Enhancing Anomaly Detection with Flexible Distribution Models

Oussama SGHAIER

A Thesis

in

The Concordia Institute for Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality Systems Engineering) at

Concordia University

Montréal, Québec, Canada

December 2023

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Oussama SGHAIER**

Entitled: **Enhancing Anomaly Detection with Flexible Distribution Models**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Yong Zeng

_____ Examiner
Dr. Jamal Bentahar

_____ Supervisor
Dr. Nizar Bouguila

_____ Co-supervisor
Dr. Manar Amayri

Approved by _____
Dr. Jun Yan, Chair
Department of The Concordia Institute for Information Systems Engineering

_____ 2023

_____
Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Enhancing Anomaly Detection with Flexible Distribution Models

Oussama SGHAIER

The performance of an anomaly detection task depends on the modeling of the input data. In the case of proportional data, Dirichlet and its general form distributions are a convenient choice to effectively capture the underlying characteristics of this kind of data.

In this thesis, we propose a normality score approach based on transformations that consist of learning a normality function. We suggest geometric transformations for image data and transformation-based neural networks for non-image data. Then, we propose an approximation of the softmax output vector of a classifier with generalized Dirichlet (GD), scaled Dirichlet (SD), shifted scaled Dirichlet (SSD), and Beta-Liouville (BL) distributions. We use a technique based on likelihood to determine its parameters.

Motivated by the salient characteristics of Liouville and Libby-Novick Beta distributions, we expand the Beta-Liouville distribution and build a new distribution called the Libby-Novick Beta-Liouville distribution. We demonstrate the efficiency of our proposed distribution through three challenging approaches. First, we develop generative models, namely finite mixture models of Libby-Novick Beta-Liouville distributions. Then, we propose two discriminative techniques: normality scores based on selecting the given distribution to approximate the softmax output vector of a deep classifier, and an improved version of the Support Vector Machine (SVM) by suggesting a feature mapping method. We test the efficiency of our suggested techniques for anomaly detection tasks using several experimental settings and five data sets: three image data sets and two non-image data sets.

Acknowledgments

I dedicate this page to thank all those who contributed directly or indirectly to the success of this master's thesis.

With much appreciation, I want to sincerely thank and express my gratitude to Professor Nizar Bouguila, my supervisor, for his genuine encouragement and support during this Master's program. At Concordia, I had the good fortune to work under one of the most accomplished, respected, and talented supervisors. I'm really grateful to you!

I would like to express my deepest acknowledgements to Dr. Manar Amayri, my co-supervisor, for her valuable guidance during my work.

To my beloved family, my parents and my two brothers, thank you for your endless support during this amazing journey. Your encouragement and love have been my anchor, and I couldn't have done it without you.

Finally, I would like to thank my relatives in Montreal and my friends; Houssem, Ahmed, Omar, Mohamed, and Jawher for their encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Anomaly detection is the identification of patterns in a data set that conflict with the normal behavior [1]. It has become the topic of extensive research thanks to its potential applications such as fraud detection in credit card transactions [2], detection of the presence of malignant tumors in MRI images [3] $\cdots$ However, the task of anomaly detection remains challenging. We can resume the challenges in these two points, 1) The exact definition of an anomaly event is still ambiguous and depends on the studied case; 2) Collecting abnormal samples is hard and costs much time due to the fact that anomaly events are rare.

Thanks to its intuitive interpretation, the normality scores approach makes it simpler to identify and understand the degree of abnormality in data points. While lower scores point to possible abnormalities, higher scores usually imply a higher possibility of being normal. In our second chapter, we develop a deep anomaly detector for both image and non-image data based on transformations and normality scores, with a generalization for the assumption of the softmax output vector. In this context, the assumption of approximating the softmax vector with a Dirichlet distribution is a weak hypothesis. Moreover, although Dirichlet distribution has been used in several applications such as human skin detection [4] and online data clustering [5], it has strong independencies between random variables, which makes it less robust in real-life applications [6],[7],[8]. Also, it has poor parameterization that limits the representation of variance and covariance in a data set. Thus, to

handle the problems cited above, we choose to approximate the output softmax vector with general forms of Dirichlet distribution as they have a more general covariance structure and have more parameters, which offer more degrees of freedom and flexibility. The selected general forms of Dirichlet distribution are generalized Dirichlet, scaled Dirichlet, shifted scaled Dirichlet, and Beta-Liouville distributions.

In the second work, motivated by the great performance of Beta-Liouville distribution in the first part of the thesis and taking the advantage of Libby-Novick Beta in modeling data on the support [0,1][9], we expand the Beta-Liouville distribution and build a new distribution called the Libby-Novick Beta-Liouville distribution. Compared to Dirichlet, it contains three more parameters, and one more parameter compared to Beta-Liouville. Therefore, it provides more degrees of freedom for data modeling. Moreover, the additional shape parameters can change the tail weights, simultaneously adjust the skewness and kurtosis, and increase the entropy of the resulting distribution [9]. Furthermore, it has almost half the number of the generalized Dirichlet parameters, which reduces significantly the complexity as well as the execution time. The main objective of creating such a distribution is to illustrate the potential of both generative approaches and discriminative methods in accomplishing excellent achievement in anomaly detection tasks. Typical generative-based approaches such as model-based reconstruction schemes like Autoencoder (AE) [10],[11],[12] and Variational Autoencoder (VAE) [13] rely only on learning the normal data during the training. Therefore, they could miss the distinctive features of outliers, which could lead to misclassification of anomalies. In this setting, mixture models [14],[15],[16],[17] are a very effective generative approach in learning the distribution of the entire dataset and are hence well-suited for anomaly identification. Furthermore, mixture models enable a formal solution to unsupervised learning [15]. Taking these benefits, we develop Libby-Novick Beta-Liouville finite mixture models for detecting anomalies. For discriminative techniques, the normality scores approach described in the first part of this introduction might be considered a decision boundary method capable of clearly separating the normal class from the anomaly class by estimating the classifier output vector with Libby-Novick Beta-Liouville distribution. Added to that, a range of classical approaches has been developed including Support Vector Machine (SVM) [18], Isolation Forest (IF) [19][20], Local Outlier Factor (LOF) [20], K-Nearest Neighbors (KNN) [21], etc. However, these approaches such as KNN suffer

from sensitivity to hyperparameters, and they do not take into account the kind of data. Thanks to its computational effectiveness, particularly in high-dimensional feature spaces, SVM has established itself as a standard learning tool producing benchmark results. However, its traditional kernels do not take into account the nature of the data. For that, a feature mapping function may be constructed using the benefits of Libby-Novick Beta-Liouville in terms of flexibility and data nature capture. This will improve understanding of the statistical properties of the data and lead to an improvement in classification accuracy.

1.2 Related Work

We can divide the previous related work into two main categories: Generative models and discriminative models.

Hidden Markov Models (HMM) were introduced in [22],[23] as an effective generative technique for data modeling and data clustering. Additionally, several previous generative methods were based on the development of Auto Encoder [10],[11],[24]. It is composed of an Encoder to transform the input into a latent vector, and a decoder that repeats the input from the latent vector. At testing time, the normal samples have small reconstruction errors, while the abnormal ones are supposed to have large reconstruction errors. The main issue with AutoEncoder is that it requires a regularized latent space, where each point in the latent space is significant, to produce data correctly. Variational Auto Encoder was introduced to solve this issue [25]. Moreover, Generative Adversarial Networks (GAN) are widely used in the task of anomaly detection especially in images [26],[27],[28], as they can easily generate detailed reconstructed images. Another generative scheme introduced in this field is mixture models based on a given distribution. Typical mixture models were based on the Gaussian distribution (Gaussian Mixture Models: GMM). In [29], the authors proposed a Deep Autoencoding Gaussian Mixture Model (DAGMM) for unsupervised anomaly detection. They generated a low representation of the input data through a deep autoencoder and fed the reconstructed data to a Gaussian mixture model. Added to that, LGMAD was introduced in [30] which is a combination of LSTM and GMM. The goal was to detect anomalies in time-series data sets. However, it has been demonstrated that the Gaussian distribution is excessively inflexible and is not the best

option for proportional data. In this context, Dirichlet and its general forms distribution mixture models [31],[32][33],[34],[35],[36] have received less attention than that of Gaussian, yet some significant research works in outlier detection have been presented for modeling proportional data. The Dirichlet process mixture is used in [37] to model information in order to detect outliers in large-scale traffic data. Moreover, to build a scalable anomaly detection system, Dirichlet mixture models serve as a decision engine [38], where the process starts by collecting network data, then analyses and filters data, and at the end, classify samples with Dirichlet mixture models. In [39][40], the work was focused on modeling proportional data for classification tasks by applying Dirichlet, generalized Dirichlet, and Beta-Liouville mixture models. Using the same distributions mixture models (Dirichlet, generalized Dirichlet, and Beta-Liouville), the work in [41] was dedicated to the application of the spacial color image segmentation.

For discriminative approaches, several architectures have been developed for the task of anomaly detection. In this section, we will consider only two kinds of approaches: Normality scores methods and techniques based on SVM.

The idea of the normality scores approach is to train the model on the normal data, and at testing time, a score is given for each sample to classify it as an anomaly or not. In [42], authors proposed an architecture that starts by applying geometric transformations on image data, then fed the transformed data to a classifier. At testing time, and after approximating the softmax output vector with Dirichlet distribution, a normality scores function is built to classify the samples. Another interesting work was presented in [43]. The normality scores were developed during the testing time after applying a semi-supervised method based on GANs and frame prediction.

SVM has been a powerful tool in several research works to achieve the task of anomaly detection. For instance, authors in [44] began by outlining the necessary terminology for the SVM classifier and intrusion detection systems. Then, they discussed how different machine learning methods have been used in conjunction with the SVM classifier to identify abnormalities. Deep learning was introduced with SVM, where the architecture presented in [45] started by training deep belief networks (DBN) to extract robust features, then training one-class SVM from the features extracted by DBN. Implementing feature mapping functions was another topic of interest for improving the performance of SVM while dealing with proportional data. The feature mapping function suggested

by [46] based on the Dirichlet distribution has proven effective for several classification and regression tasks using proportional data. As an improvement to this work, authors in [47] try to exploit the explanatory capabilities of generalized Dirichlet and Beta-Liouville distributions in modeling proportional data, to build a flexible feature mapping function.

1.3 Contributions

This thesis' primary goal is to investigate how well general types of Dirichlet distributions work when it comes to attaining excellent results in anomaly detection tasks for proportional data. We summarize our contributions as the following:

- **Dirichlet and Liouville-Based Normality Scores for Deep Anomaly Detection Using Transformations:**

  We propose a deep anomaly detection architecture based on normality scores by approximating the softmax output vector of the classifier with generalized Dirichlet (GD), scaled Dirichlet (SD), shifted scaled Dirichlet (SSD), and Beta-Liouville (BL) distributions. The proposed procedure is evaluated on both image data and non-image data. For the first kind of data, we choose the geometric transformation, while for the second kind of data, neural networks are a good option to transform them. This work has been submitted to *ieee transactions on neural networks and learning systems* and is under review [48].

- **Libby-Novick Beta-Liouville Distribution for Enhanced Anomaly Detection in Proportional Data:**

  In this work, we investigate the appropriateness of the proposed Libby-Novick Beta-Liouville distribution for modeling proportional data in developing useful approaches for anomaly detection. We develop Libby-Novick Beta-Liouville finite mixture models. We also introduce a deep anomaly detector based on a general assumption for the softmax predictions vector, applicable to both images and non-images. We provide the Libby-Novick Beta-Liouville method for approximating the classifier's output vector. Finally, we construct a novel feature mapping function in SVM using the Libby-Novick Beta-Liouville distribution.This contribution has been submitted to *ACM Transactions on Intelligent Systems and Technology* [49].

1.4 Thesis Overview

- In Chapter 1, we describe in detail the proposed deep anomaly architecture-based normality scores. We present the different generalizations of Dirichlet normality scores. It also includes the learning of the parameters. We conduct experiments on different data sets, and we implement some baseline methods for benchmarking.

- In Chapter 2, we introduce our proposed Libby-Novick Beta-Liouville distribution. Also, we present three different approaches based on it: finite mixture models, normality scores, and feature mapping in SVM. We perform several tests on three image data sets and two non-image data.

- In Chapter 3, we provide a summary of our overall contributions in closing remarks.

Chapter 2

Dirichlet and Liouville-Based Normality Scores for Deep Anomaly Detection Using Transformations

In this chapter, we describe in detail our normality scores architecture and we develop the different general forms of Dirichlet normality scores function. We use maximum likelihood to estimate the different parameters.

2.1 The Proposed Procedure

2.1.1 Problem Statement

In this part, we focus on anomaly detection based on normality score. The principle of this methodology is as follows: let $\mathcal{X}$ be the set of all data samples and each sample has its own label: 'Normal' or 'Anomaly'. Let **X** be the set of normal samples, the idea is to establish a classifier *C*(x) that takes a sample x and returns 1 if x $\in$ **X** and 0 if not. For that, we need to build a score function $n_s(x)$ and compare its value to a threshold λ, and based on this comparison, we classify our sample whether it is an anomaly or not.

$$C_s^{\lambda}(x) = \begin{cases} 1 & n_s(x) \geq \lambda \\ 0 & n_s(x) < \lambda. \end{cases}$$

In our work, the main challenge is how to learn the score function not how to detect a suitable value for the threshold λ, that's why we will focus only on how to create the score function correctly and ignore the constrained binary decision problem. For that, we need useful metrics to evaluate the score function. As mentioned in [42], a useful metric to evaluate the quality of the score function is the Area Under the Receiver Operating Characteristic (AUROC) which is used to measure the usefulness of a test or a combination of tests where a greater area means more useful test, it tells how much the model is capable of distinguishing between classes. Another metric can be used which is Area Under Precision-Recall (AUPR) which can be suitable when we have prior knowledge of the anomaly proportion.

2.1.2 General Framework of the Proposed Architecture

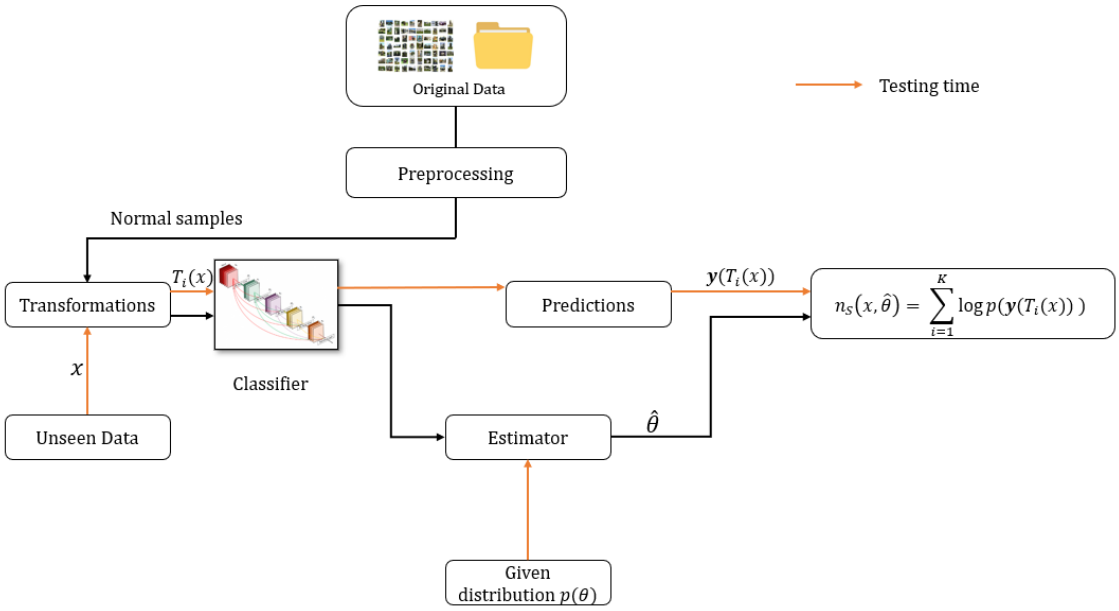


Figure 2.1: The proposed anomaly detection pipeline

In this section, we develop a supervised model, aiming at learning the normality of a given sample using transformations. An overview of the architecture is presented in Fig.2.1.
For the transformation step, let

$$\Delta = \{T_1, T_2, \dots T_K\}$$

be the set of K transformations. Each transformation will be applied on each normal sample of the data set so that at the end of the operation we get the transformed data set:

$$\mathbf{X}_T \triangleq \{(T_i(x), i) : x \in \mathbf{X}, T_i \in \Delta\}.$$

where i is the index of transformation, T_i is the corresponding transformation, x is the given sample and $\mathbf{X}$ is the set of normal samples. In this manner, each new transformed sample has a new label which is the index of transformation. As we will see later in the implementation details, we use the one-vs-all technique which consists of considering one class as normal and the rest of the classes as anomalies, we use the transformed data of the normal classes to feed a deep multi-class classifier fc_θ (K-classifier). The main goal of using such a classifier is to predict which transformation is applied to the sample. At the testing time, we take an unseen sample, and then apply all the transformations in Δ on it, after each transformation, the trained K-classifier model will output a K-soft-max vector (length = K) where each element of the vector assigns the probability of the given unseen transformed sample to belong to the class of transformation which is the index of the element. Then, starting from the prediction vectors generated after each transformation, we build our score function as the sum of the log-likelihood of the distributions of these vectors:

$$n_s(x) = \sum_{i=1}^{K} \log p(\boldsymbol{y}(T_i(x))|T_i) \tag{1}$$

where $\boldsymbol{y}(T_i(x))$ is the soft-max vector predictions outputted by the deep classifier on the i^{th} $\boldsymbol{X}_{T_i}$: data (data after being applied by the transformation T_i). Note that we assume that the conditional probabilities in the score function are independent. From the expression of the normality score function, we can assume that the higher the score of an image the more likely to be normal, in other words, if $n_s(x_1) > n_s(x_2)$, x_1 is more normal than x_2.

