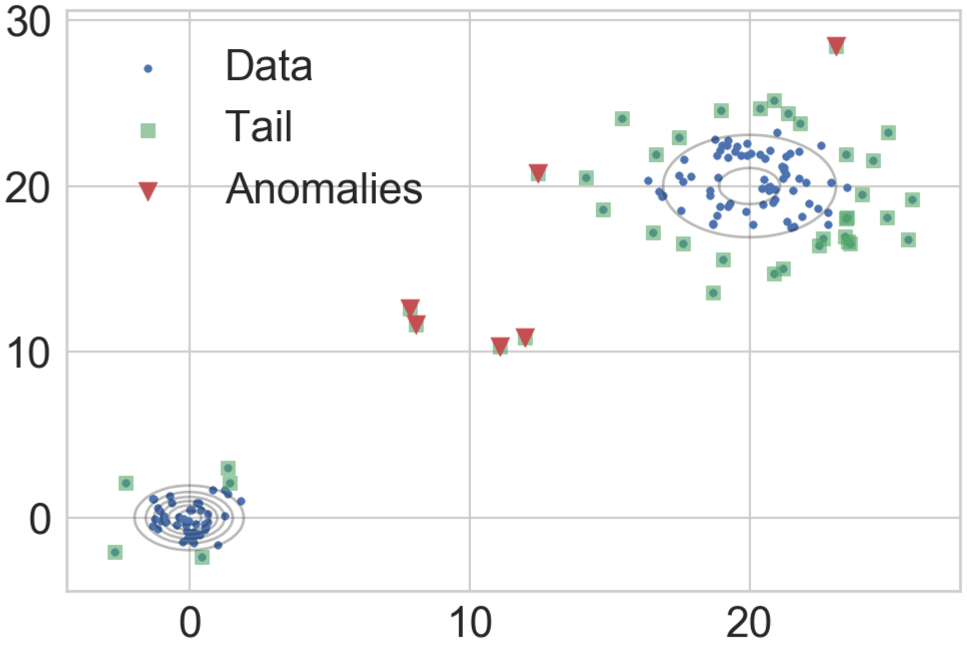


Figure 3.2: Results for a synthetic 2D case, with a fixed Gaussian mixture model as $\mathtt{G}_0$. The model identifies the anomalies (red) with respect to the tail of $\mathtt{G}_0$ (green) as well as the parameters for $\mathtt{G}_0$ (shown as contour lines).

where $P_Y^{EV}(u - y_i)$ is the probability of observing y_i in the tail of $\mathtt{G}_Y$. Since GPD is a unimodal distribution, we use the survival function value, $1 - G_Y^{EV}(y - u_i)$, instead of the exact probability. For non-tail instances, i.e., $y_i > u$, the conditional probability $p(a_i = -1| \ldots)$ is set to 0. Under this modified model, computing the posterior for θ^a in (3.10) is not needed anymore. If the form of the normal model is known, e.g., a unimodal Gaussian or a mixture of Gaussians[2] (See Figure 3.2), the anomalies and the model parameters can be inferred via Gibbs sampling, using the above mentioned conditional distributions. However, in the next section we show how the Bayesian formulation can be extended to a richer class of the base distribution, $\mathtt{G}_0$, i.e., non-parametric mixture models.

[2] In presence of multiple clusters, the prior $\mathtt{G}_0$ can be chosen as a mixture of individual priors generating the non-anomalous components ensuring that low probability or tail region of the distribution is associated with generating parameters associated with anomalous components.

Challenges If $\mathtt{G}_0$ is the conjugate prior of G, one can get an analytical form for the posterior in (3.10). The posterior for θ^a is the main challenge here, for two reasons: a). $\mathtt{G}_0^{\mathtt{EV}}$ exists only for a limited base distributions, $\mathtt{G}_0$, and, b). even for known $\mathtt{G}_0^{\mathtt{EV}}$, it is unlikely that the posterior in (3.10) will have an analytical form.

We first note that the quantity $p_G(x_i|\theta^a)$ is the probability of the observation x_i to be generated by the distribution G, parameterized by θ^a, which, in turn, is sampled from the EVD for $\mathtt{G}_0$, i.e., $\mathtt{G}_0^{\mathtt{EV}}$.

For distributions belonging to the exponential family, one can show that if $\mathtt{G}_0$ is the conjugate of G, then sampling x_i from $G(.|\theta^a)$, where $\theta^a \sim \mathtt{G}_0^{\mathtt{EV}}$, is equivalent to (under expectation): first sampling θ from $\mathtt{G}_0$, and then sampling x_i from the EVD of G (or G^{EV}), parameterized by θ, i.e., $\mathbb{E}_{\theta^a \sim \mathtt{G}_0^{\mathtt{EV}}}[p_G(x_i|\theta^a)] = \mathbb{E}_{\theta \sim \mathtt{G}_0}[p_{G^{EV}}(x_i|\theta)]$.

We show that this claim will hold for the following simple setting, and omit the general proof in the interest of space. Let $G \sim \mathcal{N}(\mu, 1)$, i.e., G is a univariate Gaussian distribution with fixed variance and the mean is generated from a Gaussian prior, i.e., $\mathtt{G}_0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$. Note that the EVD for a Gaussian distribution is a *Gumbel* distribution, i.e., $\mathtt{G}_0^{\mathtt{EV}} \sim \mathit{Gumbel}(\mu_0, \sigma_0)$.

Assuming that x_i is an anomaly, i.e., x_i is sampled from a Gaussian, $\mathcal{N}(\mu^a, 1)$, where $\mu^a \sim \mathit{Gumbel}(\mu_0, \sigma_0)$, then we can show that for any $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, the probability that x_i is *not in the tail* of $\mathcal{N}(\mu, 1)$ will be very small, since:

$$\begin{aligned}
&\mathbb{E}_{X\sim\mathcal{N}(\mu^a,1)}[\mathtt{G}_{\mathcal{N}(\mu,1)}(X)]\\
&= \int \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2}\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu^a)^2}{2}\right) dx\\
&\propto \exp\left(-\frac{(\mu-\mu^a)^2}{4}\right)
\end{aligned}$$

Thus the claim will hold in this case because the prior distribution is Gaussian, for which $|\mu - \mu^a| \gg 0$.

3.2 Extension to Data with Multiple Clusters

While the previous result is an interesting step towards explicitly modeling the anomaly distribution, it is still limited to the case where the normal data is being generated from a single cluster. A natural extension to the presented preliminary model is the scenario where the normal data could be generated from multiple clusters. The key challenge in performing anomaly detection on such data is the method to identify the generative model that is robust to anomaly presence.

Why integrate Extreme Value Theory and DPMM? Anomalies with significantly large deviations are inherently caught by most anomaly detection algorithms including traditional DPMM. The distinction between the algorithms is observed when identifying anomalies with relatively similar behavior to normal data. Such anomalies are found in the vicinity of clusters and are often clustered into being normal. Traditional DPMM algorithm can identify such anomalies by increasing the concentration parameter but the choice of the new value has the same challenges as the choice of a threshold thus arising a need for an external algorithm like EVT that studies these tail points separately and an integrated approach would ensure enhanced and robust clustering.

3.2.1 Background on Mixture Models

Finite Mixture Models (FMM) are a useful clustering tool to identify and study sub-populations within data. However, they require pre-specifying the number of clusters, which is not always known. This is especially important for anomalous data for which accurate knowledge is not available, and can lead to some significantly inaccurate (and in some cases unreliable) interpretations of the data. Non-parametric mixture models, e.g., *Dirichlet Process Mixture Models* (DPMM) (Frigyik, Kapila, and Gupta, 2010), can be used in such settings.

Dirichlet Process Mixture Models A DPMM can be thought of as an infinite extension of a finite mixture model (FMM), which is equivalent to the following distributions:

$$\boldsymbol{\pi}|\alpha \sim \text{Dir}(\alpha/K, \ldots, \alpha/K) \tag{3.13}$$

$$z_i|\boldsymbol{\pi} \sim \text{Multi}(\boldsymbol{\pi}) \tag{3.14}$$

$$\theta_k|\psi \sim \mathtt{G}_0(\psi) \tag{3.15}$$

$$x_i|z_i, \{\theta_k\}_{k=1}^K \sim \boldsymbol{G}(\theta_{z_i}) \tag{3.16}$$

Each observation x_i is generated by first sampling a cluster index, z_i from a Multinomial distribution, parameterized by a K length vector, $\boldsymbol{\pi}$. A symmetric Dirichlet prior is used to generate $\boldsymbol{\pi}$. The observations are sampled from a cluster specific distribution, $\boldsymbol{G}$, parameterized by θ_k. The cluster specific distribution parameters are also generated from a prior (or base) distribution, $\mathtt{G}_0$, parameterized by ψ.

A DPMM is an extension of FMM to the case where $K \to \infty$. While several equivalent representations of DPMM exist, we will use the *Stick Breaking* representation, which shows DPMM as a natural extension of FMM. The stick breaking representation allows sampling the mixture weights, with possibly infinite components, as follows:

- Start with a unit-length stick and break it according to β_1, where $\beta_1 \sim \text{Beta}(1, \alpha_0)$, and assign β_1 to π_1;
- Break remaining stick according to the proportion $\beta_k \sim \text{Beta}(1, \alpha_0)$ and assign β_k portion of the remaining stick to π_k.

The sequence $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^{\infty}$ satisfies $\sum_{k=1}^{\infty} \pi_k = 1$ and is typically written as $\boldsymbol{\pi} \sim \mathtt{GEM}(\alpha)$[3].

[3] named after Griffiths, Engen, and McCloskey

3.2.2 Integrated Non-parametric Clustering and Anomaly Detection (INCAD)

We propose an instance of the general Bayesian anomaly detection algorithm described in Section 3.1 which uses a DPMM as its base distribution, $\mathtt{G}_0$. The generative model (See Figure 3.3) consists of two coupled DPMM models, each corresponding to the normal and anomalous behaviors, respectively, and is equivalent to the following distributions[4]:

$$\begin{aligned}
\boldsymbol{\pi}|\alpha &\sim \mathtt{GEM}(\alpha) && (3.17)\\
\boldsymbol{\pi}^a|\alpha^* &\sim \mathtt{GEM}(\alpha^*) && (3.18)\\
\theta_k|\psi &\sim \mathtt{G}_0(\psi) && (3.19)\\
\theta_k^a|\psi &\sim \mathtt{G}_0^{EV}(\psi) && (3.20)\\
sign(z_i)|\gamma &\sim \text{Bernoulli}(\gamma) && (3.21)\\
|z_i|\Big|\boldsymbol{\pi}, \boldsymbol{\pi}^a, sign(z_i) &\sim \begin{cases} \text{Multi}(\boldsymbol{\pi}) & \text{if } sign(z_i) = 1 \\ \text{Multi}(\boldsymbol{\pi}^a) & \text{if } sign(z_i) = -1 \end{cases} && (3.22)\\
x_i|z_i, \{\theta_k\}_{k=1}^{\infty}, \{\theta_k^a\}_{k=1}^{\infty} &\sim \begin{cases} \text{G}(\theta_{|z_i|}) & \text{if } sign(z_i) = 1 \\ \text{G}(\theta_{|z_i|}^a) & \text{if } sign(z_i) = -1 \end{cases} && (3.23)
\end{aligned}$$

The key difference from the model in Section 3.1 is the additional variable, z_i, that works as the cluster labels as well as anomaly indicator. The $sign(z_i)$ represents presence of anomalous behavior where anomalous (or non-anomalous) observations are assigned

[4] GEM is a recursive process with an infinite number of clusters of which only a finite number of them are populated. The number of the populated clusters as well as the corresponding proportions are learned sequentially as seen in the stick breaking process.

Since the true number of clusters is unknown, Dirichlet process priors, like the GEM distribution, are traditionally used to sample the vectors $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^a$. When sampling from the GEM distribution, we generate a vector (of unknown but finite length) from a simplex that sums to one (as seen in the stick breaking approach). The vector length can be regulated using the concentration parameter (large concentration parameter returns more number of populated clusters i.e. vector of longer length).

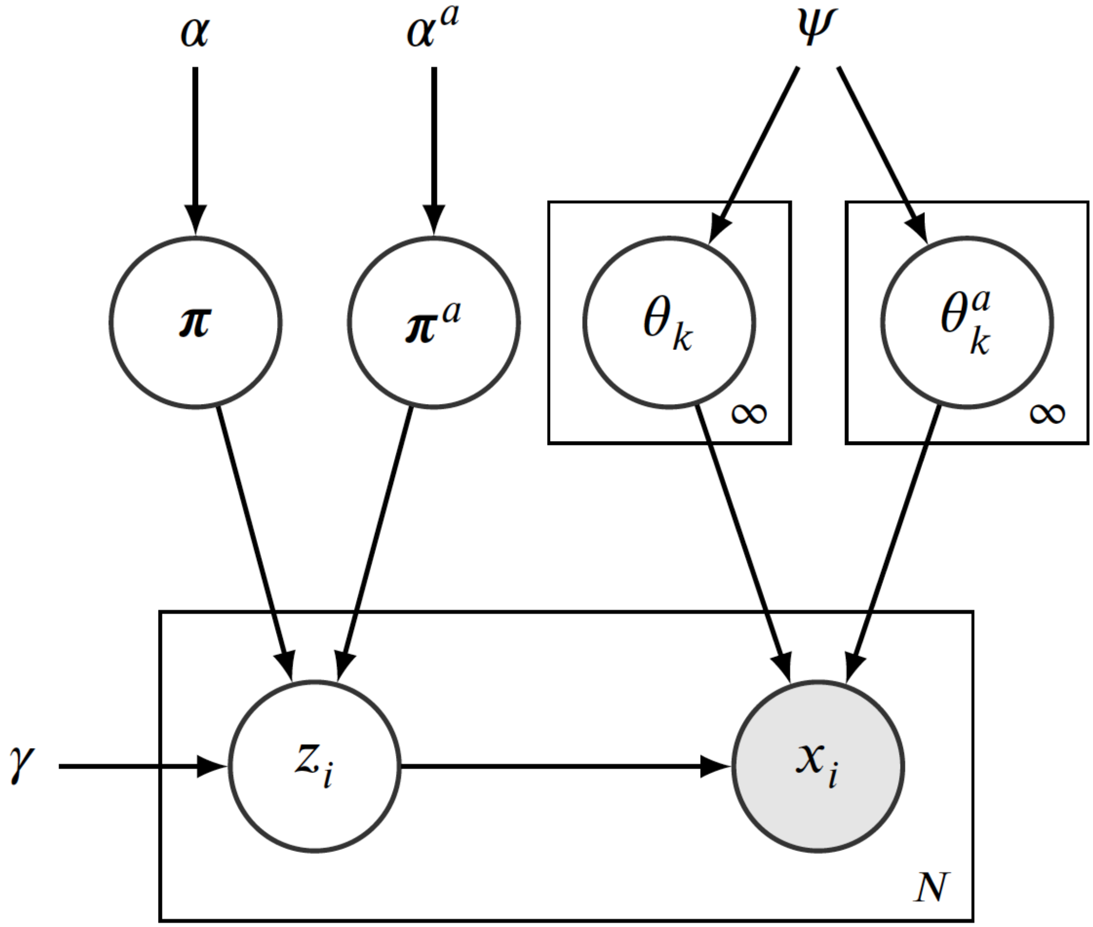


Figure 3.3: Graphical representation of the proposed INCAD model: The figure illustrates the generative process for observations in a mixture distribution.

negative (or positive) labels. Based on the observed labels, anomalies can be classified into global, local and group anomalies.

Definition 3 (Global Anomalies)**.** *A single observation is defined as a group anomaly if it is a observation with distinctly novel behavior. INCAD classifies such observations into singleton clusters with negative cluster labels.*

Definition 4 (Group Anomalies)**.** *Multiple observations with similar behavior that is distinct from existing predominant behaviors (normal clusters) are classified as group anomalies. Such observations are classified into smaller clusters with negative cluster labels.*

Definition 5 (Local Anomaly)**.** *Observations with behaviors that moderately deviate from normal clusters but not distinct enough to form individual clusters are defined as local anomalies. Such observations are classified into normal clusters with similar behavior but with negative labels to indicate diverging behavior. Anomalies that originate from an overlapping anomalous cluster are often classified as local anomalies.*

Since labels are assigned considering both clustering as well as anomaly detection, we call this model, INCAD (*Integrated Non-parametric Clustering and Anomaly Detection*). Based on $sign(z_i)$, z_i is sampled from a Multinomial distribution that is either parameterized by $\boldsymbol{\pi}$ (if $sign(z_i) = 1$) or $\boldsymbol{\pi}^a$ (if $sign(z_i) = -1$). The Multinomial parameters, $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^a$ are sampled from the *Stick Breaking* construction of a Dirichlet process, i.e., $\boldsymbol{\pi} \sim \texttt{GEM}(\alpha)$ and $\boldsymbol{\pi}^a \sim \texttt{GEM}(\alpha^*)$.

The INCAD model goes beyond the illustrated simple case where we assume multiple anomalous sources, each associated with a different concentration parameter α^*. The generative model can now be seen as a collection of multiple DPMMs of which all but one DPMM can be perceived as sources for anomalous data and the set of concentration parameters for anomalous data, $\{\alpha^*_d\}$, would dictate the corresponding DPMM's cluster proportions $\{\boldsymbol{\pi}^a_d\}$.

Inference for the INCAD model includes inferring posteriors for $(z_i)_{i=1}^n$, $(\theta_k, \theta_k^a)_{k=1}^\infty$. While this follows the general Gibbs sampling based scheme discussed in Section 3.1 (omit-ting exact details in the interest of space), there are some additional issues that are unique to the INCAD model. In particular, the dependency between z_i and $sign(z_i)$ in Figure 3.3 means that one cannot consider the model as a straightforward mixture for two DPMMs. However, the relationship between the normal and anomalous model parameters, via the EVT construct, means that we can calculate the posteriors for $sign(z_i)$ using the modification proposed earlier (See (3.12)).

Inference when $\mathtt{G}_0^{\mathtt{EV}}$ is available

MCMC and variational inference based algorithms [58, 10] have been typically used for inference of the computationally expensive infinite mixture models. For INCAD, we adopt an extension of a Gibbs sampling-based method for a fixed mixture model that allows room for additional cluster formation. The algorithm is inspired by the sampling based MCMC method for conjugate priors (Algorithm 1 [58]). Here, new clusters comprise anomalous observations identified using EVT.

Gibbs Sampling The anomaly classification variable $sign(z_.)$ is a unique feature of INCAD that distinguishes it from traditional DPMM. Thus, the posterior probabilities for the latent variables namely, the number of clusters K, cluster and anomaly indicators $\{z_i\}_{i=1}^N$ are computed using Markov property and Bayes rule:

$$\begin{aligned} P(|z_i| = k \Big| \quad & x_., z_{-i}, \alpha, \alpha^*, \boldsymbol{\pi}, \boldsymbol{\pi}^a, \psi, \left\{\theta_k\right\}, \left\{\theta_k^a\right\}, sign(z_.), \gamma) \\ = \quad & P(|z_i| = k \Big| x_., z_{-i}, \alpha, \alpha^*, \left\{\theta_k\right\}, \left\{\theta_k^a\right\}, sign(z_i)) \end{aligned}$$

$$\propto \begin{cases} P(|z_i| = k \Big| z_{-i}, \alpha, \theta_k) P(x_i \Big| |z_i| = k, z_{-i}, \theta_k, \alpha) & , sign(z_i) = 1 \\ \\ P(|z_i| = k \Big| z_{-i}, \alpha^*, \theta_k^a) P(x_i \Big| |z_i| = k, z_{-i}, \theta_k^a, \alpha^*) & , sign(z_i) = -1 \end{cases}$$

$$= \begin{cases} \frac{n_k}{(n+\alpha-1)} G(x_i \,\Big|\, \theta_k) & , sign(z_i) = 1 \\ \frac{n_k}{(n+\alpha^*-1)} G(x_i \,\Big|\, \theta_k^a) & , sign(z_i) = -1 \end{cases} \tag{3.24}$$

where $\alpha^* = \frac{1}{1-p_i}$, p_i is the probability of x_i being anomalous, n_k is the number of observations in the k^{th} cluster and K is the number of non-empty clusters. In the improved versions of INCAD, p_i is the cumulative density function for the extreme value distribution.

The posterior probability of forming a new cluster denoted by $K+1$ is given by:

$$P(|z_i| = K+1 \,\Big|\, x_., z_{-i}, \alpha, \alpha^*, \boldsymbol{\pi}, \boldsymbol{\pi}^a, \psi, \left\{\theta_k\right\}, \left\{\theta_k^a\right\}, sign(z_.), \gamma)$$
$$= P(|z_i| = K+1 \,\Big|\, x_i, z_{-i}, \alpha, \alpha^*, \psi, sign(z_i))$$

$$\propto \begin{cases} P(|z_i| = K+1 \,\Big|\, z_{-i}, \alpha, \psi) P(x_i \,\Big|\, |z_i| = K+1, z_{-i}, \alpha, \psi, sign(z_i)) & , sign(z_i) = 1 \\ \\ P(|z_i| = K+1 \,\Big|\, z_{-i}, \alpha^*, \psi) P(x_i \,\Big|\, |z_i| = K+1, z_{-i}, \alpha^*, \psi, sign(z_i)) & , sign(z_i) = -1 \end{cases}$$

$$= \begin{cases} \frac{\alpha}{n+\alpha-1} \int G(x_i \,\Big|\, \theta) \mathtt{G}_0(\theta \,\Big|\, \psi) d\theta & , sign(z_i) = 1 \\ \frac{\alpha^*}{n+\alpha^*-1} \int G(x_i \,\Big|\, \theta^a) \mathtt{G}_0^{\mathtt{EV}}(\theta^a \,\Big|\, \psi) d\theta^a & , sign(z_i) = -1 \end{cases} \tag{3.25}$$

Similarly, the parameters for clusters $k \in \{1, 2, \ldots, K\}$ are sampled from:

$$\theta_k \propto \mathtt{G}_0(\theta_k \,\Big|\, \psi) \mathcal{L}(\boldsymbol{x}_k \,\Big|\, \theta_k) \quad \text{if cluster is not anomalous} \tag{3.26}$$

$$\theta_k^a \propto \mathtt{G}_0^{\mathtt{EV}}(\theta_k^a \,\Big|\, \psi) \mathcal{L}(\boldsymbol{x}_k \,\Big|\, \theta_k^a) \quad \text{if cluster is anomalous} \tag{3.27}$$

where $\boldsymbol{x}_k = \{x_i \Big| |z_i| = k\}$ is the set of all points in cluster k. Finally, to identify the anomaly classification of the data, the posterior probability of $sign(z_i)$ is given by:

$$P\left(sign(z_i) = -1 \,\middle|\, x_., |z_.|, \alpha, \alpha^*, \boldsymbol{\pi}, \boldsymbol{\pi}^a, \psi, \left\{\theta_k\right\}, \left\{\theta_k^a\right\}, \gamma\right) = P(sign(z_i) = -1 \,|\, x_i, |z_.|, \alpha^*, \psi, \left\{\theta_k^a\right\}, \gamma)$$

$$\propto \sum_{k=1}^{K+1} P(sign(z_i) = -1 \,|\, x_i, |z_i| = k, z_{-i}, \alpha^*, \psi, \left\{\theta_k^a\right\}, \gamma) * P(|z_i| = k \,|\, x_i, z_{-i}, \alpha^*, \psi, \left\{\theta_k^a\right\}, \gamma)$$

$$= \sum_{k=1}^{K} P(x_i \,|\, \theta_k^a)\gamma \frac{n_k}{(n + \alpha^* - 1)} + \left(\int G(x_i \,|\, \theta^a)\mathtt{G}_0^{\mathtt{EV}}(\theta^a \,|\, \psi) d\theta^a\right) \gamma \frac{\alpha^*}{n + \alpha^* - 1} \quad (3.28)$$

Similarly,

$$P(sign(z_i) = 1 \,|\, x_i, |z_.|, \alpha, \psi, \left\{\theta_k\right\}, \gamma)$$

$$\propto \sum_{k=1}^{K} P(x_i \,|\, \theta_k)(1 - \gamma) \frac{n_k}{(n + \alpha - 1)} + \left(\int G(x_i \,|\, \theta)\mathtt{G}_0(\theta \,|\, \psi) d\theta\right) (1 - \gamma) \frac{\alpha}{n + \alpha - 1} (3.29)$$

Inference when $\mathtt{G}_0^{\mathtt{EV}}$ is not available

Existence of a tail distribution $\mathtt{G}_0^{\mathtt{EV}}$ is not always feasible. As the extreme value distribution might not belong to the family of the conjugate priors of G, we assume $\theta^a \sim \mathtt{G}_0$ for sampling the parameters $\{\theta_k^a\}_{k=1}^{\infty}$ for anomalous clusters. Here, we perform rejection sampling to sample observations from the tail distribution. For this, we initially sample P observations from G_0 and isolate observations with probability density less than a set threshold[5] $0 < t << 1$. The above procedure is repeated M times till sufficient samples S_{tail} from the tail distribution have been identified. The cluster means $\{\theta_k^a\}_{k=1}^{\infty}$ can be estimated by randomly sampling from the tail observations S_{tail}. However, this could result in potential

[5] The choice of threshold governs the range of values that can be considered in the tail. Larger threshold allows wider sample range and therefore, better parameter estimation. However, collecting extreme tail samples using rejection sampling could be difficult when using larger thresholds. It must be noted that optimal choice specific to the data can be made based on the data distribution. In our analysis, we set the threshold to 15% (probability density) for ease of sampling.

convergence issues. Thus, we propose the closest observation in S_{tail} to the sample estimate for the respective anomalous cluster.

The pseudo-Gibbs sampling algorithm, presented in Algorithm 2, has been designed to address the cases when $\mathtt{G}_0^{\mathtt{EV}}$ is not available. For such cases, the modified concentration parameter α^* is given by the function f where,

$$f(\alpha|x_n, \mathbf{x}, \mathbf{z}) = \begin{cases} \alpha & , if\ not\ in\ tail \\ \frac{1}{1-p_n} & , if\ in\ tail \end{cases} \tag{3.30}$$

where, p_n is the cumulative density of x_n for the extreme value distribution of the tail data[6] where, the cumulative density is given by the Extended Generalized Pareto Distribution described in Section 2.2.3.

Non-Exchangeability and Evolution Detection in Stream

Exchangeable models are robust to alterations in the order of the sequence of observations. However, for streaming data that evolves over time, it can be costly to assume exchangeability among the observations. The instances that mark the beginning of an evolution are captured and monitored in INCAD. Additionally, relapse of outdated and non-prevalent behaviors are identified and evaluated. These features are possible due to the non-exchangeable nature of the INCAD model.

To further understand the non-exchangeable nature of INCAD, one can look at the joint probability of the cluster assignments for the INCAD model,

$$P(z_1, z_2, ..z_n|\mathbf{x}) = P(z_1|\mathbf{x})P(z_2|z_1, \mathbf{x})..P(z_n|z_{1:n-1}, \mathbf{x}) \tag{3.34}$$

[6] The left and right continuous inverses of the function $\frac{1}{1-\mathtt{G}_0^{\mathtt{EV}}(.)}$ are broadly studied in Extreme Value theory to understand the behavior of the tail distributions.

Algorithm 1 Gibbs Sampling Algorithm when $\mathtt{G}_0^{\mathtt{EV}}$ is available

```
Given $z_\cdot^{(t-1)}, \left\{ \theta_k^{(t-1)} \right\}, \left\{ \theta_k^{a(t-1)} \right\}$ from iteration $(t-1)$. Let $K$ be the total number of clusters at iteration $(t-1)$.
Set $z_\cdot = |z_\cdot^{(t-1)}|$ and $a_\cdot = sign(z_\cdot^{(t-1)})$
for each observation $i$ do
    Remove $x_i$ from its cluster $z_i$..
    if $x_i$ is the only point in its cluster then
        Remove the cluster and update K to K-1.
    end if
    Drop empty clusters.
    Sample $z_i$ from the Multinomial distribution given by Equations 3.24 and 3.25
    if $z_i = K + 1$ then
        Sample new cluster parameters from the following distribution. [7]

            $$\theta \left| x_i, z_\cdot, \left\{ \theta_k^{(t-1)} \right\}, \left\{ \theta_k^{a(t-1)} \right\}, a_\cdot^{(t-1)} \right. \tag{3.31}$$

            $$\propto \begin{cases} \alpha \mathtt{G}_0(\theta|\psi)G(x_i|\theta) + \sum_{j\neq i} G(x_i|\theta_{z_j})\delta(\theta - \theta_{z_j}^{(t-1)})\delta(a_j^{(t-1)}) & , a_i^{(t-1)} = 1 \\ \alpha^* \mathtt{G}_0^{\mathtt{EV}}(\theta|\psi)G(x_i|\theta) + \sum_{j\neq i} G(x_i|\theta_{z_j})\delta(\theta - \theta_{z_j}^{(t-1)})\delta(a_j^{(t-1)} - 1) & , a_i^{(t-1)} = -1 \end{cases} \tag{3.32}$$

        Update $K = K + 1$
    end if
    for each cluster $k \in \{1, 2, \dots, K\}$ do
        Sample cluster parameters $\theta_k$ and $\theta_k^a$ using Equations 3.26 and 3.27.
    end for
    Sample the anomaly classification $a_i$ using Equations 3.28 and 3.29.
    Set $z_i^{(t)} = z_i * a_i$
end for
```

Algorithm 2 Gibbs Sampling Algorithm when $\mathtt{G}_0^{\mathtt{EV}}$ is not available

```
Given $z_{\cdot}^{(t-1)}, \left\{ \theta_k^{(t-1)} \right\}, \left\{ \theta_k^{a(t-1)} \right\}$ from iteration $(t-1)$. Let $K$ be the total number of clusters at iteration $(t-1)$.
 1: Set $z_{\cdot} = |z_{\cdot}^{(t-1)}|$ and $a_{\cdot} = sign(z_{\cdot}^{(t-1)})$
 2: for each observation $i$ do
 3:     Remove $x_i$ from its cluster $z_i$..
 4:     if $x_i$ is the only point in its cluster then
 5:         Remove the cluster and update K to K-1.
 6:     end if
 7:     Drop empty clusters.
 8:     Sample $z_i$ from the Multinomial distribution given by Equations 3.24 and 3.25
 9:     if $z_i = K + 1$ then
10:         Set the cluster distribution to be multivariate normal with the new cluster mean
    as $x_i$ and cluster variance as $\Sigma$ which is pre-defined.
11:         Update K=K+1.
12:     end if
13:     for each cluster $k \in \{1, 2, \dots, K\}$ do
14:         Sample cluster parameters $\theta_k$ and $\theta_k^a$ using Equation 3.26.
15:     end for
16:     Sample the anomaly classification $a_i$ from the Binomial($p_i$) where $p_i$ is given by

        $$p_i = p(x_i) = \begin{cases} \text{Probability of } x_i \text{ being anomalous,} & x_i \text{ in tail} \\ 0, & \text{otherwise} \end{cases} \tag{3.33}$$

17:     if most cluster instances are classified as anomalous then
18:         Classify all cluster's instances as anomalies.
19:     end if
20:     Set $z_i^{(t)} = z_i * a_i$
21: end for
```

Algorithm 3 Algorithm for Streaming Extension

```
Perform clustering on a small portion of the data ( 20%) using non-streaming model
1: for each new data point $x_N$ do
2:     Compute the mixture proportions $m\_para$ and the mixture density for all the data.
         Compute $t_1 = q^{th}$ percentile pdf value to identify the tail points
3:     For each $x_i$ $s.t.$ $g(x_i) < t_1$ repeat steps 3→19 of Algorithm 2
4:     if cluster size $\leq 0.05 * N$ then
5:         Classify all the cluster points as anomalies.
6:     end if
7: end for
```

Without loss of generality, let us assume there are K clusters. Let, for any $k < K$, the joint probability of all the points in cluster k be given by

$$\left(\frac{\alpha * p_{k,1}}{I_{k,1} + \alpha - 1} + \frac{\alpha^* * (1 - p_{k,1})}{I_{k,1} + \alpha^* - 1} \right) \prod_{n_k=2}^{N_k} \left(\frac{(n_k - 1) * p_{k,n_k}}{I_{k,n_k} + \alpha - 1} + \frac{(n_k - 1) * (1 - p_{k,n_k})}{I_{k,n_k} + \alpha^* - 1} \right) \tag{3.35}$$

where N_k is the size of the cluster k , $I_{k,i}$ is the index of the i^{th} instance joining the k^{th} cluster and $p_{k,i} = p_{I_{k,i}}$. Thus, the joint probability for complete data is then given by

$$\frac{\prod_{k=1}^{K} \left[(I_{k,1} - 1) p_{k,1} (\alpha - \alpha^*) + \alpha^* (I_{k,1} + \alpha - 1) \prod_{n_k=2}^{N_k} (n_k - 1)(I_{k,n_k} + \alpha - 1 + p_{k,n_k}(\alpha^* - \alpha) \right]}{\prod_{i=1}^{N} ((i + \alpha - 1)(i + \alpha^* - 1))} \tag{3.36}$$

which is dependent on the order of the data. This shows that the model is not exchangeable unless $\alpha = \alpha^*$ or $p_{k,n_k} = 0$ or $p_{k,n_k} = 1$. These conditions effectively reduce the prior distribution to a traditional CRP model. Hence, it can be concluded that the INCAD model cannot be modified to be exchangeable.

The non-exchangeable and non-parametric prior in the INCAD model serves as a excellent platform to capture *drift or evolution* in the behavior(s) locally and globally. Such prior can detect the following trends:

1. Instances that signify new evolutionary behavior are captured and classified as anomalous.
2. Increased prevalence in a previously rare behavior can be re-evaluated and conceived as normal[8].
3. Outdated behaviors that are no longer prevalent would be classified as anomalous. Additionally, relapse of such behaviors are also branded as anomalous till sufficient popularity is reached.

[8] As an alternate frame of reference, one can say that with sufficient surge in the instances, group anomalies can eventually grow to become normal clusters.

A clear streaming extension of the INCAD model involves exclusive re-evaluation of the tail instances as opposed to updating with entire data. The Gibbs sampling algorithm for the streaming INCAD model is given in Algorithm 3.