Back to the transformation task, and as mentioned in the introduction, we choose to apply geometric transformations to images for two main reasons. 1) Geometric transformations are a set of bijections. As a result, the original image and its transformed version will have the same geometric structure. By this way, we define the effectiveness of this kind of transformation by its ability to preserve the spatial information about normal samples [42]; 2) Non-geometric transformations such as sharpening can easily destroy the features of an image which leads to bad performance. For non-images data, the idea of neural networks based on dense layers was inspired by geometric transformations. In fact, geometric transformations are linear transformations as they are changes in the bases formula which preserve the structure of a sample. Thus, under certain circumstances, many qualitative evaluations of a vector space that is the subject of a linear transformation may hold automatically in the form of the linear transformation. So the most suitable idea is to apply neural networks based on linear layers. The difference between the neural networks is the size of the hidden layers. In this way, we obtain different representations of the data. The auto-encoder was added for a better extraction of features.

Talking now about the classifier, we decided to assign each kind of data to a specific classifier. For image data, we choose to apply Wide Residual Networks [50] as it has shown good performance in classifying images [50],[42]. Compared to traditional residual networks, WRN has more channels per convolution layer. It has two main parameters: N number of convolution blocks and D number of feature maps to increase per layer. For non-image data, we build our own classifier which is a succession of 1D Convolution layer followed by max-pooling layer and dense layer. The convolution layer takes into consideration spatial information in a way we can reduce the variation between the different features. In earlier stages of this work, we tried to build a classifier based only on dense layers, but we got bad results in training it, so we abandoned it. We hypothesize that dense layers cannot reduce the variance between the features.

The estimation of the parameters of the distribution followed by $\boldsymbol{y}(T_i(x))$ for a fixed transformation T_i is based on the predictions of the classifier for the normal transformed data $\mathbf{X}_{T_i}$. Let $\boldsymbol{C}_i = (\boldsymbol{C}_{i1}, \cdots, \boldsymbol{C}_{iN})$ with $\boldsymbol{C}_{ij}$ $(j = 1, \cdots, N)$ is the softmax output vector prediction for the sample j in $\boldsymbol{X}_{T_i}$. N is the cardinal of $\boldsymbol{X}_{T_i}$. We use maximum likelihood to estimate our parameters. Authors in [42] choose to approximate the softmax output vector predictions $\boldsymbol{y}(T_i(x))$ with

Dirichlet distribution for two main reasons: 1) It is a common choice for data defined on a simplex; 2) Since the Dirichlet distribution, as well as its general forms, belong to the exponential family, the objective function of log-likelihood will be convex, therefore, the maximum can easily be found by a simple search [51]. The expression of the likelihood is given by:

$$p(\boldsymbol{C}_i|\boldsymbol{\theta}_i) = \prod_{j=1}^{N} p(\boldsymbol{C}_{ij}|\boldsymbol{\theta}_i) \tag{2}$$

For the rest of the chapter, we fix the set of transformations as $\Delta = \{T_1, T_2, \dots T_K\}$. $n_s(x)$ is our normality score and $\boldsymbol{y}(T_i(x))$ = softmax($f_\theta(T_i(x))$) the output predictions vector of f_θ applied on $T_i(x)$ with f_θ is a K-class classification model trained on $\mathbf{X}_T$. K is the number of transformations.

2.2 Normality Scores

In this section, our intention is to build a verified version of normality score. We assume an approximation of the softmax output vector with general forms of Dirichlet distribution. The different relationships between the general forms of Dirichlet can be found in Table 2.1.

Distribution	Number of Parameters	Reduced to Dirichlet
Generalized Dirichlet	$2K$	β_i=α_{i+1}+β_{i+1}
Scaled Dirichlet	$2K$	$\boldsymbol{\beta} = (1, \cdots, 1)$
Shifted scaled Dirichlet	$2K + 1$	$\boldsymbol{\beta} = (1, \cdots, 1)$, $\boldsymbol{p} = (1, \cdots, 1)$ and $a = 1$
Beta-Liouville	$K + 2$	$\alpha = \sum_{k=1}^{K} \alpha_k$ and $\beta = \alpha_{K+1}$

Table 2.1: Different relationships between the general forms of Dirichlet distributions

2.2.1 Generalized Dirichlet Normality Score

In dimension K, the generalized Dirichlet probability density function is defined by[6]:

$$f(\boldsymbol{X}) = \prod_{i=1}^{K} \frac{\Gamma(\alpha_i + \beta_i)}{\Gamma(\alpha_i)\Gamma(\beta_i)} X_i^{\alpha_i - 1} (1 - \sum_{k=1}^{i} X_k)^{\gamma_i} \tag{3}$$

for α_i , β_i and $X_i > 0$ for $i = 0, \cdots, K$, $\sum_{i=0}^{K} X_i \leq 1$ and

$$\gamma_i = \begin{cases} \beta_i - \alpha_{i+1} - \beta_{i+1} & 1 \leq i \leq K-1 \\ \beta_K - 1 & i = K \end{cases}$$

When $\beta_i = \alpha_{i+1} + \beta_{i+1}$, the generalized Dirichlet is reduced to Dirichlet distribution which is given by:

$$p(\boldsymbol{X}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{i=1}^{d} X_i^{\alpha_i - 1} \tag{4}$$

It was an efficient tool for several applications such as market-share data mining [52] and unsual events detection [53]. The most important advantage of this general form and unlike the standard form, it eliminates the strict correlation negativity between any two random variables [34]. For more details, the mean and the variance of Dirichlet distribution are given by:

$$E(X_i) = \frac{\alpha_i}{\sum_{m=1}^{K} \alpha_m} \tag{5}$$

$$Var(X_i) = \frac{\alpha_i (\sum_{m=1}^{K} \alpha_m - \alpha_i)}{(\sum_{m=1}^{K} \alpha_m)^2 (\sum_{m=1}^{K} \alpha_m + 1)} \tag{6}$$

Thus, the covariance between two random variables X_i and X_j is given by:

$$Cov(X_i, X_j) = -\frac{\alpha_i \alpha_j}{(\sum_{m=1}^{K} \alpha_m)^2 (\sum_{m=1}^{K} \alpha_m + 1)} \tag{7}$$

From the expression in (7), we conclude that any two random variables from $\boldsymbol{X}$ are negatively correlated which is not always the case. In [6], Wong demonstrated that the covariance between two random variables in generalized Dirichlet is given by:

$$Cov(X_i, X_j) = E(X_j) \left(\frac{\alpha_i}{\alpha_i + \beta_i + 1} \prod_{k=1}^{i-1} \frac{\beta_k + 1}{\alpha_k + \beta_k + 1} - E(X_i) \right) \tag{8}$$

One other advantage of generalized Dirichlet distribution is that it has a more structured covariance matrix which makes it more practical and useful than Dirichlet. To understand very well

the difference between the two distributions, Wong [6] introduced two experiments, and from these experiments, it has been shown that the independencies between random variables in generalized Dirichlet are much weaker compared to Dirichlet, that's why the general form is more robust for realistic cases.

For this part, we approximate $\boldsymbol{y}(T_i(x)) \sim \mathrm{GD}(\boldsymbol{\theta_i})$ with GD is the generalized Dirichlet distribution and $\boldsymbol{\theta}_i$ = (α_{i1},$\cdots$,α_{ik},β_{i1},$\cdots$,β_{iK}) the parameter of the distribution, i is the index of transformation.

By injecting the expression of the density function (3) in the expression of the normality score (1), we get:

$$n_s(x) = \sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j + \sum_{i=1}^{K}\sum_{j=1}^{K}\tilde{\gamma}_{ij}\log\left(1-\sum_{m=1}^{j}[\boldsymbol{y}(T_i(x))]_m\right) + \sum_{i=1}^{K}\sum_{j=1}^{K}\log(B(\tilde{\alpha}_{ij},\tilde{\beta}_{ij})) \quad (9)$$

where $B(\alpha,\beta)$ is the beta function, $\tilde{\alpha}_{ij}$ and $\tilde{\beta}_{ij}$ are the estimators of α_{ij} and β_{ij} respectively. By eliminating the last term of the expression as it is independent of the sample, the score function becomes:

$$n_s(x) = \sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j + \sum_{i=1}^{K}\sum_{j=1}^{K}\tilde{\gamma}_{ij}\log\left(1-\sum_{m=1}^{j}[\boldsymbol{y}(T_i(x))]_m\right) \quad (10)$$

where:

$$\tilde{\gamma}_{ij} = \begin{cases} \tilde{\beta}_{ij} - \tilde{\alpha}_{i,j+1} - \tilde{\beta}_{i,j+1} & 1 \le j \le K-1 \\ \tilde{\beta}_{i,K-1} - 1 & j = K \end{cases}$$

For the parameters, and following a full study done by Wong. T in [6], we can get the following expression of the estimated vector parameters:

$$\tilde{\alpha}_{ij} = \frac{a_{ij}\mu_{ij}B_{i,j-1} - \mu_{ij}\left(\boldsymbol{S_{i}}_{jj} + \mu_{ij}^2\right)}{\mu_{ij}\left(A_{i,j-1} + \mu_{ij}^2\right) - a_{ij}\mu_{ij}B_{i,j-1}} \quad (11)$$

$$\tilde{\beta}_{ij} = \frac{\tilde{\alpha}_{ij}\left(A_{i,j-1} - \mu_{ij}\right)}{\mu_{ij}} \tag{12}$$

where $\boldsymbol{S}_i$ is the covariance matrix of the normal samples data set after being applied by the i^{th} transformation, $\boldsymbol{\mu}_i$ is the mean vector: μ_{ij} represent the mean of the prediction values for the j^{th} sample and:

$$\begin{cases} a_{ij} = \frac{\alpha_{ij}}{\alpha_{ij}+\beta_{ij}} \\ A_{ij} = \prod_{1=1}^{j} \frac{\beta_{il}}{\alpha_{il}+\beta_{il}} \\ B_{ij} = \prod_{1=1}^{j} \frac{\beta_{il}(\beta_{il}+1)}{((\alpha_{il}+\beta_{il})(\alpha_{il}+\beta_{il}+1)} \end{cases}$$

When β_{ij} = $\alpha_{i,j+1}$ + $\beta_{i,j+1}$ (the case where the generalized Dirichlet is reduced to Dirichlet), we obtain the following normality score which is the Dirichlet normality score as developed in [42]:

$$n_s(x) = \sum_{i=0}^{K-1}\left[\log\Gamma(\sum_{j=1}^{K}\tilde{\alpha}_{ij}) - \sum_{j=1}^{K}\log\Gamma(\tilde{\alpha}_{ij}) + \sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j\right] \tag{13}$$

The simplified form is given by:

$$n_s(x) = \sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j \tag{14}$$

2.2.2 Scaled Dirichlet Normality Score

In dimension K, the scaled Dirichlet probability density function is defined by [7]:

$$f(\boldsymbol{X}) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\frac{\prod_{i=1}^{K}\beta_i^{\alpha_i}X_i^{\alpha_i-1}}{(\sum_{i=1}^{K}\beta_i X_i)^{\sum_{i=1}^{K}\alpha_i}} \tag{15}$$

For a better understanding of the scaled Dirichlet distribution, we know that the Dirichlet family is the most convenient choice when it comes to choosing a suitable prior in Bayesian analysis of multinomial situations [7]. However, we need to mention that Dirichlet distribution does not take into account relative positions between categories or multinomial cells [7] [8]. The main difference between scaled Dirichlet distribution and the standard one is that in the first distribution, we

remove the requirement of the equal scaled parameters in the Gamma because the standard distribution can be obtained by normalizing a set of independent, equally scaled Gamma random variables (see Property 3.1 in [7]). As result, scaled Dirichlet can be reduced to Dirichlet when the Gamma random variables are scaled equally [54].

Now, we assume that $\boldsymbol{y}(T_i(x)) \sim \mathrm{SD}(\boldsymbol{\theta_i})$ where SD is the scaled Dirichlet distribution and $\boldsymbol{\theta}_i$ = (α_{i1},$\cdots$,α_{ik},β_{i1},$\cdots$,β_{iK}) the parameter of the distribution, i is the index of transformation.
Using the maximum likelihood method, the expressions of the estimated parameters are the following at iteration t:

$$\tilde{\alpha}_{ik,t} = \Psi^{-1}\left[\Psi(\sum_{j=1}^{K}\tilde{\alpha}_{jk,t-1}) + \log(\tilde{\beta}_{ik,t-1}) + \frac{1}{N}\sum_{j=1}^{N}\log(c_{jk}) \; - \frac{1}{N}\sum_{j=1}^{N}\log(\sum_{m=1}^{K}\tilde{\beta}_{im,t-1}c_{jm})\right] \quad k = 1\cdots K \tag{16}$$

$$\tilde{\beta}_{ik,t} = \frac{N\tilde{\alpha}_{ik,t}}{\left(\sum_{m=1}^{K}\tilde{\alpha}_{im,t}\right)\left(\sum_{j=1}^{N}\frac{c_{jk}}{\sum_{m=1}^{K}\tilde{\beta}_{ik,t-1}cjm}\right)} \quad k = 1\cdots K \tag{17}$$

where $\boldsymbol{C}$ = $(c_{ji})_{j=1\cdots N, i=1\cdots K}$ the matrix where the j^{th} raw represents the softmax output vector for the sample j in $\boldsymbol{X}_{T_i}$ Once we have the expression of the parameters we can calculate our score function:

$$\begin{aligned} n_s(x) = &\sum_{i=1}^{K}\log\left(\Gamma(\tilde{\alpha}_{ij}\right) - \sum_{i=1}^{K}\sum_{j=1}^{K}\log\left(\Gamma(\tilde{\alpha}_{ij})\right) \\ &+ \sum_{i=1}^{K}\sum_{j=1}^{K}\tilde{\alpha}_{ij}\log\left(\tilde{\beta}_{ij}\right) + \sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j \\ &- \sum_{i=1}^{K}(\tilde{\alpha}_{i,+})\log\left(\sum_{j=1}^{K}\tilde{\beta}_{ij}[\boldsymbol{y}(T_i(x))]_j\right) \end{aligned} \tag{18}$$

where $\tilde{\alpha}_{ij}$ is the estimate of α_{ij} and with $\tilde{\beta}_{ij}$ is the estimate of β_{ij}, and $\tilde{\alpha}_{i,+} = \sum_{j=1}^{K}\tilde{\alpha_{ik}}$ After removing all the terms that are not related to our observations, the new expression of $n_s(x)$ is:

$$\sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j - \sum_{i=1}^{K}(\tilde{\alpha}_{i,+})\log\left(\sum_{j=1}^{K}\tilde{\beta}_{ij}[\boldsymbol{y}(T_i(x))]_j\right) \tag{19}$$

By setting $(\beta_{i1}, \cdots, \beta_{iK}) = (1, \cdots, 1)$, the normality score in 19 becomes:

$$\sum_{i=1}^{K}\sum_{j=1}^{K}(\tilde{\alpha}_{ij}-1)\log[\boldsymbol{y}(T_i(x))]_j - \sum_{i=1}^{K}(\tilde{\alpha}_{i,+})\log\left(\sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j\right) \tag{20}$$

As $\sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j = 1$ because $\boldsymbol{y}(T_i(x))$ is the softmax output vector of the classifier, the normality score in 19 becomes the same as the Dirichlet normality score in 14. In this way, we deduce that by generalizing the distribution, we are generalizing the normality score.

2.2.3 Shifted Scaled Dirichlet Normality Score

Being in the same family as the previously discussed distributions, shifted scaled Dirichlet distribution is a modified version of Dirichlet distribution where operations of powering and perturbations are applied.

Assuming X follows a shifted scaled Dirichlet with parameters $\boldsymbol{\theta} = (\alpha_1, \cdots, \alpha_K, p_1, \cdots, p_K, a)$, its probability density function is defined by [7]:

$$f(\boldsymbol{X}) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\frac{1}{a^{K-1}}\frac{\prod_{i=1}^{K}p_i^{-\frac{\alpha_i}{a}}X_i^{\frac{\alpha_i}{a}-1}}{\left(\sum_{i=1}^{K}\left(\frac{X_i}{p_i}\right)^{\frac{1}{a}}\right)^{\sum_{i=1}^{K}\alpha_i}} \tag{21}$$

As generalized Dirichlet and scaled Dirichlet distributions, shifted scaled Dirichlet distribution has almost twice the number of parameters compared to the Dirichlet distribution $(2K+1)$ which provide the flexibility for diverse real-world applications [55],[56],[57], and also provide the ability to model the mean and the variance-covariance matrix separately [56]. It has complete permutation symmetry. The parameter a is called the scale parameter and it describes how the plotting of the density is distributed (stretching or shrinking the distribution), while the location parameter $\boldsymbol{p}$ = $(p_1, \cdots, p_K)$ follows the location of the data densities that simply shift the samples[58][59]. Note that when $a = 1$, the shifted scaled Dirichlet distribution is reduced to scaled form of Dirichlet distribution with parameters $\boldsymbol{\theta} = (\alpha_1, \cdots, \alpha_K, \frac{1}{p_1}, \cdots, \frac{1}{p_K})$, also we can move to the standard form by setting $a = 1$ and $\boldsymbol{p}$ = $(1, \cdots, 1)$.

In this sub-section, we approximate the softmax vector with shifted scaled Dirichlet distribution;

$\boldsymbol{y}(T_i(x)) \sim \text{SSD}(\boldsymbol{\theta_i})$ where SSD is the shifetd scaled Dirichlet distribution and $\boldsymbol{\theta} = (\alpha_{i1}, \cdots, \alpha_{iK}, p_{i1}, \cdots, p_{iK}, a_i)$ the parameters of the distribution, i is the index of transformation.

From the expression of the pdf in (21), the normality score, in this case, is defined by:

$$\begin{aligned} n_s(x) = \sum_{i=1}^{K} \log[\Gamma(\tilde{\alpha}_{i,+})] - \sum_{i=1}^{K}\sum_{j=1}^{K} \log(\tilde{\alpha}_{ij}) - (K-1)\sum_{i=1}^{K} \log \tilde{a}_i \\ - \sum_{i=1}^{K}\sum_{j=1}^{K}(-\frac{\tilde{\alpha}_{ij}}{\tilde{a}_i} \log \tilde{p_{ij}}) + \sum_{i=1}^{K}\sum_{j=1}^{K}(\frac{\tilde{\alpha}_{ij}}{\tilde{a}_i} - 1) \log\left([\boldsymbol{y}(T_i(x))]_j\right) \\ - \sum_{i=1}^{K} \tilde{\alpha}_{i,+} \log \left[\sum_{j=1}^{K} \left(\frac{\boldsymbol{y}(T_i(x))_j}{\tilde{p}_{ij}}\right)^{\frac{1}{\tilde{a}_i}}\right] \end{aligned} \tag{22}$$

$\tilde{\alpha}_{ij}$ is the estimate of α_{ij}, $\tilde{p}_{ij}$ is the estimate of p_{ij}, $\tilde{a}_i$ is the estimate of the parameter a_i and $\tilde{\alpha}_{i,+} = \sum_{j=1}^{K} \tilde{\alpha}_{ij}$. After removing all the terms that are not related to our observations, the new expression of $n_s(x)$ is:

$$n_s(x) = \sum_{i=1}^{K}\sum_{j=1}^{K}(\frac{\tilde{\alpha}_{ij}}{\tilde{a}_i} - 1) \log\left([\boldsymbol{y}(T_i(x))]_j\right) - \sum_{i=1}^{K} \tilde{\alpha}_{i,+} \log \left[\sum_{j=1}^{K} \left(\frac{\boldsymbol{y}(T_i(x))_j}{\tilde{p}_{ij}}\right)^{\frac{1}{\tilde{a}_i}}\right] \tag{23}$$

As in the two previous distributions, we generalize the Dirichlet normality score. By setting $a = 1$ and $\mathbf{p} = (1, \cdots, 1)$, we obtain the expression in (14).

Using the same method as for scaled Diriclet distribution, we can find the following expressions of the estimated parameters at iteration t:

$$\begin{aligned} \tilde{\alpha}_{ik,t} = \Psi^{-1}\Bigg[\Psi(\sum_{j=1}^{K} \tilde{\alpha}_{ij,t-1}) + \frac{1}{\tilde{a}_{it-1}} \log\left(\tilde{p}_{ik,t-1}\right) + \frac{1}{N\tilde{a}_{it-1}} \sum_{j=1}^{N} \log c_{jk} \\ - \frac{1}{N}\sum_{j=1}^{N} \log[\frac{c_{jk}}{\tilde{p}_{ik,t-1}}]^{\frac{1}{\tilde{a}_{it-1}}}\Bigg] \quad k = 1 \cdots K \end{aligned} \tag{24}$$

$$\tilde{p}_{ik,t} = \frac{-\frac{\tilde{\alpha}_{ik,t-1}}{\tilde{a}_{it-1}}}{\sum_{j=1}^{N} \tilde{\boldsymbol{\alpha}}_{+} \frac{(c_{jk})^{\frac{1}{\tilde{a}_{it-1}}} \frac{1}{\tilde{a}_i} \frac{1}{\tilde{p}_{ik,t-1}}^{\frac{1}{\tilde{a}_{it-1}}+1}}{\sum_{l=1}^{K} (\frac{c_{jl}}{\tilde{p}_{il,t-1}})^{\frac{1}{\tilde{a}_{it-1}}}}} \quad k = 1 \cdots K \tag{25}$$

$$\tilde{a}_{i,t} = \frac{q1}{N(K-1)} \tag{26}$$

where:

$$q1 = \sum_{j=1}^{N} \tilde{\boldsymbol{\alpha}}_{+} \frac{\sum_{l=1}^{K} \log\left(\frac{c_{jl}}{\tilde{p}_{il,t-1}}\right) (\frac{c_{jl}}{\tilde{p}_{il,t-1}})^{\frac{1}{\tilde{a}_{it-1}}}}{\sum_{l=1}^{K} (\frac{c_{jl}}{\tilde{p}_{il,t-1}})^{\frac{1}{\tilde{a}_{it-1}}}} \tag{27}$$

2.2.4 Beta-Liouville Normality Score

The last general form in this work is Beta-Liouville distribution. From the name, this distribution is a mixture of both Liouville and Beta distributions. A K-dimentional vector $\boldsymbol{X}$ is assumed to follow a Liouville distribution with parameters (α_1,$\cdots$,α_K) and density generator $g(.)$ if its pdf is defined by [39] [60]:

$$f(\boldsymbol{X}) = g(u) \prod_{i=1}^{K} \frac{X_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{28}$$

By taking the density generator of this form:

$$g(u) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{u^{\Gamma(\sum_{i=1}^{K} \alpha_i - 1)}} f(u) \tag{29}$$

where $f(.)$ is the pdf of the variable u, therefore we can obtain a new form of the pdf of Liouville distribution in Eq(28):

$$f(\boldsymbol{X}) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{u^{\Gamma(\sum_{i=1}^{K} \alpha_i - 1)}} f(u) \prod_{i=1}^{K} \frac{X_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{30}$$

As Beta distribution has a flexible shape, we can adopt it as a density for the variable u with two positive parameters α and β [61]:

$$f(u|\alpha, \beta) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} u^{\alpha - 1}(1 - u)^{\beta - 1} \tag{31}$$

By injecting Eq (31) in (30), we obtain:

$$f(\boldsymbol{X}) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\prod_{i=1}^{K}\frac{X_i^{\alpha_i-1}}{\Gamma(\alpha_i)}(\sum_{i=1}^{K}X_i)^{\alpha-\sum_{i=1}^{K}\alpha_i}(1-\sum_{i=1}^{K}X_i)^{\beta-1} \tag{32}$$

Note that in (31), when we set $\alpha = \sum_{k=1}^{K}\alpha_k$ and $\beta = \alpha_{K+1}$, equation (30) is reduced to Dirichlet distribution with parameters α_1,$\cdots$,α_{K+1}.

Compared to Dirichlet distribution, Beta-Liouville has two extra parameters to adjust the spread of the distribution. In contrast to Dirichlet distribution, its covariance matrix may be positive or negative. Its flexibility allows researchers to apply it in different applications such as text classification and texture discrimination [62] and automatic image orientation detection [39].

Now, we approximate $\boldsymbol{y}(T_i(x))$ by Beta-Liouville distribution, $\boldsymbol{y}(T_i(x)) \sim \mathrm{BL}(\boldsymbol{\theta}_i)$ with $\boldsymbol{\theta}_i = (\alpha_{i1},\ldots,\alpha_{iK},\alpha_i,\beta_i)$. We start by estimating the parameters, the expressions are given by:

$$\begin{cases} \tilde{\alpha}_{ik,t} = \Psi^{-1}\left[\Psi\left(\sum_{m=1}^{K}\tilde{\alpha}_{im,t-1}\right) + \frac{1}{N}\sum_{j=1}^{N}\log c_{jk} \right. \\ \quad \left. -\frac{1}{N}\sum_{j=0}^{N-1}\log\left(\sum_{m=1}^{K}c_{jm}\right)\right] \quad k = 1\cdots K \\ \tilde{\alpha}_{i,t} = \Psi^{-1}\left[\Psi(\tilde{\alpha}_{i,t-1}+\tilde{\beta}_{i,t-1}) \right. \\ \quad \left. +\frac{1}{N}\sum_{j=1}^{N}\log\left(\sum_{k=1}^{K}c_{jk}\right)\right] \\ \tilde{\beta}_{i,t} = \Psi^{-1}\left[\Psi(\tilde{\alpha}_{i,t-1}+\tilde{\beta}_{i,t-1}) \right. \\ \quad \left. +\frac{1}{N}\sum_{j=1}^{N}\log\left(1-\sum_{k=1}^{K}c_{jk}\right)\right] \end{cases}$$

Using the same expression of the normality score used for the previous distributions, we get:

$$
\begin{aligned}
n_s(x) = \sum_{i=1}^{K} \log\left(\Gamma(\sum_{j=1}^{K}\tilde{\alpha}_{ij})\right) + \sum_{i=1}^{K}\log\Big(\Gamma(\tilde{\alpha}_i+\tilde{\beta}_i)\Big) \\
- \sum_{i=1}^{K}\log\left(\Gamma(\tilde{\alpha}_i)\right) - \sum_{i=1}^{K}\log\Big(\Gamma(\tilde{\beta}_i)\Big) + \sum_{i=1}^{K}\sum_{j=1}^{K}\tilde{\alpha}_{ij}\log[\boldsymbol{y}(T_i(x))]_j \\
- \sum_{i=1}^{K}\sum_{j=1}^{K}\log\left(\Gamma(\tilde{\alpha}_{ij})\right) + \sum_{i=1}^{K}(\tilde{\alpha}_i - \sum_{j=1}^{K}\tilde{\alpha}_{ij})\log\sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j \\
+ \sum_{i=1}^{K}(\tilde{\beta}_i - 1)\log\left(1 - \sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j\right) \qquad (33)
\end{aligned}
$$

where $\tilde{\alpha}_{ij}$ is the estimator of α_{ij}, $\tilde{\alpha}_i$ is the estimator of α_i and $\tilde{\beta}_i$ is the estimator of β_i. As we see, the expression of the score function is a bit complicated that's why we are going to simplify it in the same way as the previous two distributions. We remove all the terms that are independent of the observations (prediction vectors). So, the simplified expression of the score function for Beta-Liouville distribution is :

$$
\begin{aligned}
n_s(x) = \sum_{i=1}^{K}\sum_{j=1}^{K}\tilde{\alpha}_{ij}\log[\boldsymbol{y}(T_i(x))]_j + \sum_{i=1}^{K}(\tilde{\alpha}_i - \sum_{j=1}^{K}\tilde{\alpha}_{ij})\log\sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j \\
+ \sum_{i=1}^{K}(\tilde{\beta}_i - 1)\log\left(1 - \sum_{j=1}^{K}[\boldsymbol{y}(T_i(x))]_j\right) \qquad (34)
\end{aligned}
$$

By taking $\alpha = \sum_{k=1}^{K}\alpha_k$ and $\beta = \alpha_{K+1}$, the Beta-Liouville normality score in (34) is reduced to Dirichlet normality score (Equation (14)).

2.3 Experimental Results

In this section, we carry out experiments to investigate and demonstrate the performance of the proposed general forms of normality scores using the general forms of Dirichlet distributions. Our

suggested distributions' effectiveness is validated through image and non-image data. We demonstrate through experiments that they outperform the baseline methods such as One-Class Support Vector Machine (OC-SVM) as well as the standard form based on Dirichlet. Note that the technique used for evaluation in our experiments is the one-vs-all technique. It considers one class as an anomaly and the rest as normal categories.

2.3.1 Learning the Normality Score for Image Data

Implementation Details

In the first application, we investigate the proposed distributions to an anomaly detection problem in two image data sets. The first data set is CIFAR10 previously used in [63]. It contains 50000 training samples of 32×32 color images divided into 10 classes: *0:airplanes, 1:cars, 2:birds, 3:cats, 4:deer, 5:dogs, 6:frogs, 7:horses, 8:ships, and 9:trucks* with 5000 samples for each class, and 10000 samples for testing (1000 samples for each class). The second data is Fashion MNIST developed in [64]. As CIFAR10, it contains 50000 training 32×32 image samples partitioned equally over 10 categories: *0:T-shirt/top, 1:Trouser, 2:Pullover, 3:Dress, 4:Coat, 5:Sandal, 6:Shirt, 7:Sneaker, 8:Bag, 9:Ankle boot* and 10000 for testing.

We fix the number of geometric transformations $K = 72$. Due to the high number of transformed training samples ($50000 \times 72 = 3600000$), we choose to reduce the number of the training samples to 10000 (number of transformed samples, in this case, $10000 \times 72 = 720000$) and the number of the testing samples to 1000 (number of transformed samples $1000 \times 72 = 72000$), because training a deep learning classifier model on the whole data set costs in terms of time and hardware (powerful server). The distribution of samples over classes in training set and testing set for both data sets: CIFAR10 and Fashion MNIST are shown in Table 2.2.

Now, we show the effectiveness of the chosen classifier: Wide Residual Network. Although authors in [42] chose to work with 10 and 4 as depth and width parameters for the classifier, we select, in our work, these parameters to be 16 and 8, respectively as in [50] because the anomaly detection results improved by setting these values. It is noteworthy that the measure of the usefulness of the

Data Set	CIFAR10		Fashion MNIST	
Class	Train	Test	Train	Test
0	1005	103	942	107
1	974	89	1027	105
2	1032	100	1016	111
3	1016	103	1019	93
4	999	90	974	115
5	937	86	989	87
6	1030	112	1021	97
7	1001	102	1022	95
8	1025	106	990	95
9	981	109	1000	95

Table 2.2: distribution of samples over classes in the training set and testing set for both data sets: CIFAR10 and Fashion MNIST

proposed classifier should take into consideration the fact that we have balanced data (almost 1000×72 training samples for each class and almost 100×72 testing samples for each category). That's why, we can consider, in this case, the accuracy as a metric to measure the capability of the classifier in distinguishing the different transformations applied to the normal samples. Fig 2.2 shows the training accuracy for each class for both data sets. The batch size is fixed and equals 128. We can notice that the classifier fits well for all classes ranging from 88% to 99% for Fashion MNIST and from 92% to 100% for CIFAR10.

The initial values of the parameters for each distribution are shown in Table 2.3. More specifically,

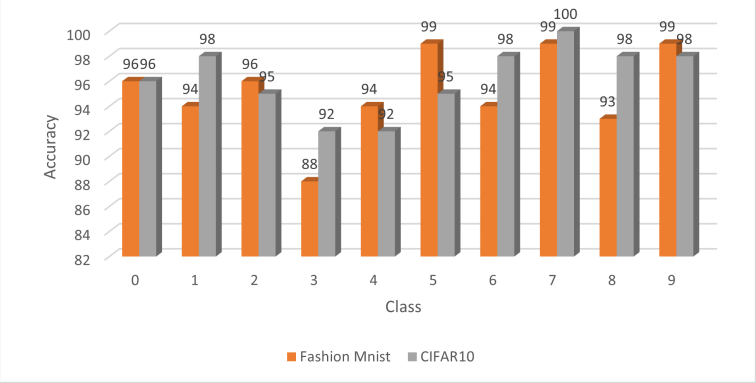


Figure 2.2: Performance of the WRN on both image data sets

they were chosen through two possible methods: 1) Use the Wicker Initialization [65] to estimate

Dirichlet's parameters using maximum likelihood approximation; 2) We fix them to a constant C. Usually, the second method is applied when the Wicker Initialization for estimating the parameter presents several iterations and operations to execute which can affect the time and the complexity. For Dirichlet and Beta-Liouville distributions, the estimation of the parameter vector $\boldsymbol{\alpha}$ is based on Wicker Initialization, while we fix the parameters α and β for Beta-Liouville to (α,β) = (0.1,0.1). Note that there is no initialization for the parameters of generalized Dirichlet, as their expressions are dependent on the covariance matrix of the observed data. This will decrease the complexity of the estimation procedure as well as the execution time. For scaled Dirichlet and shifted scaled Dirichlet distributions, Wicker Initialization didn't give us great results, therefore, we have investigated the following values for the initialization: $\boldsymbol{\alpha} = \boldsymbol{\beta} = (0.05, \cdots, 0.05)$ for scaled Dirichlet, and $\boldsymbol{\alpha} = \boldsymbol{p} = (0.2, \cdots, 0.2)$ and $a = 0.5$ for shifted scaled Dirichlet.