Choice of Priors

For computational ease, the base distribution that generates the parameters for the normal clusters, $\mathtt{G}_0$, is chosen to be the conjugate of the generative distribution for the actual data, $\boldsymbol{G}$. This makes the inference task considerably simpler, though approximate methods have been discussed for non-conjugate prior choices as well [58, 41]. In this part, we use a *Multivariate Normal Distribution* (`MVN`) as the data distribution, $\boldsymbol{G}$, and the *Normal Inverse Wishart* (`NIW`) as the base distribution, $\mathtt{G}_0$. It must be noted that the model is not limited to MVN distribution. In particular, any univariate data distribution that satisfies the necessary conditions in Theorem 2 could be used. For multivariate data, distributions from exponential family satisfy the necessary conditions needed for the Ext-GPD approach. The required conditions for the multivariate case have been presented in the supplementary section and in Theorem 3.

The concentration parameter, α, and the prior for the base distributions, ψ, are treated as hyper-parameters, though suitable vague priors maybe set to make the model more robust to the choice of the hyper-parameters. α controls the final number of normal clusters, while α_d^* controls the final number of anomalous clusters from the d^{th} DPMM. To ensure that a larger number of populated non-anomalous clusters are formed with few instances assigned to them, α's can be typically set to a higher values.

The parameter γ influences the number of anomalous instances in the data set, and is initialized based on the expected proportion of anomalies in the given context. For the results listed in this part, we have used a standard set of the parameter and hyper-parameter choices to show the results in a generalized setting (detailed in Section 4.1.1). But in other contexts, one can use the information from the data to determine the hyperparameters. For instance, the γ value can be initially set to the proportion of anomalies known in the data, and the concentration parameter α can be set higher if the true number of clusters is known to be high. It must be noted that the choice of hyper-parameters $\{\alpha_d^*\}$ and parameter γ is updated and optimized using Extreme Value distributions and Bayesian updates over iterations.

Chapter 4

Detecting Anomalies in Streaming Data - A Comparative Evaluation

4.1 Experimental Setup

To comprehensively evaluate the capabilities of the proposed INCAD model, results on both synthetically generated and publicly available benchmark data sets are provided. We evaluate the ability of the proposed model to identify both clusters and anomalies, in both batch and streaming settings. We also compare the model performance with existing methods for anomaly detection and clustering. Additionally, we study the role of various user-defined parameters on the model performance.

4.1.1 Model Initialization

The INCAD model has the following user-defined hyper-parameters: the initial number of clusters (K), the concentration parameter (α), the initial mean and covariance matrices for the clusters and the prior for the proportion of anomalies (γ). For the experiments, we set K to 10 and α to 1. For each data set, the sample mean and covariance are used as the initial values for the cluster parameters. The proportion of anomalies (γ) is set to 0.1. In the batch phase, the model is run until convergence is achieved, with a maximum iteration limit of 1000.

4.1.2 Data description

We consider a variety of publicly available benchmark data sets from different domains (See Table 4.1) for the experimental evaluation. Additionally, a synthetically generated 2-dimensional data set, SD, with 4 normal clusters and scattered anomalies was generated to evaluate the joint clustering and anomaly detection performance. Each cluster consisted of 100 observations, sampled from a 2-D Gaussian distribution with means in $\{(-40, -40)$, $(-30, 10)$, $(40, -60)$, $(45, 30)\}$, for each cluster, respectively. The covariance matrix for each cluster was set to $5I$, where I is the 2×2 identity matrix. 23 anomalies were added by sampling from a Gaussian distribution with mean at $(0, 0)$ and covariance as $100I$. For a qualitative evaluation of the joint clustering and anomaly detection performance, we use the MNIST handwritten digits data set [52], which consists of 60000 28×28 images, corresponding to 10 digits (clusters). We use a 10% sample of the original data set and use principal component analysis (PCA) to reduce the dimensionality of the data from 784 to 25.

Finally, we use the Gas Sensor Array Drift data set [78] to understand the performance of the INCAD model in a streaming setting. The data set consists of 470 readings from an array of 16 chemical sensors exposed to gas mixtures at three different concentration levels. First, two concentration levels were used as the batch data set and the third concentration level was injected in a streaming fashion.

4.1.3 State-of-the-art Methods

We compare the performance of INCAD with several existing state-of-art anomaly detection and clustering methods, as well as one method that has been proposed for joint clustering and anomaly detection [19]

Name	N	d	c
Pageb	5473	11	2
Wine-Cluster	6497	12	2
Heart Statlog	270	13	2
Zoo	101	16	7
Abalone	4177	8	2
Magic Gamma	19020	10	2
Iono	351	33	2
Ecoli	336	7	8
Haberman	306	2	12
Concrete	1030	9	2
German	1000	7	9
Segment	2310	18	7
Iris	150	4	3
Yeast	1484	8	10
WDBC	569	31	2
Vehicle	846	18	4
Glass	214	9	6
Tae	151	3	3
Balance Scale	625	4	3
Vowel	990	10	11

(a) Clustering

Name	N	d	a
Annthyroid	7200	6	7.42 %
Pen Global	809	16	11.12%
Cardio	1831	21	9.61 %
Mammography	11183	6	2.32 %
Letter	1600	32	6.25 %
Seismic Bumps	2584	11	6.58 %
Cover	217	10	9.22 %
Breast Cancer	367	30	2.72 %
Smtp	113	3	11.5 %
Wine-AD	129	13	7.75 %
Pendigits	6870	16	2.27 %

(b) Anomaly detection

Table 4.1: Description of the benchmark data sets used for evaluation of the clustering (*source*: UCI-ML repository [24]) and anomaly detection (*source*: Outlier Detection DataSets /ODDS (Rayana, 2016)) capabilities of the proposed model. N - number of instances, d - number of attributes, c - number of true clusters and a - the fraction of known anomalies in the data set.

Anomaly Detection: For anomaly detection, we consider four existing methods: *k nearest neighbor outlier detection* (kNN) [62], *local outlier factor* (LOF) [12], *one-class Support Vector Machines* (oc-SVM) [72] and *k-means*−− (Chawla and Gionis, 2013). The first two methods assign an anomaly score for each data instance, while the last two methods assign an anomaly label. Both kNN and LOF have been previously shown to outperform other existing methods [39] and are considered state-of-art methods. The *k-means*−− method performs joint clustering and anomaly detection and thus is the most similar to INCAD. All methods have one or more user-defined parameters. We investigated a range of values for each parameter and report the mean results.

Clustering: We compare the clustering performance of INCAD with k-means, *k-means*−− and a Bayesian Gaussian Mixture model with a Dirichlet prior (BGM-DP). While both k-means and *k-means*−− are *hard* clustering algorithms that require specifying the number of clusters as a user-defined parameter, BGM-DP is a *soft* clustering algorithm that does not need the number of clusters to be provided in advance. Thus it is similar to INCAD in that regard.

4.1.4 Evaluation Metrics

For the anomaly detection methods that assign an anomaly label to a test instance, i.e., oc-SVM, *k-means*−− and INCAD, *f-measure*[1] on the anomaly class is used as the evaluation metric. For the scoring methods, i.e., kNN, LOF and the scoring version of INCAD, the instances with top *p* anomaly scores are labeled as anomalies and these labels are then used to calculate the f-measure. For the clustering evaluation, we use *average cluster purity* (Chawla and Gionis, 2013), as the evaluation metric, where the purity of a cluster is defined as the fraction of the majority class of the cluster with respect to the size of the cluster.

4.2 Results

In this section, we discuss the overall performance of the INCAD model against the state-of-the-art algorithms with respect to clustering and anomaly detection, in both streaming and batch settings on simulated as well as benchmark datasets.

4.2.1 Simulated Data

Batch scenario: For a given batch dataset, INCAD produces two types of outputs. First, it assigns every data instance to either a *normal* cluster (with a positive index) or an *anomalous*

[1] the class-specific f-measure is defined as the harmonic mean of the recall and precision on the given data set for that class.

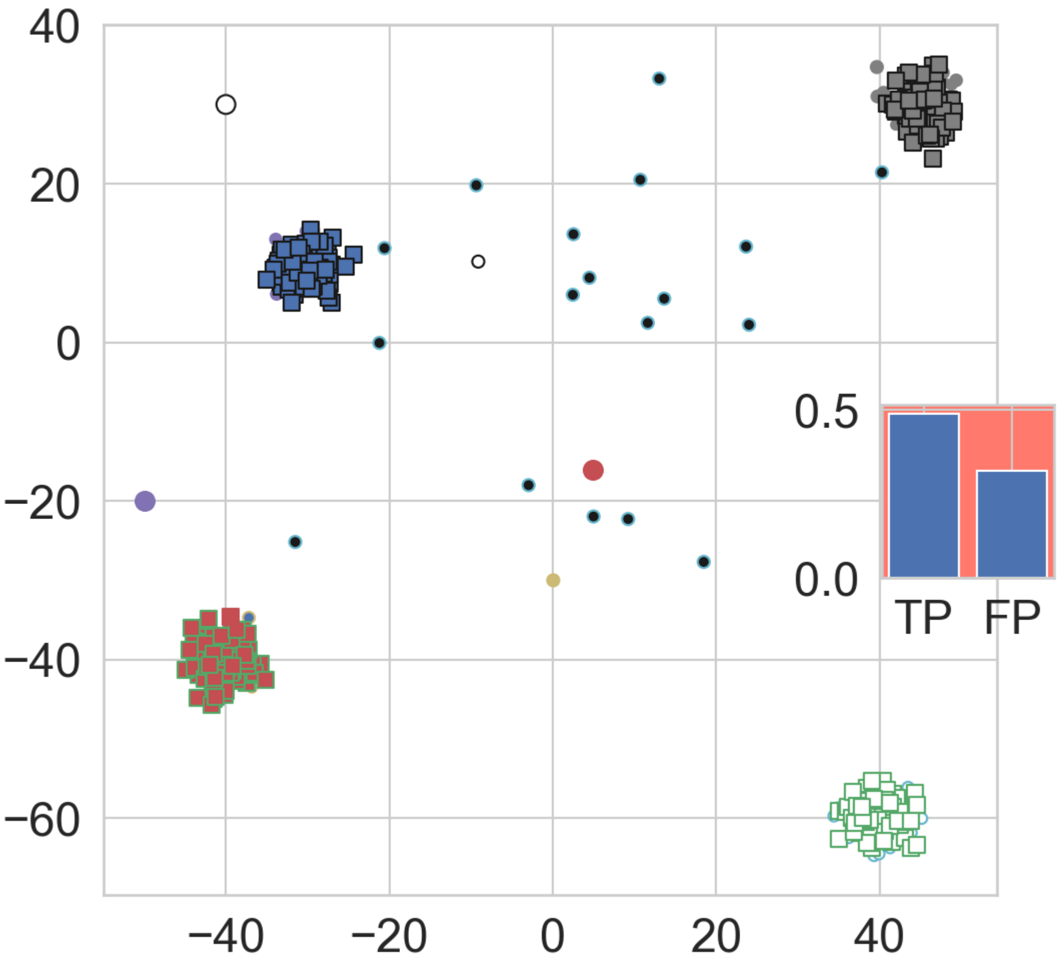


Figure 4.1: INCAD output for the synthetic data, SD. Instances belonging to the normal clusters are shown as □ and instances belonging to anomalous clusters are shown as ○. The size of the anomalous instances indicates the probabilistic anomaly score. *Inset*: the average anomaly score for truly anomalous instances (TP) and false positives (FP).

cluster (with a negative index). The sign of the cluster index is used as the *anomaly label*. Additionally, the method also assigns a probability for each instance to be in the tail of the overall data distribution, which is used as the *probabilistic anomaly score*. For the SD data set, the identified normal and anomalous clusters, as well as the anomaly scores, are shown in Figure 4.1. We first note that INCAD identifies the four main clusters in the data, without the need to initially specify the number of clusters. Additional anomalous clusters, with negative index, were identified as well. While the method correctly labels all the 23 anomalous instances, it also identified some peripheral instances of the normal clusters as anomalies; these would constitute as the false positives. However, the probability score is higher for the true anomalies (See Figure 4.1: *Inset*). Thus, simple heuristics, such as a low threshold on the anomaly probability, can be potentially employed, as a post-processing step, to filter out these false positives.

Streaming Scenario: To study the performance of INCAD in a streaming mode, we simulate the following streaming scenario: We first create a batch of data consisting of instances belonging to three of the four clusters in SD and present it to INCAD for batch learning. INCAD identifies the three primary clusters and some of the peripheral instances as local anomalies, after the batch phase (See Figure 2.1a). The instances belonging to the fourth cluster and the anomalies are sequentially presented to the model. With each incoming streaming instance, the tail data is re-evaluated and the overall identified data distribution is updated. At the beginning of the streaming phase, the new instances are identified as group anomalies, as shown in Figure 2.1b. However, a fourth normal cluster is identified after a sufficient number of instances belonging to the fourth cluster are observed in the stream, as shown in Figure 2.1c. Finally, the remaining truly anomalous instances are identified as global anomalies, as they do not form a tight enough group to become a normal cluster, as shown in Figure 2.1d.

4.2.2 Anomaly Detection Performance on Benchmark Datasets

The f-measure performance of INCAD and the competing algorithms is shown in Table 4.2. For all the listed algorithms, results for the best parameter settings are reported. The proposed INCAD model outperforms other methods on 4 out of 11 data sets. While other methods, especially LOF and KNN are better on other data sets, it should be noted that these methods are highly sensitive to the parameter settings. The k-means$--$ method, which is capable of both clustering and anomaly detection, shows the best average performance. However, this model requires specifying the proportion of true anomalies in the data set, which might not be feasible in a real-world setting[2].

A specific behavior noticed in the score based INCAD model is the ranking of the anomalies. As INCAD is a conservative algorithm that identifies more anomalies, it can be seen that the model recall is relatively higher than the rest of the methods. However, the true anomalies might not always be ranked as the most anomalous observations. This behavior can be best observed in two particular datasets, namely Pen-Global and Wine data, where the score based model has failed to rank most true anomalies in the top while the classification model still identified some of the true anomalies.

4.2.3 Clustering Performance on Benchmark Datasets

Table 4.3 summarizes the performance of INCAD and other competing clustering methods on the benchmark data sets. Overall, INCAD has the best average performance compared to others, which is significant, despite not having to provide a prior specification of the expected number of clusters, unlike k-means and k-means$--$. Looking at both anomaly detection and clustering performance, it is clear that INCAD is effective in identifying both anomalies and clusters in the data and is superior to k-means$--$, which also does the joint detection.

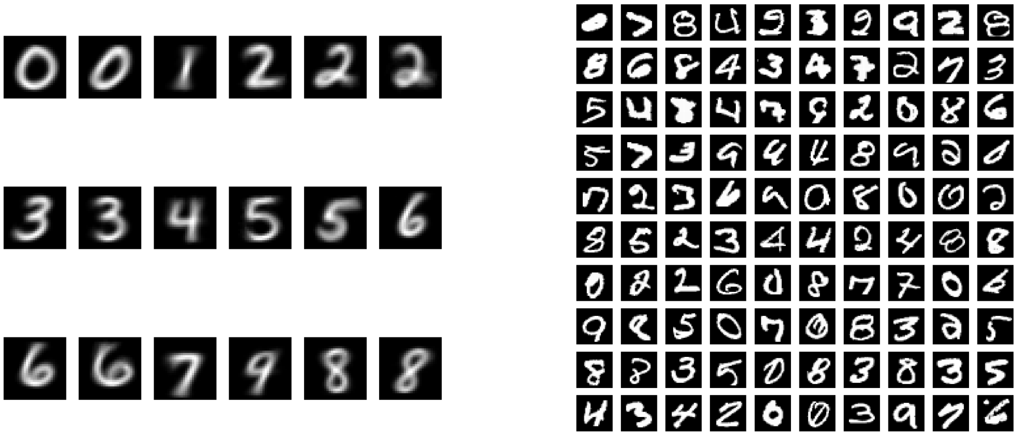
[2] For some real datasets with >30% anomalies, smaller clusters identified by INCAD can be manually reclassified as anomalous.

Dataset	LOF	KMeans−−	KNN	OCSVM	INCAD	INCAD (score)
COVER	**0.36 (± 0.0331)**	0.15 (± 0.0316)	0.15 (± 0.0)	0.15 (± 0.0554)	0.3 (± 0.1613)	0.18 (± 0.0714)
WINE	0.24 (± 0.08)	0.3 (± 0.0)	0.23 (± 0.0943)	0.1 (± 0.0419)	**0.41 (± 0.1941)**	0.1 (± 0.0)
SMTP	**0.59 (± 0.1674)**	0.54 (± 0.0)	0.53 (± 0.0921)	0.21 (± 0.0915)	0.31 (± 0.0669)	0.32 (± 0.102)
PENDIGITS	0.08 (± 0.0075)	**0.19 (± 0.1537)**	0.1 (± 0.0152)	0.06 (± 0.0124)	0.09 (± 0.0365)	0.07 (± 0.0138)
BREAST-CANCER	0.44 (± 0.0165)	**0.6 (± 0.0)**	0.39 (± 0.0598)	0.05 (± 0.0479)	0.19 (± 0.0638)	0.4 (± 0.015)
LETTER	0.44 (± 0.0409)	0.07 (± 0.04)	0.4 (± 0.0779)	0.11 (± 0.0162)	0.28 (± 0.0354)	**0.45 (± 0.0265)**
ANNTHYROID	0.21 (± 0.0121)	0.17 (± 0.0817)	0.3 (± 0.0084)	0.11 (± 0.019)	0.36 (± 0.0254)	**0.39 (± 0.0455)**
PEN-GLOBAL	0.23 (± 0.0365)	0.34 (± 0.0627)	0.25 (± 0.0278)	0.21 (± 0.0497)	**0.53 (± 0.0662)**	0.25 (± 0.0358)
CARDIO	0.21 (± 0.0173)	**0.36 (± 0.3145)**	0.31 (± 0.0772)	0.15 (± 0.0297)	0.2 (± 0.1045)	0.2 (± 0.0838)
MAMMOGRAPHY	0.19 (± 0.0455)	0.12 (± 0.1276)	0.22 (± 0.03)	0.05 (± 0.0354)	0.12 (± 0.0131)	**0.24 (± 0.0216)**
SEISMIC-BUMPS	0.07 (± 0.0113)	0.1 (± 0.0766)	0.15 (± 0.0068)	0.13 (± 0.0304)	**0.23 (± 0.0191)**	0.17 (± 0.0189)

Table 4.2: Comparing INCAD with existing anomaly detection algorithms using f-measure on the anomaly class as the evaluation metric. For score based methods, instances with top k scores are labeled as anomalous, where k is the actual number of anomalies in the data set. The average precision and recall on the anomaly class, across all data sets, is shown in the last two rows.

Dataset	k-means	k-means−−	BGM (DP Prior)	INCAD
PAGEB	0.9	0.9	0.94	**0.99 (± 0.0114)**
ABALONE	0.75	**0.81**	0.76	**0.81 (± 0.0139)**
ZOO	**0.87**	0.41	0.64	0.79 (± 0.0913)
WINE	0.63	0.63	0.69	**0.79 (± 0.0719)**
HEART-STATLOG	**0.84**	0.71	0.61	0.79 (± 0.033)
IONO	0.71	0.64	**0.83**	0.79 (± 0.0156)
MAGIC.GAMMA	0.65	0.73	0.77	**0.78 (± 0.0103)**
ECOLI	**0.83**	0.43	0.57	0.76 (± 0.0079)
HABERMAN	**0.75**	0.74	**0.75**	**0.75 (± 0.0069)**
SEGMENT	0.55	0.14	0.52	**0.71 (± 0.0989)**
GERMAN	**0.7**	**0.7**	**0.7**	**0.7 (± 0.0036)**
CONCRETE	0.6	**0.87**	0.65	0.69 (± 0.0324)
IRIS	**0.81**	0.33	0.76	0.67 (± 0.0096)
YEAST	**0.66**	**0.66**	**0.66**	**0.66 (± 0.002)**
WDBC	**0.91**	0.91	0.82	0.63 (± 0.0021)
GLASS	**0.56**	0.36	0.51	0.55 (± 0.0296)
TAE	0.44	0.4	0.44	**0.54 (± 0.0145)**
VEHICLE	0.37	0.35	**0.5**	**0.49 (± 0.0416)**
BALANCE-SCALE	**0.65**	**0.65**	0.59	0.46 (± 0.0016)
VOWEL	0.33	0.09	0.34	**0.37 (± 0.0587)**
Avg. Purity	0.68	0.57	0.66	**0.69**

Table 4.3: Comparing INCAD with existing clustering algorithms using purity score as the evaluation metric.



(a) Clusters

(b) Anomalies

Figure 4.2: Output of INCAD for the MNIST 10% sample data. (a). Cluster centers identified by INCAD. Note that the number of clusters (18) is automatically inferred by the model, (b). Anomalies identified by INCAD.

To further show the effectiveness of INCAD for the joint detection task, we visualize the detected clusters and anomalies for the MNIST hand-written digit data set. INCAD identified 18 clusters in the data. The cluster centroids are shown in Figure 4.2a. The most interesting outcome of clustering using INCAD was the identification of subtle writing behaviors identified in the data. For instance, three different writing styles of digits '2' and '6' were identified, which corresponded to distinctive *slants*, presence of *loops*, etc. The anomalous digits (See Figure 4.2b) identified by INCAD include unrecognizable and ill-written digits.

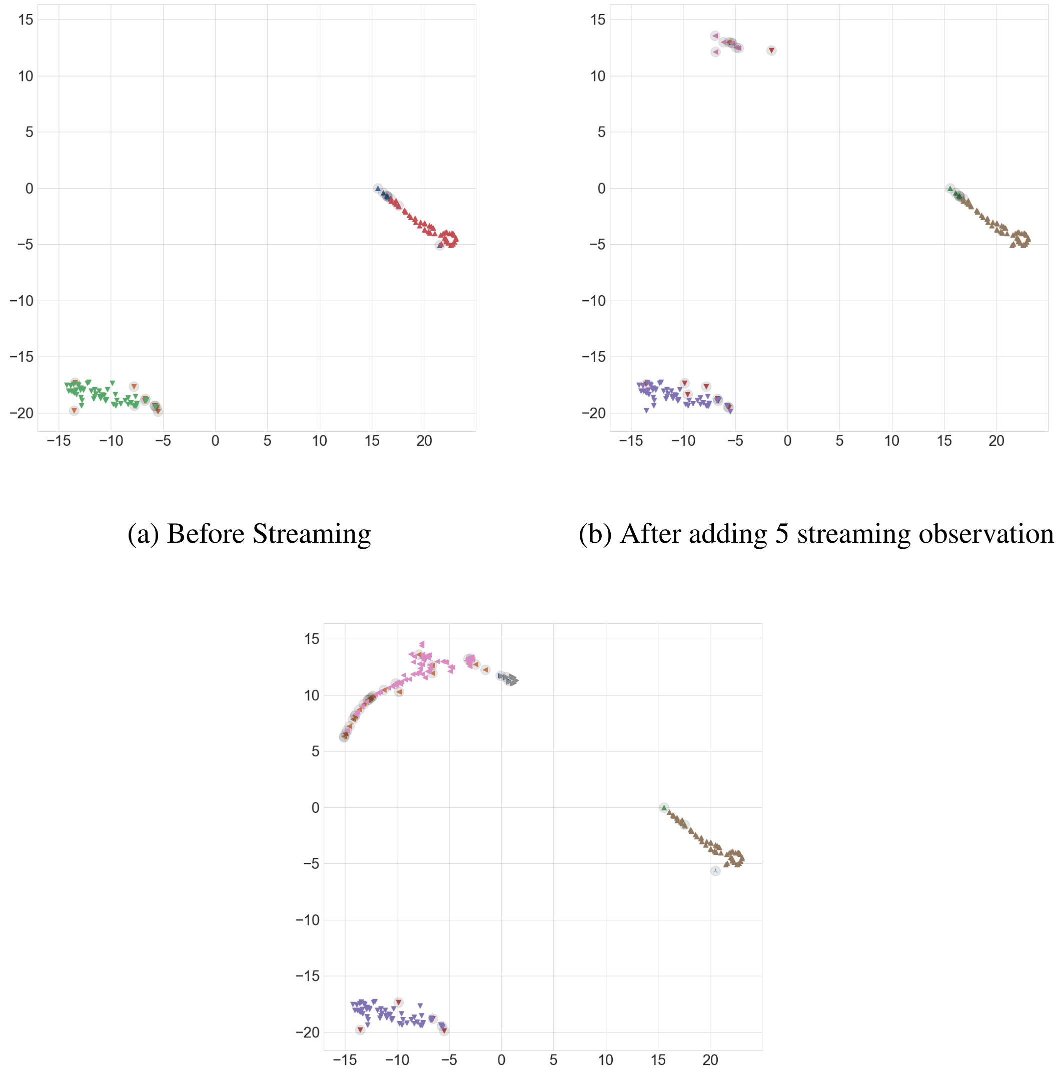
4.2.4 Streaming Anomaly Detection and Clustering: Gas Sensor Array Drift Data

The experiment for the Gas Sensor Array Drift data set simulates a streaming scenario in which a gas at different concentrations is being introduced into a chamber and the concentration levels are being measured by an array of 16 chemical sensors. For these experiments, the observations corresponding to two concentration levels are provided for batch learning and observations corresponding to the third concentration level are added as a stream. The monitoring output of INCAD, at different phases of the stream, are shown in Figure 4.3. At the end of the batch learning, INCAD is able to identify the two gas concentrations (See Figure 4.3a) present in the batch data set. After the start of the streaming phase, the new instances are identified as anomalies (See Figure 4.3b), as they belong to a previously unseen concentration. However, as more data is observed in the stream, a new novel cluster is identified (See Figure 4.3c) and all the instances belonging to the third concentration are now considered normal.

4.2.5 Sensitivity to Batch Proportion

Previous results on streaming data show that INCAD can identify anomalies and new clusters in a stream. The performance, however, depends on the size of the initial batch data set. Figure 4.4 shows the performance of the model, both in terms of computing time and accuracy in identifying anomalies for the synthetic data set, SD. While the total size of the data set is fixed, the proportion of the instances in the batch is varied from 10% to 90%. The computing time[3] for processing the batch increases linearly with the increase in the batch size. At the same time, the time taken to process a single stream instance also increases as the size of the batch increases. This is because the INCAD model has to update the tail probabilities for the data observed so far. The quality of the detected anomalies (shown using the F-measure for the anomalies detected after all of the data is observed), improves as the size of the batch increases. Additionally, the performance is more stable (lower variance across multiple runs) when the batch size is higher because the batch phase is able to learn a stable clustering structure in the data.

[3] All the methods are implemented in Python and all experiments were conducted on a 2.7 GHz Quad-Core Intel Core i7 processor with a 16 GB RAM.



(a) Before Streaming

(b) After adding 5 streaming observations

(c) After adding all streaming observations

Figure 4.3: Evolving anomalies and clusters identified by INCAD for the Gas Sensor Array Drift data. Cluster assignments are shown using colored symbols, anomalous observations are labeled using colored circles. While the original data has 16 dimensions, the data is mapped to 2-D using the t-SNE algorithm [54].

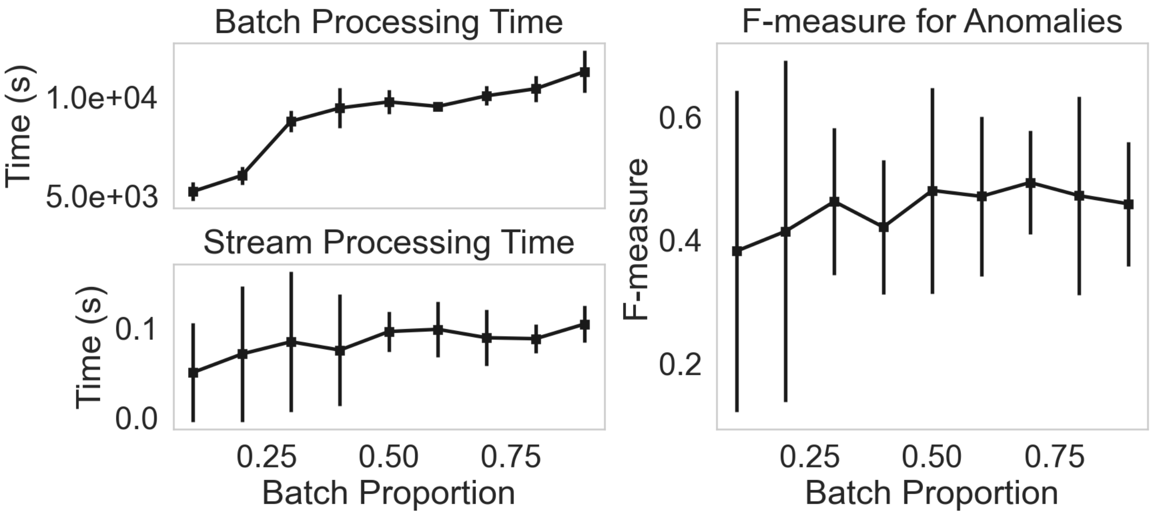


Figure 4.4: Impact of the size of the batch data set on INCAD performance on the synthetic data set (SD). For each batch size, mean and standard deviation across 5 different runs are shown.

4.3 Conclusions and Future Work

We have introduced a Bayesian framework for anomaly detection that explicitly models the normal and anomalous data. While in the past, lack of labeled anomalies has prevented such solutions, we adopt concepts from Extreme Value Theory (EVT), to model the anomalous data with respect to the extremes of the model for the normal data. This is a fundamental breakthrough in anomaly detection as it permits probabilistic reasoning for both types of instances, without the need for a non-intuitive threshold, as is the case for existing methods. Additionally, the proposed INCAD algorithm combines EVT with another powerful modeling tool - DPMM which allows identifying clusters and anomalies at the same time. The non-parametric prior on the number of clusters ensures that the model is not handicapped by the need to know the exact number of clusters. Moreover, this sets the model up to be adapted for a streaming scenario, where the number of clusters can change over the stream.

As the results show, INCAD outperforms existing methods that have been proposed exclusively for anomaly detection or clustering, on each of the tasks, for most of the data sets (See Tables 4.2 and 4.3). Moreover, while existing methods rely on carefully specified, problem-specific, parameters, INCAD requires specifying relaxed Bayesian priors and infers key parameters, such as the number of clusters, from the data. Additionally, the probabilistic output of INCAD, allows for an interpretable setting of thresholds, on the anomaly score, something that is not possible with most existing score based anomaly detection algorithms. INCAD is especially effective in dealing with streaming data, where the notion of normal clusters and anomalies evolve over the duration of the stream, as shown in Figure 4.3. This makes INCAD highly suitable for monitoring the behavior of complex systems over time, without the need to explicitly retrain the underlying model.

One of the key shortcomings of the model is the complexity of the iterative Gibbs algorithm. Variational inference methods that have been proposed for inference in DPMM clustering (Blei and Jordan, Huynh, Phung, and Venkatesh, 2004, 2016) can be used to improve the complexity and will be explored in the future.

Part II

Anomaly Detection in High-Dimensional Evolving Data

Chapter 5

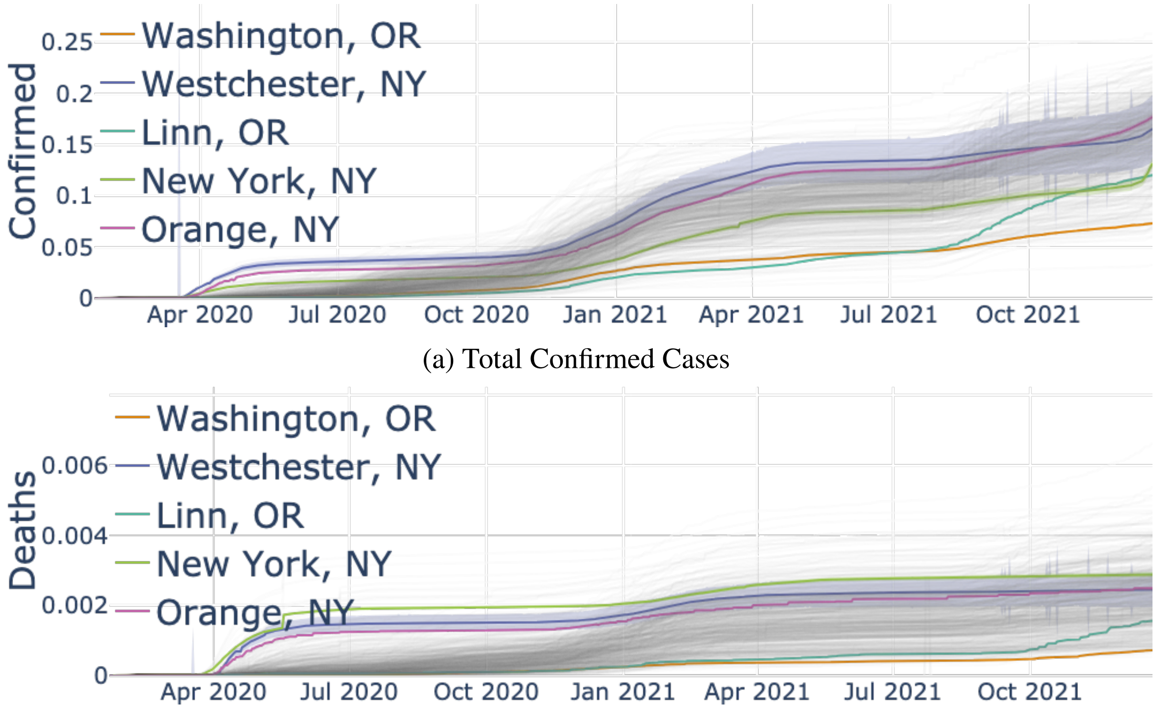
Large Deviations Principle

5.1 Introduction

Anomaly detection has been extensively studied over many decades across many domains [16, 46]. Among the most useful applications of anomaly detection is to simultaneously monitor multiple systems' behaviors and identify the system that exhibits anomalous behavior due to external or internal stress factors. These include the study of multiple evolving streams like a time series database.

Time series analysis metrics are expensive for large collections of streams or are difficult to extend to do a relative study across a database of time series. However, the study of anomalous behaviors in one time series as relative to others in a database is, by comparison, less researched. In particular, such a perspective is important in response to pandemic propagation, economic issues, social justice issues, climate change adaptation, public health etc.