	$\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)$	$\boldsymbol{\beta} = (\beta_1, \cdots, \beta_K)$	α	β	$\boldsymbol{p}$	a
Dirichlet	Wicker Initialization	–	–	–	–	–
Generalized Dirichlet	No Initialization	No Initialization	–	–	–	–
Scaled Dirichlet	$(0.05, \cdots, 0.05)$	$(0.05, \cdots, 0.05)$	–	–	–	–
Shifted Scaled Dirichlet	$(0.2, \cdots, 0.2)$	–	–	–	$(0.2, \cdots, 0.2)$	0.5
Beta-Liouville	Wicker Initialization	–	0.1	0.1	–	–

Table 2.3: Parameters Initialization for each used distribution

Results

To demonstrate the merits of our proposed general forms of Dirichlet distribution in the construction of the normality score, they are compared with the standard Dirichlet distribution and other two baseline methods based on One Class Support Vector Machine (OCSVM) including Raw OCSVM and Convolutional Auto Encoder One Class Support Vector Machine (CAE OCSVM) [66],[67]. OCSVM model learns the boundary for the normal data samples so that it is able to classify the points outside the boundary as anomalies. The hyperparameters of OCSVM were set to this range of values $\nu \in \{0.1, \cdots, 0.9\}$ and $\gamma \in \{2^{-7}, 2^{-6}, \cdots, 2^2\}$. We report the normality score performance of different methods in Table 2.4 and Table 2.5 in terms of AUROC.

From the two tables, we notice that the Dirichlet distribution family (the standard form as well as

	RAW-OC-SVM	CAE-OC-SVM	Dirichlet	Generalized Dirichlet	Scaled Dirichlet	Shifted Scaled Dirichlet	Beta-Liouville
airplane	63.0	58.23±2.0	71.2±1.9	68.26±2.0	66.7±1.08	67.43±3.0	**76.76±3.6**
cars	50.0	51.4±0.5	88.73±1.7	**93.93±0.5**	93.6±0.2	93.2±0.4	93.46±1.2
bird	68.1	66.43±0.3	**73.5±0.7**	68.1±4.9	67.66±4.0	68.76±0.9	72±1.3
cat	54.3	51.53±0.3	**73.76±1.1**	68.93±0.9	70.03±2.29	67.4±1.0	63.53±1.2
deer	71.6	68.4±0.4	74.06±1.4	74.23±0.6	77.73±2.38	**81.73±0.9**	74.66±2.4
dog	50.0	58.66±0.4	**82.63**±2.4	78±1.6	78.83±3.06	74.46±2.1	79.6±3.1
frog	77.4	79.83±0.4	72.66±0.8	74.93±2.1	**81.76±7.16**	78.76±2.7	74.8±2.8
horse	52.2	55.33±0.8	90.06±1.6	89.66±0.1	90.86±0.63	90.1±1.8	**92.36±1.2**
ship	70.7	68.3±1.0	89.53±1.2	**91.03±0.8**	89.66±1.47	90.93±0.2	89.33±3.6
truck	52.4	60.56±1.1	85.2±0.5	88.83±0.8	**89.56±1.09**	89.1±0.4	86.6±2.0
mean	60.34	60.87	80.13	79.59	**80.64**	80.19	80.31

Table 2.4: Average AUC with standard deviation over 3 runs for CIFAR10 dataset

the general forms) outperforms the baseline methods, especially in the CIFAR10 dataset where the difference in AUC reaches 20% (60.34% for Raw OCSVM and 80.64% for scaled Dirichlet). This confirms (our assumptions) that data defined on simplex are better discriminated by Dirichlet distributions. Overall, the scaled Dirichlet distribution has the better AUC outperforming the rest of the methods whether in CIFAR 10 dataset (80.64%) or Fashion MNIST dataset (93.21%).

As shown in Table 2.4, for CIFAR10 dataset, the proposed general forms of Dirichlet outperform the standard one in 7 out of 10 classes. Generalized Dirichlet marks AUC = 93.93% for class 1 (automobile) and AUC = 91.03% for class 8 (ship) as the best scores compared to other models. The same thing for scaled Dirichlet and Beta-Liouville distributions, they have the best AUC scores for classes $6, 9$ (frog,truck) and $0, 7$ (airplane,horse) respectively.

Inspecting the results in Table 2.5, we notice that the baseline methods can perform much better on smaller size datasets (the size of an image in Fashion MNIST is 32×32×1, while the size in CIFAR10 is 32×32×3), reaching the best score for three classes (1:Trouser, 3:Dress, 7:Sneaker). However, the general forms excel in six classes (two classes by generalized Dirichlet, three classes by scaled Dirichlet, and one class by shifted scaled Dirichlet).

Another interesting point we can notice from Table 2.5 is the performance of Dirichlet distribution. Compared to other methods, the best performance for Dirichlet was reported in class 8 (Bag). Also, it was outperformed by the general forms (generalized Dirichlet, scaled Dirichlet, and shifted scaled Dirichlet) as well as the baseline methods (Raw-OC-SVM). To interpret this, in Fig2.3, we present

	RAW-OC-SVM	CAE-OC-SVM	Dirichlet	Generalized Dirichlet	Scaled Dirichlet	Shifted Scaled Dirichlet	Beta-Liouville
T-shirt/top	92.0	89.1±0.95	89.3±1.83	92.16±0.55	**95.33±0.5**	91.8±0.26	90.06±0.32
Trouser	**99.1**	97.1±0.55	97.86±0.63	98.7±0.55	96.83±1.5	96.6±1.3	98.53±0.73
Pullover	89.5	86.1±0.2	86.16±0.55	89.1±1.75	**93.06±0.37**	90.5±0.25	88.26±1.45
Dress	**92.0**	85.7±1.0	88.56±1.2	87.63±2.88	86.6±1.67	82.0±2.2	79.46±0.8
Coat	90.9	88.4±0.76	86.13±1.97	**91.46±0.45**	91.16±0.15	91.0±0.58	87.9±1.15
Sandal	93.2	92.8±1.2	96.16±0.9	**96.23±2.05**	95.73±0.96	96.2±0.75	93.33±3.14
Shirt	82.1	81.7±0.8	78.8±0.26	80.26±4.07	**83.9±0.72**	83.3±0.43	78.53±0.75
Sneaker	**98.5**	97.0±0.4	97.26±0.75	98.23±0.28	97.76±0.2	98.1±0.37	97.83±0.87
Bag	91.2	94.5±1.8	**97.73±0.66**	94.23±2.65	92.26±0.65	91.0±1.2	93.86±1.51
Ankle boot	98.2	96.2±0.7	99.33±0.05	**99.5±0.1**	99.46±0.11	**99.5±0.05**	99.1±0.6
mean	92.67	90.9	91.73	92.75	**93.21**	92.09	90.69

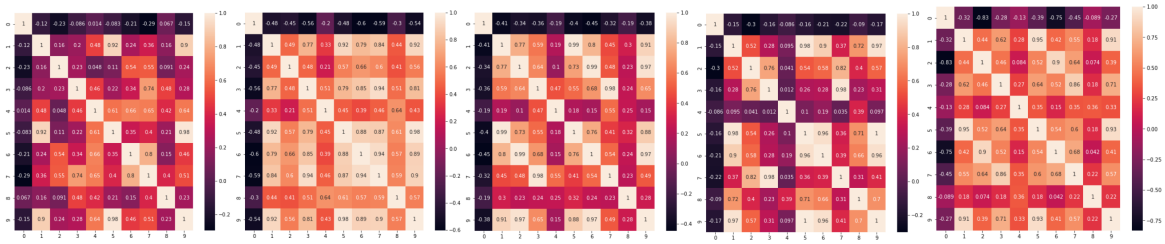
Table 2.5: Average AUC with standard deviation over 3 runs for Fashion MNIST dataset

the correlation matrices for the predicted vectors after each experiment. Note that in experiment i, the classifier and after being trained on the transformed form of samples of class i (class i is the normal class and anomalous are instances from other classes), predicts the softmax vectors for testing data being applied by all the transformations (Normally, we should analyze the CM of the predicted vectors of the testing data being transformed by all the transformations, but, we can't show all of them as we have 72 transformations, so we choose to show a random transformation which is the first transformation: transformation with index 0, note that each correlation matrix is 72×72, so we take the first 10×10 for display). From the correlation matrices, we see that the elements of the softmax output vector are highly correlated as well as are mostly positively related, which contracts the properties of Dirichlet, when we assume that it works better for independents random variables and negatively correlated random variables.

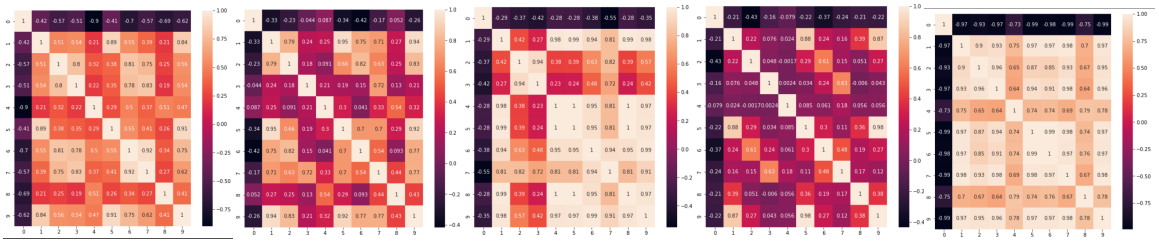
2.3.2 Learning the Normality Score for Non-image Data

Implementation Details

Now, we evaluate our extension work performance on non-image data. We choose to work with NSL-KDD Cup dataset [68]. It contains 125973 samples as train set and 22544 samples as test set. The names of the labels are *normal, neptune,back, land, pod, smurf, teardrop,mailbomb, apache2, processtable, udpstorm, worm, ipsweep, nmap, portsweep, satan ,mscan, saint, ftp-write,*



(a) CM for experiment 0 (b) CM for experiment 1 (c) CM for experiment 2 (d) CM for experiment 3 (e) CM for experiment 4



(f) CM for experiment 5 (g) CM for experiment 6 (h) CM for experiment 7 (i) CM for experiment 8 (j) CM for experiment 9

Figure 2.3: Correlation matrices of the first 10 elements of the softmax output vector obtained by 10 experiments

guess-passwd, imap, multihop, phf, spy, warezclient, warezmaster, sendmail, named ,snmpgetattack, snmpguess, xlock, xsnoop ,httptunnel, buffer-overflow, loadmodule, perl, rootkit, ps ,sqlattack, xterm. In order to simplify the work, we restrict the names of the labels to only two classes: we put all the labels that are different from the *normal* class into one class named *attack*. As a result, we have at the end two classes: normal class and attack class.

In this section, we build our own classifier. The summary of the classifier can be found in Table 2.6. For the transformations, we choose the number of transformations $K = 10$. So, the dimension of the hidden layer is in this set: $h\_dim \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. The autoencoder is established with a code size equal to 32.

The initialization of the parameters of all the distributions remains the same as in image data experiments.

Layer (type)	Output shape
conv1d (Conv1D)	(None, 114, 128)
max pooling1d	(None, 38, 128)
lstm (LSTM)	(None, 70)
dropout (Dropout)	(None, 70)
dense26 (Dense)	(None, 10)
Total params: 56,942	–
Trainable params: 56,942	–
Non-trainable params: 0	–

Table 2.6: Summary of the proposed classifier

Results

In this section, we demonstrate the effectiveness of using general forms of Dirichlet normality score in an anomaly detection problem for non-image data.

Table 2.7 illustrates the AUC results on NSL-KDD Cup data set by different general forms of Dirichlet distribution as well as the standard form. As shown in this table and except for the generalized Dirichlet, all the general forms outperform the standard one. The AUC reaches 84.59% for the scaled Dirichlet normality score, followed by shifted scaled Dirichlet with 79.05%. According to these results, we can see that Dirichlet and Beta-Liouville have the same performance for the Normal class, and the same thing for scaled Dirichlet and shifted scaled Dirichlet. For the Attack class, we notice that scaled Dirichlet marks a very high score compared to other distributions (87.4% with a difference of more than 5% from the nearest second score).

Another metric can be used especially when it comes to dealing with skewed data which is Area Under Precision-Recall. We can treat this metric in two different ways: the first is to calculate the score by considering the anomalies the positive class (AUPR-pos), and the second is to consider the anomalies of the negative class (AUPR-neg). Table 2.8 shows the different scores obtained by our methods for building the normality score. We can see the great performance of scaled Dirichlet in Attack class for both cases of AUPR. For normal class, the best score when anomalies are the positive class is marked by scaled Dirichlet normality score while Beta-Liouville outperforms all the other distributions in the case when anomalies are the negative class. Overall, scaled Dirichlet has the best scores compared to other methods. However, we can confirm that the other distributions succeeded too in creating good results ($> 75\%$). Fig 2.4 represents the different correlation

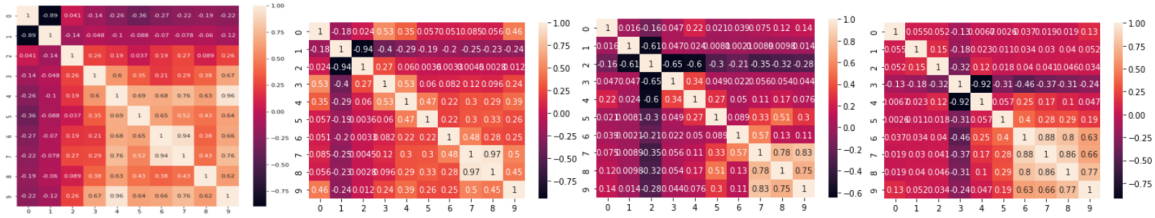
	Dirichlet	Generalized Dirichlet	Scaled Dirichlet	Shifted Scaled Dirichlet	Beta-Liouville
Normal	75.62	79.09	81.79	**82.12**	75.12
Attack	76.91	72.67	**87.4**	75.98	81.59
mean	76.26	75.88	**84.59**	79.05	78.35

Table 2.7: AUC for NSL-KDD Cup dataset

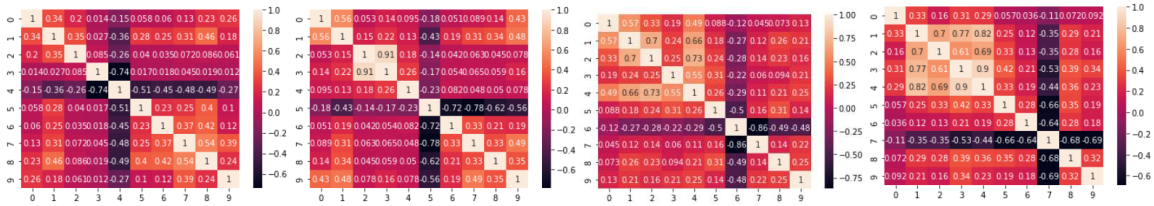
Distribution	Dirichlet		Generalized Dirichlet		Scaled Dirichlet		Shifted Scaled Dirichlet		Beta-Liouville	
Class	AUPR pos	AUPR neg	AUPR pos	AUPR neg	AUPR pos	AUPR neg	AUPR pos	AUPR neg	AUPR pos	AUPR neg
Normal	75.9	67.93	85.0	68.0	**85.42**	76.92	84.37	77.19	68.14	**77.23**
Attack	80.71	76.46	78.78	66.65	**87.15**	**87.25**	72.46	68.17	70.14	85.42
mean	78.3	72.19	81.91	67.32	**86.28**	**82.08**	78.41	72.68	69.14	81.32

Table 2.8: AUPR NSL-KDD Cup dataset

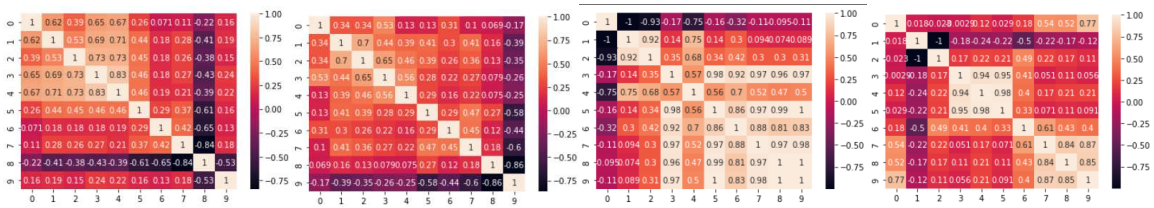
matrices of the predicted vectors by each transformation for both classes. We can notice that several elements are highly positively related which demonstrates the need to use the general forms of Dirichlet rather than Dirichlet.



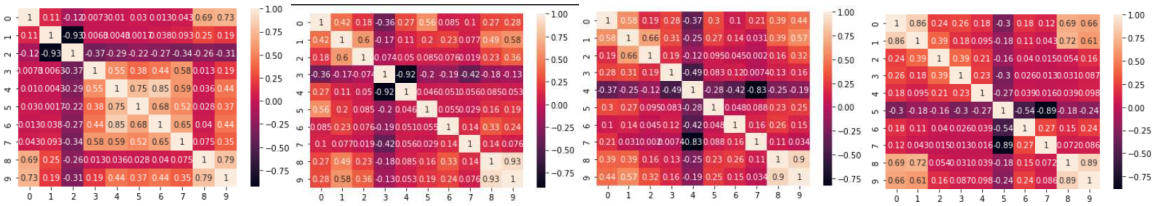
(a) CM for Class At-tack and Transforma-tion 0 (b) CM for Class At-tack and Transforma-tion 1 (c) CM for Class At-tack and Transforma-tion 2 (d) CM for Class At-tack and Transforma-tion 3



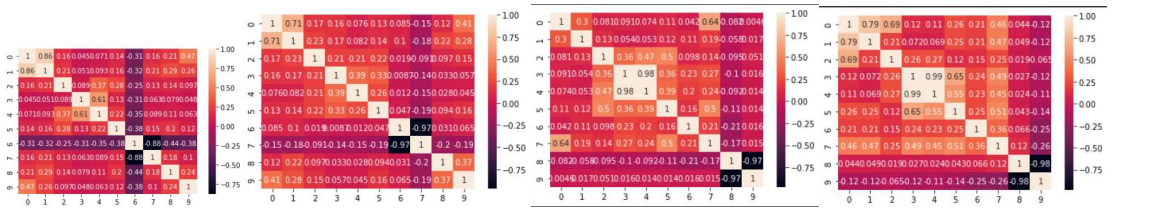
(e) CM for Class At-tack and Transforma-and Transformation 5 (f) CM for Class Attack tion 4 (g) CM for Class At-tack and Transforma-tion 6 (h) CM for Class At-tack and Transforma-tion 7



(i) CM for Class Attack and Transformation 8 (j) CM for Class Attack and Transformation 9 (k) CM for Class Nor-mal and Transforma-tion 0 (l) CM for Class Nor-mal and Transforma-tion 1



(m) CM for Class Nor-mal and Transforma-tion 2 (n) CM for Class Nor-mal and Transforma-tion 3 (o) CM for Class Nor-mal and Transforma-tion 4 (p) CM for Class Nor-mal and Transforma-tion 5



(q) CM for Class Nor-Normal and Trans-formation 6 (r) CM for Class Nor-mal and Transforma-tion 7 (s) CM for Class Nor-mal and Transforma-tion 8 (t) CM for Class Nor-mal and Transforma-tion 9

Figure 2.4: Correlation matrices for the softmax output vector obtained by 10 transformations

Chapter 3

Libby-Novick Beta-Liouville Distribution for Enhanced Anomaly Detection in Proportional Data

In this chapter, we focus on development the Libby-Novick Beta-Liouville distribution. We investigate it in three different methods: Finite mixture models, normality scores, and feature mapping in SVM. Comparisons with comparable recent approaches have shown the worth of our proposed distribution.

3.1 Libby-Novick Beta-Liouville Distribution

A K-dimensional vector $\boldsymbol{X}$ follows a Liouville distribution with parameters (α_1,$\cdots$,α_K) and density generator $g(.)$ if its pdf (probability density function) is defined by [39] [60]:

$$p(\boldsymbol{X}|\alpha_1,\cdots,\alpha_K) = g(u)\prod_{i=1}^{K}\frac{X_i^{\alpha_i-1}}{\Gamma(\alpha_i)} \tag{35}$$

where $u = \sum_{i=1}^{K} X_i < 1$, and $0 < X_i < 1, i = 1,\cdots,K$. One common choice of the generator function is:

$$g(u) = \frac{\Gamma(\sum_{i=1}^{K}\alpha_i)}{u^{\Gamma(\sum_{i=1}^{K}\alpha_i-1)}}f(u) \tag{36}$$

where $f(.)$ is the pdf of the variable u, as a result, we can obtain a new expression of the pdf of Liouville distribution:

$$p(\boldsymbol{X}) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)}{u^{\sum_{i=1}^{K} \alpha_i - 1}} f(u) \prod_{i=1}^{K} \frac{X_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{37}$$

A convenient choice as a distribution for u is Beta distribution which can approximate any arbitrary distribution thanks to its two shape parameters [69]. However, in this context, Libby-Novick Beta (LNB) [9], which is an extended form of the Beta distribution, contains more shape parameters than the ordinary version, it has three shape parameters. Therefore, it can fit data with more flexibility. Because of the additional feature, LNB can accurately represent skewness and kurtosis in data, especially when modeling real-world data [70]. Added to that, the third additional shape parameter modifies tail weights and increases the produced distribution's entropy. In our work, we choose LNB as a distribution for modeling the random variable u, its pdf is given by [9]:

$$f(u|\alpha, \beta, \lambda) = \frac{\lambda^{\alpha} u^{\alpha-1} (1-u)^{\beta-1}}{B(\alpha, \beta)(1-(1-\lambda)u)^{\alpha+\beta}} \tag{38}$$

with:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{39}$$

represents the Beta function and $\Gamma(.)$ denotes the Gamma function. Note that when $\lambda = 1$, LNB is reduced to standard Beta with shape parameters α and β.

We obtain the expression of the pdf for our proposed distribution for work, which is the Libby-Novick Beta-Liouville distribution, by using the Libby-Novick Beta as the density function for u in Eq(36) and injecting Eq(38) in Eq(37).

$$p(\boldsymbol{X}|\alpha_1, \cdots, \alpha_K, \alpha, \beta, \lambda) = \frac{\Gamma(\sum_{i=1}^{K} \alpha_i)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\lambda^{\alpha} (\sum_{i=1}^{K} X_i)^{\alpha - \sum_{i=1}^{K} \alpha_i} (1 - \sum_{i=1}^{K} X_i)^{\beta-1}}{(1-(1-\lambda)\sum_{i=1}^{K} X_i)^{\alpha+\beta}} \prod_{i=1}^{K} \frac{X_i^{\alpha_i - 1}}{\Gamma(\alpha_i)} \tag{40}$$

Figure(3.1) displays some examples of Libby-Novick Beta-Liouville distribution for different parameters.

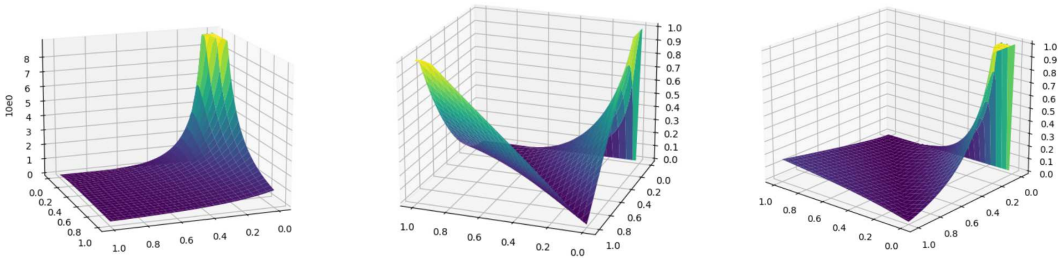


Figure 3.1: Examples of Libby-Novick Beta-Liouville distribution

When the density generator function has a LNB distribution with parameters $\sum_{k=1}^{K} \alpha_k$, α_{K+1}, and 1:

$$f(u|\alpha, \beta, \lambda) = \frac{u^{\sum_{k=1}^{K} \alpha_k - 1}(1-u)^{\alpha_{K+1}-1}}{B(\alpha, \beta)} \tag{41}$$

Eq(40) is is reduced to the Dirichlet distribution with parameters $\alpha_1, \cdots, \alpha_K, \alpha_{K+1}$. We can confirm that Dirichlet is a special case of Libby-Novick Beta-Liouville distribution.

3.2 Libby-Novick Beta-Liouville Finite Mixture Models

Let $\mathcal{X} = (\boldsymbol{X_1}, \cdots, \boldsymbol{X_N})$ a set of training examples where N is the number of samples. Each sample $X_i (i = 1, \cdots, N)$ can be represented by finite mixture of distributions:

$$p(\boldsymbol{X_i}|\Theta) = \sum_{j=1}^{M} p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j}) \tag{42}$$

where $\Theta = (p_1, \cdots, p_M, \boldsymbol{\theta_1}, \cdots, \boldsymbol{\theta_M})$ is the set of parameters, $p_j > 0, j = 1, \cdots, M$, $\sum_{j=1}^{M} p_j = 1$, and M is the number of components. As a generative model, finite mixture allows the generation of a vector by selecting a component density j with probability p_j, and then creating a vector from that distribution $p(\boldsymbol{X}|\boldsymbol{\theta_j})$. However, deploying finite mixture models remains challenging. The main challenge is the choice of the appropriate distribution of each component $p(X|\theta_j)$ that should fit well the data. Numerous studies have been conducted on techniques for choosing the distribution and estimating the parameters of mixture models [14],[39], etc. In our work, we adopt an Expectation-Maximization (EM) algorithm for Libby-Novick Beta-Liouville mixture models.

When dealing with mixture models, an important task that must be achieved is the estimation of the associated parameters. In this paper, we choose the Maximum Likelihood approach (ML), which consists of estimating the parameter values that maximize the likelihood function: $\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, P(\mathcal{X}|\Theta)$, where $P(\mathcal{X}|\Theta)$ is the likelihood function:

$$P(\mathcal{X}|\Theta) = \prod_{i=1}^{N}\sum_{j=1}^{M} p_j \frac{\Gamma(\sum_{k=1}^{K}\alpha_{jk})\Gamma(\alpha_j+\beta_j)}{\Gamma(\alpha_j)\Gamma(\beta_j)} \frac{\lambda_j^{\alpha_j}(\sum_{k=1}^{K}X_{ik})^{\alpha_j-\sum_{k=1}^{K}\alpha_{jk}}(1-\sum_{k=1}^{K}X_{ik})^{\beta_j-1}}{(1-(1-\lambda_j)\sum_{k=1}^{K}X_{ik})^{\alpha_j+\beta_j}} \prod_{k=1}^{K}\frac{X_{ik}^{\alpha_{jk}-1}}{\Gamma(\alpha_{jk})} \quad (43)$$

ML can be combined with the EM algorithm to estimate the mixture parameters through an iterative process [71]. During the Expectation step, we define $Z_i = (Z_{i1}, \cdots, Z_{iM})$ a multi-Bernoulli distributed random vector, to indicate the cluster of the vector $\boldsymbol{X_i}$: $\boldsymbol{Z_{ij}} = 1$ if $\boldsymbol{X_i}$ belongs to class j, 0 otherwise. By the end of the Expectation step, we use the expected values of the class assignments with are the posterior probabilities ($\hat{Z}_{ij} = \frac{p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j})}{\sum_{j=1}^{M} p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j})}$) to replace the missing data $Z = (Z_1, \cdots, Z_N)$.

The Maximization step is used to update the estimates of the parameters. The idea consists of maximizing the following expression:

$$L(\mathcal{X}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M}\hat{Z}_{ij}\log(p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j})) \quad (44)$$

The expressions of the first derivatives of $L(\mathcal{X}|\Theta)$ with respect to the mixture components parameters are the following:

$$\frac{\partial L(\mathcal{X}|\Theta)}{\partial \alpha_j} = (\Psi(\alpha_j+\beta_j)-\Psi(\alpha_j))\sum_{i=1}^{N}\hat{Z}_{ij} + \sum_{i=1}^{N}\hat{Z}_{ij}\left(\log(\lambda_j) + \log(\sum_{k=1}^{K}X_{ik}) + \log(1-(1-\lambda_j)\sum_{k=1}^{K}X_{ik})\right) \quad (45)$$

$$\frac{\partial L(\mathcal{X}|\Theta)}{\partial \beta_j} = (\Psi(\alpha_j + \beta_j) - \Psi(\beta_j)) \sum_{i=1}^{N} \hat{Z}_{ij} + \sum_{i=1}^{N} \hat{Z}_{ij} \left(\log(1 - \sum_{k=1}^{K} X_{ik}) \right.$$

$$\left. - \log(1 - (1 - \lambda_j) \sum_{k=1}^{K} X_{ik}) \right) \quad (46)$$

$$\frac{\partial L(\mathcal{X}|\Theta)}{\partial \lambda_j} = \frac{\alpha_j}{\lambda_j} \sum_{i=1}^{N} \hat{Z}_{ij} - (\alpha_j + \beta_j) \sum_{i=1}^{N} \hat{Z}_{ij} \frac{\sum_{k=1}^{K} X_{ik}}{(1 - (1 - \lambda_j) \sum_{k=1}^{K} X_{ik})} \quad (47)$$

$$\frac{\partial L(\mathcal{X}|\Theta)}{\partial \alpha_{jk}} = \left(\Psi(\sum_{k=1}^{K} \alpha_{jk}) - \Psi(\alpha_{jk}) \right) \sum_{i=1}^{N} \hat{Z}_{ij} + \sum_{i=1}^{N} \hat{Z}_{ij} \left(\log(X_{ik}) - \log(\sum_{k=1}^{K} X_{ik}) \right) \quad (48)$$

where $\Psi(x) = \frac{\partial \log \Gamma(x)}{\partial x}$ denotes the digamma function. From the previous equations, we notice that it is evident that the θ_j parameters lack a closed-form solution. Therefore, and starting from a given set of initial estimates, we adopt Newton-Raphson method to estimate these parameters.

$$\boldsymbol{\theta_j}^{(t+1)} = \boldsymbol{\theta_j}^{(t)} - H(\boldsymbol{\theta_j}^{(t)})^{-1} \frac{\partial L(\mathcal{X}|\Theta^{(t)})}{\partial \boldsymbol{\theta_j}^{(t)}} \quad (49)$$

where $H(\boldsymbol{\theta_j}^{(t)})^{-1}$ is the inverse of the hessian matrix. More details about how to calculate the inverse of the hessian matrix can be found in Appendix A. For the p_j, a closed form expression can be given by:

$$p_j = \frac{1}{N} \sum_{i=1}^{N} \hat{Z}_{ij} \quad (50)$$

We decided to initialize the parameters in our work with random values drawn from a uniform distribution. Furthermore, if we randomly start with values within the specified range (a, b), we are less likely to find ourselves trapped in local minima or maxima. This uncertainty might facilitate early investigation of a wider search field and result in better solutions.

Once the estimation equations are in hand, the complete learning algorithm can be found in Algorithm 1:

Algorithm 1 Expectation Maximization Algorithm

1: Initialization
2: Expectation step: Compute the posterior probabilities: $\hat{Z}_{ij} = \frac{p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j})}{\sum_{j=1}^{M} p_j p(\boldsymbol{X_i}|\boldsymbol{\theta_j})}$
3: Maximization step:
4: a) Update $\boldsymbol{\theta_j}$ using Eq(49), j=1,$\cdots$,M
5: b) Update p_j using Eq(50), j=1,$\cdots$,M
6: Repeat Steps 2 and 3 until convergence

3.3 Libby-Novick Beta-Liouville Normality Score

The problem statement as well as the proposed procedure are the same as described in section 2.1. In this section, we approximate $\boldsymbol{y}(T_i(x))$ with Libby-Novick Beta-Liouville distribution: $\boldsymbol{y}(T_i(x)) \sim \text{LNBL}(\boldsymbol{\theta}_i)$ with $\boldsymbol{\theta}_i = (\alpha_{i1}, \cdots, \alpha_{iK}, \alpha_i, \beta_i, \lambda_i)$. Injecting the expression of LNBL pdf in Eq(40) into the normality scores expression in Eq(1), we obtain the following expression of $n_s(x)$:

$$\begin{aligned} n_s(x) = &\sum_{i=1}^{K} \log\left(\Gamma(\sum_{k}^{K} \tilde{\alpha}_{ik})\right) + \sum_{i=1}^{K} \log\left(\Gamma(\tilde{\alpha}_i + \tilde{\beta}_i)\right) - \sum_{i=1}^{K} \log\left(\Gamma(\tilde{\alpha}_i)\right) - \sum_{i=1}^{K} \log\left(\Gamma(\tilde{\beta}_i)\right) \\ &+ \sum_{i=1}^{K} \tilde{\alpha}_i \log(\tilde{\lambda}_i) + \sum_{i=1}^{K} \left(\tilde{\alpha}_i - \sum_{k=1}^{K} \tilde{\alpha}_{ik}\right) \log\left(\sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) \\ &+ \sum_{i=1}^{K} (\tilde{\beta}_i - 1) \log\left(1 - \sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) - \sum_{i=1}^{K} (\tilde{\alpha}_i + \tilde{\beta}_i) \log\left(1 - (1 - \tilde{\lambda}_i) \sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) \\ &+ \sum_{i=1}^{K} \sum_{k=1}^{K} (\tilde{\alpha}_{ik} - 1) \log\left([\boldsymbol{y}(T_i(x))]_k\right) - \sum_{i=1}^{K} \sum_{k=1}^{K} \log\left(\Gamma(\tilde{\alpha}_{ik})\right) \end{aligned} \quad (51)$$

where $\tilde{\alpha}_{ik}$ is the estimator of α_{ik}, $\tilde{\alpha}_i$ is the estimator of α_i, $\tilde{\beta}_i$ is the estimator of β_i, and $\tilde{\lambda}_i$ is the estimator of λ_i. By eliminating all the terms that are independent of the sample values, we can simplify the expression of $n_s(x)$ to:

$$\begin{aligned} n_s(x) = &\sum_{i=1}^{K} \left(\tilde{\alpha}_i - \sum_{k=1}^{K} \tilde{\alpha}_{ik}\right) \log\left(\sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) + \sum_{i=1}^{K} (\tilde{\beta}_i - 1) \log\left(1 - \sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) \\ &- \sum_{i=1}^{K} (\tilde{\alpha}_i + \tilde{\beta}_i) \log\left(1 - (1 - \tilde{\lambda}_i) \sum_{k=1}^{K} [\boldsymbol{y}(T_i(x))]_k\right) + \sum_{i=1}^{K} \sum_{k=1}^{K} (\tilde{\alpha}_{ik} - 1) \log\left([\boldsymbol{y}(T_i(x))]_k\right) \end{aligned} \quad (52)$$