For instance, consider the example of the COVID-19 infection data. Studying the confirmed case and death trends across various countries, states or counties could highlight and identify the most (or least) significant public policies. One possible approach to study the data could be to monitor each time series [13, 55, 81] and identify sudden outbreaks or



(a) Total Confirmed Cases

(b) Total Deaths

Figure 5.1: Top 5 anomalous counties identified by the proposed LAD algorithm based on the daily multivariate time-series, consisting of cumulative COVID-19 per-capita infections and deaths. At any time instance, the algorithm analyzes the bivariate time series for all the counties to identify anomalies. The time-series for the non-anomalous counties are plotted (light-gray) in the background for reference. For the counties in New York (Westchester, New York and Orange), the number of confirmed cases (*top*), and the significant rise during early 2021 along with the consistently high death rates, is the primary cause for anomaly[1]. On the other hand, Washington and Linn County in Oregon were identified as anomalous primarily due to their steady low rates compared to the rest of the counties.

significant causal events. However, such methods study each time series individually and cannot be used to detect the gradual divergence from the normal trends or initial signs of such drift.

In this part of the thesis, we propose a new anomaly detection algorithm called *Large deviations Anomaly Detection* (LAD), for large/high-dimensional data and multivariate time series data. LAD uses the rate function from *large deviations principle* (LDP) [23, 77,

[1] In early January 2021, a highly contagious variant from the UK was found in NY state. In addition to the above, the post holiday surge was seen in the form of increased hospitalizations during this period - `https://abc7ny.com/uk-lockdown-covid-variant-nyc-vaccine-hospitalizations-19-deaths/9340767/`

76] to deduce anomaly scores for the underlying data. Core ideas for the algorithm are inspired by the large deviation theory's projection theorem that allows better handling of high dimensional data. Unlike most high dimensional anomaly detection models, LAD does not incorporate feature selection or dimensionality reduction, which makes it ideal to study multiple time series in an online mode. The intuition behind the LAD model allows it to naturally segregate the anomalous observations at each time step while comparing multiple multivariate time series simultaneously. The key contributions of this part are following:

1. We propose the *Large deviations Anomaly Detection* (LAD) algorithm, a novel and highly scalable LDP based methodology, for scoring based anomaly detection.

2. The proposed LAD model is capable of analyzing large and high dimensional datasets without additional dimensionality reduction procedures thereby allowing more accurate and cost effective anomaly detection.

3. An online extension of the LAD model is presented to detect anomalies in a multivariate time series database using an evolving anomaly score for each time series. The anomaly score varies with time and can be used to track developing anomalous behavior.

4. We perform an empirical study on publicly available anomaly detection benchmark datasets to analyze the robustness and performance of the proposed method on high dimensional and large datasets.

5. We present a detailed analysis of COVID-19 trends for US counties where we identify counties with anomalous behavior (See Figure 5.1 for an illustration).

5.2 Related Work

In this section, we provide a brief overview of relevant anomaly detection methods which have been proposed for high-dimensional data and for multivariate time-series data. We also discuss other works that have used the large deviations principle for detecting anomalies.

A large body of research exists on studying anomalies in high dimensional data [1, 5] but challenges remain. Many anomaly detection algorithms use dimensionality reduction techniques as a pre-processing step to anomaly detection. However, many high dimensional anomalies can only be detected in high dimensional problem settings and dimensionality reduction in such settings can lead to false negatives. Many methods exist that identify anomalies on high-dimensional data without dimensional reduction or feature selection, e.g. by using distance metrics. *Elliptic Envelope* (EE) [65] fits an ellipse around data centers by fitting robust covariance estimates. *Isolation Forest* (I-Forest) [53] uses recursive partitioning by random feature selection and isolating outlier observations. *k nearest neighbor outlier detection* (kNN) [62] uses distance from nearest neighbor to get anomaly scores. *local outlier factor* (LOF) [12] uses deviation in local densities with respect to its neighbors to detect anomalies. *k-means*$--$ (Chawla and Gionis, 2013) method uses distance from nearest cluster centers to jointly perform clustering and anomaly detection. *Concentration Free Outlier Factor* (CFOF) [4] uses a "reverse nearest neighbor-based score" which measures the number of nearest neighbors required for a point to have a set proportion of data within its envelope. In particular, methods like I-Forest and CFOF are targeted towards anomaly detection in high dimensional datasets.

In most settings, real time detection of anomalies is needed to dispatch necessary preventive measures for damage control. Such problem formulation requires collectively monitoring a high dimensional time series database to identify anomalies in real time. Recently, large deviations theory has been widely applied in the fields of climate models [22], statistical mechanics [74], networks [60], etc. Specially for analysis of time series, the theory of large deviations has proven to be of great interest over recent decades [11, 57]. However, these methods are data specific, often study individual time series and are difficult to generalize to other areas of research.

Anomaly detection for time series have been extensively explored in the literature [43], though most focus has been on identifying anomalous events in a single time-series. While, the task of detecting anomalous time series in a collection of time series has been studied in the past [80, 15, 17], most of these works have focused on univariate time series and have not easy to scale to long time series data. Our proposed method addresses this issue by using the large deviation principle.

5.3 Large Deviation Principle

Large deviations theory provides techniques to derive the probability of rare events[2] that have an asymptotically exact exponential approximation[23, 77, 76]. In this section, we briefly go over the large deviation theory and different ways to generate the rate functions required for the large deviations principle.

The key concept of this theory is the Large Deviations Principle (LDP). The principle describes the exponential decay of the probabilities for the mean of random variables. The rate of decay is characterized by the rate function $\mathcal{I}$. The theorem is detailed below:

Theorem 4. *A family of probability measures $\{\mu_\epsilon\}_{\epsilon>0}$ on a Polish space $\mathcal{X}$ is said to satisfy large deviation principle (LDP) with the rate function $\mathcal{I} : \mathcal{X} \rightarrow [0, \infty]$ if:*

1. *$\mathcal{I}$ has compact level sets and is not identically infinite*

2. *$liminf_{\epsilon\rightarrow 0} \epsilon log \mu_\epsilon(\mathcal{O}) \geq -\mathcal{I}(\mathcal{O}) \quad \forall \mathcal{O} \subseteq \mathcal{X}$ open sets*

3. *$limsup_{\epsilon\rightarrow 0} \epsilon log \mu_\epsilon(C) \leq -\mathcal{I}(C) \quad \forall C \subseteq \mathcal{X}$ closed sets*

where, $\mathcal{I}(\mathcal{S}) = inf_{x\in\mathcal{S}} \mathcal{I}(x),\ \mathcal{S} \subseteq \mathcal{X}$

To implement LDP on known data with known distributions, it is important to decipher the rate function $\mathcal{I}$. Cramer's Theorem provides the relation between the rate function $\mathcal{I}$ and the logarithmic moment generating function Λ.

[2] In our context, these rare events include outlier/anomalous behaviors.

Definition 6. *The logarithmic moment generating function of a random variable X is defined as*

$$\Lambda(t) = \log E[\exp(tX)] \tag{5.1}$$

Theorem 5 (Cramer's Theorem)**.** *Let $X_1, X_2, \dots X_n$ be a sequence of iid real random variables with finite logarithmic moment generating function, e.g. $\Lambda(t) < \infty$ for all $t \in \mathbb{R}$. Then the law for the empirical average satisfies the large deviations principle with rate $\epsilon = 1/n$ and rate function is given by*

$$\mathcal{I}(x) := \sup_{t\in\mathbb{R}} (tx - \Lambda(t)) \quad \forall t \in \mathbb{R} \tag{5.2}$$

Thus, we get,

$$\lim_{n\to\infty} \frac{1}{n} \log\left(P\left(\sum_{i=1}^{n} X_i \geq nx \right)\right) = -\mathcal{I}(x), \quad \forall x > E[X_1] \tag{5.3}$$

For more complex distributions, identifying the rate function using the logarithmic moment generating function can be challenging. Many methods like the contraction principle and the exponential tilting exist that extend rate functions from one topological space that satisfies LDP to the topological spaces of interest[23]. For our work, we are interested in the Dawson-Gärtner Projective LDP, that generates the rate function using nested family of projections.

Theorem 6. *Dawson-Gärtner Projective LDP: Let $\{\pi^N\}_{N\in\mathbb{N}}$ be a nested family of projections acting on $\mathcal{X}$ s.t. $\cup_{N\in\mathbb{N}}\pi^N$ is the identity. Let $\mathcal{X}^N = \pi^N\mathcal{X}$ and $\mu_\epsilon^N = \mu_0 \circ (\pi^N)^{-1}, N \in \mathbb{N}$. If $\forall N \in \mathcal{N}$, the family $\{\mu_\epsilon^N\}_{\epsilon>0}$ satisfies the LDP on $\mathcal{X}^N$ with rate function $\mathcal{I}^N$, then $\{\mu_\epsilon\}_{\epsilon>0}$ satisfies the LDP with rate function I given by,*

$$\mathcal{I}(x) = sup_{N\in\mathbb{N}}\mathcal{I}^N(\pi^N x) \quad x \in \mathcal{X}$$

Since $\mathcal{I}^N(y) = inf_{\{x \in \mathcal{X} | \pi^N(x) = y\}} \mathcal{I}(x),\ y \in \mathcal{Y}$*, the supremum defining* $\mathcal{I}$ *is monotone in N because projections are nested.*

The theorem allows extending the rate function from a lower projection to a higher projection space. The implementation of this theorem in the LAD model is discussed in Section 6.1.

Chapter 6

Large Deviations Based Anomaly Detection (LAD) for Time Series Databases

6.1 Methodology

Consider the case of multivariate time series data. Let $\{\mathbf{t_n}\}_{\mathbf{n=1}}^{\mathbf{N}}$ be a set of multivariate time series datasets where $\mathbf{t_n} = (\mathbf{t_{n,1}}, \dots, \mathbf{t_{n,T}})$ is a time series of length T and each $\mathbf{t_{n,t}}$ has d attributes. The motivation is to identify anomalous $\mathbf{t_n}$ that diverge significantly from the non-anomalous counterparts at any given time steps or a time window.

The main challenge is to design a score for individual time series that evolves in a temporal setting as well as enables tracking the initial time of deviation as well as the scale of deviation from the normal trend.

As shown in the following sections, our model addresses the problem through the use of rate functions derived from the large deviations principle. We use the Dawson-Gärtner Projective LDP (See Section 6.1.2) for projecting the rate function to a low dimensional setting while preserving anomalous instances.

The extension to temporal data (See Section 6.1.3) is done by collectively studying each time series data as one observation.

6.1.1 Large Deviations for Anomaly Detection

Our approach uses a direct implementation of LDP to derive the rate function values for each observation. As the theory focuses on extremely rare events, the raw probabilities associated with them are usually very small [77, 23, 76]. However, the LDP provides a rate function that is useful as a scoring metric for our LAD model.

Consider a dataset X of size n. Let $\mathbf{a} = \{\mathbf{a_1}, \dots, \mathbf{a_n}\}$ and $\mathbf{I} = \{\mathbf{I_1}, \dots, \mathbf{I_n}\}$ be anomaly score and anomaly label vectors for the observations respectively such that $a_i \in [0, 1]$ and $I_i \in \{0, 1\}$ $\forall i \in \{1, 2, \dots, n\}$.

By large deviations principle, we know that for a given dataset X of size n, $P(\bar{X} = p) \approx e^{-n\mathcal{I}(p)}$. Assuming that the underlying data is standard Gaussian distribution with mean 0 and variance 1, we can use the rate function for Gaussian data where $\mathcal{I}(p) = \frac{p^2}{2}$. Then the resulting probability that the sample mean is p is given by:

$$P(\bar{X} = p) \approx e^{-n\frac{p^2}{2}} \tag{6.1}$$

Now, in presence of an anomalous observation x_a, the sample mean is shifted by approximately x_a/n for large n. Thus, the probability of the shifted mean being the true mean is given by,

$$P(\bar{X} = x_a/n) \approx e^{-\frac{x_a^2}{2n}} \tag{6.2}$$

However, for large n and $|x_a| << 1$, the above probabilities decay exponentially which significantly reduces their effectiveness for anomaly detection. Thus, we use $\frac{x_a^2}{2n}$ as the anomaly score for our model. Thus generalizing this, the anomaly score for each individual observation is given by:

$$a_i = n\mathcal{I}(x_i) \quad \forall i \in \{1, 2, \dots, n\} \tag{6.3}$$

6.1.2 LDP for High Dimensional Data

High dimensional data pose significant challenges to anomaly detection. The presence of redundant or irrelevant features acts as noise, making anomaly detection difficult. However, dimensionality reduction can impact anomalies that arise from less significant features of the datasets. To address this, we use the Dawson-Gärtner Projective theorem in the LAD model to compute the rate function for high dimensional data. The theorem records the maximum value across all projections which preserves the anomaly score making it optimal to detect anomalies in high dimensional data. The model algorithm is presented in Algorithm 4.

Algorithm 4 Algorithm 1: LAD Model

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th.
Output: Anomaly score **a**
Initialization: Set initial anomaly score and labels **a** and **I** to zero vectors and, entropy matrix $E = 0_{(n,d)}$ where $0_{(n,d)}$ is a zero matrix of size (n, d).

```
for each $s <= N_{iter}$ do
    Subset $X_{sub} = X[I_i == 0]$
    $X_{normalized}[:, d_i] = \frac{X[:,d_i]-X_{sub}\bar{[}:,d_i]}{cov(X_{sub}[:,d_i])}, \quad \forall d_i \in \{1, \dots, d\}$
    $E[i, :] = -X_{normalized}[i]^2/2n, \quad \forall i$
    $a_i = -max(E[i, :])$
    $\mathbf{a} = \frac{\mathbf{a-min(a)}}{\mathbf{max(a)-min(a)}}$
    $th = min(th, quantile(\mathbf{a}, \mathbf{0.95})$
    $I_i = 1$ if $a_i > th, \quad \forall i$
end for
```

6.1.3 LAD for Time Series Data

The definition of an anomaly is often contingent on the data and the problem statement. Broadly, time series anomalies can be categorized into two groups [17]:

1. **Divergent trends/Process anomalies**: Time series with divergent trends that last for significant time periods fall into this group. Here, one can argue that the generative process of such time series could be different from the rest of the non-anomalous counterparts.

2. **Subsequence anomalies**: Such time series have temporally sudden fluctuations or deviations from expected behavior which can be deemed as anomalous. These anomalies occur as a subsequence of sudden spikes or fatigues in a time series of relatively non-anomalous trend.

The online extension of the LAD model is designed to capture anomalous behavior at each time step. Based on the mode of analysis of the temporal anomaly scores, one can identify both divergent trends and subsequence anomalies. In this part of the thesis, we focus on the divergent trends (or process anomalies). In particular, we try to look at the anomalous trends in COVID-19 cases and deaths in US counties. Studies to collectively identify divergent trends and subsequence anomalies are being considered as prospective future work.

In this section, we present an extension of the LAD model to multivariate time series data. Here, we wish to preserve the temporal dependency as well as dependency across different features of the time series. Thus, as shown in Algorithm 5, a horizontal stacking of the data is performed. This allows collective study of temporal and non-temporal features. To preserve temporal dependency, the anomaly scores and labels are carried on to the next time step where the labels are then re-evaluated.

As long term anomalies are of interest, time series with temporally longer anomalous behaviors are ranked more anomalous. The overall time series anomaly score A_n for each time series $\mathbf{t_n}$ can be computed as:

$$A_n = \frac{\sum_{t=1}^{T} I[n,t]}{T} \quad \forall n \tag{6.4}$$

For a database of time series with varying lengths, the time series anomaly score is computed by normalizing with respective lengths.

Similarly, the method can be extended to studying anomalies within an individual time series by breaking the series into a database of sub-sequences of a time series extracted via a

Algorithm 5 Algorithm 2: LAD for Time series anomaly detection

Input: Time series dataset $\{\mathbf{t_n}\}_{\mathbf{n=1}}^{\mathbf{N}}$ of size (N, T, d), number of iterations N_{iter}, threshold th, window w.
Output: An array of temporal anomaly scores **a**, an array of temporal anomaly labels I
Initialization: Set initial anomaly score and labels **a** and **I** to zero matrices of size (N, T) and, entropy matrix E to a zero matrix of size (N, T, d).

```
for each $t <= T$ do
    $X = hstack(\bar{t}_{n,t})$ where $\bar{t}_{n,t} = \{t_{n,t-w}, \dots t_{n,t}\}$
    $I[i,t] = I[i,t-1]$
    $\mathbf{a[:,t] = a[:,t-1]}$
    for each $s <= N_{iter}$ do
        Subset non-anomalous time series $X_{sub} = \{X[i,:] | I[i,t] == 0, \forall i\}$
        $X_{normalized}[:,d_i] = \frac{X[:,d_i] - \bar{X}_{sub}[:,d_i]}{cov(X_{sub}[:,d_i])}, \quad \forall d_i \in \{1, 2, \dots, d * w\}$
        $E[i,:] = -X_{normalized}[i]^2/2n, \quad \forall i$
        $\mathbf{a[i,t] = -max(E[i,:])}$
        $\mathbf{a[:,t] = \frac{a[:,t] - min(a[:,t])}{max(a[:,t]) - min(a[:,t])}}$
        $th = min(th, quantile(\mathbf{a[:,t], 0.95}))$
        $I[i,t] = 1$ if $\mathbf{a[i,t] > th, \quad \forall i}$
    end for
end for
```

sliding window. It must be noted that this approach allows for a retrospective classification of anomalies.

Chapter 7

Anomaly Detection for High Dimensional Data

7.1 Experiments

In this section, we evaluate the performance of the LAD algorithm on multi-aspect datasets. The following experiments have been conducted to study the model:

1. Anomaly Detection Performance: LAD's ability to detect real-world anomalies as compared to state-of-the-art anomaly detection models is evaluated using the ground truth labels.

2. Handling Large Data: Scalability of the LAD model on large datasets (high observation count or high dimensionality) are studied.

3. Speed: The computation and execution times of different algorithms are studied and evaluated.

4. Time series Anomaly Detection: The LAD model is used to classify anomalies in time series (retrospective study). The model's performance is compared with similar scoring based anomaly detection algorithms.

5. COVID-19 Time Series Data: We study the performance of the LAD model on multiple multivariate time series datasets to identify anomalous instances within each time step as well anomalous time series amongst many.

7.1.1 Datasets

We consider a variety of publicly available benchmark data sets from Outlier Detection DataSets/ODDS (Rayana, 2016) (See Tables 7.1) for the experimental evaluation. For anomaly detection within individual time series, we study univariate time series data from Numenta Benchmark Datasets [59] (See Tables 7.2). Additionally, for the time series data, we use COVID-19 deaths and confirmed cases for US counties from John Hopkins COIVD-19 Data Repository (Dong, Du, and Gardner, 2020). The country level global data for COVID-19 trends was taken from the Our World in Data Repository (**owidcoronavirus**, **owidcoronavirus**).

Name	N	d	a
HTTP	567498	3	0.39%
MNIST	7603	100	9.207%
Arrhythmia	452	274	14.602%
Shuttle	49097	9	7.151%
Letter	1600	32	6.25%
Musk	3062	166	3.168%
Optdigits	5216	64	2.876%
Satellite Image	6435	36	31.639%
Speech	3686	400	1.655%
SMTP	95156	3	0.032%
Satellite Image-2	5803	36	1.224%
Forest Cover	286048	10	0.96%
KDD99	620098	29	29 0.17%

Table 7.1: High Dimensional and Large Sample Datasets: Description of the benchmark data sets used for evaluation of the anomaly detection capabilities of the proposed model. N - number of instances, d - number of attributes and a - the fraction of known anomalies in the data set.

Dataset	N	a
EC2 CPU UTILIZATION 825CC2	4032	0.09%
EC2 NETWORK IN 257A54	4032	0.1%
EC2 CPU UTILIZATION 5F5533	4032	0.1%
EC2 CPU UTILIZATION AC20CD	4032	0.1%
EC2 CPU UTILIZATION 24AE8D	4032	0.1%
SPEED 7578	1127	0.1%
SPEED 6005	2500	0.1%
OCCUPANCY 6005	2380	0.1%
SPEED T4013	2495	0.1%
ART LOAD BALANCER SPIKES	4032	0.1%
EXCHANGE-3 CPM RESULTS	1538	0.1%
EXCHANGE-4 CPM RESULTS	1643	0.1%
TWITTER VOLUME KO	15851	0.1%
TWITTER VOLUME CVS	15853	0.1%
TWITTER VOLUME CRM	15902	0.1%
MACHINE TEMPERATURE SYSTEM FAILURE	22695	0.1%
EC2 REQUEST LATENCY SYSTEM FAILURE	4032	0.09%
CPU UTILIZATION ASG MISCONFIGURATION	18050	0.08%

Table 7.2: Benchmark Time Series: Description of the benchmark time series used for evaluation. N - number of instances, d - number of attributes and a - the fraction of known anomalies in the data set.

7.1.2 Baseline Methods and Parameter Initialization

As described in Section 6.1, LAD falls under the unsupervised learning regime targeted for high dimensional data, we do not compare with supervised algorithms. For this we consider *Elliptic Envelope* (EE) [65], *Isolation Forest* (I-Forest) [53][1], *local outlier factor* (LOF) [12], and *Concentration Free Outlier Factor* CFOF [4]. The CFOF and LOF models assign an anomaly score for each data instance, while the rest of the methods provide an anomaly label. As above mentioned methods have one or more user-defined parameters, we investigated a range of values for each parameter, and report the best results. For Isolation Forest, Elliptic Envelope and CFOF, the contamination value is set to the true proportion

[1] The I-Forest model returns both anomaly scores and anomaly labels. As the classification model outperforms its score based counterpart on the above discussed datasets, we only present results on the classification model.

of anomalies in the dataset.

To study anomaly detection in time series, the LAD model is compared with other score based time series anomaly detection algorithms like Twitter AD Vec, Skyline, Earthgecko Skyline (E.Skyline), Numenta, Relative Entropy (RE), Random Cut Forest (RCF), Windowed Gaussian (WG).

The LAD model relies on a threshold value to classify observations with scores over the value as strictly anomalous. Though this value is iteratively updated, an initial value is required by the algorithm. For the LAD model in this part of the thesis, the initial threshold value for the experiment is set to 0.95 for all datasets.

All the methods for anomaly detection benchmark datasets are implemented in Python and all experiments were conducted on a 2.7 GHz Quad-Core Intel Core i7 processor with a 16 GB RAM.

7.1.3 Evaluation Metrics

As LAD is a score based algorithm, we study the ROC curves by comparing the True Positive Rate (TPR) and False Positive Rate (FPR), across various thresholds. The final ROC-AUC (Area under the ROC curve) is reported for evaluation.

For anomaly detection within individual time series, we use the F-measure as the evaluation metric to study the overall performance of the model. Since all the models return anomaly scores, thresholds were used to classify observations as anomalous vs non-anomalous. Threshold was set to be the maximum score in the truly non-anomalous data for each model and the observations with scores higher than the set threshold were labeled anomalous. This is to ensure that the model is able to distinguish anomalies from the rest of the data.

For time series database anomaly detection, we present the final outliers and study their deviations from normal baselines under different model settings.

7.1.4 Anomaly Detection Performance

We present the results on small and large datasets. The Table 7.3 presents data description of small datasets for evaluation. Table 7.4 shows the performance of the LAD model on these datasets. It can be seen that LAD outperforms other algorithms in most datasets.

Name	N	d	a
Pima	768	8	34.896%
Wine	129	13	7.752%
Cardio	1831	21	9.612%
Pendigits	6870	16	2.271%
Thyroid	3772	6	2.466%
Vowels	1456	12	3.434%
Breast cancer	683	9	34.993%
Lympho	148	18	4.054%
Annthyroid	7200	6	7.417%
WBC	378	30	5.556%
Mammography	11183	6	2.325%
Glass	214	9	4.206%
Cover	286048	10	0.96%
Vertebral	240	6	12.5%

Table 7.3: Description of the benchmark data sets used for evaluation of the anomaly detection (*source*: Outlier Detection DataSets /ODDS (Rayana, 2016)) capabilities of the proposed model. N - number of instances, d - number of attributes, and a - the fraction of known anomalies in the data set.

Table 7.5 shows the performance of LOF, I-Forest, EE, CFOF and LAD on anomaly detection benchmark datasets. Due to relatively large run-time[2], CFOF results are shown for datasets with samples less than 10k. For all the listed algorithms, results for the best parameter settings are reported. The proposed LAD model outperforms other methods on most data sets. For larger and high-dimensional datasets, it can be seen from Table 7.5 that the LAD model outperforms all the models in most settings.[3]

To study the LAD model's computational effectiveness, we study the computation time

[2] The CFOF model is computationally expensive relative to the rest of the algorithms. As it is aimed to study high-dimensional data, only results on datasets with <10k observations are presented.

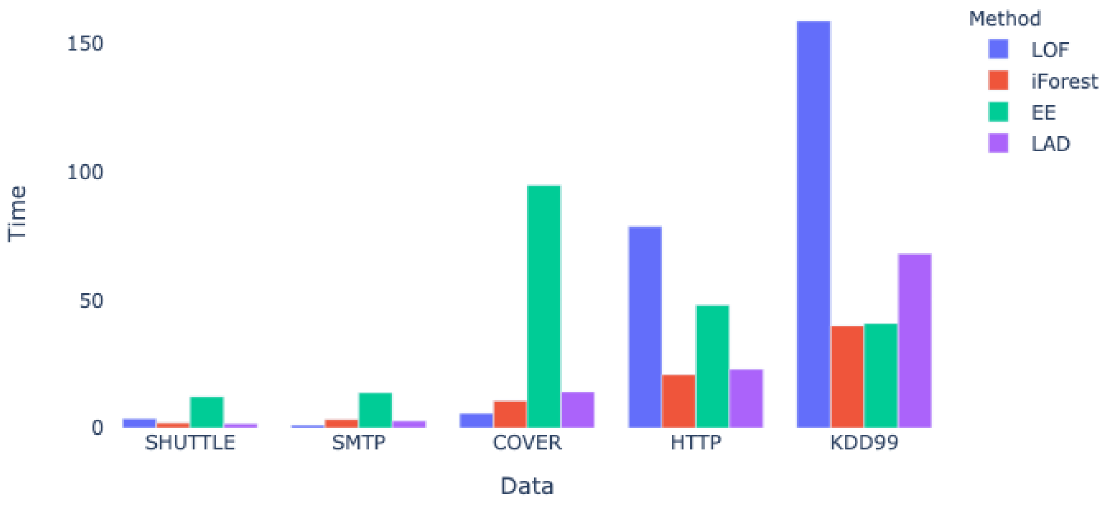
[3] The lowest AUC values for the LAD model are observed for Speech and Optdigits data where multiple true clusters are noted.

Table 7.4: Comparing LAD with existing anomaly detection algorithms using ROC-AUC as the evaluation metric.

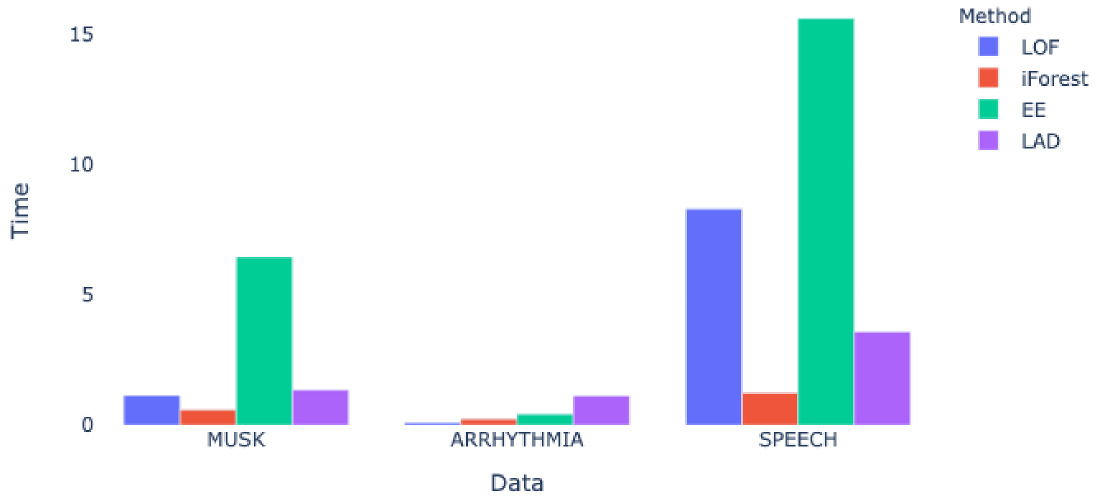
Data	LOF	KMeans–	KNN	OCSVM	iForest	EE	LAD
PIMA	0.59	0.64	0.64	0.52	0.62	0.62	0.68
VERTEBRAL	0.51	0.43	0.36	0.61	0.45	0.43	0.35
THYROID	0.62	0.56	0.95	0.51	0.77	0.82	0.92
WBC	0.95	0.8	0.94	0.47	0.8	0.7	0.95
CARDIO	0.59	0.81	0.87	0.57	0.76	0.71	0.96
MAMMOGRAPHY	0.74	0.63	0.85	0.55	0.6	0.49	0.87
GLASS	0.72	0.54	0.87	0.73	0.54	0.48	0.73
BREASTW	0.35	0.23	0.99	0.81	0.95	0.95	0.96
VOWELS	0.89	0.58	0.81	0.64	0.6	0.51	0.77
PENDIGITS	0.54	0.59	0.74	0.51	0.71	0.54	0.91
AVERAGE RANK	4.75	5.6	2.85	6.2	5.1	6.2	2.7

Table 7.5: Comparing LAD with existing anomaly detection algorithms for large/ high dimensional datasets using ROC-AUC as the evaluation metric.

Data	LOF	I-Forest	EE	CFOF	LAD
SHUTTLE	0.52	0.98	0.96	-	0.99
SATIMAGE-2	0.57	0.95	0.96	0.70	0.99
SATIMAGE	0.51	0.64	0.65	0.55	0.6
KDD99	0.51	0.85	0.54	-	1.0
ARRHYTHMIA	0.61	0.67	0.7	0.56	0.71
OPTDIGITS	0.51	0.52		0.49	0.48
LETTER	0.54	0.54	0.6	0.90	0.6
MUSK	0.5	0.96	0.96	0.49	0.96
HTTP	0.47	0.95	0.95	-	1.0
MNIST	0.5	0.61	0.65	0.75	0.87
COVER	0.51	0.63	0.52	-	0.96
SMTP	0.84	0.83	0.83	-	0.82
SPEECH	0.5	0.53	0.51	0.47	0.47



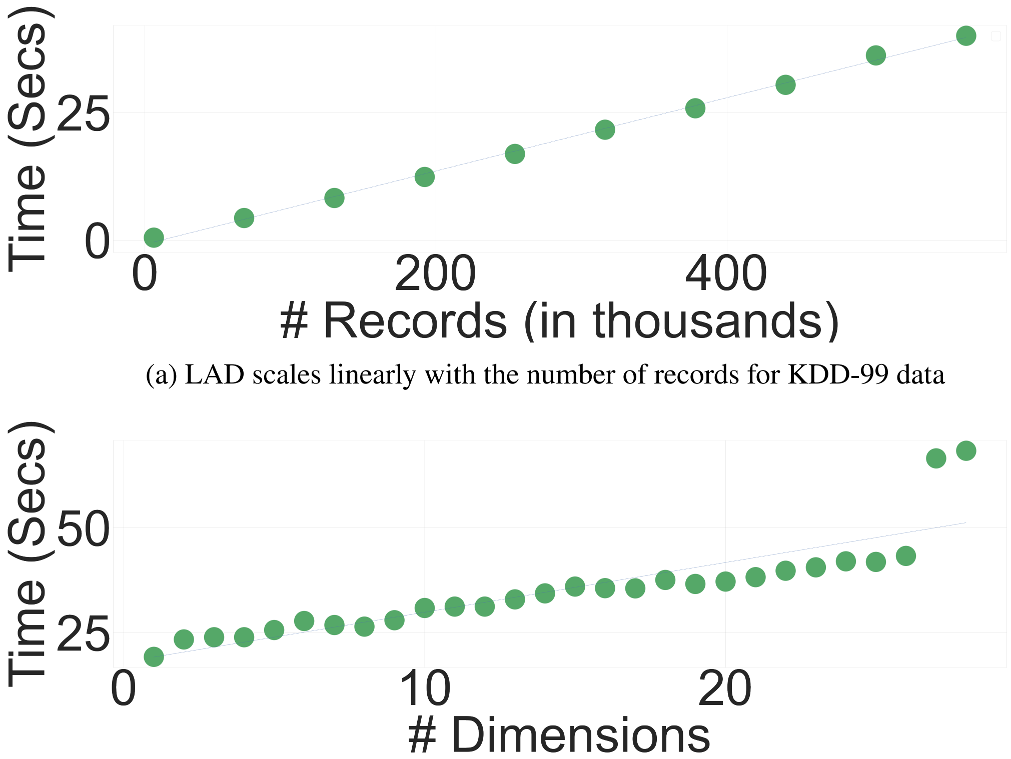
(a) Computation time for large datasets



(b) Computation time for high dimensional datasets

Figure 7.1: Computation time for large and high-dimensional datasets: The figure shows the execution time in seconds for different datasets. The LAD model presents a significant advantage over other state-of-the-art models as illustrated here.

and scaling of the LAD model on large and high dimensional datasets. We consider datasets with more than 10k observations or over 100 features for our analysis. Figures 7.1a and 7.1b show the computation time in seconds for benchmark datasets. It can be seen that the LAD model has a relatively low computation time, second only to Isolation Forest in most datasets. In fact, the computation time is more stable for our model as opposed to others in high dimensional datasets.



(a) LAD scales linearly with the number of records for KDD-99 data

(b) LAD scales linearly with the number of dimensions in KDD-99 data.

Figure 7.2: Computation time for KDD-99 data: The figure shows the scaling of the LAD model for different number of records and increasing dimensionality of the data.

Figure 7.2a shows the scalability of the LAD with respect to the number of records in the data. We plot the time needed to run on the first k records of the KDD-99 dataset. Each record has 29 dimensions. Figure 7.2b shows the scalability of the LAD with respect to the number of dimensions (linear-scale). We plot the time needed to run on the first 1, 2, ..., 29 dimensions of the KDD-99 dataset. The results confirm the linear scalability of the LAD with the number of records as well as the number of dimensions.