Using the Fixed Point Iteration, the expressions of the estimated parameters are the following at

iteration t:

$$\tilde{\alpha}_{ik,t} = \Psi^{-1}\left[\Psi(\sum_{j=1}^{K}\tilde{\alpha}_{jk,t-1}) + \frac{1}{N}\sum_{j=1}^{N}\log(c_{jk}) - \frac{1}{N}\sum_{j=1}^{N}\log(\sum_{m=1}^{K}c_{jm})\right] \quad k = 1\cdots K \tag{53}$$

$$\tilde{\alpha}_{i,t} = \Psi^{-1}\left[\Psi\left(\tilde{\alpha}_{i,t-1} + \tilde{\beta}_{i,t-1}\right) + \frac{1}{N}\sum_{j=1}^{N}\log\left(\sum_{k=1}^{K}c_{jk}\right)\right.$$

$$\left. - \frac{1}{N}\sum_{j=1}^{N}\log\left(1-(1-\tilde{\lambda}_{i,t-1})\sum_{k=1}^{K}c_{jk}\right)\right] \tag{54}$$

$$\tilde{\beta}_{i,t} = \Psi^{-1}\left[\Psi\left(\tilde{\alpha}_{i,t-1} + \tilde{\beta}_{i,t-1}\right) + \frac{1}{N}\sum_{j=1}^{N}\log\left(1-\sum_{k=1}^{K}c_{jk}\right)\right.$$

$$\left. - \frac{1}{N}\sum_{j=1}^{N}\log\left(1-(1-\tilde{\lambda}_{i,t-1})\sum_{k=1}^{K}c_{jk}\right)\right] \tag{55}$$

$$\tilde{\lambda}_{i,t} = \frac{N\tilde{\alpha}_{i,t-1}}{\sum_{j=1}^{N}\left[\left(\tilde{\alpha}_{i,t-1} + \tilde{\beta}_{i,t-1}\right)\frac{\sum_{k=1}^{K}c_{jk}}{1-(1-\tilde{\lambda}_{i,t-1})\sum_{k=1}^{K}c_{jk}}\right]} \tag{56}$$

with $\boldsymbol{C} = (c_{jk})_{j=1\cdots N, k=1\cdots K}$ is the matrix where the j^{th} raw equals to $\boldsymbol{y}(T_i(x_j))$, x_j is the sample j in the normal samples set $\boldsymbol{X}$.

By setting the values of the estimated parameters of α_i, β_i, and λ_i to $\sum_{k=1}^{K}\alpha_{ik}$, α_{K+1}, and 1 respectively, we get this expression of $n_s(x)$:

$$n_s(x) = \sum_{i=1}^{K}\left[\log\Gamma(\sum_{k=1}^{K}\tilde{\alpha}_{ik}) - \sum_{k=1}^{K}\log\Gamma(\tilde{\alpha}_{ik}) + \sum_{k=1}^{K}(\tilde{\alpha}_{ik}-1)\log[\boldsymbol{y}(T_i(x))]_k\right] \tag{57}$$

The expression in Eq(57) is Dirichlet normality score [42]. This indicates that we are generalizing the normality score from Dirichlet normality scores to Libby-Novick Beta-Liouville normality

scores.

3.4 Libby-Novick Beta-Liouville Feature Mapping in SVM

3.4.1 Support Vector Machines Classifier

A well-known and popular option for the supervised learning is SVM. Empirically, it has demonstrated strong generalization abilities in a variety of investigations and applications [72],[73]. Finding the best decision boundaries that optimize the margin between several classes in a dataset is particularly successful with SVMs [74]. The primal representation of the SVM optimization problem is given by:

$$\min_{\boldsymbol{w},b,\epsilon} \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_i \epsilon_i \tag{58}$$

subject to

$$y^{(i)}(w^t \phi(\boldsymbol{X_i}) + b) > 1 - \epsilon_i, i = 1 \cdots, N \tag{59}$$

$$\epsilon_i > 0, i = 1 \cdots, N \tag{60}$$

where N is the number of samples, $\boldsymbol{X_i}$ is the normalized version of the i^{th} sample in the data set, and y_i is the corresponding label.

The hyperparameter C controls the regularization strength in SVM. By adding a penalty to the misclassifying data points, it reduces the overfitting of the model. It manages the trade-off between minimizing the classification error (greater complexity) and getting a wider margin (lower complexity). In (59), $\phi(\boldsymbol{X_i})$ is the feature mapping function from the input space χ to the feature space H. It is equal to the input data if no extra features are added or extracted from the data. In this case, the kernel K which is the inner product between data points, becomes $< \boldsymbol{X_i}, \boldsymbol{X_j} >$ instead of $< \phi(\boldsymbol{X_i}), \phi(\boldsymbol{X_j}) >$. $\sum_i \epsilon_i$ is the upper bound of the generalization error, where the slack variable ϵ_i is given to solve the case of non-lineary separated data. More details about the properties of the primal representation of the optimization problem of SVM can be found in [47]. For large datasets, it is computationally convenient to solve the dual problem. Using Lagrange multipliers to loosen

the restrictions, the dual solution becomes,

$$\max_{\gamma} \sum_{i}^{N} \gamma_i - \frac{1}{2} \sum_{i}^{N} \sum_{j}^{N} \gamma_i \gamma_j y^{(i)} y^{(j)} < \phi(\boldsymbol{X_i}), \phi(\boldsymbol{X_j}) > \tag{61}$$

subject to

$$0 < \gamma_i < C, \ \sum_{i}^{N} \gamma_i y^{(i)} = 0, i = 1 \cdots, N \tag{62}$$

In this case, the decision function of SVM becomes:

$$f(\boldsymbol{X}) = \sum_{i}^{N} \gamma_i y^{(i)} < \phi(\boldsymbol{X_i}), \phi(\boldsymbol{X}) > \tag{63}$$

3.4.2 Libby-Novick Beta-Liouville SVM Feature Mapping Function

In this section, we focus on how to choose the feature mapping function to solve both the primal and dual problems given respectively in (58) and (61). For better modeling, the choice of $\phi(X)$ depends on the structure of the data. Taking the advantage of Libby-Novick Beta-Liouville distribution in modeling proportional data, a possible feature mapping function can be structured as follows:

$$\phi_j(\boldsymbol{X_i}) = \begin{cases} X_{ij}, \quad j = 1, \cdots, K \\ \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\lambda^{\alpha} (\sum_{k=1}^{K} X_{ik})^{\alpha - \sum_{k=1}^{K} \alpha_k} (1 - \sum_{k=1}^{K} X_{ik})^{\beta - 1}}{(1-(1-\lambda)\sum_{k=1}^{K} X_{ik})^{\alpha+\beta}} \prod_{k=1}^{K} \frac{X_{ik}^{\alpha_k - 1}}{\Gamma(\alpha_k)}, \ j = K+1 \end{cases} \tag{64}$$

From the proposed feature mapping function in (64), we notice that the input dimension is increased by 1. The added feature introduces diversity in the data set. Added to that, it provides additional details regarding the overall distribution of the original features. This new representation knowledge can be useful for categorizing aberrant samples. Using the initial input data set, we can use Newton Raphson's method to estimate the parameters in a similar way as in [46]. More details can be found in Appendix B. We obtained a new formulation of Libby-Novick Beta-Liouville SVM (LNBL

SVM), after getting the trained parameters, as follows:

$$\min_{w,b,\epsilon} \frac{1}{2} \sum_{k}^{K+1} w_k^2 + C \sum_{i}^{N} \epsilon_i \tag{65}$$

subject to

$$y^{(i)}(w^t \phi(\boldsymbol{X_i}) + b) > 1 - \epsilon_i, i = 1 \cdots, N \tag{66}$$

$$\epsilon_i > 0, i = 1 \cdots, N \tag{67}$$

A new expression of LNBL SVM can be formulated as follows:

$$\min_{w,b,\epsilon} \frac{1}{2} \sum_{k}^{K+1} w_k^2 + C \sum_{i}^{N} \epsilon_i \tag{68}$$

subject to

$$y^{(i)}(\sum_{k=1}^{K} w_k X_{ik} + w_{K+1} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\lambda^{\alpha}(\sum_{k=1}^{K} X_{ik})^{\alpha - \sum_{k=1}^{K} \alpha_k}(1 - \sum_{k=1}^{K} X_{ik})^{\beta-1}}{(1-(1-\lambda)\sum_{k=1}^{K} X_{ik})^{\alpha+\beta}} \prod_{k=1}^{K} \frac{X_{ik}^{\alpha_k - 1}}{\Gamma(\alpha_k)} + b) > 1 - \epsilon_i, i = 1 \cdots, N \tag{69}$$

$$\epsilon_i > 0, i = 1 \cdots, N \tag{70}$$

3.5 Results

In this section, we conduct experiments to look into and gauge how well our suggested distribution works for the various proposed approaches. We demonstrate through the implementation results that the Libby-Novick Beta-Liouville outperforms the typical distribution used for proportional data such as Dirichlet, Beta-Liouville, and Multivariate Beta.

3.5.1 Data Sets

Our evaluation was based on five different data sets: three image data sets and two non-image data sets. For the image data, the first two data sets are CIFAR10 and fashion MNIST described in section2.2.The third one is the MNIST data set [75], a popular handwritten digit data set. It has 50000 training samples (5000 samples for each digit from zero to nine) and 10000 testing images. For non-image data, a widely used data set for the task of anomaly detection is the NSL-KDD Cup data set [68] described in section2.3.2. The last data set is bank additional full which is related to direct marketing campaigns of a Portuguese banking institution [76]. It has 41188 samples with two possible labels: *yes* and *no*, to indicate whether the client will subscribe to a term deposit or not. The statistics of the data sets are summarized in Table 3.1.

Data Set	Dimensions	Instances
CIFAR 10	784	60000
Fashion MNIST	784	60000
MNSIT	784	60000
NSL-KDD Cup	41	148517
Bank data	20	41188

Table 3.1: Data Sets Summary

Given that each image dataset contains 784 features for each sample, it presents a significant computational challenge. Principal Component Analysis (PCA) is used to lessen this by reducing the feature dimensionality to 30, which makes the process of anomaly identification easier to handle.

3.5.2 Mixture Models Results

In this section, our assessment is predicated on three subsets of data built from the NSL-KDD data set. Each one of them contains only two classes. The first one is established by considering the samples belonging to *normal* class as the normal samples (67343 samples) and the samples belonging to *Neptune* class as the anomaly samples (41212 samples), therefore, the anomaly rate in this case is 37%. Using an identical methodology, the second one establishes the anomaly rate in

this instance as 5%. This is achieved by classifying the samples from the *normal* class as normal samples (67343 samples) and the samples from the *Satan* class as anomaly samples (3633 samples). The third one is by taking the *NMAP* class (1493) as the outlier category, making the anomaly rate equal to 2%. Furthermore, we set the number of components $M = 2$.

Fig(3.2) displays an overview of the anomaly detection outcomes for the three subsets as determined by the accuracy and f1 score values. In this figure, we present results obtained by One-Class Support Vector Machines (OC-SVM), by setting the contamination parameter to the anomaly rate, and Isolation Forest (IF), using the outlier rate as the ν parameter of IF. Additionally, we show the results of generative finite mixture models with different distributions, namely Libby-Novick Beta-Liouville mixture models (LNBLMM), Beta-Liouville mixture models (BLMM), Dirichlet mixture models (DMM), Multivariate Beta mixture models (MBMM), and Gaussian mixture models (GMM).

We can observe that LNBLMM outperforms the other methods for low anomaly rates (2% and 5%). For 2% and 5% outlier rates, respectively, LNBLMM achieves 99.65 and 99.04 as f1 scores, and 99.32 and 98.23 as accuracy. Furthermore, we observe that DMM, BLMM, and LNBLMM perform noticeably better than GMM overall. The reason behind this is that the Dirichlet family distributions based on our generative models are highly relevant options because the resultant vectors following normalization are defined on the simplex with a unit-sum constraint. For this reason, the Dirichlet distribution and two more flexible generalizations, the Beta-Liouville and the Libby-Novick Beta-Liouville distributions, are among the best choices. For high anomaly rates, GMM has the highest scores, outperforming the other generative models which themselves perform better than OC-SVM and IF.

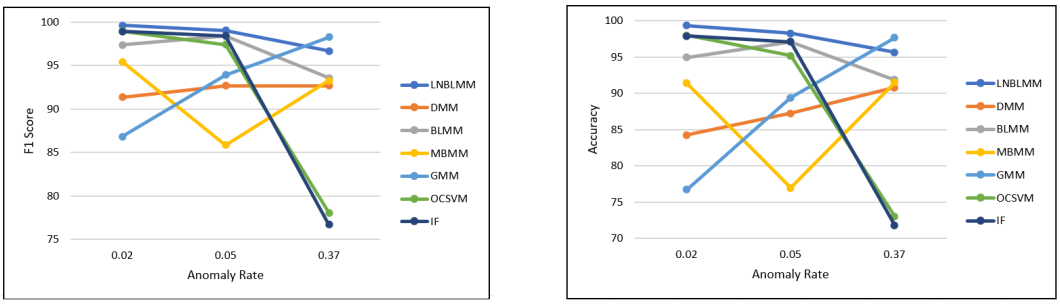


Figure 3.2: F1 score and Accuracy for the three subsets built from NSL KDD Data Set

The confusion matrices of LNBLMM, BLMM, and DMM for the two scenarios—when *nmap* is

the outlier category (Anomaly rate = 2%) and *satan* is the anomaly class (Anomaly rate = 5%)—are shown in Figs. (3.3) and (3.4). Inspecting the matrices, we notice that LNBLMM and BLMM score better performance than DMM in detecting anomalies and classifying correctly the normal samples. This can be assumed by the higher values of True Positive (TP) and True Negative (TN) for LNBLMM and BLMM compared to DMM. Between LNBLMM and BLMM, we can affirm that both of them have almost the same performance in detecting anomalies reaching 0 as False Positive (FP) for LNBLMM and 1 as FP for BLMM, in the case where the anomaly rate is set to 2%. For classifying the normal samples, LNBLMM has higher values of TP for both cases (2% and 5%) compared to BLMM (TP=9645 for LNBLMM to 9217 for BLMM in case of 2% anomaly, and TP=9541 for LNBLMM to 9405 for BLMM in case of 5% anomaly).

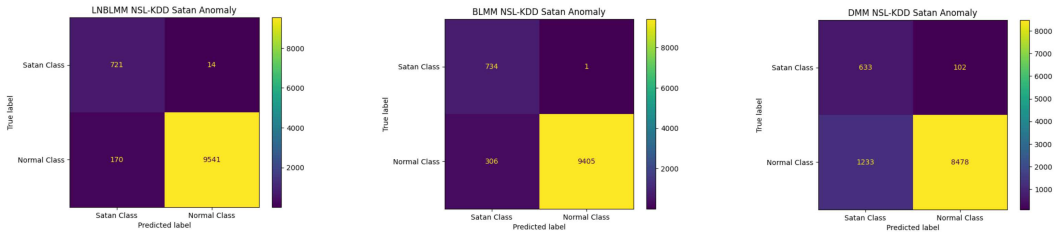


Figure 3.3: Confusion Matrices in case where *Satan* is the anomaly class (Anomaly rate = 5%)

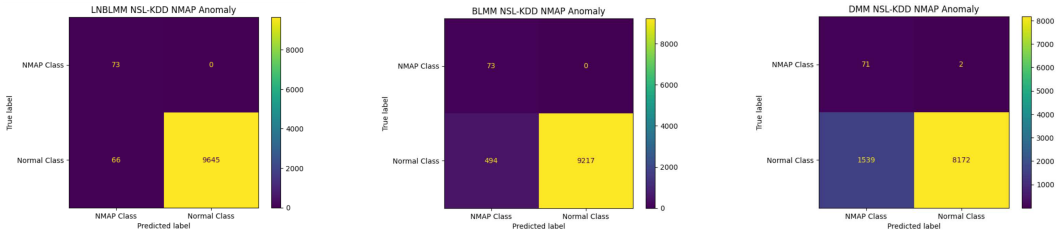


Figure 3.4: Confusion Matrices in case where *NMAP* is the anomaly class (Anomaly rate = 2%)

Table 3.2 shows the results of the different generative models as well as OCSVM and IF for the bank data. It is clear that LNBLMM excels over the other approaches, reaching 98.89 as the f1 score and 98.16 as accuracy. The performance between LNBLMM and DMM is remarkably significant whereas the results between LNBLMM and BLMM are less noticeable. This affirms the quality of

the LNBL and BL distributions in capturing the feature data distribution.