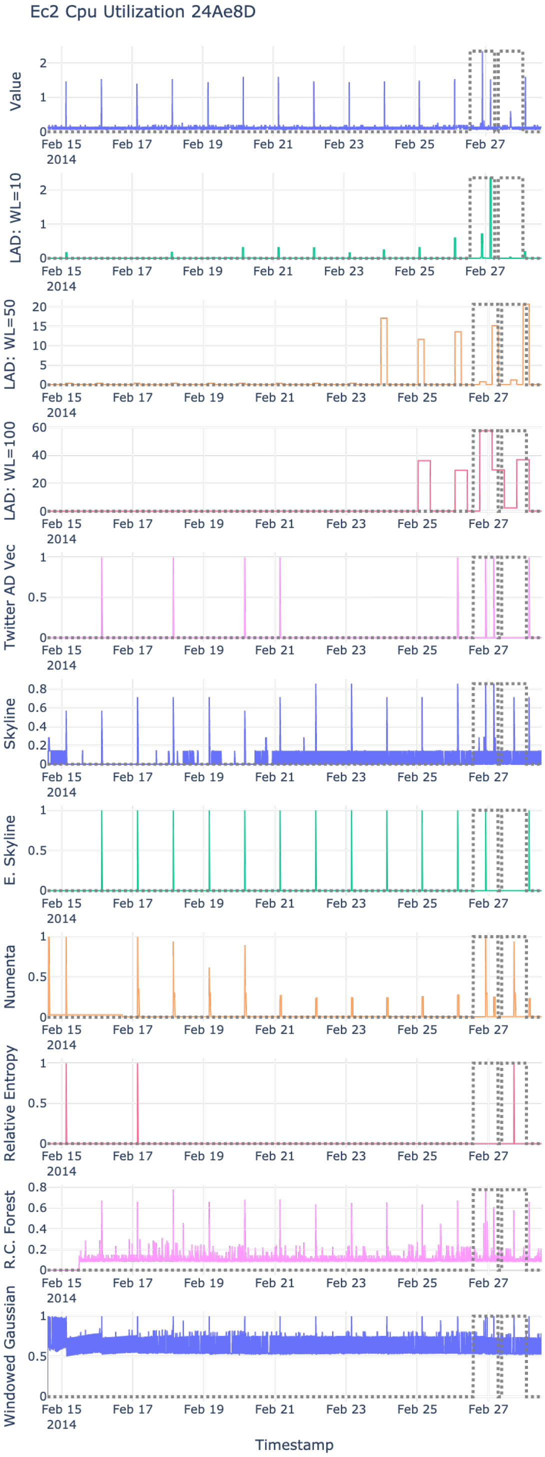
7.1.5 Anomaly Detection in Time Series

In Table 7.6, we compare the performance of the LAD model as compared to other score-based algorithms. In particular, it can be seen that the LAD model with a window length of 100 has the best anomaly detection performance as compared to other methods in most datasets.

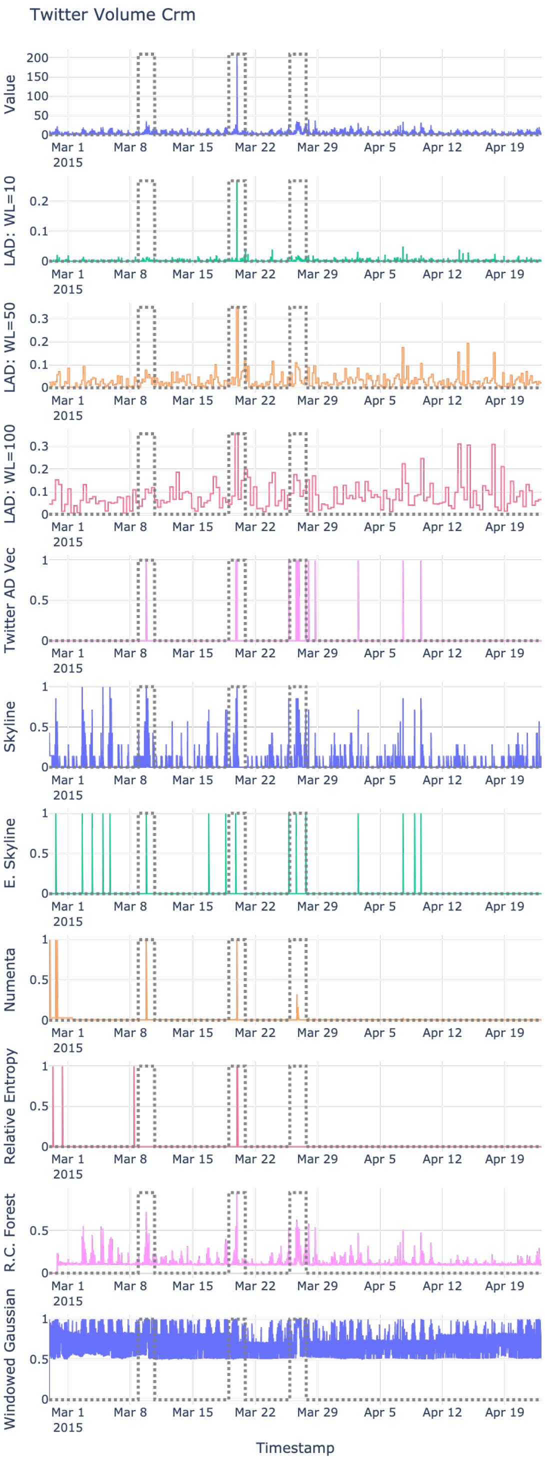
In particular, we can see that the model has higher anomaly scores for truly anomalous data as compared to the rest of the time series as seen in Figures 7.3.

Table 7.6: Comparing LAD with existing anomaly detection algorithms for time series datasets using F-measure as the evaluation metric.

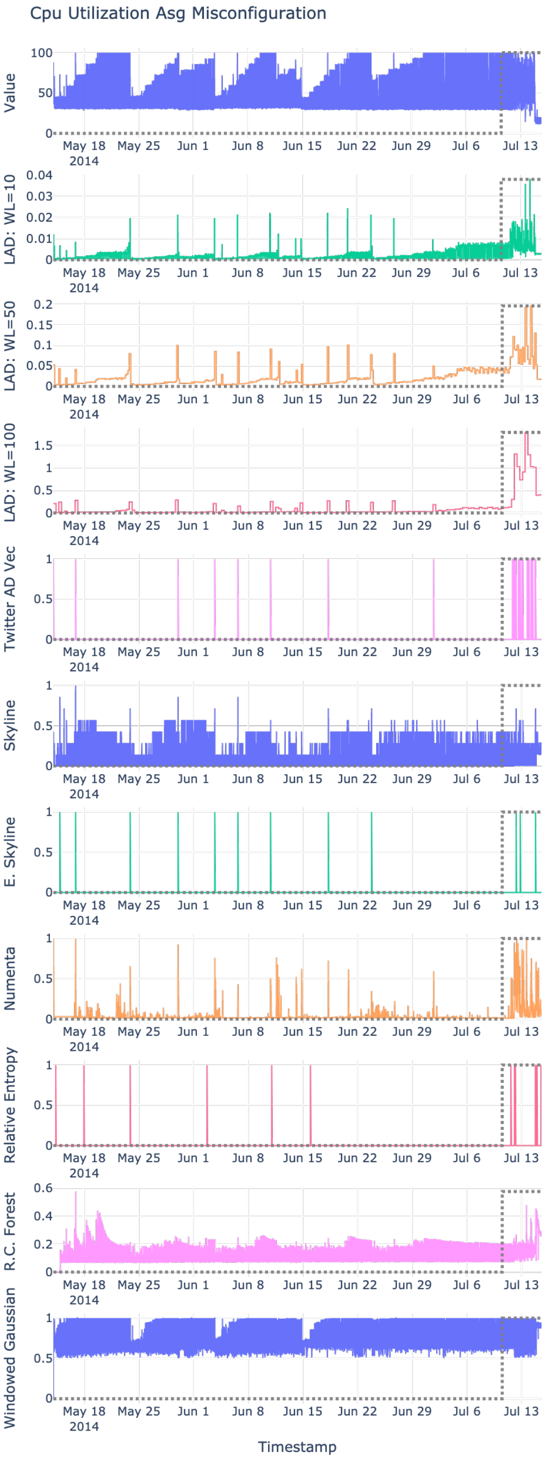
Data	LAD: WL=10	LAD: WL=50	LAD: WL=100	Twitter AD Vec	Skyline	E.Skyline	Numenta	RE	RCF	WG
EC2 CPU UTILIZATION 825CC2	0.0	0.1	0.37	0.16	**0.45**	0.16	0.03	0.05	0.13	0.19
EC2 NETWORK IN 257A54	0.14	0.25	**0.33**	0.03	0.04	0.18	0.02	0.01	0.03	0.02
EC2 CPU UTILIZATION 5F5533	0.14	0.36	**0.57**	0.18	0.03	0.18	0.01	0.03	0.04	0.0
EC2 CPU UTILIZATION AC20CD	0.0	0.31	**0.33**	0.03	0.02	0.01	0.01	0.03	0.0	0.11
EC2 CPU UTILIZATION 24AE8D	0.09	0.12	**0.59**	0.01	0.01	0.0	0.0	0.0	0.0	0.01
SPEED 7578	0.26	0.29	**0.54**	0.19	0.08	0.05	0.05	0.08	0.02	0.17
SPEED 6005	0.15	**0.59**	**0.59**	0.04	0.11	0.11	0.03	0.04	0.04	0.01
OCCUPANCY 6005	0.08	0.29	**0.5**	0.01	0.01	0.01	0.01	0.01	0.01	0.0
SPEED T4013	0.27	**0.88**	0.45	0.15	0.16	0.02	0.04	0.03	0.13	0.14
ART LOAD BALANCER SPIKES	0.08	**0.16**	0.15	0.02	0.01	0.0	0.0	0.01	0.0	0.08
EXCHANGE-3 CPM RESULTS	0.0	0.4	**0.77**	0.01	0.01	0.01	0.01	0.03	0.01	0.01
EXCHANGE-4 CPM RESULTS	**0.21**	0.21	0.17	0.02	0.04	0.04	0.05	0.19	0.05	0.05
TWITTER VOLUME KO	0.01	0.06	**0.11**	0.01	0.01	0.0	0.01	0.0	0.0	0.03
TWITTER VOLUME CVS	0.04	0.06	**0.12**	0.03	0.01	0.01	0.01	0.01	0.01	0.03
TWITTER VOLUME CRM	0.01	0.06	**0.11**	0.03	0.01	0.0	0.0	0.01	0.01	0.01
MACHINE TEMPERATURE SYSTEM FAILURE	0.02	0.04	0.08	**0.18**	0.03	0.01	0.0	0.02	0.03	0.0
EC2 REQUEST LATENCY SYSTEM FAILURE	0.2	**0.62**	0.35	0.15	0.04	0.15	0.02	0.15	0.03	0.02
CPU UTILIZATION ASG MISCONFIGURATION	0.03	0.24	**0.83**	0.04	0.0	0.0	0.0	0.02	0.0	0.0



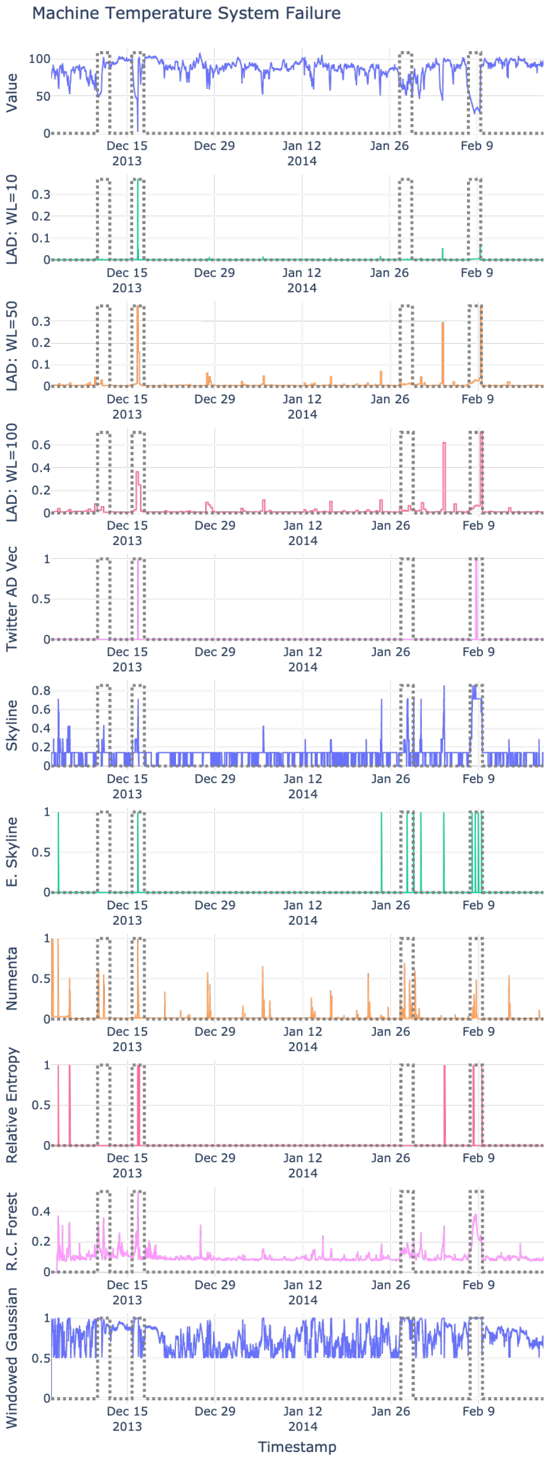
(a) Time Series Performance in CPU Utilization data



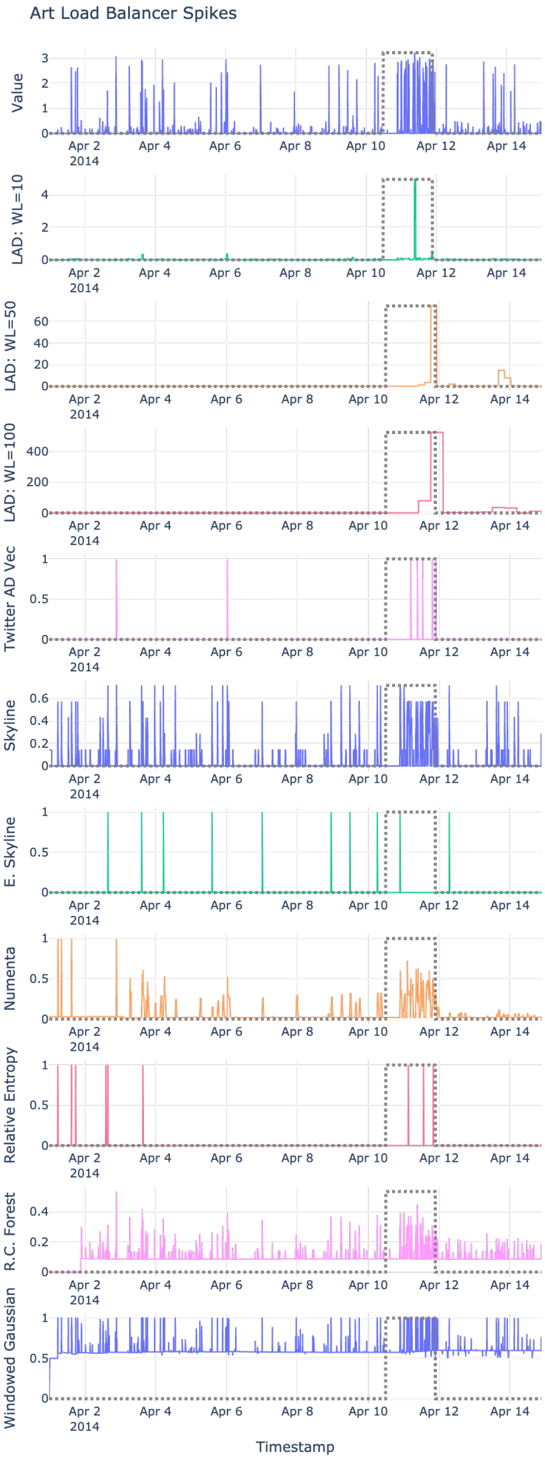
(b) Time Series Performance in Twitter Volume data



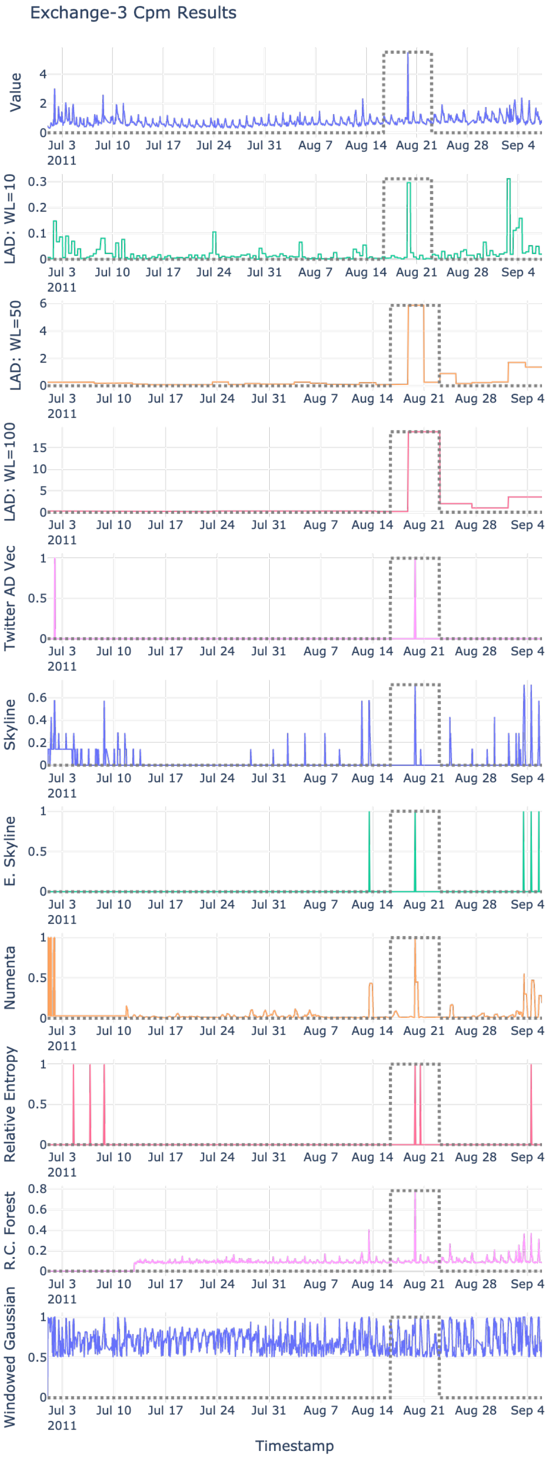
(c) Time Series Performance in CPU Utilization Misconfiguration



(d) Time Series Performance in Machine Temperature System Failure data



(e) Time Series Performance in Art Load Balancer Spikes data



(f) Time Series Performance in Exchange-3 CPM Results data

Figure 7.3: Anomaly detection within individual time series: The above images illustrate the performance of LAD in comparison to the state of the art time series anomaly detection algorithms. The anomaly scores for all time series observations are plotted for each algorithm. The dotted grey line indicates the observations labeled true anomalies within the data.

Chapter 8

Anomaly Detection in COVID-19 Trends

8.1 Anomaly Detection in Time Series Data

This section presents the results of the LAD model on COVID-19 time series data at the US county level. Multiple settings were used to understand the data:

1. Deaths and confirmed case trends were considered for analysis

2. Daily New vs Total Counts: Both total cases as well daily new cases were analyzed for anomaly detection.

3. Complete history vs One Time Step: Two versions of the model were studied where data from previous time steps were and were not considered. By this, we tried to distinguish the impact of the history of the time series on identifying anomalous trends.

4. Univariate vs Multivariate Time Series data: To further understand the LAD model, the deaths and case trends were studied individually as a univariate time series as well as collectively in a multivariate time series data setting.

5. Time Series of Uniform vs Varying Lengths: Finally, all the above analyses were conducted on time series data with varying lengths. Here, for each county level time

series, the time of the first event was considered as initial time step to objectively study the relative temporal changes in trends.

To bring all the counts to a baseline, the total counts in each time series were scaled to the respective county population. Missing information was replaced with zeros and counties with a population less than 100k were eliminated from the study.

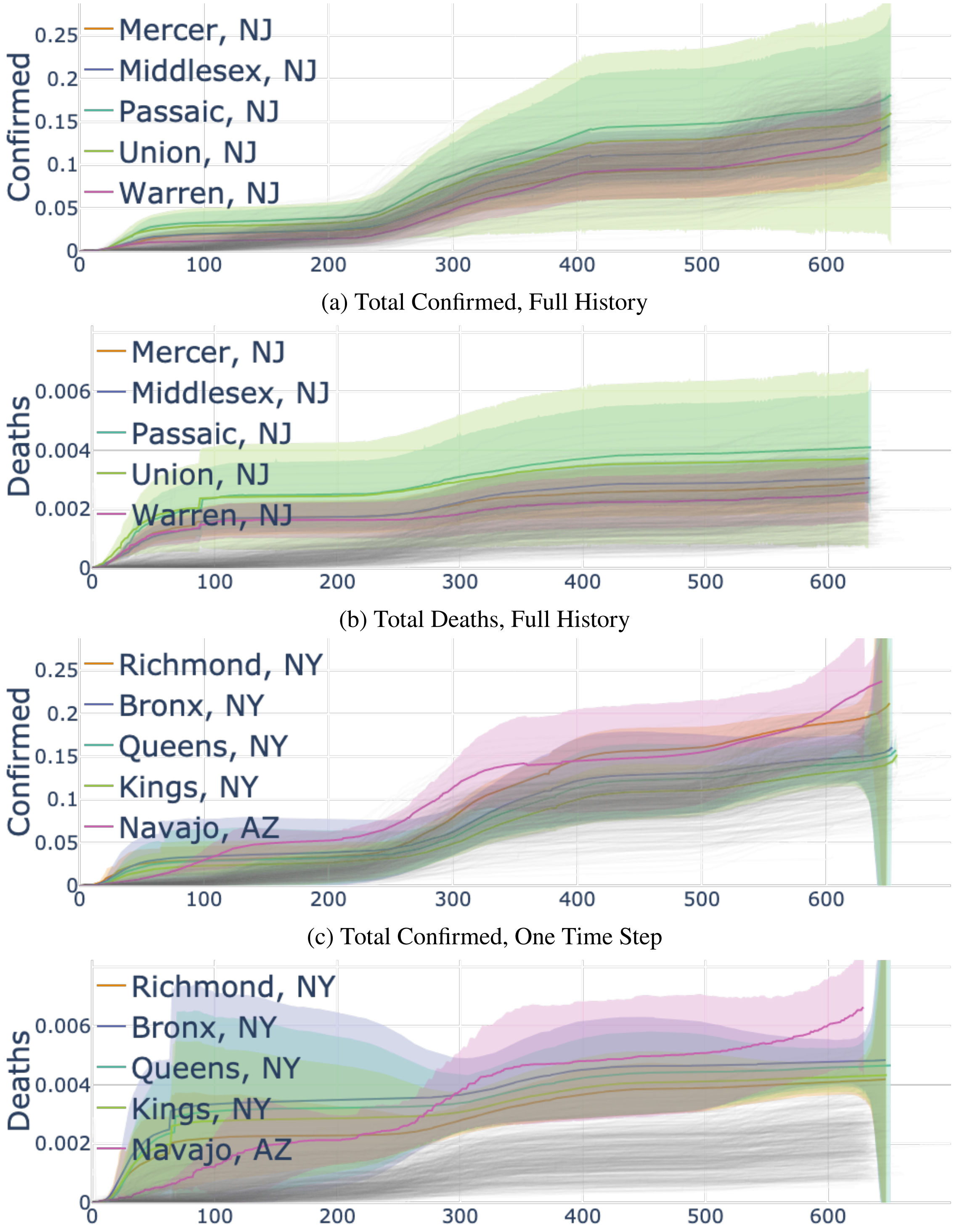
8.1.1 Discoveries: US COVID-19 Trends

In this section, we look at the trends since the start of 2020. We look at the daily new case and daily deaths in US counties. To rank the counties, anomaly scores between January 1 2020 - December 22 2021 were considered to identify most anomalous counties.

Complete history vs One Time Step The full history setting considers the complete history of the time series and is aimed to capture the most deviant trends over time. The one time step (or any smaller window) setting is more suitable to study deviations within the specific window. As we target long term deviating trends, the one time step setting returns trends that have stayed most deviant throughout the entire time range. This can be seen in Figures 8.1 and 8.2 where the one time step setting returns trends that have stayed deviant almost throughout the duration while the full history setting is able to capture significantly higher overall deviations from normal trends and therefore higher anomaly score. For instance, counties like Mercer(NJ), Union (NJ), that had extensive testing conducted[1] were captured in the one time step model as seen in Figures 8.1c and 8.1d. Similarly, counties in NY observed a peak in early 2021 [2], which was not captured as anomalous in the one time step model as seen in Figures 8.2a and 8.2b.

[1] https://www.nj.com/coronavirus/2021/12/more-covid-testing-sites-opening-as-cases-
html

[2] https://www.newsday.com/news/health/coronavirus/
coronavirus-long-island-deaths-vaccinations-1.50200404



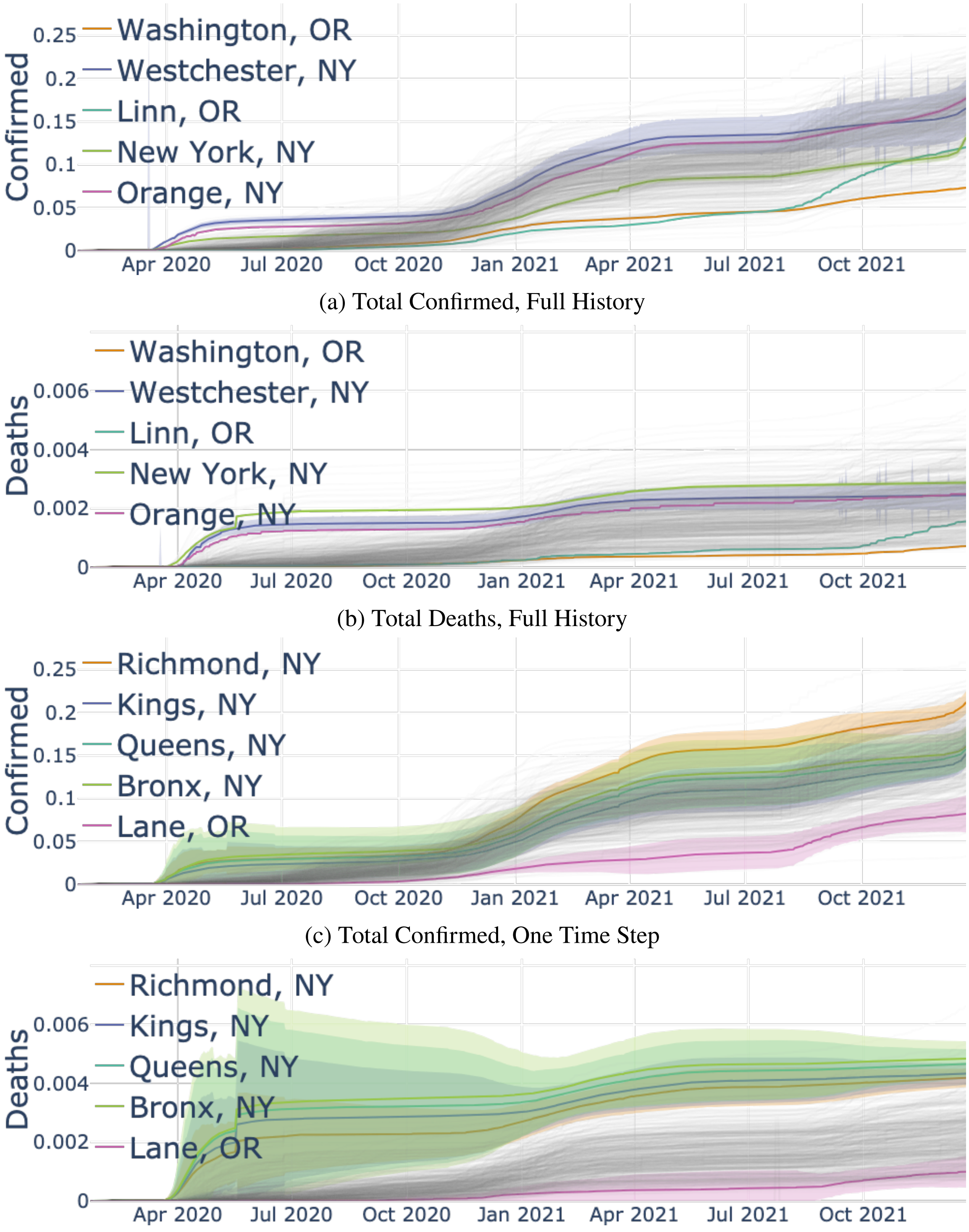
(a) Total Confirmed, Full History

(b) Total Deaths, Full History

(c) Total Confirmed, One Time Step

(d) Total Deaths, One Time Step

Figure 8.1: Top 5 Counties with Anomalous Trends: Varying lengths, Total Counts, Multivariate Time Series



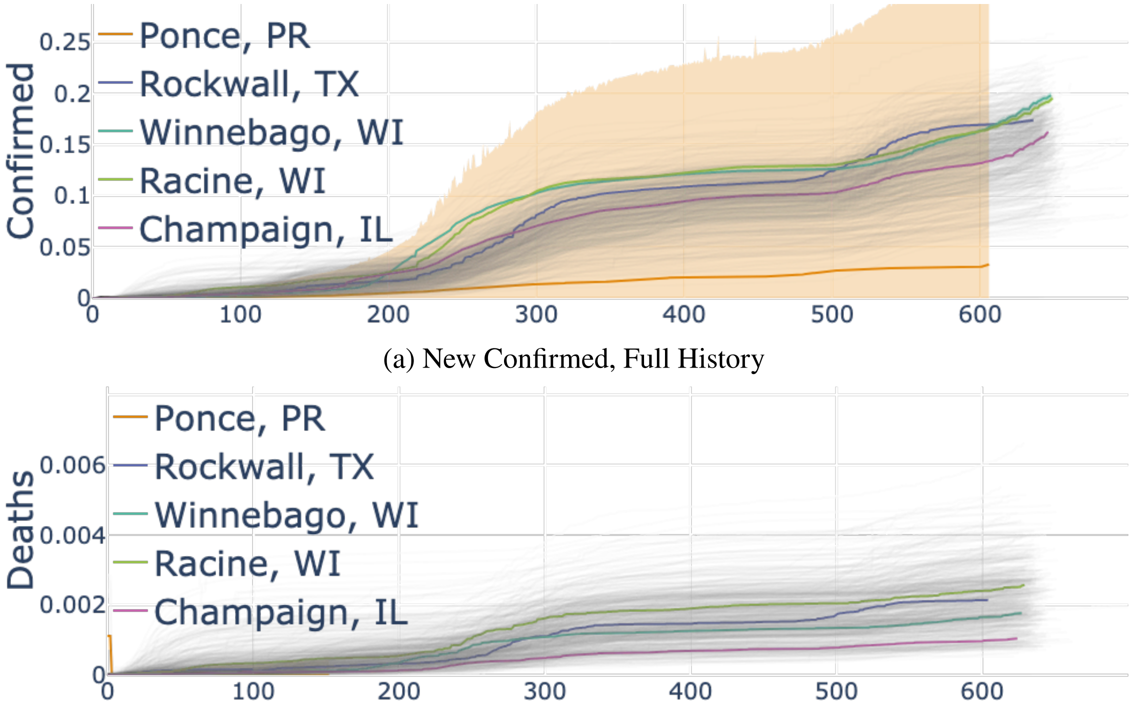
(a) Total Confirmed, Full History

(b) Total Deaths, Full History

(c) Total Confirmed, One Time Step

(d) Total Deaths, One Time Step

Figure 8.2: Top 5 Counties with Anomalous Trends: Uniform lengths, Total Counts, Multivariate Time Series



(a) New Confirmed, Full History

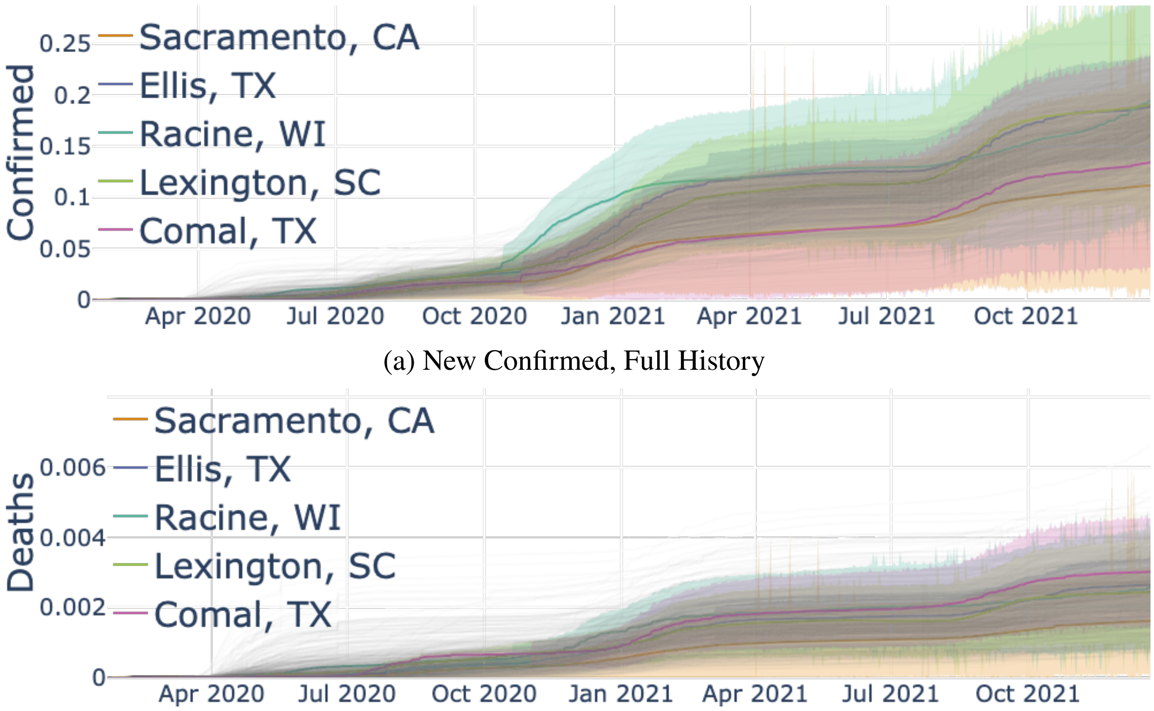
(b) New Deaths, Full History

Figure 8.3: Top 5 Counties with Anomalous Trends: Varying lengths, Daily New Counts, Multivariate Time Series

Univariate vs Multivariate Time series In Figures 8.1, 8.2, 8.3 and 8.4 we see the anomalous trends in multivariate time series, where total confirmed cases and deaths were collectively evaluated for anomaly detection. For instance, despite the near-normal trends in confirmed cases, Kings, Queens and Bronx (NY)[3] in Figures 8.1a- 8.1b, were identified anomalous due to their deviant death trends which significantly contributed to the anomaly scores. This setting enables identification of time-series with at least one deviating feature.