Data Set	Bank Data	
Approach	F1 Score	Accuracy
LNBLMM	**98.89**	**98.16**
DMM	89.16	82.97
BLMM	95.64	92.5
MBMM	85.29	78.24
GMM	90.31	82.33
OCSVM	94.13	90.59
IF	87.11	81.2

Table 3.2: F1 score and Accuracy for different approaches on Bank Data Set

3.5.3 Normality Scores Results

In this section, we implement our proposed Libby-Novick Beta-Liouville (LNBL) normality scores approach on the NSL-KDD Cup as well as the three image data sets: Fashion MNIST, MNIST, and CIFAR10. Regarding the NSL-KDD Cup dataset, and to make the task easier, we limit the label names to only two classes: we group all the labels that differ from the *normal* class into one class namely the *attack* class. Consequently, we have two labels at the end: the *normal* class and the *attack* class. We evaluate its efficacy by contrasting our LNBL distribution-based normality scores with Dirichlet and Beta-Liouville distributions-based normality scores, and two baseline approaches, namely Raw One-Class Support Vector Machine (RAW-OCSVM) and Convolutional AutoEncoder One-Class Support Vector Machine (CAE OCSVM) [66],[67].

The chosen classifier for our architecture is Wide Residual Networks (WRN) [50]. We set its depth and width to 16 and 8 respectively. Furthermore, the hyperparameters of OCSVM were adjusted to $\gamma \in \left\{2^{-7}, 2^{-6}, \cdots, 2^{2}\right\}$ and $\nu \in \{0.1, \cdots, 0.9\}$. We can initialize the parameters in one of two ways when it comes to estimation:1) Apply the maximum likelihood to the initialization suggested by Wicker in [65]; 2) Fix them to a constant C. In our study, we fix the starting values of LNBL to $(\alpha_1, \cdots, \alpha_K, \alpha, \beta, \lambda) = (0.05, \cdots, 0.05, 1.4, 1.2, 1.6)$ and initialize the Dirichlet and Beta-Liouville parameters using Wicker initialization. Lastly, we set $K = 72$ as the number of transformations.

We show the normality scores performance of several the techniques in terms of AUROC for image

data sets in Tables 3.3, 3.4, and 3.5. Note that the evaluation approach used in our trials is the one-vs-all strategy. It sees the other classes as typical, and one class as abnormal.

We can confirm that Dirichlet, Beta-Liouville, and Libby-Novick Beta-Liouville perform better than the baseline techniques based on the results in the three tables. This validates (our conjectures) that distributions like Libby-Novick Beta-Liouville and Beta-Liouville are better at differentiating between data that are defined on a simplex.

For the MNIST data set, the LNBL normality score (AUC=96.23) is superior to the Dirichlet normality scores (AUC=95.68), Beta-Liouville normality scores (AUC=93.94), RAW OCSVAM (AUC=93.85), and CAE OCSVM (AUC=85.31), as shown in Table 3.3. Furthermore, LNBL achieves AUC=99.9 for class *6* and has the highest scores for five of the 10 classes: *3,4,5,6,7*. Nevertheless, Dirichlet performs exceptionally well in three classes: *2,6,9*, whereas class *8* receives the highest AUC=93.19 according to Beta-Liouville.

	Raw-OC-SVM	CAE-OC-SVM	Dirichlet	Beta-Liouville	Libby-Novick Beta-Liouville
0	**99.54**	97.25	91.4	88.99	90.24
1	**99.91**	99.28	85.5	73.78	93.8
2	88.5	82.33	**99.4**	99.31	98.88
3	89.07	76.14	96.01	95.96	**99.5**
4	95.46	79.39	98.72	96.69	**98.94**
5	91.17	78.35	98.35	97.82	**99.56**
6	97.06	86.69	**99.9**	**99.9**	**99.9**
7	95.05	86.83	95.58	94.94	**96.03**
8	86.47	74.41	92.94	**93.19**	86.83
9	96.3	92.48	**99.09**	98.91	98.7
mean	93.85	85.31	95.68	93.94	**96.23**

Table 3.3: AUROC MNIST dataset

Upon examining Tables 3.4 and 3.5, it is evident that the baseline methods perform remarkably well when dealing with smaller data samples (the size of an image in Fashion MNIST is 32×32×1, whereas it is 32×32×3 in CIFAR10). The fashion MNIST has the best AUC in four classes: *Trouser,*

Dress, Coat, Sneaker, and the highest test AUC is achieved by Raw OCSVM (AUC=$92,67$). Moreover, it is clear that overall LNBL outperforms both BL and Dirichlet. It has the highest scores across 4 out of 10 classes for Fashion MNIST and has the best average AUC for the CIFAR10 data set (AUC = 80.35). One explanation for this could be that the three extra parameters of LNBL over Dirichlet provide a more stable covariance structure that can address the problem of feature independence in a data set.

	Raw-OC-SVM	CAE-OC-SVM	Dirichlet	Beta-Liouville	Libby-Novick Beta-Liouville
T-shirt/top	92.0	89.1	89.3	90.06	**92.4**
Trouser	**99.1**	97.1	97.86	98.53	97.05
Pullover	89.5	86.1	86.16	88.26	**92.31**
Dress	**92.0**	85.7	88.56	79.46	81.29
Coat	**90.9**	88.4	86.13	87.9	87.4
Sandal	93.2	92.8	**96.16**	93.33	92.08
Shirt	82.1	81.7	78.8	78.53	**84.48**
Sneaker	**98.5**	97.0	97.26	97.83	97.34
Bag	91.2	94.5	**97.73**	93.86	96.84
Ankle boot	98.2	96.2	99.33	99.1	**99.46**
mean	**92.67**	90.86	91.73	90.68	92.06

Table 3.4: AUROC Fashion MNIST dataset

Table 3.6 illustrates the AUC results on the NSL-KDD Cup data set by Dirichlet, BL, and LNBL. As shown in this table, LNBL excels in the two classes with 10% difference in AUC from the nearest score (AUC = 88.14 for LNBL to AUC = 78.35 and AUC = 76.26 for BL and Dirichlet respectively). The AUC of the *attack* class reaches 85.96 for the LNBL normality score, followed by BL with 81.59. For the *normal* class, we can see that Dirichlet and BL have the same performance for the Normal class. Furthermore, we notice that LNBL marks a very high score compared to other distributions (90.33 with a difference of 15% from the nearest score).

	Raw-OC-SVM	CAE-OC-SVM	Dirichlet	Beta-Liouville	Libby-Novick Beta-Liouville
airplane	63	58.23	71.2	**76.76**	72.79
automobile	50	51.4	88.73	**93.46**	89.91
bird	61.8	66.43	73.5	72	**75.18**
cat	54.3	51.53	**73.76**	63.53	69.78
deer	71.6	68.4	**74.06**	74.66	73.27
dog	50	48.66	**82.63**	79.6	77.5
forg	77.4	**79.83**	72.66	74.8	76.84
horse	52.2	55.33	**90.06**	92.36	89.24
ship	70.7	68.3	89.53	89.33	**89.91**
truck	52.4	60.56	85.2	86.6	**89.12**
mean	60.34	60.56	80.13	80.31	**80.35**

Table 3.5: AUROC CIFAR10 dataset

	Dirichlet	Beta-Liouville	Libby-Novick Beta-Liouville
Normal	75.62	75.12	**90.33**
Attack	76.91	81.59	**85.96**
mean	76.26	78.35	**88.14**

Table 3.6: AUROC NSL-KDD Cup dataset

3.5.4 Feature Mapping SVM Results

In this part, our evaluation was based on three subsets of data from MNIST data and three others from Fashion MNIST data. We consider the class 0 the normal class (10000 samples) and we take 2000 samples from each of the following classes: 1, 2, and 3 to serve as anomaly classes for the three subsets respectively (Class 1 is the anomaly class of the first subset, class 2 is the anomaly class of the second subset, and class 3 is the anomaly class of the third subset). Therefore, we fix the anomaly rate to 16%. We choose to work with three kernels: Linear, RBF, and Polynomial. For the Polynomial kernel, we set the degree to 3. The hyperparameter C in the objective function is varied from 0.001 to 50 depending on the experiment. We choose the value that gives us the best performance. We compare the proposed Libby-Novick Beta-Liouville SVM (LNBLSVM) to the baseline SVM, Dirichlet SVM (DSVM), and Beta-Liouville SVM (BLSVM).

Fig(3.5) and Fig(3.6) illustrate the f1 scores for each subset data of both Fashion MNIST and

MNIST datasets, for different proposed kernels. We can observe that the f1 score for the anomalous challenge has greatly increased using our suggested feature mapping strategy. The enhanced separability between the support vectors from every image category is the cause.

For the fashion MNIST data set, in linear feature map, LNBLSVM shows a 4% and 2% improvement in f1 score for the first and the third subsets. In contrast, BLSVM has the best performance for the second subset with f1 score = 89.3. In general, non-linear kernels show a competitive performance between LNBLSVM and BLSVM, while the DSVM outperforms the other approaches only in the polynomial kernel for the third subset with f1 score = 70.38.

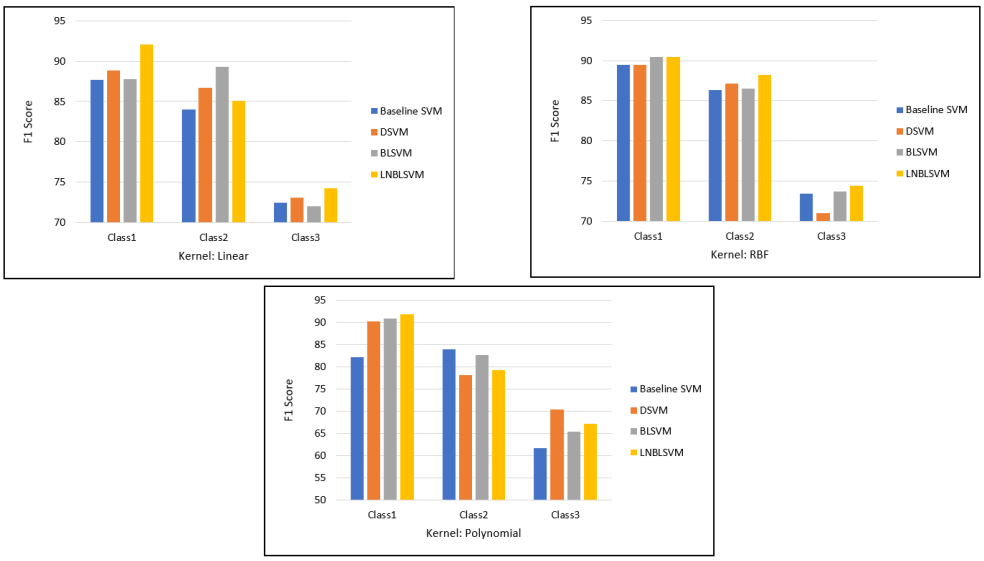


Figure 3.5: F1 score over subsets for the different approaches on Fashion MNIST Data Set

Examining the MNIST data findings, we can confirm that the first subset exhibits a notable level of perfection for LNBLSVM, with a f1 score of 98.49 in the RBF kernel scenario. However, BLSVM performs better than the other methods for the other two subsets, particularly in the non-linear kernel. For example, in the polynomial kernel situation, it scores f1 score = 82.74 for the third data, which is more than 8% from the nearest score. In general, by inspecting the nine scenarios (three kernels with three subsets), it is clear that LNBLSVM performs better than DSVM and BLSVM. The Libby-Novick Beta-Liouville distribution's superior generalization abilities to capture data distribution with a greater coherence covariance structure [9] may be the reason behind

this improvement.

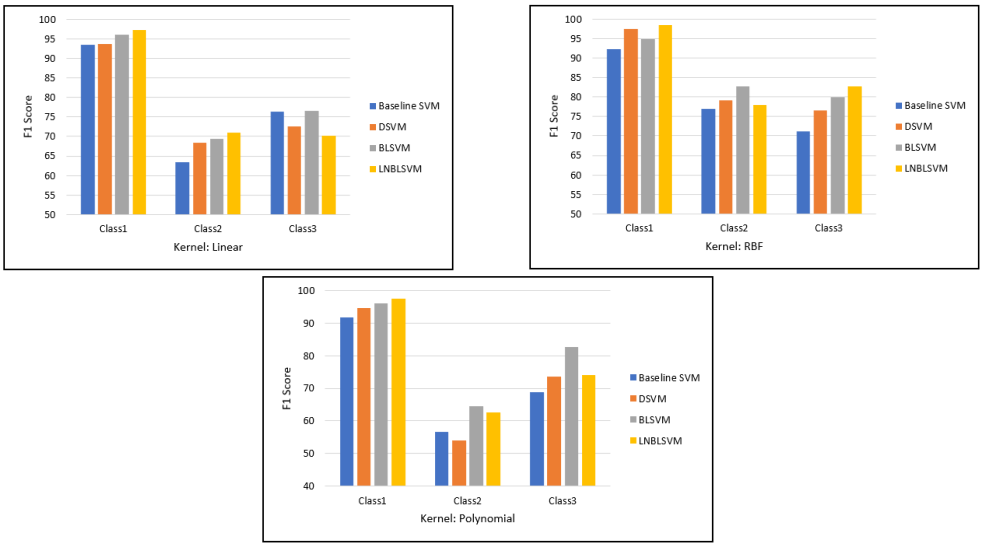


Figure 3.6: F1 score over subsets for the different approaches on MNIST Data Set

Another data set used in our work to measure the performance of our feature mapping strategy for SVM is the bank data. According to Table 3.7, for all the kernels, LNBLSVM outperforms baseline SVM, BLSVM, and DSVM except for the polynomial kernel where baseline SVM achieves a higher accuracy of 86.22 and a higher f1 score of 90.87. Considering the RBF SVM, LNBLSVM gives the highest accuracy of 97.55 and the highest f1 score of 98.52 whereas linear SVM achieves 93.43 accuracy and 95.84 f1 score as the best results. Fig(3.7) shows the distribution of accuracy scores for different approaches. Note that a larger area surrounding the mean in the violin plot indicates a greater likelihood of obtaining a consistent average result. Consequently, it is clear that feature mapping based on the Libby-Novick Beta-Liouville and Beta-Liouville distributions can be employed with classical kernel functions with greater confidence than the Dirichlet distribution. In this case, we consider the LNBLSVM as it has the best performance.

Kernel	Linear		RBF		Polynomial	
Approach	F1 Score	Accuracy	F1 Score	Accuracy	F1 Score	Accuracy
Baseline SVM	94.86	91.95	94.97	92.11	**90.87**	**86.22**
DSVM	94.77	91.82	97.65	96.08	79.93	72.45
BLSVM	94.77	91.82	97.68	96.15	87.51	81.72
LNBLSVM	**95.84**	**93.43**	**98.52**	**97.55**	86.04	79.82

Table 3.7: F1 score and Accuracy for different kernels on Bank Data Set

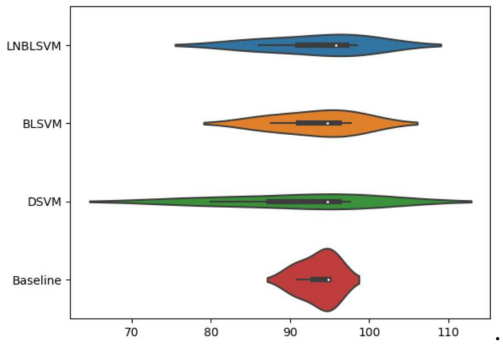


Figure 3.7: Violin plots of experimental results for Bank dataset

Chapter 4

Conclusion

In this thesis, we have addressed the problem of anomaly detection in proportional data using the salient properties of different general forms of Dirichlet distribution in modeling such kind of data.

In Chapter 2, we proposed a deep anomaly detection architecture based on normality score using four general forms of Dirichlet for approximating the softmax output vector predictions of the classifier. In our framework, we choose the appropriate transformations as well as the the appropriate classifier for each kind of data. In order to estimate the parameters of each distribution, we developed the maximum likelihood estimators and we used the Wicker initialization for initializing the parameters. The advantages of the proposed deep anomaly detector were demonstrated through experiments on two image data sets and one non-image data set. A possible future work is to optimize and detect a threshold λ to be compared with the normality score in a way we can have a complete anomaly detection system.

The focus of the second part of this work was the development of generative and discriminative methods for proportional data modeling and anomaly detection tasks. These approaches were based on the Libby-Novick Beta-Liouville distribution produced from the Liouville family which includes the Dirichlet as a specific case. Moreover, these techniques' main benefit is the effective statistical representation of the data's properties through the use of the Libby-Novick Beta-Liouville distribution, which is driven by the abundance of data mining, pattern recognition, and computer vision applications that naturally generate this kind of data. Specifically, a novel mixing model founded on

the Libby-Novick Beta-Liouville distribution has been presented. Additionally, we have suggested a technique that uses the EM approach for estimation to learn the new distribution of finite mixture models. Furthermore, we suggested a deep anomaly detection architecture based on normality score, which approximates the classifier's softmax output vector predictions using Libby-Novick Beta-Liouville. In the third and the last approach, three feature mapping functions for proportional data have been introduced in the SVM learning method. Above all, the three methods' results have unequivocally demonstrated that the Libby-Novick Beta-Liouville outperforms the commonly used Dirichlet and it can be a great substitute for the recently suggested Beta-Liouville. A possible future work is to develop hybrid generative discriminative approaches by deploying SVM kernel generated from Libby-Novick Beta-Liouville mixture models.