Daily New vs Total Counts Figures 8.2 and 8.4, show anomalous trends in multivariate time series for total and daily new counts respectively. It can be seen that the anomaly score is relatively more erratic for trends new case counts. This is due to the fact that the data for new case and death counts is more erratic leading to fluctuating normal average as well as non-smooth anomaly scores. Similar behavior can be seen across Figures 8.1 and 8.3.

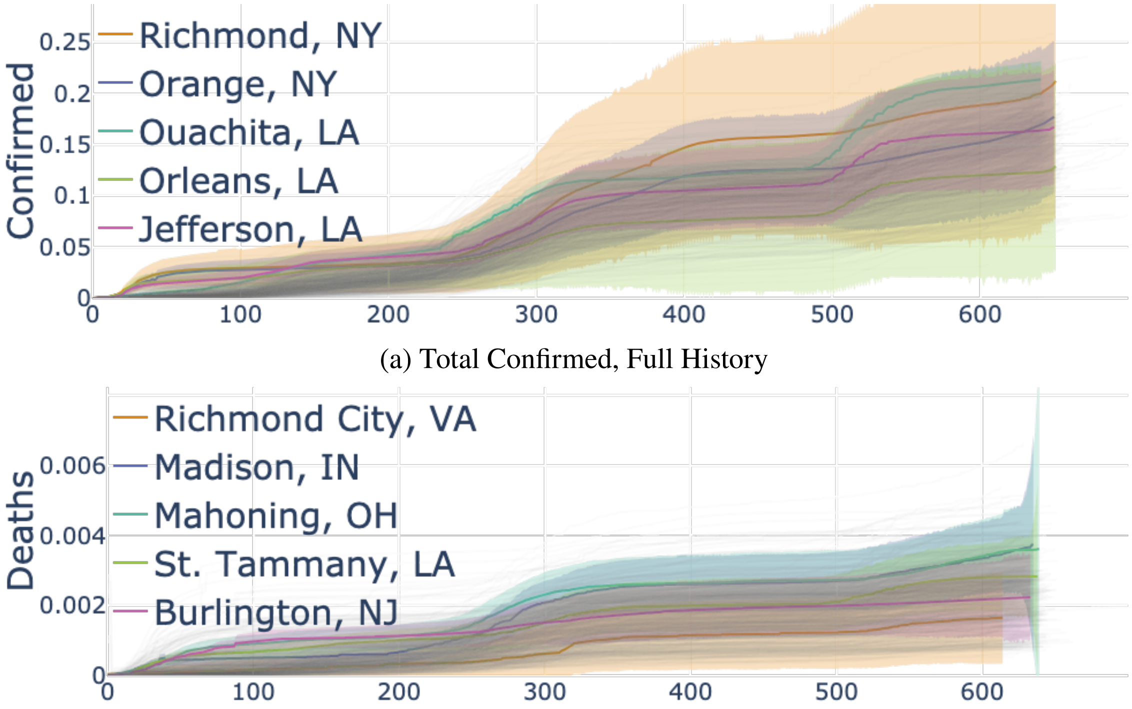
[3] https://www.nbcnewyork.com/news/coronavirus/nyc-mask-mandate-indoors-an-option-if
3428102/



(a) New Confirmed, Full History

(b) New Deaths, Full History

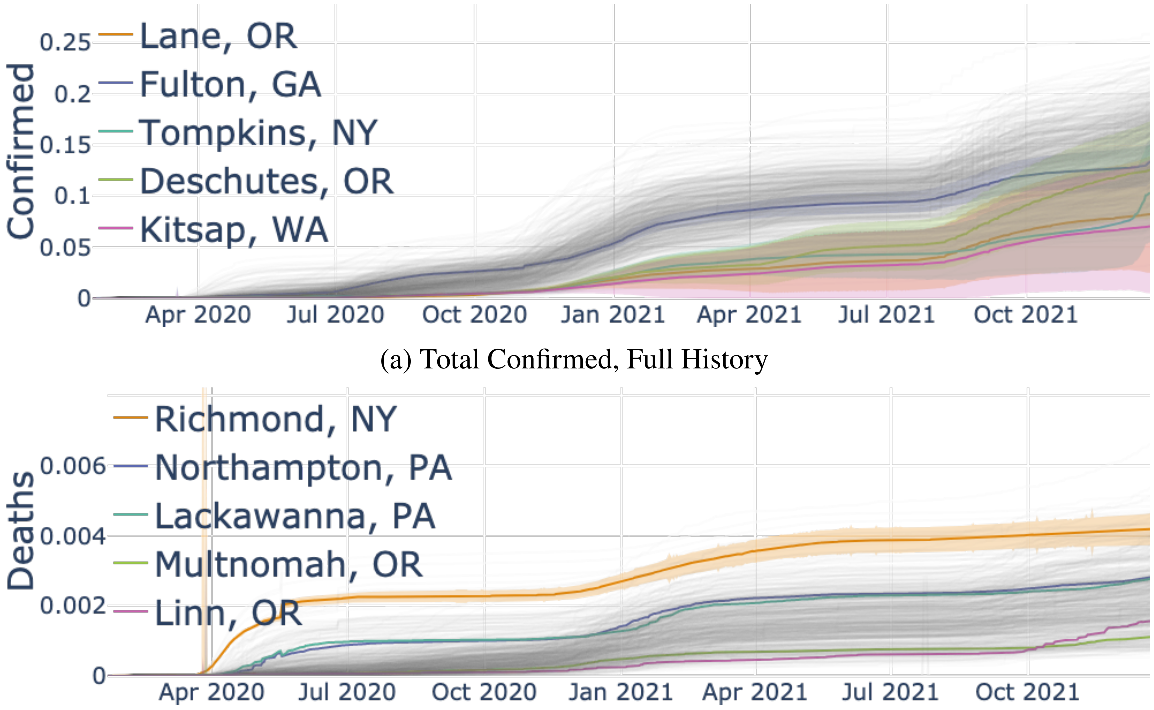
Figure 8.4: Top 5 Counties with Anomalous Trends: Uniform lengths, Daily New Counts, Multivariate Time Series



(a) Total Confirmed, Full History

(b) Total Deaths, Full History

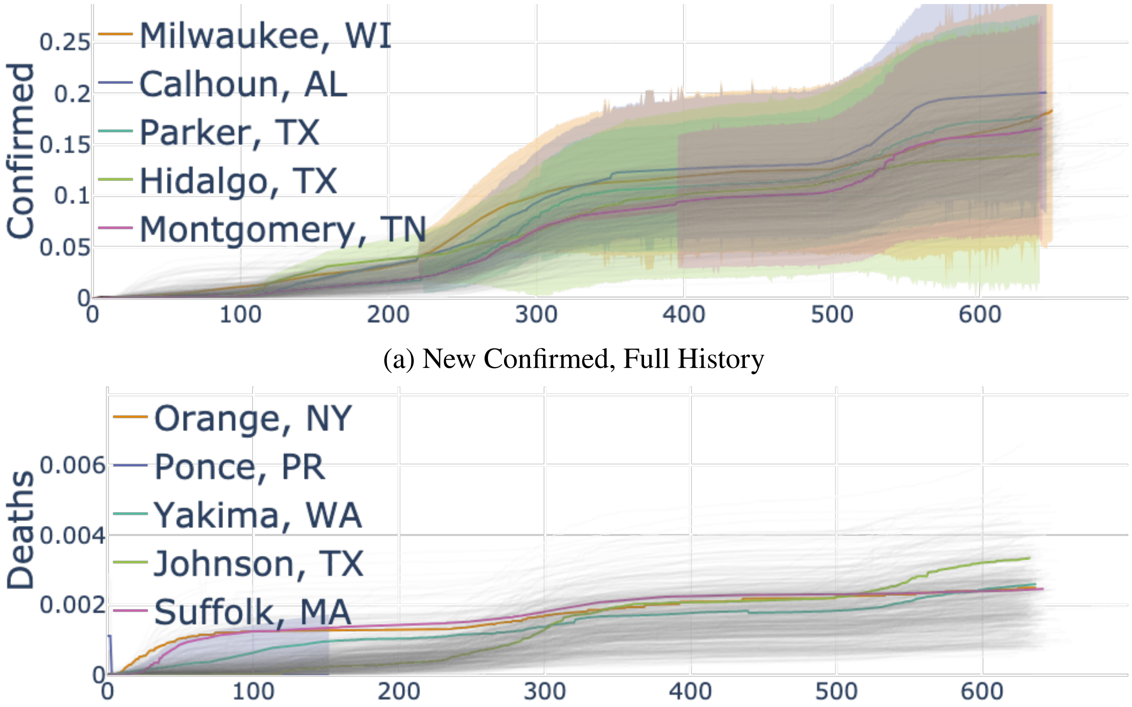
Figure 8.5: Top 5 Counties with Anomalous Trends: Varying lengths, Total counts



(a) Total Confirmed, Full History

(b) Total Deaths, Full History

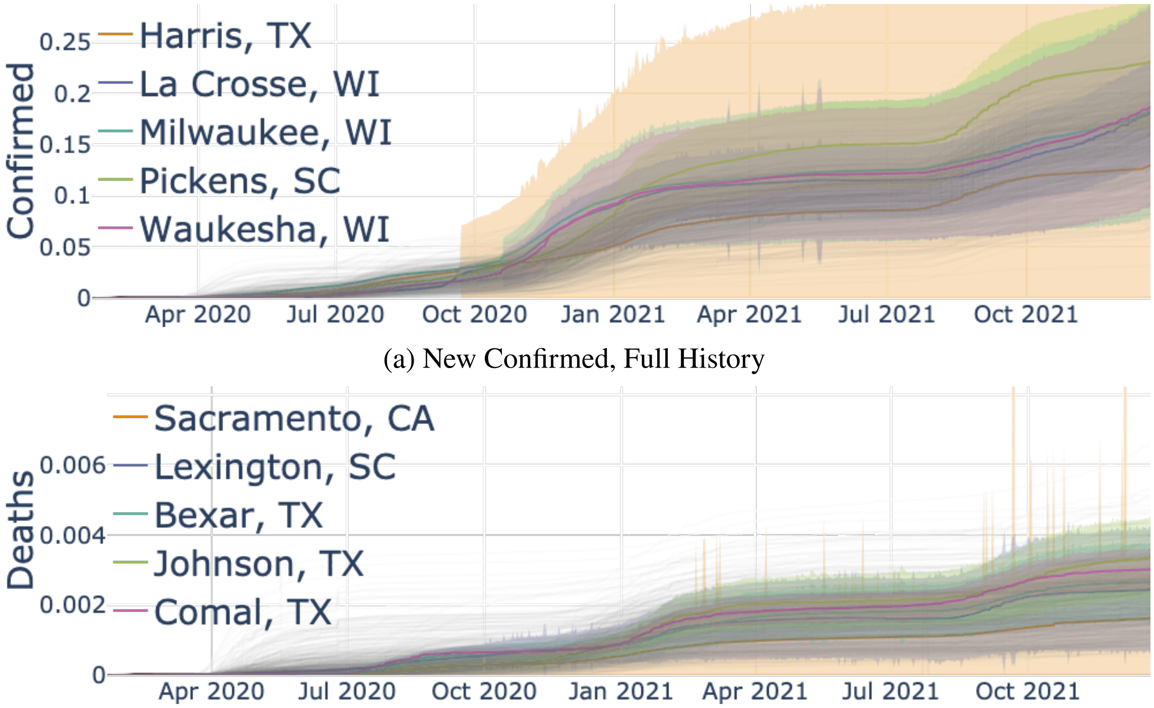
Figure 8.6: Top 5 Counties with Anomalous Trends: Uniform lengths, Total counts



(a) New Confirmed, Full History

(b) New Deaths, Full History

Figure 8.7: Top 5 Counties with Anomalous Trends: Varying lengths, Daily New Counts



(a) New Confirmed, Full History

(b) New Deaths, Full History

Figure 8.8: Top 5 Counties with Anomalous Trends: Uniform lengths, Daily New Counts

The LAD model on the daily new counts data was able to capture the escalation in Racine, Wisconsin in Figure 8.4a and 8.4b during late 2020 when multiple meatpacking plants were tied to COVID-19 cases[4].

Uniform Length vs Varying Length Time Series The US county cases and deaths data consists of time series of uniform lengths. However, not all counties have events recorded in the early stages. Thus, studying the non-synchronized database creates a bias against counties with early reported cases. On the other hand, counties with longer reporting on trends or earlier outbreaks tend to be associated with higher anomaly scores towards the most recent data due to lack of equally long time series.

This can be seen in Figures 8.2 where counties like Lane, Oregon that was flagged anomalous due to distinctively low cases due to later outbreak of the pandemic much after

[4] `https://www.jsonline.com/story/news/2020/11/25/meatpacking-plants-tied-more-covid 6376197002/`

many counties in NY, unlike in Figures 8.1 which reports counties in NY with an early start as highly anomalous in the later stages[5].

8.1.2 Global Trends and Emergence of Other COVID-19 Variants

In this section, we study the COVID-19 trends across countries globally. The Coronavirus Pandemic (COVID-19) Data from Our World in Data [**owidcoronavirus**] was used for the analysis. The study includes countries with population more than 5 million only. Trends in the daily new deaths and confirmed cases (7 day rolling average, right-aligned), biweekly growth rates in deaths and confirmed cases and case fatality rates were considered collectively as multivariate time series. Two sets of end dates were studied to analyze the onset of new variants namely the Delta and Omicron variants. Additionally, vaccination trends among countries were also analyzed.

Delta Variant

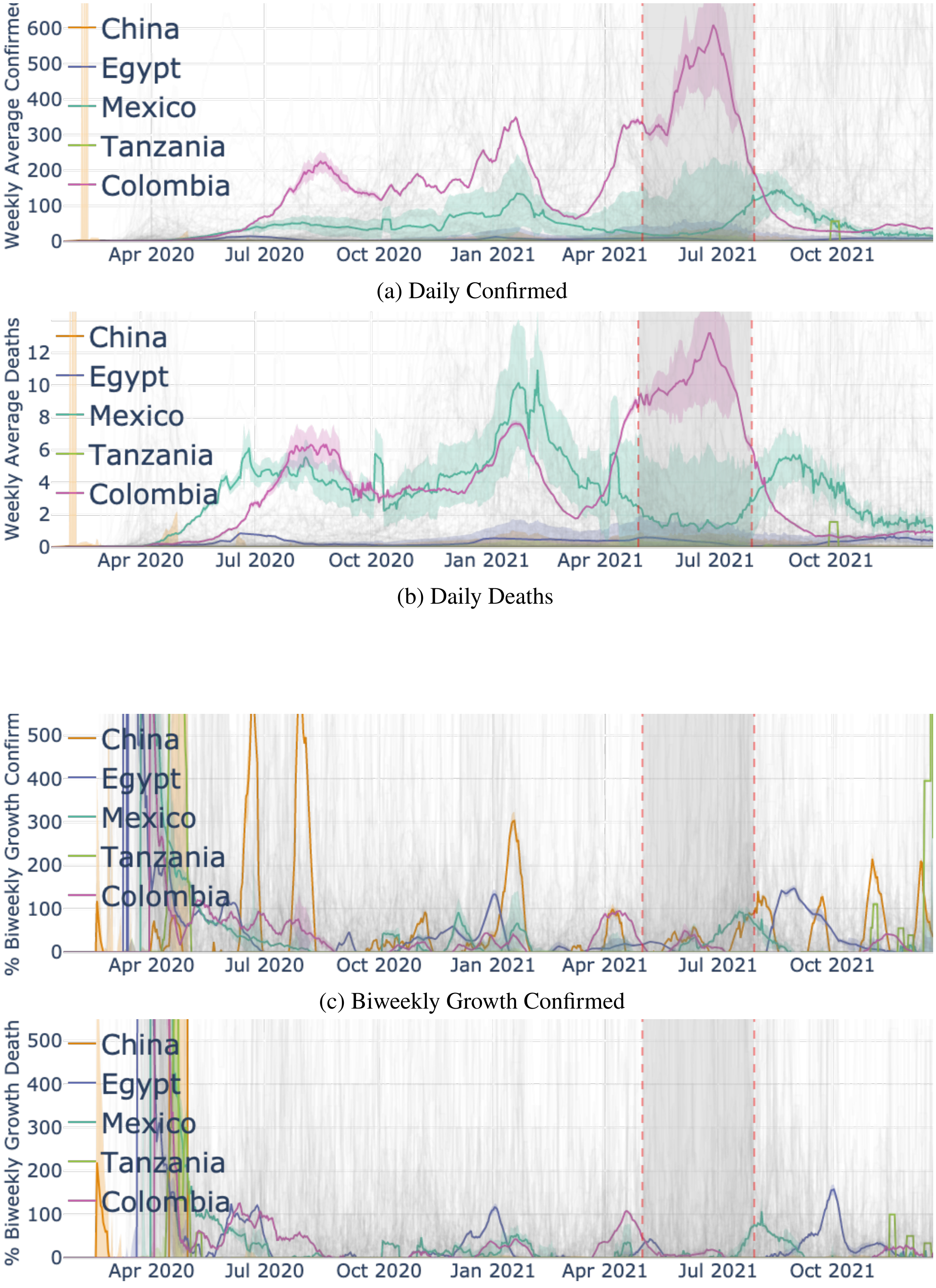
We start by considering the global trends post the incidence of the Delta variant. To rank the trends, we considered behaviors during the 90 day period between May 1 2021 - July 29 2021. It can be seen that China, Egypt, Mexico, Tanzania and Columbia were found most anomalous. In particular, China and Mexico had a very low per capita weekly average deaths and confirmed cases. However, the case fatality rate was consistently high [6] indicating that additional investigation is required to understand the root cause which can be underreporting or reporting issues or presence of a new variant.

Omicron Variant

To study the Omicron variant, we looked at the 90 day period data between September 23 2021 - December 21 2021. It can be seen that the countries with the most anomalous trends

[5] https://time.com/5812569/covid-19-new-york-morgues/

[6] https://www.reuters.com/business/healthcare-pharmaceuticals/china-reports-smallest-number-local-covid-19-cases-since-july-2021-08-13/, https://www.marketwatch.com/story/new-daily-covid-19-cases-and-deaths-spike-to-6-wee

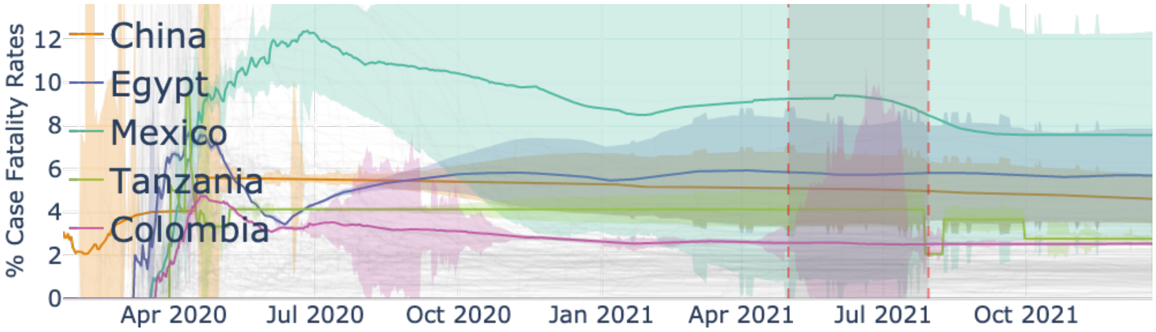


(a) Daily Confirmed

(b) Daily Deaths

(c) Biweekly Growth Confirmed

(d) Biweekly Growth Deaths



(e) Fatality Rates

Figure 8.9: Top 5 Countries with Anomalous Trends: Pandemic surge due to Delta variant. The grey shaded region indicates the time-period of interest used to identify the most anomalous trends.

include the UK and China. In particular, growth in cases in Egypt, UK and Russia has been significantly responsible for them being identified as anomalous [7]. However, in the case of Egypt and Russia, the surge in cases were not accounted to the Omicron variant but due to the COVID wave in their region that happens to coincide with the Omicron emergence [8].

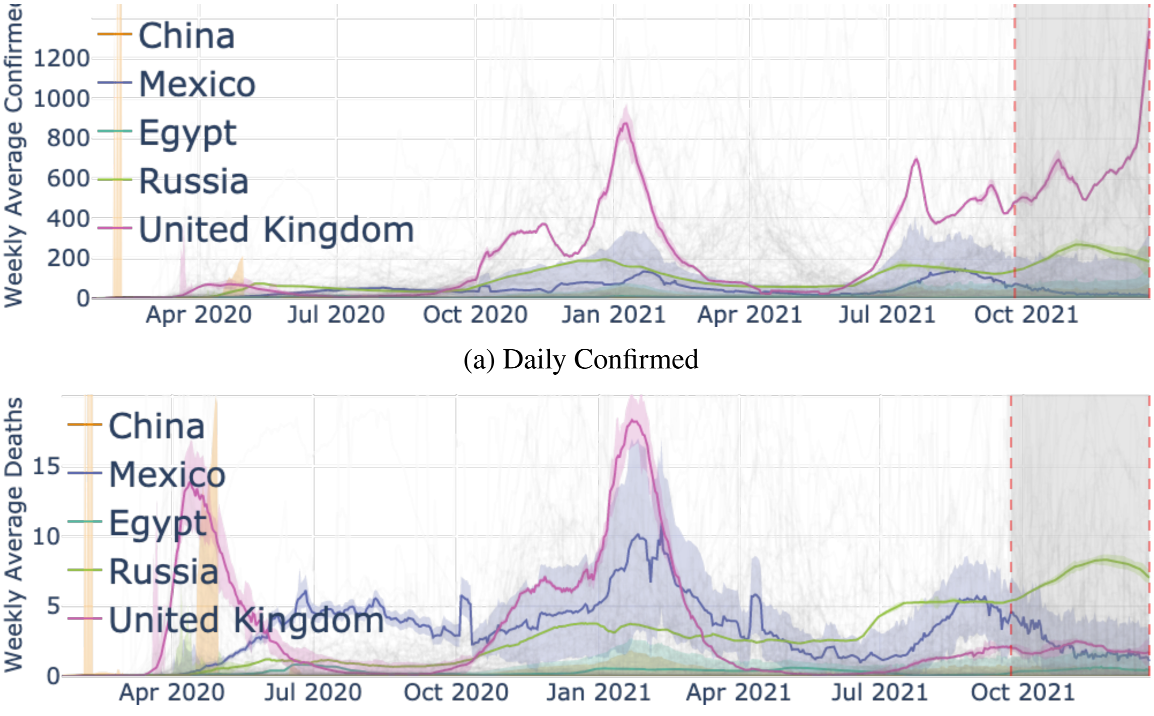
Vaccination Rates

To study trends in vaccination rates, we look at the total vaccinations, total boosters and people fully vaccinated (per hundred) in the 30 day period November 22 2021 - December 22 2021. We can see that China, South Korea, Italy, UK and Bangladesh were found most extreme due to the high vaccination rates found in these countries. In particular, the recent daily vaccination trend in Bangladesh has been relatively higher than the rest of the countries due to their 'no vaccine, no service' policy promoting the extreme trend[9].

[7] https://www.cnn.com/2021/12/13/uk/uk-omicron-infections-tidal-wave-gbr-intl/index.html

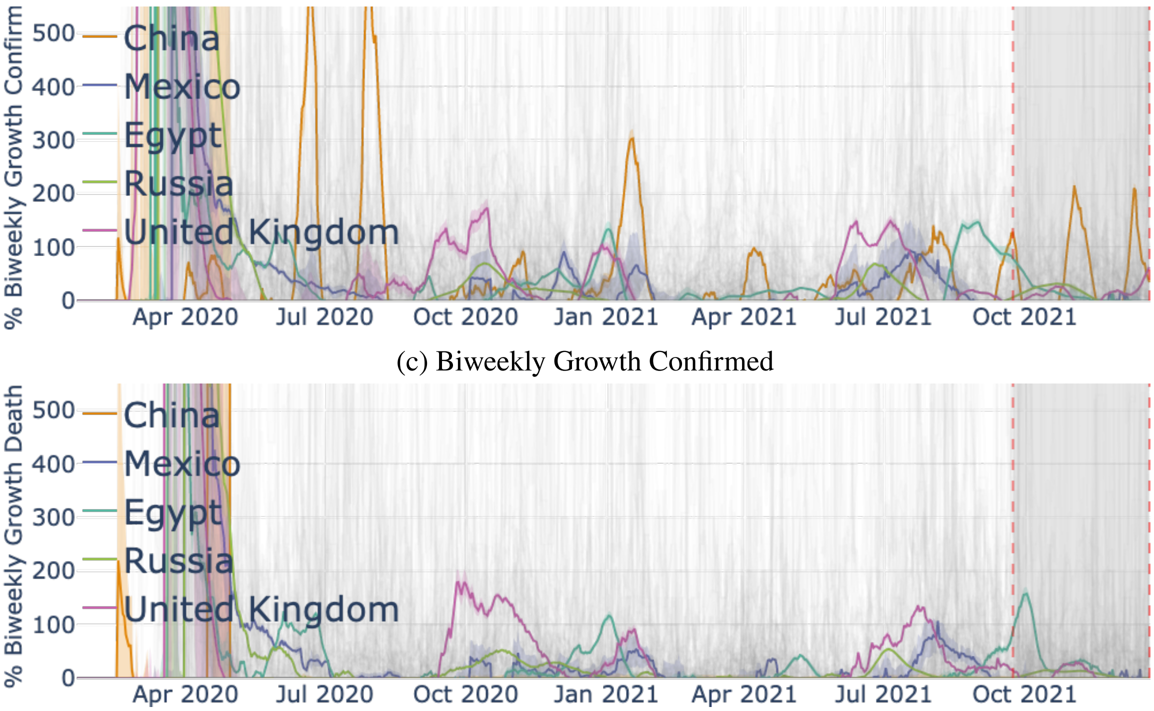
[8] https://www.egyptindependent.com/egypt-has-not-passed-the-peak-of-the-covid-19-fo
https://tass.com/society/1370957

[9] https://www.thedailystar.net/health/disease/coronavirus/news/no-vaccine-no-service-2906766



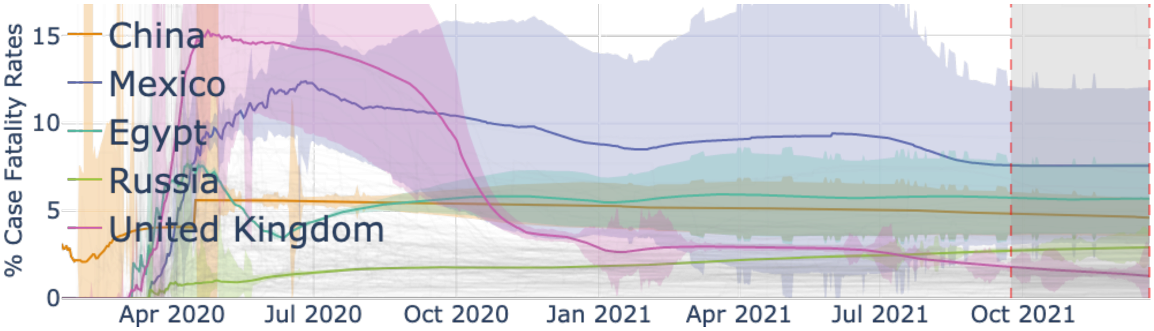
(a) Daily Confirmed

(b) Daily Deaths



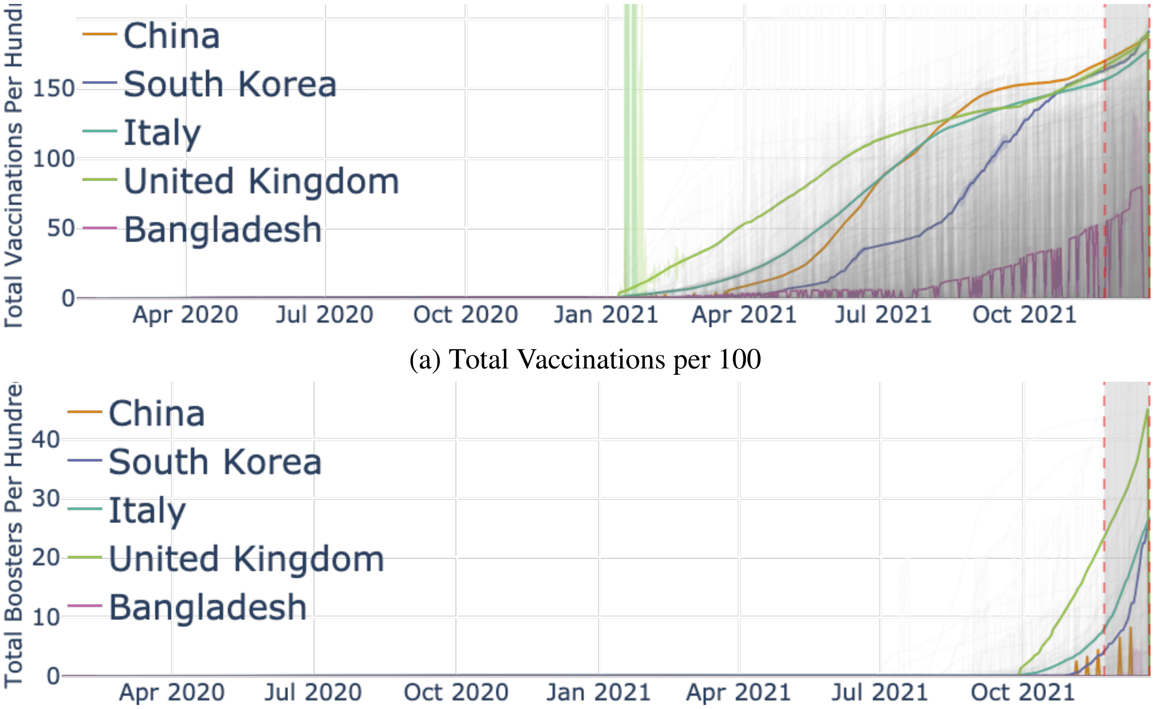
(c) Biweekly Growth Confirmed

(d) Biweekly Growth Deaths



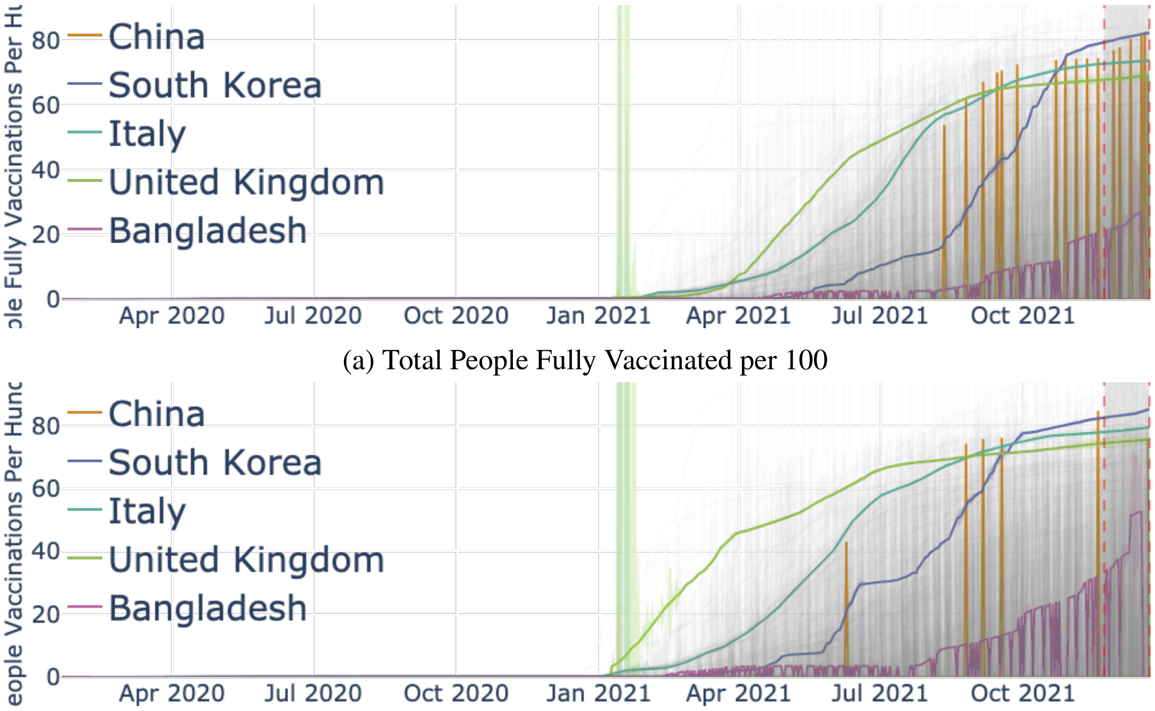
(e) Fatality

Figure 8.10: Top 5 Countries with Anomalous Trends: Pandemic surge during the Omicron variant. The grey shaded region indicates the time-period of interest used to identify the most anomalous trends.



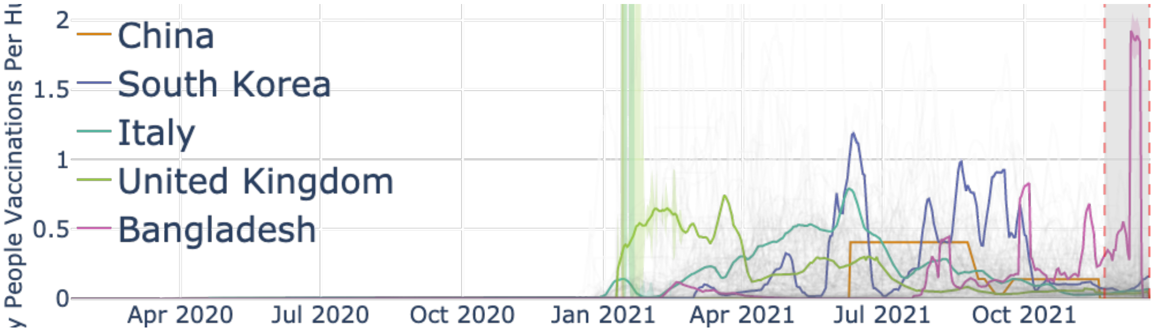
(a) Total Vaccinations per 100

(b) Total Boosters per 100



(a) Total People Fully Vaccinated per 100

(b) Total People Vaccinated per 100



(a) Daily People Vaccinated per 100

Figure 8.13: Top 5 Countries with Anomalous Trends in Vaccinations: Most extreme trends in vaccinations are illustrated in the figures. The grey shaded region indicates the time-period of interest used to identify the most anomalous trends.

8.2 Conclusion

In this part of the thesis, we propose LAD, a novel scoring algorithm for anomaly detection in large/high-dimensional data. The algorithm successfully handles high dimensions by implementing the large deviation theory. Our contributions include reestablishing the advantages of the large deviations theory to large and high dimensional datasets. We also present an online extension of the model that is aimed to identify anomalous time series in a multivariate time series data. The model shows vast potential in scalability and performance against baseline methods. The online LAD returns a temporally evolving score for each time series that allows us to study the deviations in trends relative to the complete time series database.

A potential extension to the model could include anomalous event detection for each individual time series. Another possible future work could be extending the model to enable anomaly detection in multi-modal datasets. Additionally, the online LAD model could be enhanced to use temporally weighted scores prioritizing recent events.

Part III

Large Deviations for Accelerating Neural Networks Training

Chapter 9

Neural Networks : Sensitivity to Training Samples

9.1 Introduction

Artificial Neural Networks (ANNs) are assumption free models that gather information from the provided training data. Due to their design, they are ideal to study complex functional dependencies between input and output layers. In contrast to traditional statistical models that use metrics like mean, covariance matrices, probability and confidence intervals, ANNs rely on patterns observed in training data for model development and fitting. Though this can be considered useful to develop better fitting architectures than their statistical counterparts, specially in datasets with complex data structures, the effect of incorrect or deficient training data is profound.

In this part of the thesis, we propose a statistically enhanced sampling of the training data in combination with a novel training method that ensures faster training of the neural network. The following are the contributions of this research:

1. We propose the LAD Improved Iterative Training (LIIT) strategy that uses a Modified Training Sample (MTS) to train the neural network.

2. We present 4 LAD score based sampling strategies to design the MTS. The LAD score is a large deviations based approach that is computationally inexpensive. Therefore, one can analyze large and high dimensional datasets without additional dimensionality reduction procedures thereby allowing more accurate and cost effective scoring schema.

3. The use of MTS which is a smaller training sample reduces the cost of computational time significantly for large datasets.

4. We perform an empirical study on publicly available classification benchmark datasets to analyze the performance of the proposed method.

The research is limited to simple classification based neural networks in this thesis. The future works include extending it to more complex ANNs.

9.2 Related Work

In this section, we provide a brief overview of sensitivity to training samples and speed of neural network training.

Artificial neural networks are powerful for general classification. However, its excellent performance often depends largely on a huge training set. A large body of research exists that study the impact of training data size on neural network learning [70, 25]. In particular, it is evident that smaller training data leads to less efficient models. However, the vast computational expense associated with training on large sets of data makes the need to improve training practices essential, specially for online or real-time models.

Many methods that try to model faster neural networks exist. For instance, Wang et al. (2019) use batch normalization in deep neural networks to improve the convergence rates. Zhong et al. (2017) work on image classification using their agile convolution neural network SatCNN, for quick and effective learning with small convolutional kernels and deep convolutional layers. However, these works are limited to the domain problems and cannot be easily scaled to other data types.

Another alternative to improve the training speed can be by modifying the training samples. For instance, studies like Shanker, Hu, and Hung (1996) look at the effect of standardization of data on the learning of the neural network. Kavzoglu (2009) emphasizes on characteristics of training samples and uses representative training to improve the classification. These methods, however, fail to study the impact of smaller data on model performance and efficiency.

In this part of the thesis, we propose a novel training strategy that can be generalized across domains. The method is used to replicate the true representation of the training features in a smaller sample which can be in turn used for faster training and convergence. Due to the proper representation of even the most extreme observations, this method ensures faster learning with competitive performance.

Chapter 10

LAD Improved Sequential Training for Neural Networks

10.1 Methodology

The most important aspect of classification models is the adequacy of the representative training samples for each class. Although the size of the training data is of considerable importance, acquiring a large number of representative training data may be impractical where a large number of classes are involved. In particular, since most observations within each true class have similar features, multiple samples add low value in terms of novel information/pattern. In this section, we describe the traditional batch training approach in brief followed by the LAD Improved Iterative Training approach. We present 4 sampling strategies used in the LIIT training and their respective algorithms.

10.1.1 Definitions and Terminology

Before describing the detailed methodology, we list out the terminology and corresponding definitions that are used for this study.

Definition 7. ***LAD Score*** *is the Large deviations Anomaly Detection (LAD) generated* *anomaly score for each observation in the data.*

Definition 8. ***Full-Training Data*** *is the available complete training dataset for the ANN. It must be noted that only a subset of the Full Training Data might be used to train the ANN in the LIIT approach. Hence we present a different terminology to differentiate it from the training data.*

Definition 9. ***Batch Training*** *is the traditional ANN training method using mini-batches of training data.*

Definition 10. ***Modified Training Sample (MTS)*** *is a smaller sample generated from the training data using a specific sampling algorithm.*

Definition 11. ***LAD Improved Iterative Training (LIIT)*** *is the novel improved batch training approach to train the ANN.*

10.1.2 Classification Neural Network

For this analysis, we look at the most basic classification algorithm. Figure 10.1 shows the architecture of the simple three layer dense neural network.

The model is trained using full training samples with the convergence criterion set to zero validation loss for 5 epochs with the maximum number of epochs is set to 180. Three different activation functions, RELU, Tanh, Softmax are used for the three consecutive dense layers respectively.

A simple model was chosen to study the proof of concept of the representative sampling strategy presented in the part of the thesis. Further studies are needed to understand the relation between the model choice and training sampling techniques.

10.1.3 LAD Improved Iterative Training of The Neural Network

Traditionally, in batch training, the full training data is divided into smaller samples or batches. The ANN learns from each batch sequentially till all the observations from the

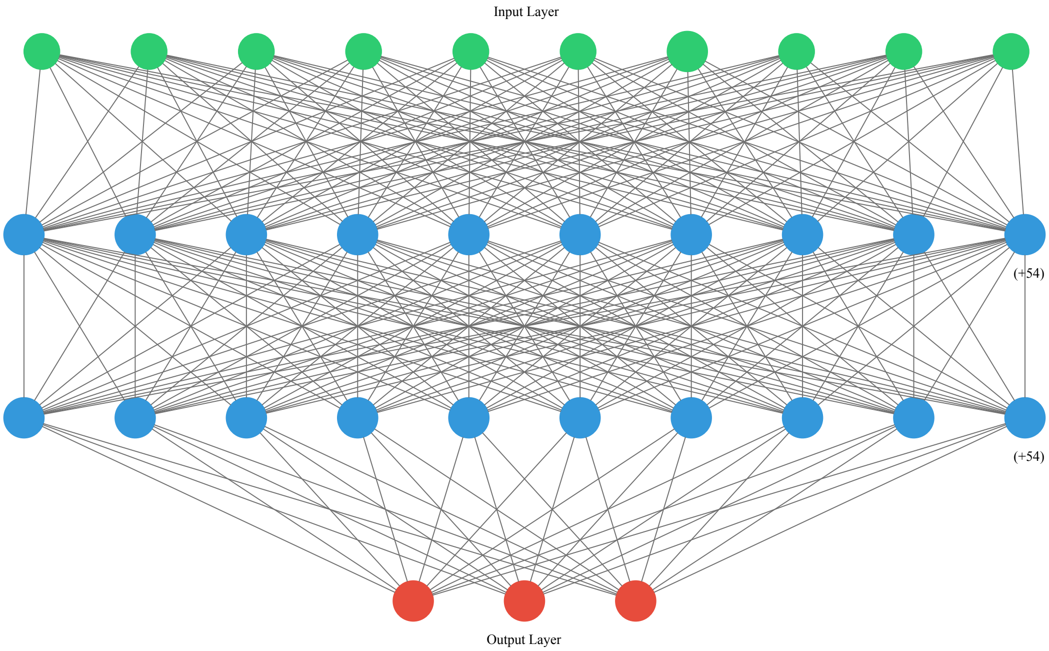


Figure 10.1: Simple Classification Neural Network: The figure illustrates a dense neural network to classify data into 3 classes. The network takes an input of 10 dimensions and returns scores for being assigned to each class.

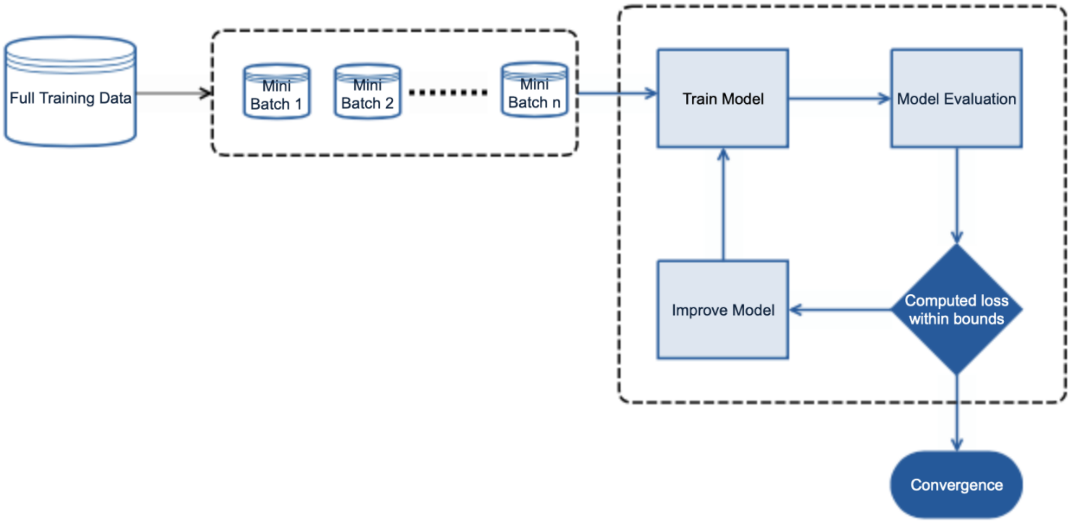


Figure 10.2: Mini-Batch Training Algorithm

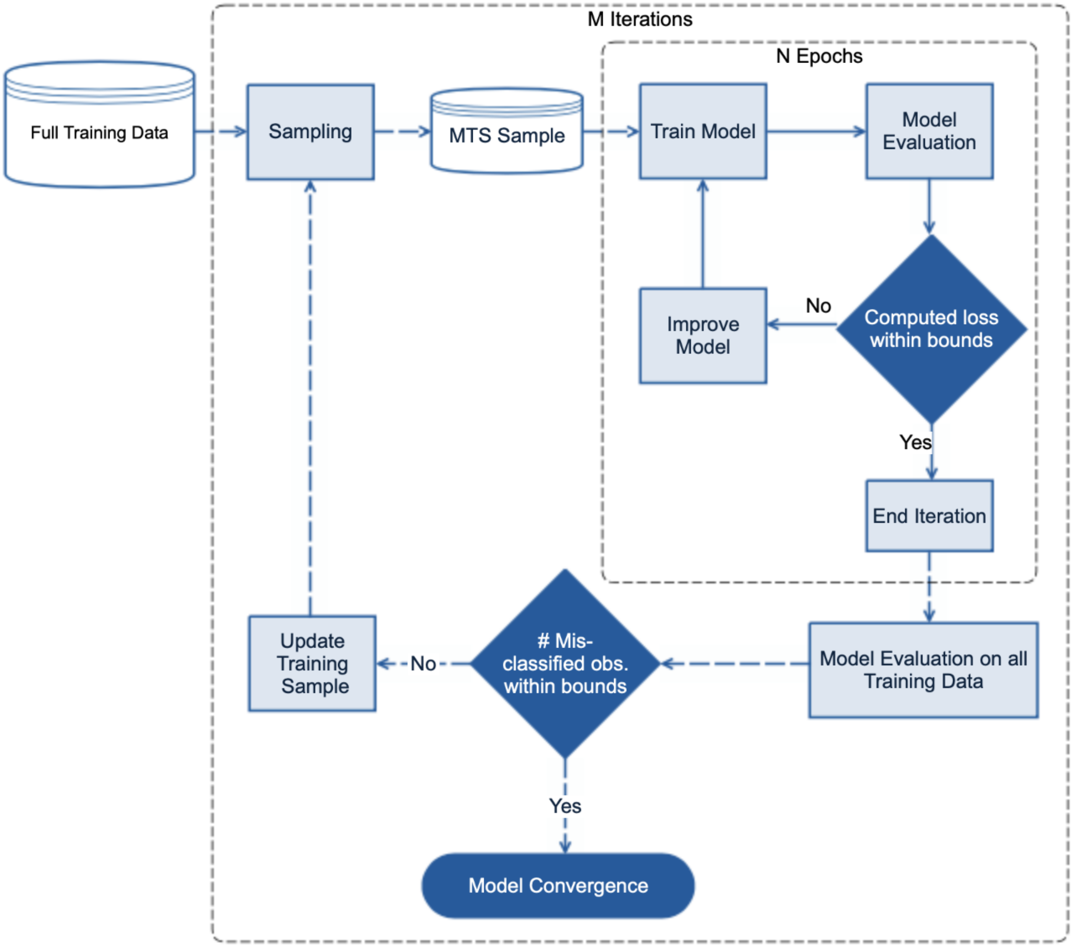


Figure 10.3: LIIT Training Algorithm

full training data are exhausted, as demonstrated in Figure 10.2. In the LIIT training, we iteratively design and update the modified training samples, MTS, from the full training data. At each iteration, we train the ANN using batch training on MTS till convergence. This partially trained model is then tested on the full training data to identify potential learning flaws. Since the current work is limited to classification models, the learning flaws include the misclassified data. The misclassified data is then used to derive the updated MTS which is used to retrain the ANN. The process is illustrated in Figure 10.3. This is inspired by Boosting techniques [66] where the subset creation depends on the previous model. However, unlike in boosting setting, we retrain the same ANN. [1]

To determine and extract the MTS sample, any sampling algorithm can be used. However, to ensure a just representation, we designed 4 LAD score based sampling algorithms along with the random sampling approach which is used as a baseline. The following are the sampling strategies used in our analysis:

1. **LAD Anomaly only (Repeated Entry)**: Observations with the highest anomaly scores in each true class are added to the training batch. Multiple copies of the observation can be added over iterations when the model fails to classify them after numerous re-training. See Algorithm 6.

2. **LAD Anomaly + Normal (Unique Entry)**: Equal parts of the high and low anomaly score observations are sampled for each true class. The final training batch contains a unique set of observations with no duplicate entries. See Algorithm 7.

3. **LAD Anomaly only (Unique Entry)**: This is similar to the **LAD Anomaly only (Repeated Entry)** approach. Observations with the highest anomaly scores in each true class are added to the training batch. However, the final training batch contains a unique set of observations with no duplicate entries. See Algorithm 8.

4. **LAD Quantile Samples (Repeated Entry)**: The observations are sampled using

[1] The LIIT approach is very similar to batch training within the epoch of a neural network.

different quantiles of the anomaly score for each true class. Multiple copies are maintained in the training batch to ensure weighting from under-represented latent classes within each known true class. See Algorithm 9.

5. **Random**: In this model, we use random sampling from the available data. See Algorithm 10.

For this part of the thesis, we sample $\sim 5-6\%$ of the full training data at each iteration that is later added to the modified training sample. We ensure equal weights for all true classes for the analysis. The LIIT approach is implemented with 6 iterations (1 initial and 5 updates) which brings to $\sim 30\%$ of the full training sample used in the LIIT approach.

Algorithm 6 LAD Anomaly only (Repeated Entry)

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th, number of true classes in the data K, sample size from each class c_{size}, number of iterations i_{iter}, ANN classification model $model_{liit}$.

Initialization:

Split data into $x_{train}, x_{test}, x_{val}, y_{train}, y_{test}, y_{val}$ (train, test and validation)

Derive LAD score ana_{score} for all observations in training data i.e.

$ana_{score} = LAD(x_{train}, y_{train})$

```
$MTS = []$ (create empty MTS sample indices list)
for each class $k$ do
    Generate list of indices of all observations in class $k$, $ind_k$
    Subset anomaly scores for each class

        $ana_{score_k} = ana_{score}[ind_k]$

    Identify top $c_{size}$ observations with least anomaly scores and add them samples to
  the $MTS$ sample i.e. (most non-anomalous observations)
    for each iteration $i \leq i_{iter}$ do
        Fit the ANN on $MTS$ using batch training,

            $model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$

        Predict model classification on $x_{train}$, $z_{pred} = model_{liit}.predict(x_{train})$
        Identify all miss-classified observations' indices in training data

            $err_{inds} = np.where(z_{pred}! = y_{train})$

        for each class $k$ do
            Identify all miss-classified observations $ind_{err_k}$
            Subset anomaly scores for miss-classified data in class $k$

                $ana_{err_k} = ana_{score}[ind_{err_k}]$

            Identify $c_{size}$ observations with highest anomaly scores from $ind_{err_k}$ i.e.
  (most anomalous observations) and add them to $MTS$ sample.
        end for
    end for
end for
```

Algorithm 7 LAD Anomaly + Normal (Unique Entry)

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th, number of true classes in the data K, sample size from each class c_{size}, number of iterations i_{iter}, ANN classification model $model_{liit}$.

Initialization:

Split data into $x_{train}, x_{test}, x_{val}, y_{train}, y_{test}, y_{val}$ (train, test and validation)

Derive LAD score ana_{score} for all observations in training data i.e.

$ana_{score} = LAD(x_{train}, y_{train})$

```
$MTS = []$ (create empty MTS sample indices list)
for each class $k$ do
    Generate list of indices of all observations in class $k$, $ind_k$
    Subset anomaly scores for each class
        $ana_{score_k} = ana_{score}[ind_k]$
    Identify top $c_{size}$ observations with least anomaly scores and add them samples to the $MTS$ sample i.e. (most non-anomalous observations)
    for each iteration $i \leq i_{iter}$ do
        Fit the ANN on $MTS$ using batch training,
            $model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$
        Predict model classification on $x_{train}$, $z_{pred} = model_{liit}.predict(x_{train})$
        Identify all miss-classified observations' indices in training data
            $err_{inds} = np.where(z_{pred}! = y_{train})$
        for each class $k$ do
            Identify all miss-classified observations $ind_{err_k}$
            Subset anomaly scores for miss-classified data in class $k$
                $ana_{err_k} = ana_{score}[ind_{err_k}]$
            Identify $c_{size}/2$ observations each for the lowest and highest anomaly scores from $ind_{err_k}$ i.e. (most anomalous as well as least anomalous observations) and add them to the $MTS$ sample indices.
        end for
        Remove repeated indices in the updated modified training sample,
            $MTS = unique(MTS)$
    end for
end for
```

Algorithm 8 LAD Anomaly only (Unique Entry)

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th, number of true classes in the data K, sample size from each class c_{size}, number of iterations i_{iter}, ANN classification model $model_{liit}$.

Initialization:

Split data into $x_{train}, x_{test}, x_{val}, y_{train}, y_{test}, y_{val}$ (train, test and validation)
Derive LAD score ana_{score} for all observations in training data i.e.
$ana_{score} = LAD(x_{train}, y_{train})$

```
$MTS = []$ (create empty MTS sample indices list)
for each class $k$ do
    Generate list of indices of all observations in class $k$, $ind_k$
    Subset anomaly scores for each class
        $ana_{score_k} = ana_{score}[ind_k]$
    Identify top $c_{size}$ observations with least anomaly scores and add them samples to the $MTS$ sample i.e. (most non-anomalous observations)
    for each iteration $i \leq i_{iter}$ do
        Fit the ANN on $MTS$ using batch training,
            $model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$
        Predict model classification on $x_{train}$, $z_{pred} = model_{liit}.predict(x_{train})$
        Identify all miss-classified observations' indices in training data
            $err_{inds} = np.where(z_{pred}! = y_{train})$
        for each class $k$ do
            Identify all miss-classified observations $ind_{err_k}$
            Subset anomaly scores for miss-classified data in class $k$
                $ana_{err_k} = ana_{score}[ind_{err_k}]$
            Identify $c_{size}$ observations with highest anomaly scores from $ind_{err_k}$ i.e. (most anomalous observations) and add them to $MTS$ sample.
        end for
        Remove repeated indices in the updated modified training sample,
            $MTS = unique(MTS)$
    end for
end for
```

Algorithm 9 LAD Quantile Samples (Repeated Entry)

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th, number of true classes in the data K, sample size from each class c_{size}, number of iterations i_{iter}, ANN classification model $model_{liit}$.

Initialization:

Split data into $x_{train}, x_{test}, x_{val}, y_{train}, y_{test}, y_{val}$ (train, test and validation)

Derive LAD score ana_{score} for all observations in training data i.e.

$ana_{score} = LAD(x_{train}, y_{train})$

```
$MTS = []$ (create empty MTS sample indices list)
for each class $k$ do
    Generate list of indices of all observations in class $k$, $ind_k$
    Subset anomaly scores for each class
        $$ana_{score_k} = ana_{score}[ind_k]$$
    Identify top $c_{size}$ observations with least anomaly scores and add them samples to the $MTS$ sample i.e. (most non-anomalous observations)
    for each iteration $i \leq i_{iter}$ do
        Fit the ANN on $MTS$ using batch training,
            $$model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$$
        Predict model classification on $x_{train}$, $z_{pred} = model_{liit}.predict(x_{train})$
        Identify all miss-classified observations' indices in training data
            $$err_{inds} = np.where(z_{pred}! = y_{train})$$
        for each class $k$ do
            Identify all miss-classified observations $ind_{err_k}$
            Subset anomaly scores for miss-classified data in class $k$
                $$ana_{err_k} = ana_{score}[ind_{err_k}]$$
            Identify observations corresponding to $c_{size}$ quantiles in $ana_{err_k}$ scores and add them to the $MTS$ sample indices.
        end for
    end for
end for
```

Algorithm 10 LAD Anomaly only (Repeated Entry)

Input: Dataset X of size (n, d), number of iterations N_{iter}, threshold th, number of true classes in the data K, sample size from each class c_{size}, number of iterations i_{iter}, ANN classification model $model_{liit}$.

Initialization:

Split data into $x_{train}, x_{test}, x_{val}, y_{train}, y_{test}, y_{val}$ (train, test and validation)
Derive LAD score ana_{score} for all observations in training data i.e.
$ana_{score} = LAD(x_{train}, y_{train})$

```
$MTS = []$ (create empty MTS sample indices list)
for each class $k$ do
    Generate list of indices of all observations in class $k$, $ind_k$
    Subset anomaly scores for each class
        $ana_{score_k} = ana_{score}[ind_k]$
    Randomly sample indices of $c_{size}$ observations and add them samples to the $MTS$ sample
    for each iteration $i \leq i_{iter}$ do
        Fit the ANN on $MTS$ using batch training,
            $model_{liit}.fit(x_{train}[MTS], y_{train}[MTS])$
        Predict model classification on $x_{train}$, $z_{pred} = model_{liit}.predict(x_{train})$
        Identify all miss-classified observations' indices in training data
            $err_{inds} = np.where(z_{pred}! = y_{train})$
        for each class $k$ do
            Identify all miss-classified observations $ind_{err_k}$
            Subset anomaly scores for miss-classified data in class $k$
                $ana_{err_k} = ana_{score}[ind_{err_k}]$
            Randomly sample indices of $c_{size}$ observations and add them samples to the $MTS$ sample
        end for
    end for
end for
```

Chapter 11

Neural Networks : Training and Stability to Perturbations

11.1 Experiments

In this section, we evaluate the classification performance of the simple neural networks on real data when trained using LAD sub-sampled data. We focus on the performance of the neural networks under different training and sampling settings.

The following experiments have been conducted to study the model:

1. Computational Expense: The LIIT trained ANN model's ability to train on a smaller set of training samples and converge faster is compared to the fully trained model.

2. Classification Performance: The overall performance of the sub-sampled models on multiple benchmark datasets is studied. For this analysis, we consider Area Under the Curve (AUC) as the performance metric to study classification.

3. Stability to Perturbations: Perturbations upto 8% are added to the test data which is used to study the change in performance in all models.

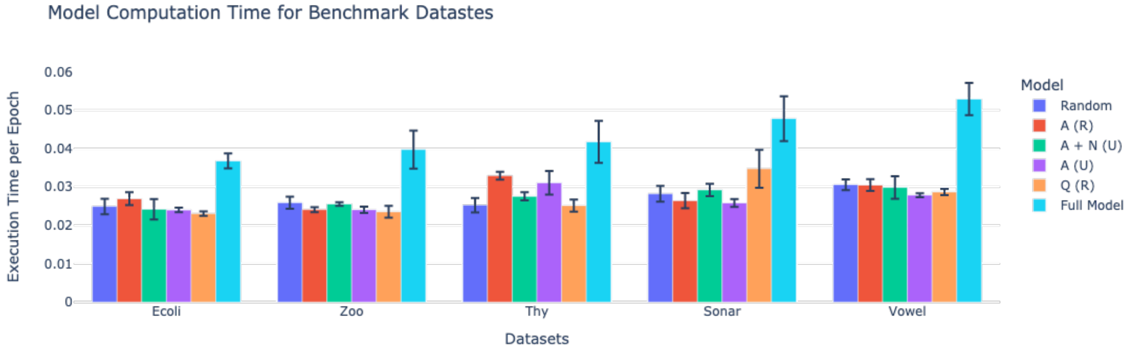
To maintain fair comparison, the number of epochs is fixed to a maximum count of 180 for the ANN model trained on the full training data a.k.a. the full model and 30 per iteration of all the LIIT trained ANNs (totaling to 180 epochs for complete training). For each trained ANN, we evaluate performance on 5 independent reruns. The average results are presented for all evaluations.

11.1.1 Datasets

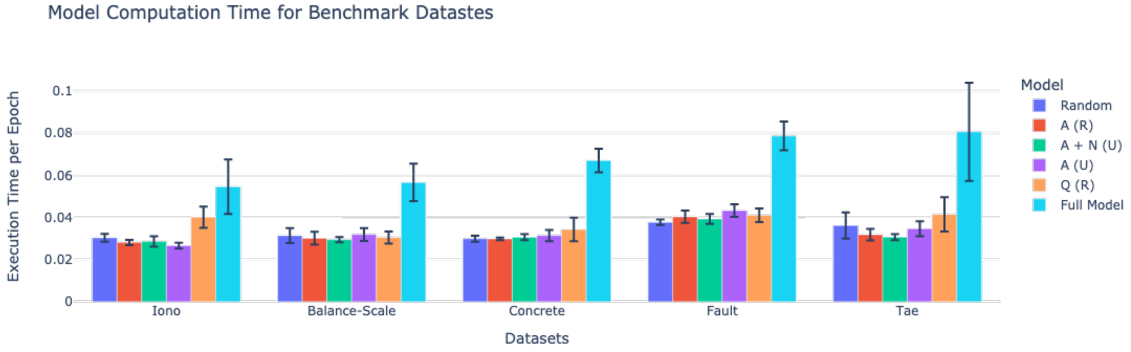
We consider a variety of publicly available benchmark data sets from the UCI-ML repository [24]) (See Table 11.1) for the experimental evaluation. For training, test and validation, the data was randomly split into 80%, 10% and 10% of the data respectively.

Name	N	d	c
Ecoli	336	7	8
Imgseg	2310	18	7
Skin	245057	4	2
Shuttle	58000	10	2
Wisc	699	9	2
Iono	351	33	2
Zoo	101	16	7
Letter	20000	16	26
Comm And Crime	1994	102	2
Vowel	990	10	11
Fault	1941	28	2
Sonar	208	60	2
Balance-Scale	625	4	3
Pageb	5473	11	2
Spambase	4601	58	2
Wave	5000	22	2
Tae	151	3	3
Thy	215	5	3
Opt Digits	5620	63	2
Concrete	1030	9	2

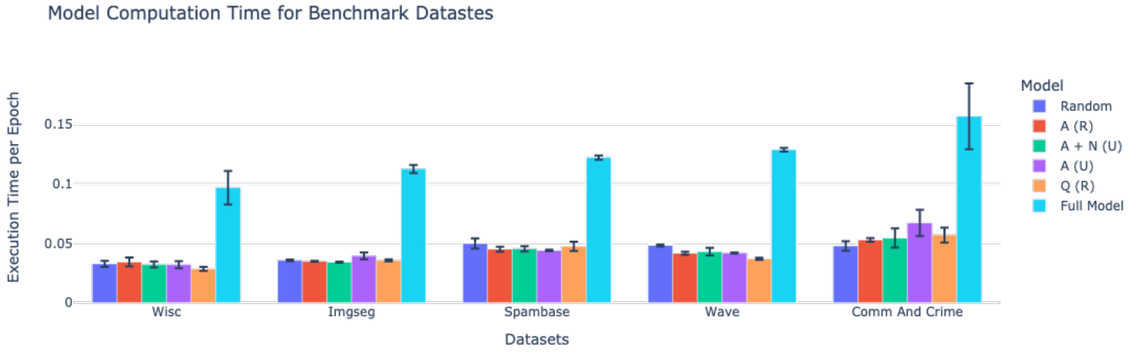
Table 11.1: classification Benchmark Datasets: Description of the benchmark data sets used for evaluation of the classification detection capabilities of the proposed model. N - number of instances, d - number of attributes, c - number of true classes in the data set.


Model Computation Time for Benchmark Datastes
Execution Time per Epoch
0.06
0.05
0.04
0.03
0.02
0.01
0
Ecoli
Zoo
Thy
Sonar
Vowel
Datasets
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

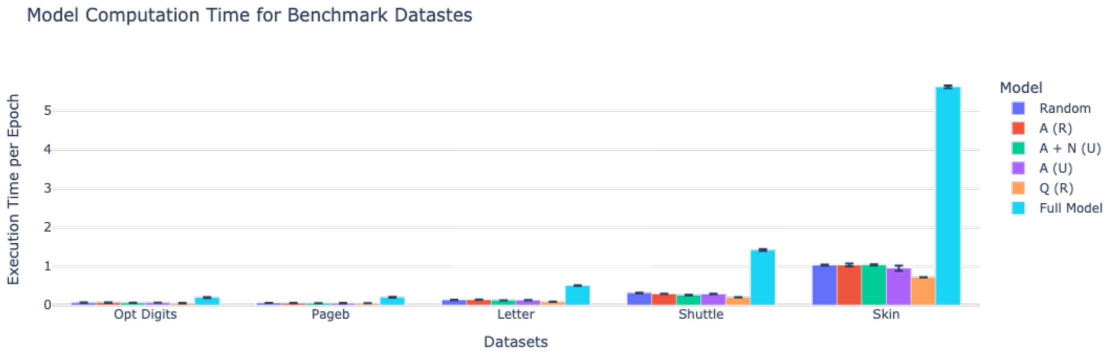
(a)


Model Computation Time for Benchmark Datastes
Execution Time per Epoch
0.1
0.08
0.06
0.04
0.02
0
Iono
Balance-Scale
Concrete
Fault
Tae
Datasets
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

(b)


Model Computation Time for Benchmark Datastes
Execution Time per Epoch
0.15
0.1
0.05
0
Wisc
Imgseg
Spambase
Wave
Comm And Crime
Datasets
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

(c)



(d)

Figure 11.1: Computation time for different datasets: The figures illustrate the computation time for different LIIT trained ANN models in comparison to the ANN trained on full training data (Full model).

Computational Time

In this section, we look at the time taken by each ANN to train on the datasets. Since the LIIT trained ANNs use only one-third of the full training data, the training time is evidently lower as compared to training for the full model. This can be clearly seen in the Figures 11.1.

Model Performance

Now, since the LIIT trained ANN models have a clear computational advantage over the full model, we look at the overall classification performance on a multitude of benchmark classification datasets. Table 11.2 shows the performance of the models on each of these datasets. We use the Area Under the Curve (AUC) as the evaluation metric to study the classification performance of the models. It is discernible that the Quantile Sampling along with LIIT trained ANN model is on par with the fully trained model.

Stability to Perturbations

Since the training samples have a significant influence on the model's learning and performance, we try to look at the stability of the model to various perturbations in the test data. For this, random noise is sampled from a multivariate normal distribution with the $0-8\%$ of the training data mean and variance and is added to all the observations in the test data. Each ANN's performance is evaluated in these settings for all benchmark datasets. The final classification performances are seen in Figures 11.2. It was interesting to note that different datasets had better and relatively more stable performances using different sampling strategies.

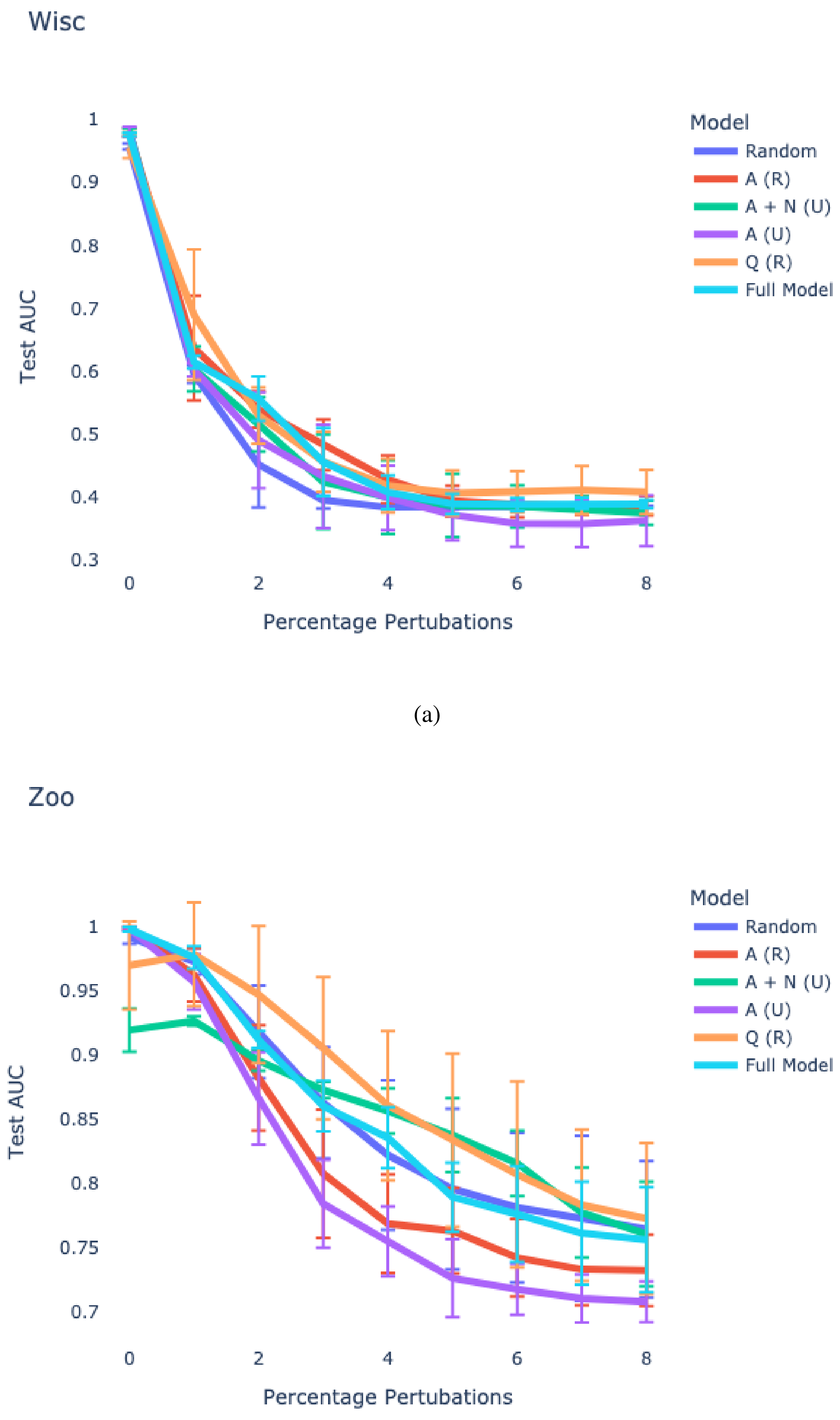
Now, to see the individual changes in performance to perturbations, we look at the raw change in AUC values due to the addition of perturbations for all models. Figures 11.3 show the change in performance for different datasets. In particular, Figures 11.3a and 11.3b show a group of datasets that show better performance using Quantile (Repeated), while Figures 11.3c - 11.3e show performance on datasets where Anomaly (Unique), Anomaly + Normal (Repeated) and Anomaly (Repeated) sampling approaches have respectively outperformed.

It can be seen that the Quantile Sample Trained Model has a higher mean AUC as well as lower deviation in AUC than the fully trained model in most datasets.

Here, we can see that different LIIT models outperform for different datasets. We hypothesize that the data distribution and heterogeneity play important role in the overall performance and stability. We intend to continue the study of the proposed hypothesis as future research.

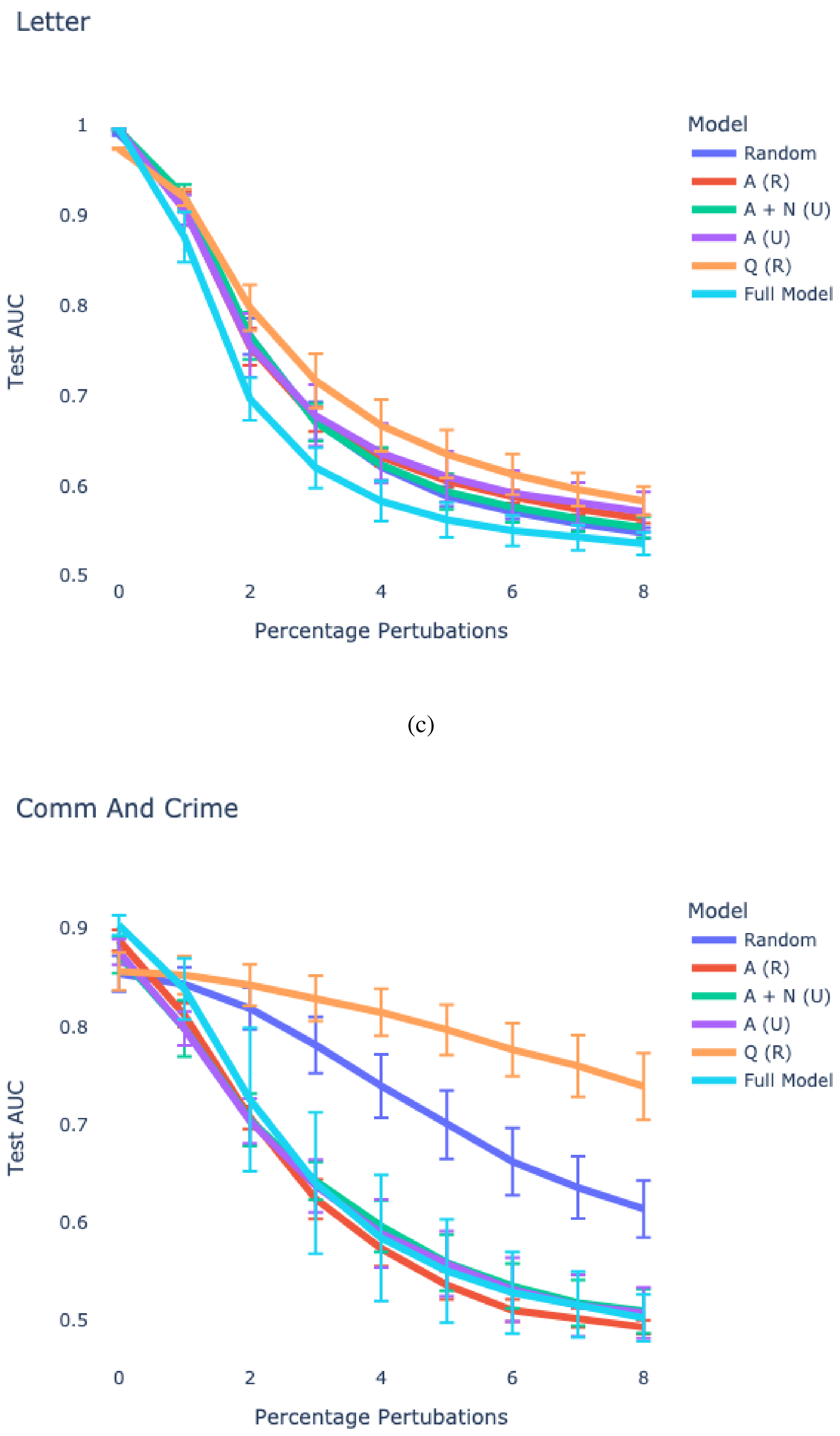
11.2 Conclusion

We present a new training strategy for enhancing the learning speed of a neural network whilst maintaining the performance of the model. We present the LAD Improved Iterative Training (LIIT) which is an improved iterative training version of the traditional batch


Wisc
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations
Zoo
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
Percentage Pertubations

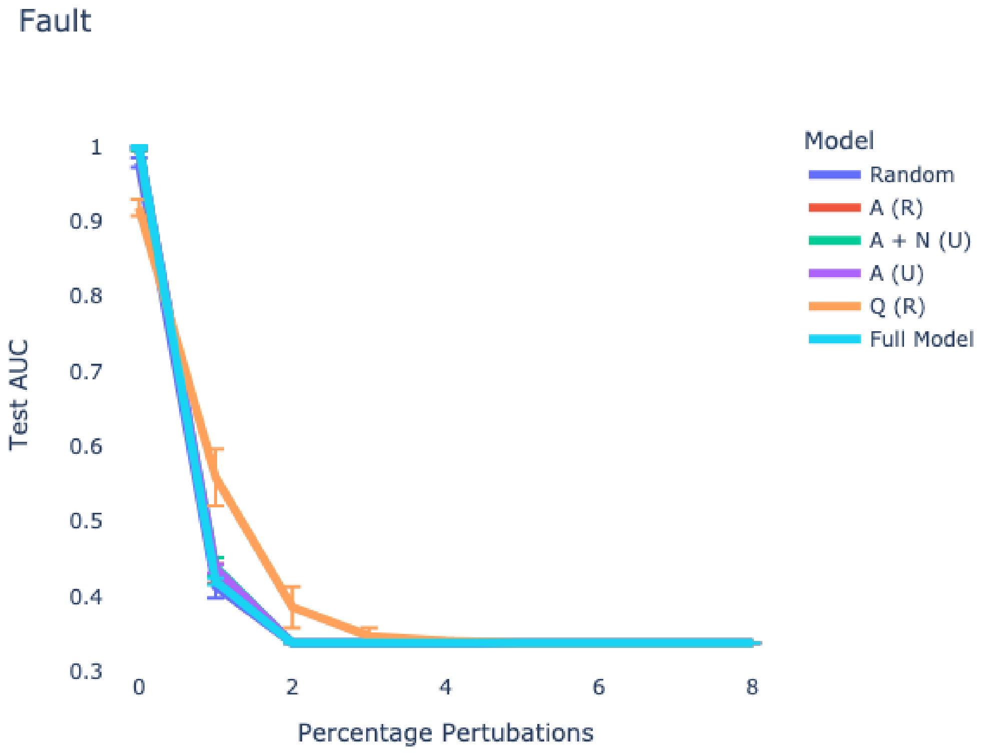
(a)

(b)

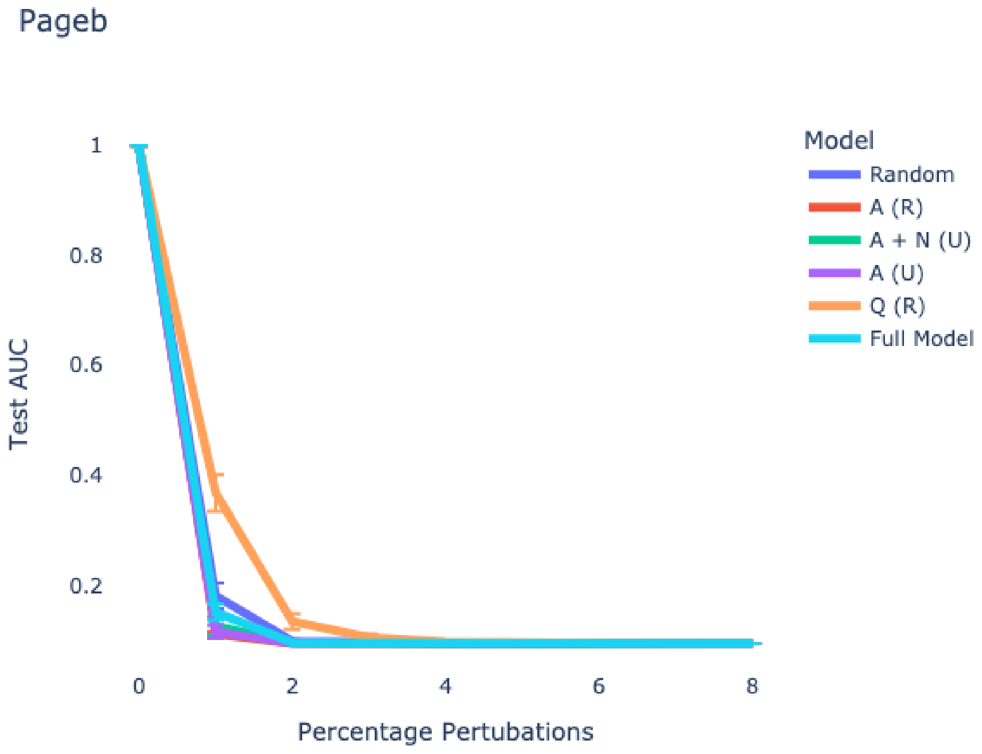

Letter
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Comm And Crime
Test AUC
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

(c)

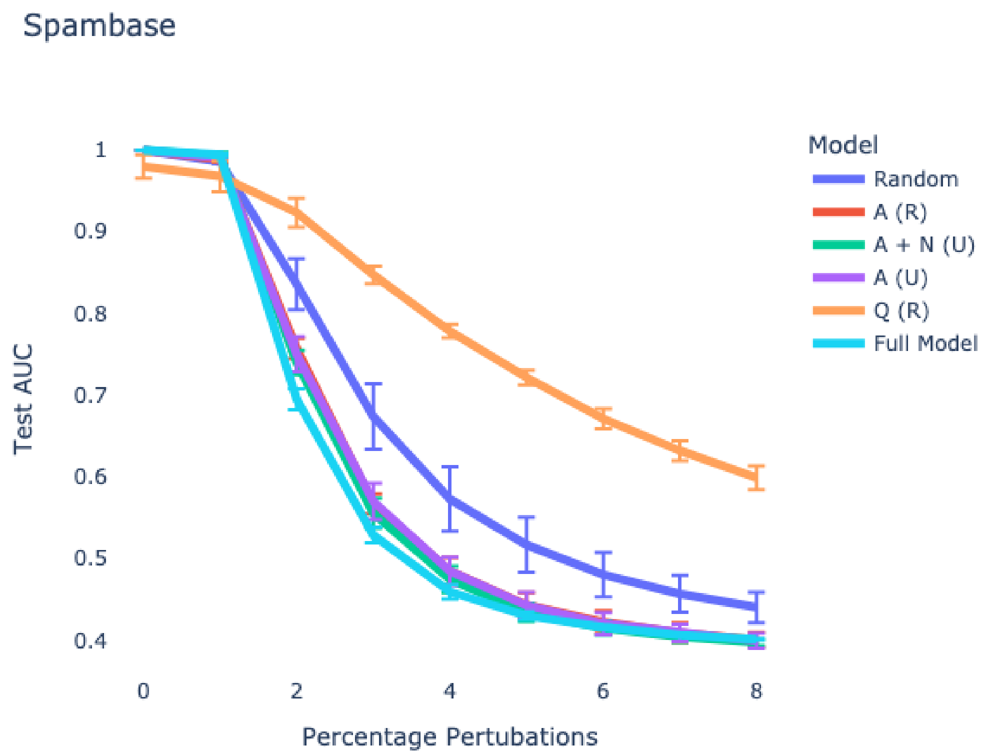
(d)


Fault
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations

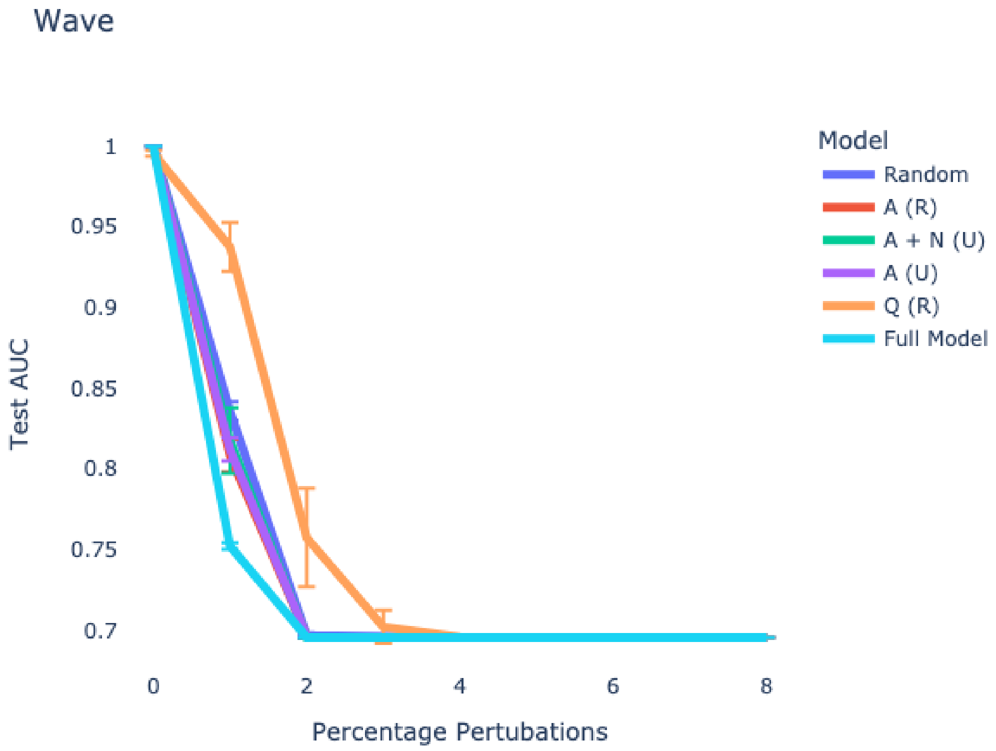
(e)


Pageb
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.8
0.6
0.4
0.2
0
2
4
6
8
Percentage Pertubations

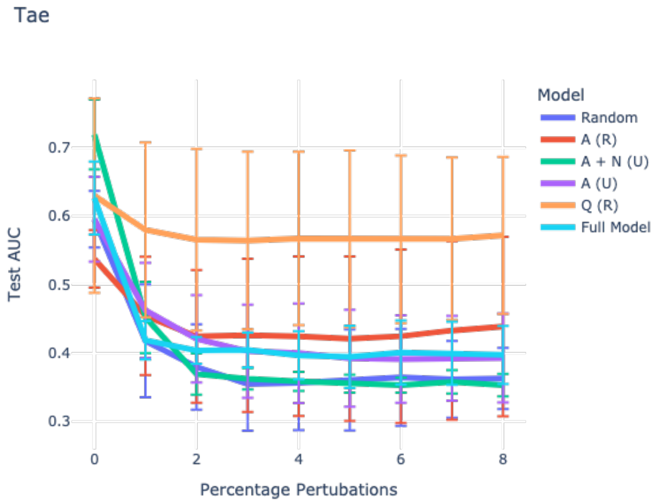
(f)


Spambase
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0
2
4
6
8
Percentage Pertubations

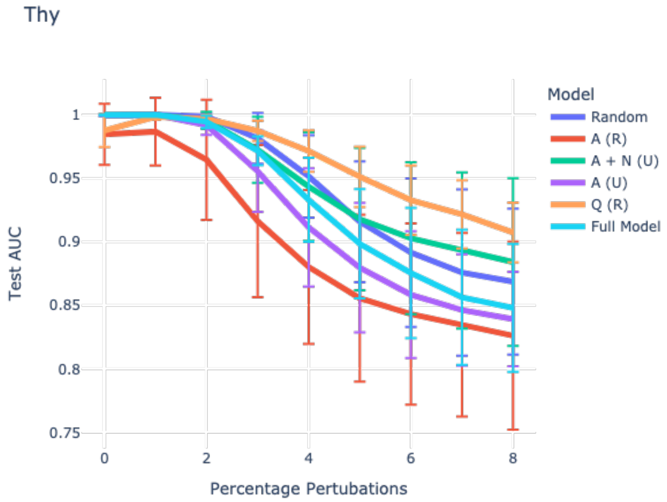
(g)


Wave
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
Percentage Pertubations

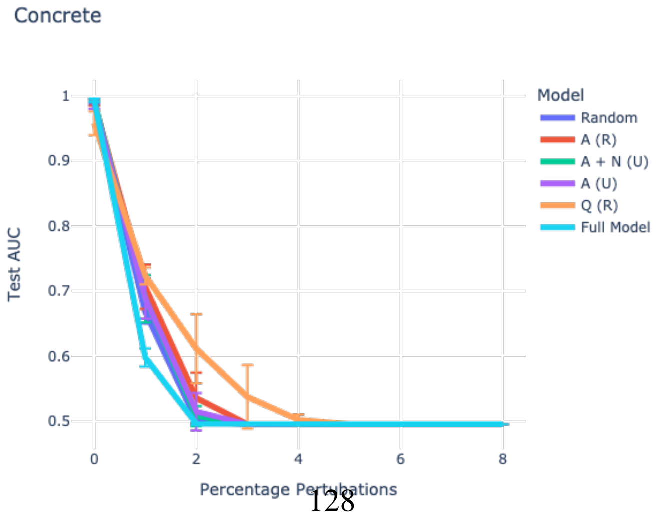
(h)


Tae
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations

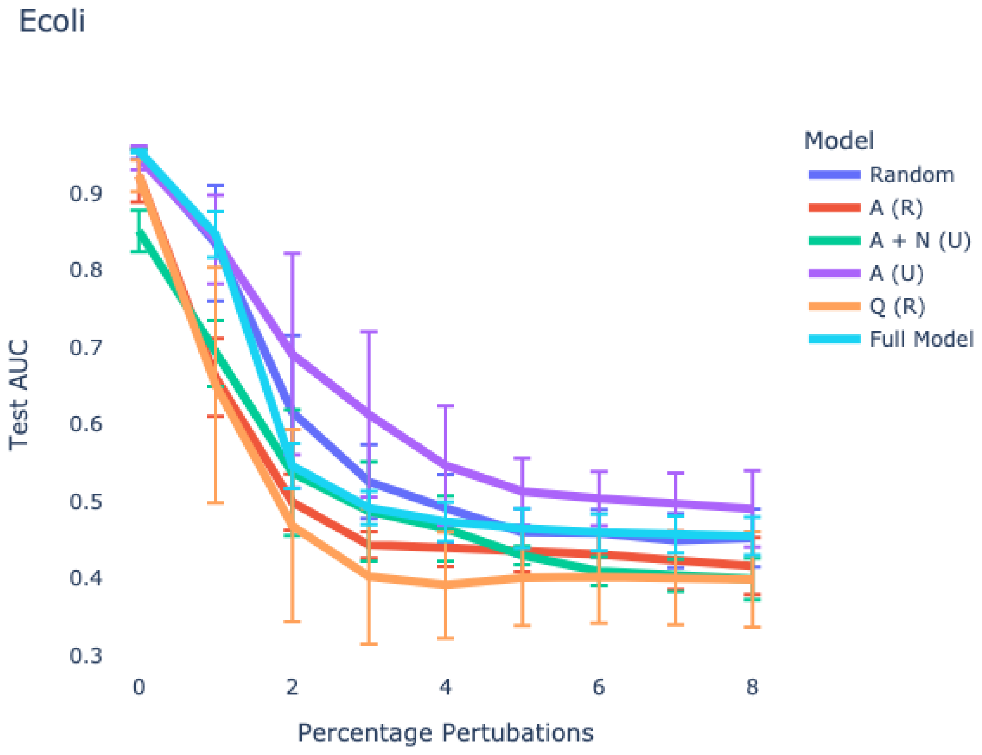
(i)


Thy
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0
2
4
6
8
Percentage Pertubations

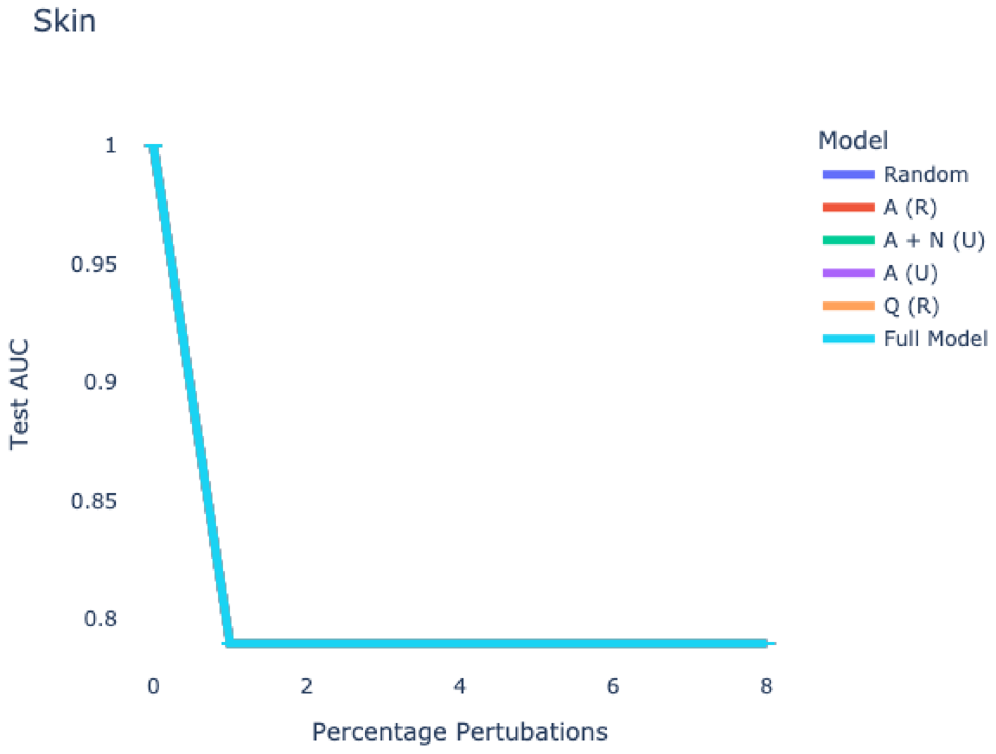
(j)


Concrete
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations

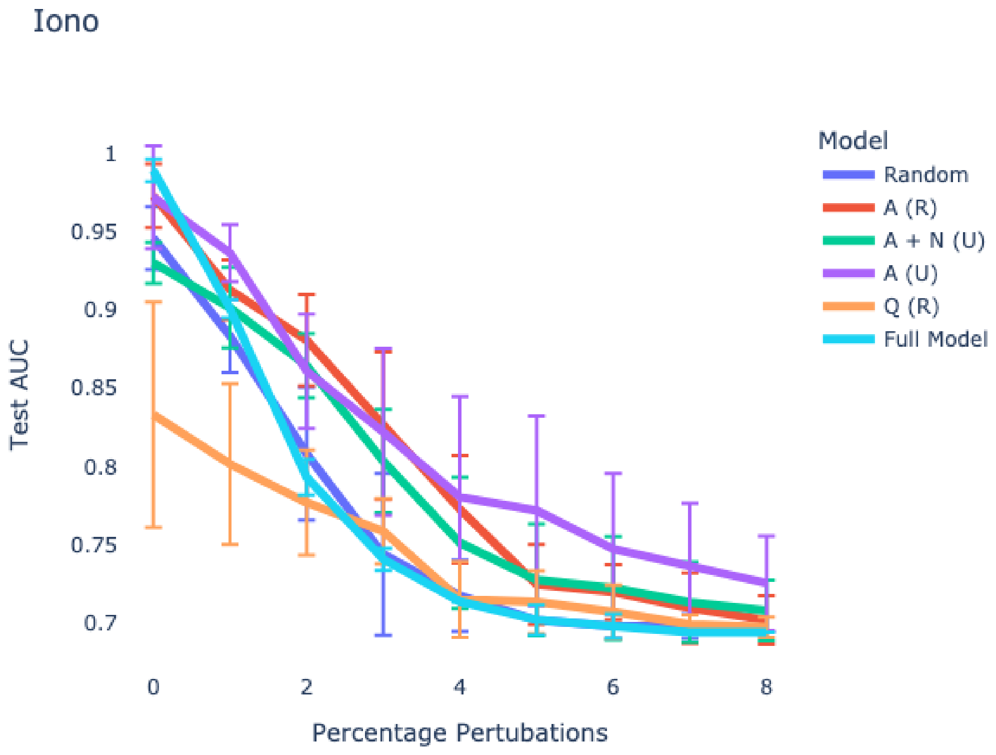
(k)


Ecoli
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
2
4
6
8
Percentage Pertubations

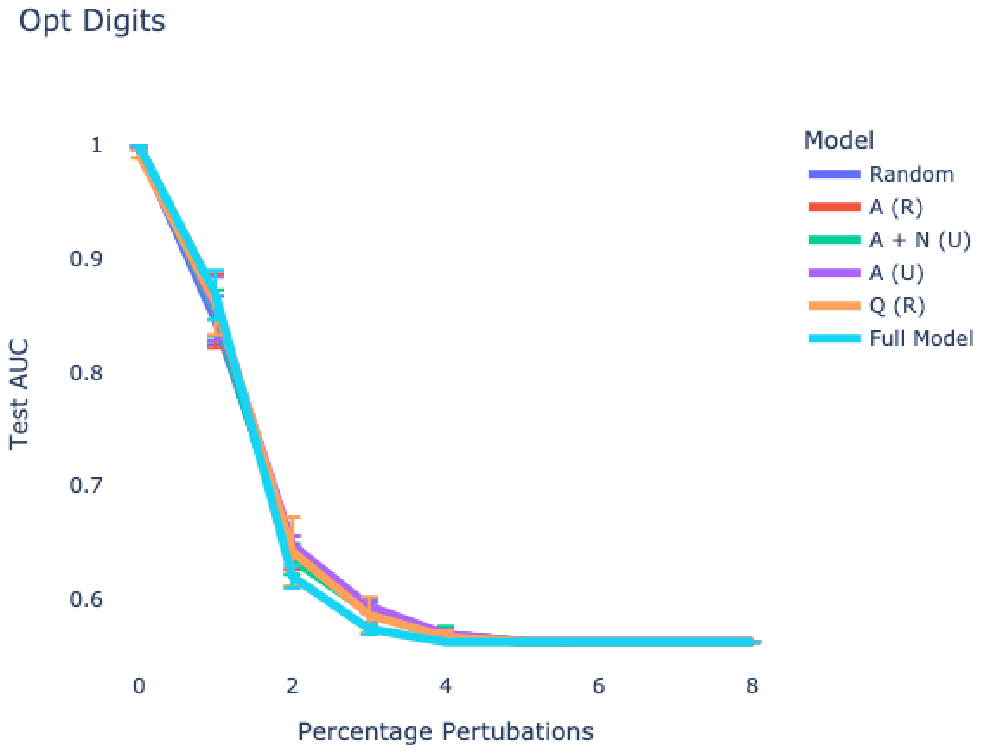
(l)


Skin
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0
2
4
6
8
Percentage Pertubations

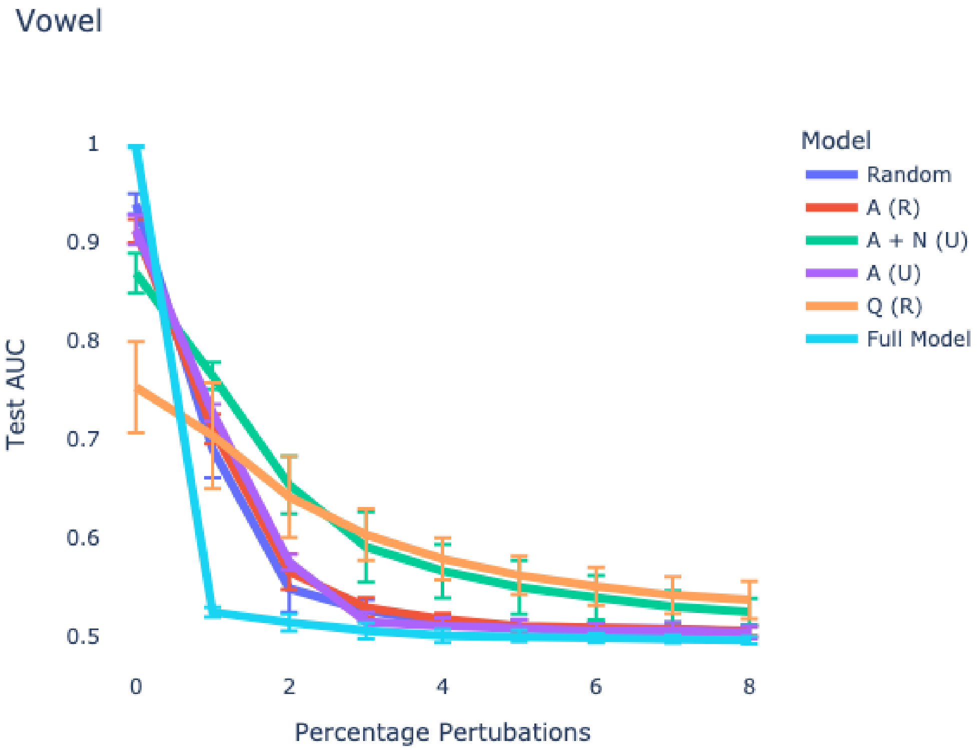
(m)


Iono
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.95
0.9
0.85
0.8
0.75
0.7
0
2
4
6
8
Percentage Pertubations

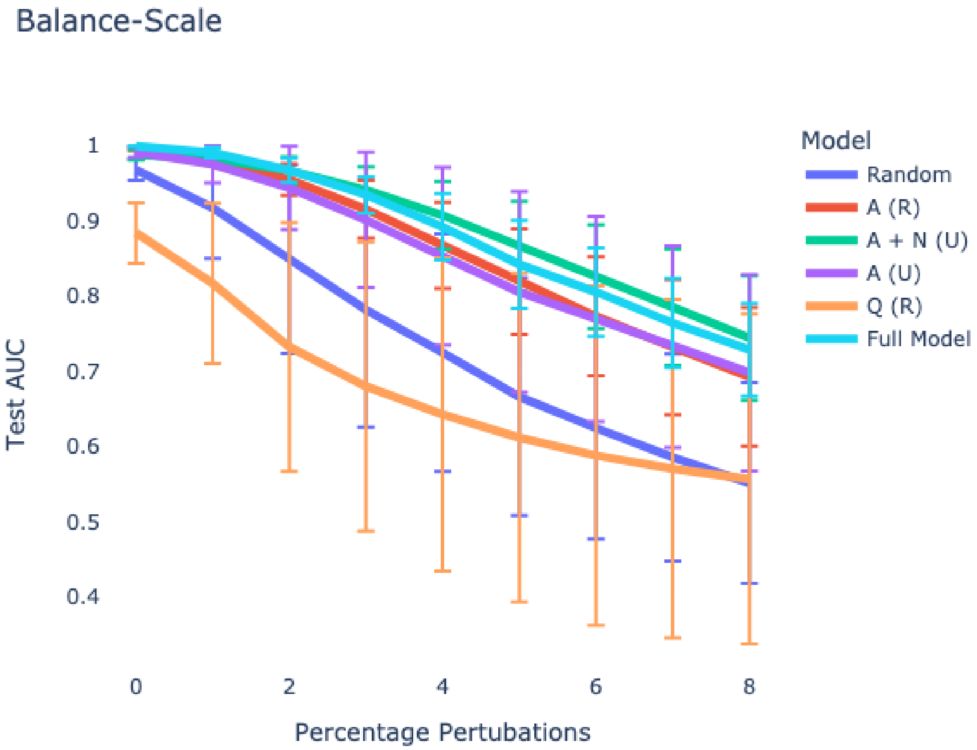
(n)


Opt Digits
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0
2
4
6
8
Percentage Pertubations

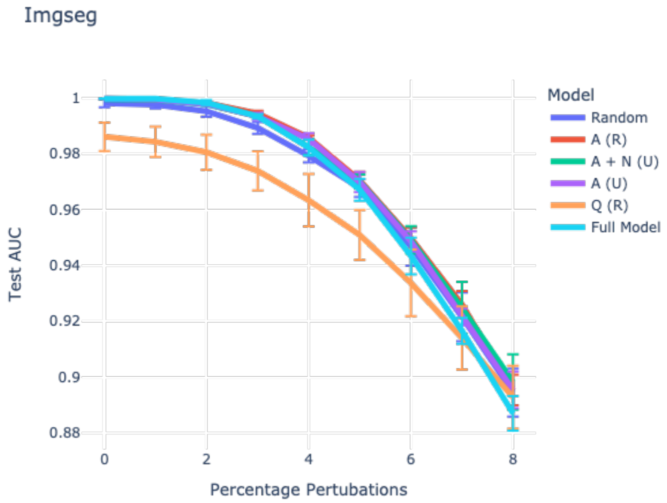
(o)


Vowel
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations

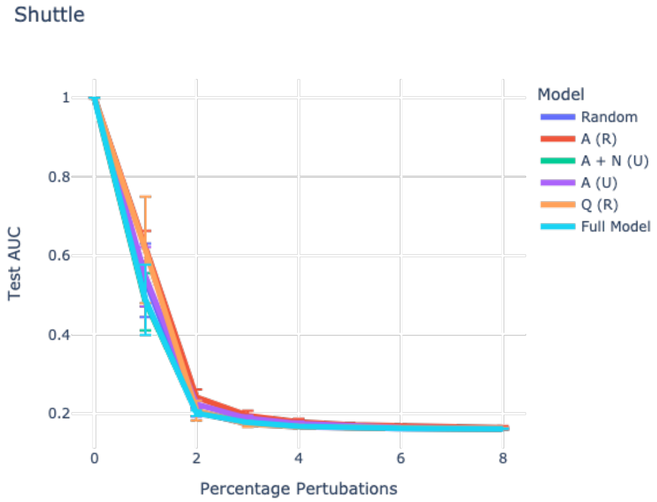
(p)


Balance-Scale
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model
Test AUC
1
0.9
0.8
0.7
0.6
0.5
0.4
0
2
4
6
8
Percentage Pertubations

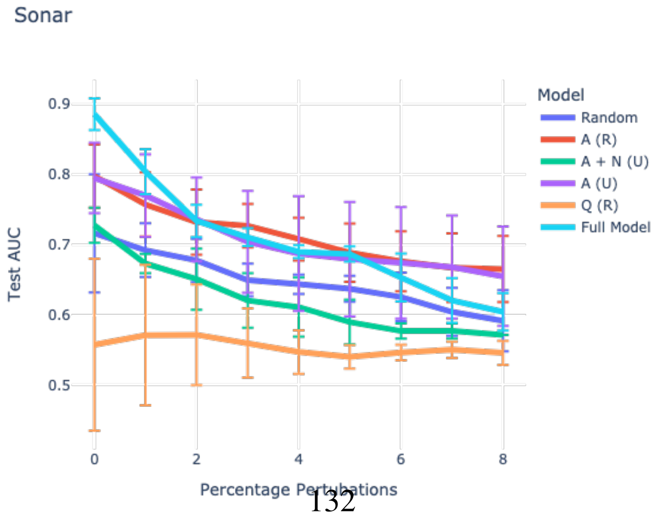
(q)


Imgseg
Test AUC
1
0.98
0.96
0.94
0.92
0.9
0.88
0
2
4
6
8
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

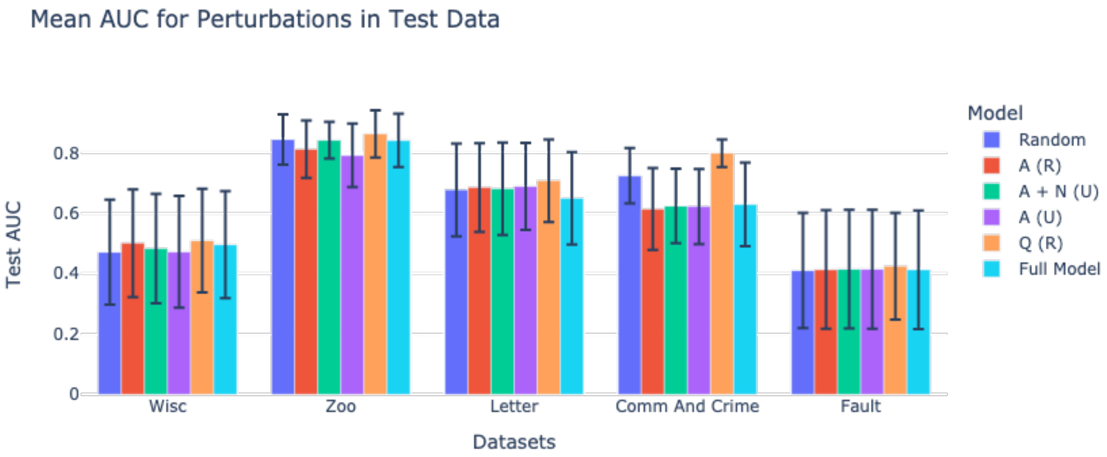
(r)


Shuttle
Test AUC
1
0.8
0.6
0.4
0.2
0
2
4
6
8
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

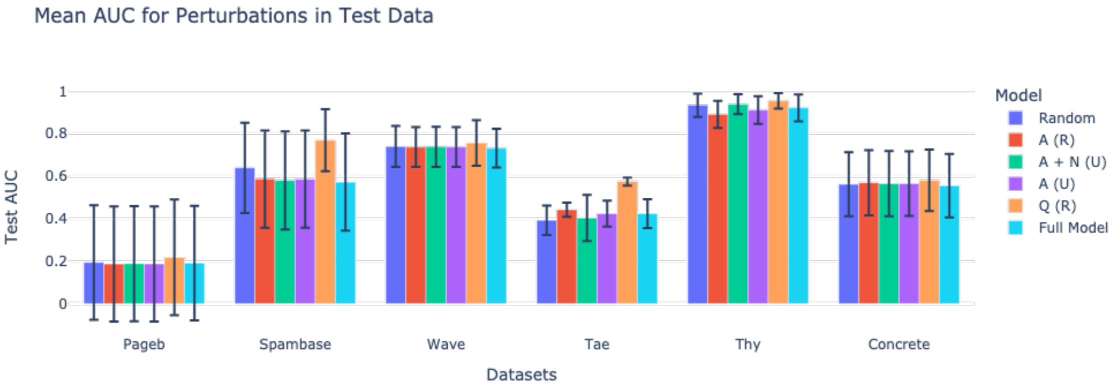
(s)


Sonar
Test AUC
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
Percentage Pertubations
Model
Random
A (R)
A + N (U)
A (U)
Q (R)
Full Model

(t)

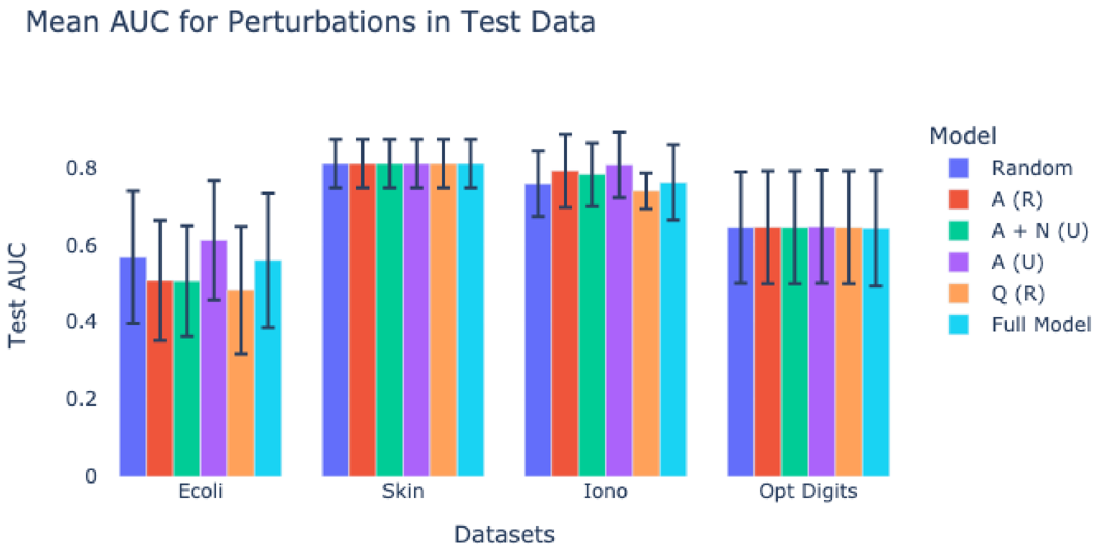


(a) Datasets where Quantile Sampling LIIT trained model shows the best performance



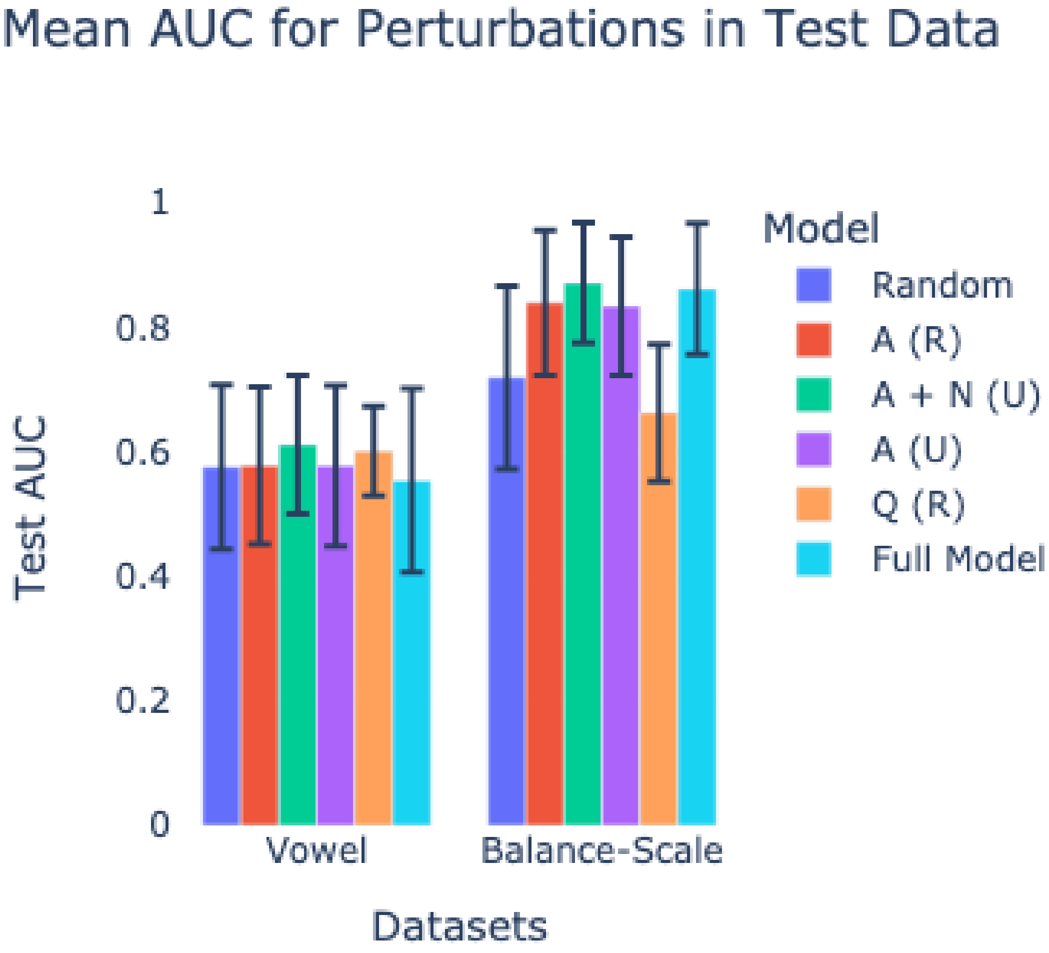
(b) Datasets where Quantile Sampling LIIT trained model shows the best performance

training approach. The LIIT approach uses a modified training sample (MTS) generated and updated using a LAD score based sampling approach that ensures enough representation of extreme and rare behaviours. In particular, the LAD score based Quantile Sampling approach allows ample heterogeneity within the sample data. We study the classification performance of the LIIT trained ANN in comparison with ANN trained on full training data on real benchmark datasets. Though the current research is limited to simple classification neural networks, the work has immense research potential. The LIIT training approach combined with specific LAD sampling methodology might draw out the best performance in a



(c) Datasets where Anomaly (Unique) LIIT trained model show best performance

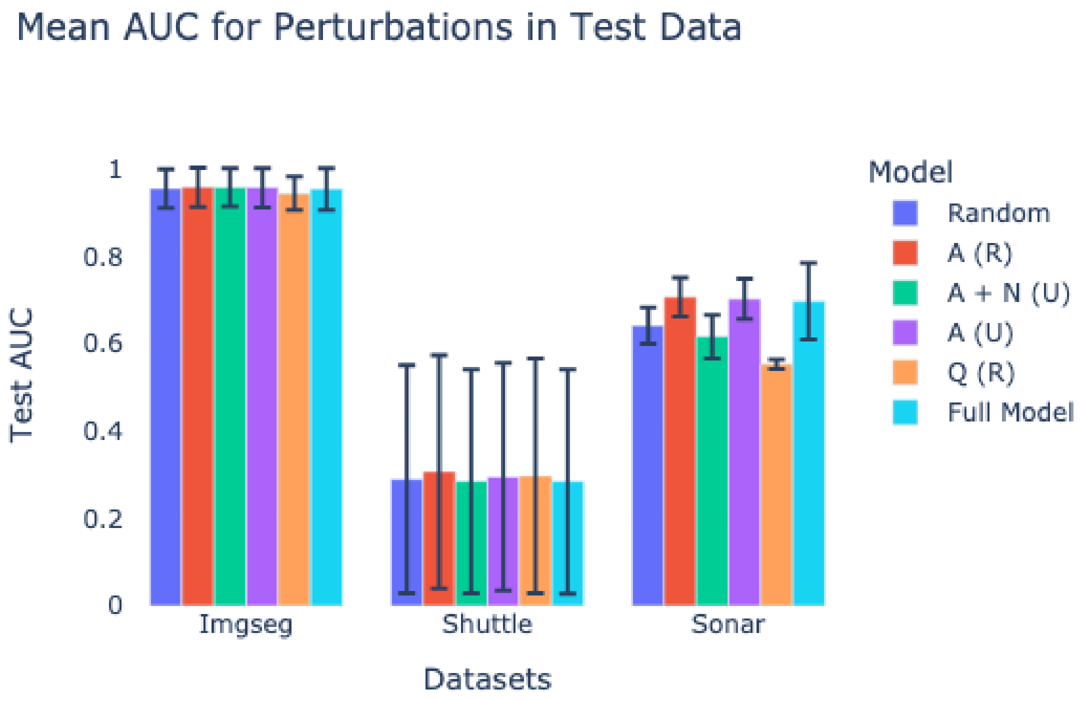
dataset based on the data characteristics. Future studies might help understand the impact of data heterogeneity and sampling method on the performance of ANN.



(d) Datasets where Anomaly + Normal (Repeated) LIIT trained model show best performance

Data	Random	Anomaly (Repeated)	Anomaly + Normal (Unique)	Anomaly (Unique)	Quantile Samples (Repeated)	Full Model
Tae	0.6 (± 0.04)	0.54 (± 0.04)	0.72 (± 0.05)	0.6 (± 0.06)	0.63 (± 0.14)	0.63 (± 0.05)
Spambase	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.98 (± 0.01)	1.0 (± 0.0)
Comm And Crime	0.85 (± 0.02)	0.89 (± 0.01)	0.87 (± 0.02)	0.88 (± 0.01)	0.86 (± 0.02)	0.9 (± 0.01)
Wisc	0.96 (± 0.0)	0.98 (± 0.01)	0.98 (± 0.0)	0.98 (± 0.01)	0.96 (± 0.02)	0.98 (± 0.0)
Letter	0.99 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.97 (± 0.0)	1.0 (± 0.0)
Vowel	0.94 (± 0.01)	0.91 (± 0.01)	0.87 (± 0.02)	0.91 (± 0.01)	0.75 (± 0.05)	1.0 (± 0.0)
Pageb	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Thy	1.0 (± 0.0)	0.98 (± 0.02)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.01)	1.0 (± 0.0)
Zoo	0.99 (± 0.01)	1.0 (± 0.0)	0.92 (± 0.02)	1.0 (± 0.0)	0.97 (± 0.03)	1.0 (± 0.0)
Concrete	0.99 (± 0.0)	0.99 (± 0.01)	0.99 (± 0.0)	0.99 (± 0.01)	0.96 (± 0.02)	0.99 (± 0.0)
Wave	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Fault	0.98 (± 0.01)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.92 (± 0.01)	1.0 (± 0.0)
Shuttle	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Opt Digits	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.0)	1.0 (± 0.0)
Skin	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)
Imgseg	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	1.0 (± 0.0)	0.99 (± 0.01)	1.0 (± 0.0)
Iono	0.95 (± 0.02)	0.97 (± 0.02)	0.93 (± 0.01)	0.97 (± 0.03)	0.83 (± 0.07)	0.99 (± 0.01)
Balance-Scale	0.97 (± 0.01)	1.0 (± 0.0)	0.99 (± 0.01)	0.99 (± 0.01)	0.88 (± 0.04)	1.0 (± 0.0)
Sonar	0.72 (± 0.08)	0.8 (± 0.05)	0.73 (± 0.03)	0.8 (± 0.05)	0.56 (± 0.12)	0.89 (± 0.02)
Ecoli	0.95 (± 0.01)	0.92 (± 0.03)	0.85 (± 0.03)	0.95 (± 0.02)	0.92 (± 0.02)	0.95 (± 0.0)

Table 11.2: Model Performance on Test Data: The mean AUC for each model on different datasets is shown here. The standard deviation in AUC on different re-runs is shown here.



(e) Datasets where Anomaly (Repeated) LIIT trained model shows the best performance

Figure 11.3: Total change in AUC over % perturbations added to the data: The figure illustrates the mean change in AUC with the increase in % perturbations to different datasets. The error-bars indicate the standard deviations in AUC values across perturbations.

Chapter 12

Conclusion

The existing research on anomaly detection for evolving data is mostly limited to studying anomalies within individual time series. With the explosive growth of temporal data and collections of time series data over the last decade, the need to develop methods to study evolving data has been emphasized. This thesis presents two statistically sound approaches to examine anomalies in evolving settings.

The thesis starts by examining the problem of studying multi-modal evolving data in streaming settings. We present the INCAD model that performs simultaneous clustering and anomaly detection for streaming datasets. The model is an amalgamation of the Bayesian non-parametric model and extreme value theory. By using the Chinese Restaurant process as a prior, the model can capture new clusters evolving in the streaming without additional interventions. Being data driven and completely unsupervised, the model is sensitive to changes in data distribution. Our study reveals that the model has exceptional performance in capturing anomalies in streams as well as reclassifying labels based on popular behaviors. However, due to the high complexity associated with the INCAD model, applicability to large datasets is limited. So, we propose the LAD algorithm.

The LAD approach is a large deviations based anomaly detection algorithm that aims to capture anomalies in a wide variety of data types. The model uses the rate function to derive a probabilistic anomaly score for observations. Due to the ease and scalability of the large deviations rate functions, the LAD model is particularly fast and efficient. This approach can be easily extended to studying individual time series as well as a collection of time series. We present extensive results for this approach in many real benchmark datasets. In particular, a significant contribution of the thesis includes studying multivariate time series databases for COVID-19 pandemic data to identify geographical locations with extreme trends. The research presents a potential channel to study various geographical streams as a collective set.

Despite its vast versatility, the LAD model is still limited to unimodal data. Extension to multi-modal data is achievable provided true cluster centers are known for the analysis. Further future work is needed to extend an unsupervised clustering component to the LAD model.

Additionally, due to its computational ease and scalability to diverse data types, another application of the LAD model is to enhance training methodology for neural networks.

Since artificial neural networks (ANNs) are sensitive to the size of training data, a generous number of samples are required to train an ANN. This demands considerable computational time and resources. We conclude the thesis by providing a preliminary report on a novel training approach for ANN. The LAD Improved Iterative Training (LIIT) is an improvised batch training algorithm that ensures a fast and efficient training of ANN. This approach uses a modified training sample (MTS) derived and updated from the training data to train the neural network. Multiple LAD score based sampling techniques have been listed that are used to generate the MTS samples. This thesis presents the performance results of the multiple LIIT trained ANNs in comparison to the ANN trained on full training data.

We conclude the thesis with a hypothesis that the data heterogeneity plays important role in the performance of LIIT trained mode. In particular, each LAD score based sampling algorithm is best suited with a specific data characteristics. However, detailed studies are recommended to derive a conclusive understanding of the relationship between sampling methods and data characteristics. This future work could be critical in developing sampling algorithms with better representation of the data distribution which can in turn be used to develop more stable methods.

Reference

[1] Charu C Aggarwal and Philip S Yu. “Outlier detection for high dimensional data”. In: *Proceedings of the 2001 ACM SIGMOD international conference on Management of data*. 2001, pp. 37–46.

[2] Subutai Ahmad et al. “Unsupervised real-time anomaly detection for streaming data”. In: *Neurocomputing* 262 (2017), pp. 134–147.

[3] Mennatallah Amer and Markus Goldstein. “Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer”. In: *Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012)*. 2012, pp. 1–12.

[4] Fabrizio Angiulli. “CFOF: a concentration free measure for anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14.1 (2020), pp. 1–53.

[5] Fabrizio Angiulli and Clara Pizzuti. “Fast outlier detection in high dimensional spaces”. In: *European conference on principles of data mining and knowledge discovery*. Springer. 2002, pp. 15–27.

[6] Charles E. Antoniak. “Mixtures of dirichlet processes with applications to Bayesian nonparametric problems”. In: *Annals of Statistics* 2.6 (1974).

[7] Stephen D Bay and Mark Schwabacher. “Mining distance-based outliers in near linear time with randomization and a simple pruning rule”. In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. 2003, pp. 29–38.

[8] Husam Al-Behadili et al. “Semi-supervised learning using incremental support vector machine and extreme value theory in gesture data”. In: *Computer Modelling and Simulation (UKSim), 2016 UKSim-AMSS 18th International Conference on*. IEEE. 2016, pp. 184–189.

[9] Abhijit Bendale and Terrance Boult. “Towards Open World Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[10] David M. Blei and Michael I. Jordan. “Variational Methods for the Dirichlet Process”. In: *ICML*. 2004.

[11] Carl Boettiger and Alan Hastings. “No early warning signals for stochastic transitions: insights from large deviation theory”. In: *Proceedings of the Royal Society B: Biological Sciences* 280.1766 (2013), p. 20131372.

[12] Markus M. Breunig et al. “LOF: identifying density-based local outliers”. In: *Proceedings of 2000 ACM SIGMOD International Conference on Management of Data*. 2000, pp. 93–104.

[13] Zeynep Ceylan. “Estimation of COVID-19 prevalence in Italy, Spain, and France”. In: *Science of The Total Environment* 729 (2020), p. 138817.

[14] Philip K Chan, Matthew V Mahoney, and Muhammad H Arshad. *A machine learning approach to anomaly detection*. Tech. rep. 2003.

[15] V. Chandola and V. Kumar. “A Comparative Evaluation of Anomaly Detection Techniques for Sequence Data”. In: *Proceedings of International Conference on Data Mining*. Pisa, Italy, 2008.

[16] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *ACM Computing Surveys* 41.3 (2009).

[17] Varun Chandola, Deepthi Cheboli, and Vipin Kumar. *Detecting Anomalies in a Time-series Database*. Tech. rep. 09-004. University of Minnesota, Computer Science Department, Feb. 2009.

[18] Myriam Charras-Garrido and Pascal Lezaud. “Extreme value analysis: an introduction”. In: *Journal de la Société Française de Statistique* 154.2 (2013), pp–66.

[19] Sanjay Chawla and Aristides Gionis. “k-means–: A unified approach to clustering and outlier detection”. In: *SDM*. 2013.

[20] David A Clifton et al. “Extending the Generalised Pareto Distribution for Novelty Detection in High-Dimensional Spaces”. In: *Journal of Signal Processing Systems* 74.3 (2014), pp. 323–339.

[21] Laurens De Haan and Ana Ferreira. *Extreme value theory: an introduction*. Springer Science & Business Media, 2007.

[22] Giovanni Dematteis, Tobias Grafke, and Eric Vanden-Eijnden. “Rogue waves and large deviations in deep sea”. In: *Proceedings of the National Academy of Sciences* 115.5 (2018), pp. 855–860.

[23] Frank Den Hollander. *Large deviations*. Vol. 14. American Mathematical Soc., 2008.

[24] Dua Dheeru and Efi Karra Taniskidou. *UCI Machine Learning Repository*. 2017. URL: `http://archive.ics.uci.edu/ml`.

[25] Josip Djolonga et al. "On robustness and transferability of convolutional neural networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 16458–16468.

[26] Ensheng Dong, Hongru Du, and Lauren Gardner. "An interactive web-based dashboard to track COVID-19 in real time". In: *The Lancet infectious diseases* 20.5 (2020), pp. 533–534.

[27] Eleazar Eskin et al. "A geometric framework for unsupervised anomaly detection". In: *Applications of data mining in computer security*. Springer, 2002, pp. 77–101.

[28] R. A. Fisher and L. H. C. Tippett. "Limiting forms of the frequency distribution of the largest or smallest member of a sample". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 24.2 (1928), pp. 180–190. DOI: `10.1017/S0305004100015681`.

[29] Jessica Franzen. *Bayesian Inference for a Mixture Model using the Gibbs Sampler*. Research Report RR 2006:1. Stockholm University, 2006.

[30] Joshua French et al. "Quantifying the risk of heat waves using extreme value theory and spatio-temporal functional data". In: *Computational Statistics & Data Analysis* 131 (2019), pp. 176–193.

[31] Bela A. Frigyik, Amol Kapila, and Maya R. Gupta. *Introduction to the Dirichlet Distribution and Related Processes*. Tech. rep. 206. 2010.

[32] Zhouyu Fu, Weiming Hu, and Tieniu Tan. "Similarity based vehicle trajectory clustering and anomaly detection". In: *Image Processing, 2005. ICIP 2005. IEEE International Conference on*. Vol. 2. IEEE. 2005, pp. II–602.

[33] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. "An approach to spacecraft anomaly detection problem using kernel feature space". In: *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*. 2005, pp. 401–410.

[34] João Gama et al. "A survey on concept drift adaptation". In: *ACM Computing Surveys* 46.4 (2014), p. 44.

[35] Pedro Garcia-Teodoro et al. "Anomaly-based network intrusion detection: Techniques, systems and challenges". In: *computers & security* 28.1-2 (2009), pp. 18–28.

[36] ZongYuan Ge et al. *Generative OpenMax for Multi-Class Open Set Classification*. 2017. arXiv: `1707.07418 [cs.CV]`.

[37] B. Gnedenko. “Sur La Distribution Limite Du Terme Maximum D’Une Serie Aleatoire”. In: *Annals of Mathematics* 44.3 (1943), pp. 423–453.

[38] Markus Goldstein and Seiichi Uchida. “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data”. In: *PloS one* 11.4 (2016), e0152173.

[39] Markus Goldstein and Seiichi Uchida. “A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data”. In: *PloS one* 11.4 (2016), e0152173.

[40] Nico Görnitz et al. “Toward supervised anomaly detection”. In: *Journal of Artificial Intelligence Research* 46 (2013), pp. 235–262.

[41] Dilan Görür and Carl Edward Rasmussen. “Dirichlet Process Gaussian Mixture Models: Choice of the Base Distribution”. In: *J. Comput. Sci. Technol.* 25.4 (2010), pp. 653–664.

[42] Sreelekha Guggilam et al. “Integrated Clustering and Anomaly Detection (INCAD) for Streaming Data (Revised)”. In: *arXiv preprint arXiv:1911.00184* (2019).

[43] Manish Gupta et al. “Outlier detection for temporal data: A survey”. In: *IEEE Transactions on Knowledge and data Engineering* 26.9 (2013), pp. 2250–2267.

[44] Zengyou He, Xiaofei Xu, and Shengchun Deng. “Discovering cluster-based local outliers”. In: *Pattern Recognition Letters* 24.9-10 (2003), pp. 1641–1650.

[45] N. Hjort et al. *Bayesian Nonparametrics: Principles and Practice*. Cambridge, UK: Cambridge University Press, 2010.

[46] Victoria Hodge and Jim Austin. “A Survey of Outlier Detection Methodologies”. In: *Artificial Intelligence Review* 22.2 (2004), pp. 85–126.

[47] Viet Huynh, Dinh Phung, and Svetha Venkatesh. “Streaming Variational Inference for Dirichlet Process Mixtures”. In: *ACML*. 2016.

[48] Jialiang Jiang et al. “Improving Quality of Care using Data Science Driven Methods”. In: *UNYTE Scientific Session - Hitting the Accelerator: Health Research Innovation through Data Science*. 2015.

[49] Nan Jiang and Le Gruenwald. “Research issues in data stream association rule mining”. In: *ACM Sigmod Record* 35.1 (2006), pp. 14–19.

[50] Taskin Kavzoglu. “Increasing the accuracy of neural network classification using refined training data”. In: *Environmental Modelling & Software* 24.7 (2009), pp. 850–858.

[51] Christopher Kruegel and Giovanni Vigna. “Anomaly detection of web-based attacks”. In: *Proc. of the 10th ACM conference on Computer and communications security*. 2003, pp. 251–261.

[52] Yann LeCun and Corinna Cortes. “MNIST handwritten digit database”. In: (2010). URL: `http://yann.lecun.com/exdb/mnist/`.

[53] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation-based anomaly detection”. In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.1 (2012), pp. 1–39.

[54] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[55] Mohsen Maleki et al. “Time series modelling to forecast the confirmed and recovered cases of COVID-19”. In: *Travel medicine and infectious disease* 37 (2020), p. 101742.

[56] David J Marchette. “A Statistical Method for Profiling Network Traffic.” In: *Workshop on Intrusion Detection and Network Monitoring*. 1999, pp. 119–128.

[57] Thomas Mikosch and Olivier Wintenberger. “A large deviations approach to limit theory for heavy-tailed time series”. In: *Probability Theory and Related Fields* 166.1 (2016), pp. 233–269.

[58] Radford M. Neal. “Markov chain sampling methods for Dirichlet process mixture models”. In: *Journal of Computational and Graphical Statistics* 9.2 (2000), pp. 249–265.

[59] *Numenta Anomaly Benchmark Evaluates Anomaly Detection Techniques For Real-Time, Streaming Data*. `http://numenta.com/press/numenta-anomaly-benchmark-nab-evaluates-anomaly-detection-techniques.html`. 2015.

[60] Ioannis Ch Paschalidis and Georgios Smaragdakis. “Spatio-temporal network anomaly detection by assessing deviations of empirical measures”. In: *IEEE/ACM Transactions On Networking* 17.3 (2008), pp. 685–697.

[61] James Pickands. “Statistical Inference Using Extreme Order Statistics”. In: *Annals of Statistics* 1 (1975), pp. 119–131.

[62] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. “Efficient algorithms for mining outliers from large data sets”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. Dallas, Texas, United States: ACM Press, 2000, pp. 427–438. ISBN: 1-58113-217-4.

[63] Carl Edward Rasmussen. “The Infinite Gaussian Mixture Model”. In: *NIPS*. MIT Press, 2000, pp. 554–560.

[64] Shebuti Rayana. *ODDS Library*. 2016. URL: `http://odds.cs.stonybrook.edu`.

[65] Peter J Rousseeuw and Katrien Van Driessen. “A fast algorithm for the minimum covariance determinant estimator”. In: *Technometrics* 41.3 (1999), pp. 212–223.

[66] Robert E Schapire. “The boosting approach to machine learning: An overview”. In: *Nonlinear estimation and classification* (2003), pp. 149–171.

[67] Murali Shanker, Michael Y Hu, and Ming S Hung. “Effect of data standardization on neural network training”. In: *Omega* 24.4 (1996), pp. 385–397.

[68] Matthew S. Shotwell and Elizabeth H. Slate. “Bayesian Outlier Detection with Dirichlet Process Mixtures”. In: *Bayesian Anal.* 6.4 (Dec. 2011), pp. 665–690.

[69] Alban Siffer et al. “Anomaly Detection in Streams with Extreme Value Theory”. In: *ACM KDD*. 2017, pp. 1067–1075.

[70] Deepak Soekhoe, Peter Van Der Putten, and Aske Plaat. “On the impact of data set size in transfer learning using deep neural networks”. In: *International symposium on intelligent data analysis*. Springer. 2016, pp. 50–60.

[71] Swee Chuan Tan, Kai Ming Ting, and Tony Fei Liu. “Fast Anomaly Detection for Streaming Data”. In: *IJCAI*. 2011, pp. 1511–1516.

[72] David M. J. Tax and Robert P. W. Duin. “Support Vector Data Description”. In: *Mach. Learn.* 54.1 (2004), pp. 45–66.

[73] Yee Whye Teh et al. “Hierarchical Dirichlet Processes”. In: *Journal of the American Statistical Association* 101.476 (2006), pp. 1566–1581.

[74] Hugo Touchette. “The large deviation approach to statistical mechanics”. In: *Physics Reports* 478.1-3 (2009), pp. 1–69.

[75] J. Varadarajan et al. “Active Online Anomaly Detection Using Dirichlet Process Mixture Model and Gaussian Process Classification”. In: *WACV*. 2017, pp. 615–623.

[76] SR Srinivasa Varadhan. “Large deviations”. In: *Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures*. World Scientific. 2010, pp. 622–639.

[77] SR Srinivasa Varadhan. *Large deviations and applications*. SIAM, 1984.

[78] Alexander Vergara et al. “Chemical gas sensor drift compensation using classifier ensembles”. In: *Sensors and Actuators B: Chemical* 166 (2012), pp. 320–329.

[79] Jinrui Wang et al. “Batch-normalized deep neural networks for achieving fast intelligent fault diagnosis of machines”. In: *Neurocomputing* 329 (2019), pp. 53–65.

[80] Dragomir Yankov, Eamonn Keogh, and Umaa Rebbapragada. “Disk aware discord discovery: Finding unusual time series in terabyte sized datasets”. In: *Knowledge and Information Systems* 17.2 (2008), pp. 241–262.

[81] Abdelhafid Zeroual et al. “Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study”. In: *Chaos, Solitons & Fractals* 140 (2020), p. 110121.

[82] Yanfei Zhong et al. “SatCNN: satellite image dataset classification using agile convolutional neural networks”. In: *Remote sensing letters* 8.2 (2017), pp. 136–145.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346 USA