Appendix A

Inverse of Hessian Matrix

The second and mixed derivatives of $L(\mathcal{X}|\Theta)$ are given by:

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_{jk_1}\partial\alpha_{jk_2}} = \begin{cases} \sum_{i=1}^{N}\hat{Z}_{ij}\left(\Psi'(\sum_{k=1}^{K}\alpha_{jk}) - \Psi'(\alpha_{jk})\right) & if \quad k_1 = k_2 = k \\ \sum_{i=1}^{N}\hat{Z}_{ij}\Psi'(\sum_{k=1}^{K}\alpha_{jk}) & otherwise \end{cases} \tag{71}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_{jk}\partial\alpha_j} = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_{jk}\partial\beta_j} = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_{jk}\partial\lambda_j} = 0 \tag{72}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\alpha_j} = \left(\Psi'(\alpha_j+\beta_j) - \Psi'(\alpha_j)\right)\sum_{i=1}^{N}\hat{Z}_{ij} \tag{73}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\beta_j} = \left(\Psi'(\alpha_j+\beta_j) - \Psi'(\beta_j)\right)\sum_{i=1}^{N}\hat{Z}_{ij} \tag{74}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\lambda_j} = (\alpha_j+\beta_j)\sum_{i=1}^{N}\left(\hat{Z}_{ij}\frac{(\sum_{k=1}^{K}X_{ik})^2}{\left(1-(1-\lambda_j)\sum_{k=1}^{K}X_{ik}\right)^2}\right) - \frac{\alpha_j}{\lambda_j^2}\sum_{i=1}^{N}\hat{Z}_{ij} \tag{75}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\beta_j} = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\beta_j\partial\alpha_j} = \Psi'(\alpha_j+\beta_j)\sum_{i=1}^{N}\hat{Z}_{ij} \tag{76}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\lambda_j} = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\lambda_j\partial\alpha_j} = \sum_{i=1}^{N}\hat{Z}_{ij}\left(\frac{1}{\lambda_j} - \frac{\sum_{k=1}^{K}X_{ik}}{\left(1-(1-\lambda_j)\sum_{k=1}^{K}X_{ik}\right)}\right) \tag{77}$$

$$\frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\beta_j\partial\lambda_j} = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\lambda_j\partial\beta_j} = \sum_{i=1}^{N} \frac{-\hat{Z}_{ij}\sum_{k=1}^{K} X_{ik}}{\left(1-(1-\lambda_j)\sum_{k=1}^{K} X_{ik}\right)} \tag{78}$$

where Ψ and Ψ' are the digamma and trigamma functions respectively. In this way, the hessian matrix has a block-diagonal structure: $H(\theta_j) = \text{block-diag}(H(\alpha_{j1},\cdots,\alpha_{jK}), H(\alpha_j,\beta_j,\lambda_j))$ with:

$$H(\alpha_j,\beta_j,\lambda_j) = \begin{pmatrix} \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\alpha_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\beta_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_j\partial\lambda_j} \\ \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\beta_j\partial\alpha_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\beta_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\beta_j\partial\lambda_j} \\ \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\lambda_j\partial\alpha_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\lambda_j\partial\beta_j} & \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial^2\lambda_j} \end{pmatrix},$$

$$\text{H}(\alpha_{j1},\cdots,\alpha_{jK}) = \frac{\partial^2 L(\mathcal{X}|\Theta)}{\partial\alpha_{jk_1}\partial\alpha_{jk_2}} \quad k_1,k_2 = 1\cdots,K \tag{79}$$

From [[77], Theorem 8.8.16], we have: $H(\theta_j)^{-1} = \text{block-diag}(H(\alpha_{j1},\cdots,\alpha_{jK})^{-1}$, $H(\alpha_j,\beta_j,\lambda_j)^{-1})$. We can write $H(\alpha_{j1},\cdots,\alpha_{jK})$ as $H(\alpha_{j1},\cdots,\alpha_{jK}) = S + \gamma aa^T$, where:

$$S = diag\left[-\Psi'(\alpha_{j1})\sum_{i=1}^{N}\hat{Z}_{ij},\cdots,\Psi'(\alpha_{jK})\sum_{i=1}^{N}\hat{Z}_{ij}\right] \tag{80}$$

$$\gamma = \Psi'(\sum_{k=1}^{K}\alpha_{jk})\sum_{i=1}^{N}\hat{Z}_{ij},\ \ a^T = \mathbf{1},\ \ \gamma \neq \sum_{k=1}^{K}\left(\frac{a_k^2}{S_{kk}}\right)^{-1} \tag{81}$$

Then, the inverse of $H(\alpha_{j1},\cdots,\alpha_{jK})$ is given by: [[77], Theorem 8.3.3]

$$H(\alpha_{j1},\cdots,\alpha_{jK})^{-1} = S^* + \delta a^* a^{*T} \tag{82}$$

$$S^* = S^{-1} = diag\left[\frac{-1}{\Psi'(\alpha_{j1})\sum_{i=1}^{N}\hat{Z}_{ij}},\cdots,\frac{-1}{\Psi'(\alpha_{jK})\sum_{i=1}^{N}\hat{Z}_{ij}}\right] \tag{83}$$

$$a^{*T} = (\frac{a_1}{S_1},\cdots,\frac{a_K}{S_K}) = \left(\frac{-1}{\Psi'(\alpha_{j1})\sum_{i=1}^{N}\hat{Z}_{ij}},\cdots,\frac{-1}{\Psi'(\alpha_{jK})\sum_{i=1}^{N}\hat{Z}_{ij}}\right) \tag{84}$$

$$\delta^* = -\gamma \sum_{k=1}^{K} \left(1 + \gamma \frac{a_k^2}{S_{kk}}\right)^{-1}$$

$$= \Psi'(\sum_{k=1}^{K} \alpha_{jk}) \sum_{i=1}^{N} \hat{Z}_{ij} \left(1 + \Psi'(\sum_{k=1}^{K} \alpha_{jk}) \sum_{i=1}^{N} \hat{Z}_{ij} \sum_{k=1}^{K} \frac{-1}{\Psi'(\alpha_{jk}) \sum_{i=1}^{N} \hat{Z}_{ij}}\right)^{-1} \quad (85)$$

Appendix B

Parameter Estimation of LNBL in SVM Approach

Let $D_{obs} = (X_1, \cdots, X_N)$ be the observed proportional data. Assuming $\boldsymbol{X_i} \sim \text{LNBL}(\boldsymbol{\theta})$ with $\boldsymbol{\theta} = (\alpha_1, \cdots, \alpha_K, \alpha, \beta, \lambda)$. The entire dataset's joint probability function can be calculated in the manner shown below:

$$
\begin{aligned}
p(D_{obs}|\boldsymbol{\theta}) &= \prod_{i=1}^{N} p(\boldsymbol{X_i}|\boldsymbol{\theta}) \\
&= \prod_{i=1}^{N} \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\lambda^{\alpha}(\sum_{k=1}^{K} X_{ik})^{\alpha-\sum_{k=1}^{K}\alpha_k}(1-\sum_{k=1}^{K} X_{ik})^{\beta-1}}{(1-(1-\lambda)\sum_{k=1}^{K} X_{ik})^{\alpha+\beta}} \prod_{k=1}^{K} \frac{X_{ik}^{\alpha_k-1}}{\Gamma(\alpha_k)}
\end{aligned} \tag{86}
$$

Our goal is to maximize Eq(86), for that, we calculate the gradient and set it to zero. We tackle the Maximum Likelihood Estimation (MLE) approach to estimate the parameters. Taking the logarithm of the distribution stated above will make it a convex optimization problem since it comes from the exponential family [78].

$$l = \log p(D_{obs}|\boldsymbol{\theta}) = N\left(\log\Gamma(\sum_{k=1}^{K}\alpha_k) + \log\Gamma(\alpha+\beta) - \log\Gamma(\alpha) - \log\Gamma(\beta)\right) + N\alpha\log\lambda$$
$$+ (\alpha - \sum_{k=1}^{K}\alpha_k)\sum_{i=1}^{N}\left(\log\sum_{k=1}^{K}X_{ik}\right) + (\beta-1)\sum_{i=1}^{N}\log(1-\sum_{k=1}^{K}X_{ik})$$
$$- (\alpha+\beta)\sum_{i=1}^{N}\log\left(1-(1-\lambda)\sum_{k=1}^{K}X_{ik}\right) + \sum_{i=1}^{N}\sum_{k=1}^{K}\log X_{ik} - N\sum_{k=1}^{K}\log\Gamma(\alpha_k) \quad (87)$$

The dataset's gradient is K×1, and it may be expressed in the following way:

$$\nabla l = \begin{pmatrix} \frac{\partial l}{\partial \alpha_1} \\ \vdots \\ \frac{\partial l}{\partial \alpha_K} \\ \\ \frac{\partial l}{\partial \alpha} \\ \\ \\ \frac{\partial l}{\partial \beta} \\ \\ \\ \frac{\partial l}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} N\left(\Psi(\sum_{k=1}^{K}\alpha_k) - \Psi(\alpha_1)\right) + \sum_{i=1}^{N}\left(\log(X_{ik}) - \log(\sum_{k=1}^{K}X_{ik})\right) \\ \vdots \\ N\left(\Psi(\sum_{k=1}^{K}\alpha_k) - \Psi(\alpha_K)\right) + \sum_{i=1}^{N}\left(\log(X_{ik}) - \log(\sum_{k=1}^{K}X_{ik})\right) \\ \\ N\left(\Psi(\alpha+\beta) - \Psi(\alpha)\right) + N\log(\lambda_j) \\ + \sum_{i=1}^{N}\left(\log(\sum_{k=1}^{K}X_{ik}) + \log(1-(1-\lambda)\sum_{k=1}^{K}X_{ik})\right) \\ \\ N\left(\Psi(\alpha+\beta) - \Psi(\beta)\right) + \sum_{i=1}^{N}\left(\log(1-\sum_{k=1}^{K}X_{ik})\right. \\ \left. + \log(1-(1-\lambda)\sum_{k=1}^{K}X_{ik})\right) \\ \\ N\frac{\alpha}{\lambda} - (\alpha+\beta)\sum_{i=1}^{N}\frac{\sum_{k=1}^{K}X_{ik}}{(1-(1-\lambda)\sum_{k=1}^{K}X_{ik})} \end{pmatrix} \quad (88)$$

Using Newton Raphson algorithm, we can update the parameters in the following way:

$$\boldsymbol{\theta_{new}} = \boldsymbol{\theta_{old}} - H^{-1}\nabla l(\boldsymbol{\theta_{old}}) \quad (89)$$

where H is the hessian of the log-likelihood function. The inverse of H can be calculated in the same way as in Appendix A.

An important step for estimating the parameters is the setting of the initial values of the parameters accurately. Method of moments was previously used to provide an initial guess of the parameters

in [42] and [47]. However, it may affect the robustness of the algorithm. If the Newton-Raphson method is started, for instance, at a point where the derivative is close to zero, it may not converge or converge slowly. In our work, we choose to initialize the parameters with random values from a uniform distribution. Moreover, we are less likely to become caught in local minima or maxima if we randomly initialize with numbers within the given range (a,b). This unpredictability may aid in the early exploration of a larger search field and lead to more effective solutions.

Bibliography

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[2] Emin Aleskerov, Bernd Freisleben, and Bharat Rao. Cardwatch: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 computational intelligence for financial engineering (CIFEr)*, pages 220–226. IEEE, 1997.

[3] Clay Spence, Lucas Parra, and Paul Sajda. Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In *Proceedings IEEE workshop on mathematical methods in biomedical image analysis (MMBIA 2001)*, pages 3–10. IEEE, 2001.

[4] Nizar Bouguila and Djemel Ziou. Dirichlet-based probability model applied to human skin detection [image skin detection]. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–521. IEEE, 2004.

[5] Wentao Fan and Nizar Bouguila. Online data clustering using variational learning of a hierarchical dirichlet process mixture of dirichlet distributions. In *Database Systems for Advanced Applications: 19th International Conference, DASFAA 2014, International Workshops: BDMA, DaMEN, SIM^3, UnCrowd; Bali, Indonesia, April 21–24, 2014, Revised Selected Papers 19*, pages 18–32. Springer, 2014.

[6] Tzu-Tsung Wong. Generalized dirichlet distribution in bayesian analysis. *Applied Mathematics and Computation*, 97(2-3):165–181, 1998.

[7] Gianna Serafina Monti, Glòria Mateu i Figueras, Vera Pawlowsky-Glahn, and Juan José Egozcue. The shifted-scaled dirichlet distribution in the simplex. In © *International Workshop on Compositional Data Analysis (4th: 2011: Sant Feliu de GuÍxols, Girona). CODAWORK 2011: International Workshop on Compositional Data Analysis, hold on May 9-13rd. 2011, Sant Feliu de Guíxols, Girona*. Universitat Politècnica de Catalunya. Centre Internacional de Mètodes . . . , 2011.

[8] Robert H Lochner. A generalized dirichlet distribution in bayesian life testing. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 37(1):103–113, 1975.

[9] Gauss M Cordeiro, Luís H de Santana, Edwin MM Ortega, and Rodrigo R Pescim. A new family of distributions: Libby-novick beta. *International Journal of Statistics and Probability*, 3(2):63, 2014.

[10] Christoph Baur, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4*, pages 161–169. Springer, 2019.

[11] Chong Zhou and Randy C Paffenroth. Anomaly detection with robust deep autoencoders. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 665–674, 2017.

[12] Chunyong Yin, Sun Zhang, Jin Wang, and Neal N Xiong. Anomaly detection based on convolutional recurrent autoencoder for iot time series. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(1):112–122, 2020.

[13] Jinwon An and Sungzoon Cho. Variational autoencoder based anomaly detection using reconstruction probability. *Special lecture on IE*, 2(1):1–18, 2015.

[14] G.J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley, 2004.

[15] N. Bouguila and W. Fan. *Mixture Models and Applications*. Unsupervised and Semi-Supervised Learning. Springer International Publishing, 2020.

[16] Wentao Fan and Nizar Bouguila. Online learning of a dirichlet process mixture of beta-liouville distributions via variational inference. *IEEE transactions on neural networks and learning systems*, 24(11):1850–1862, 2013.

[17] Wentao Fan and Nizar Bouguila. Modeling and clustering positive vectors via nonparametric mixture models of liouville distributions. *IEEE transactions on neural networks and learning systems*, 31(9):3193–3203, 2019.

[18] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[19] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[20] Zhangyu Cheng, Chengming Zou, and Jianwei Dong. Outlier detection using isolation forest and local outlier factor. In *Proceedings of the conference on research in adaptive and convergent systems*, pages 161–168, 2019.

[21] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[22] Samr Ali and Nizar Bouguila. Variational learning of beta-liouville hidden markov models for infrared action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019.

[23] Samr Ali and Nizar Bouguila. Maximum a posteriori approximation of hidden markov models for proportional sequential data modeling with simultaneous feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, 33(10):5590–5601, 2021.

[24] Xiaoran Chen and Ender Konukoglu. Unsupervised detection of lesions in brain mri using constrained adversarial auto-encoders. *arXiv preprint arXiv:1806.04972*, 2018.

[25] David Zimmerer, Simon AA Kohl, Jens Petersen, Fabian Isensee, and Klaus H Maier-Hein. Context-encoding variational autoencoder for unsupervised anomaly detection. *arXiv preprint arXiv:1812.05941*, 2018.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[27] Emily L Denton, Soumith Chintala, Rob Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. *Advances in neural information processing systems*, 28, 2015.

[28] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[29] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[30] Nan Ding, HaoXuan Ma, Huanbo Gao, YanHua Ma, and GuoZhen Tan. Real-time anomaly detection based on long short-term memory and gaussian mixture model. *Computers & Electrical Engineering*, 79:106458, 2019.

[31] Wentao Fan and Nizar Bouguila. Learning finite beta-liouville mixture models via variational bayes for proportional data clustering. In *Twenty-Third International Joint Conference on Artificial Intelligence*. Citeseer, 2013.

[32] Wentao Fan and Nizar Bouguila. Expectation propagation learning of a dirichlet process mixture of beta-liouville distributions for proportional data clustering. *Engineering Applications of Artificial Intelligence*, 43:1–14, 2015.

[33] Nizar Bouguila, Djemel Ziou, and Jean Vaillancourt. Unsupervised learning of a finite mixture model based on the dirichlet distribution and its application. *IEEE Transactions on Image Processing*, 13(11):1533–1543, 2004.

[34] Nizar Bouguila and Djemel Ziou. High-dimensional unsupervised selection and estimation of a finite generalized dirichlet mixture model based on minimum message length. *IEEE transactions on pattern analysis and machine intelligence*, 29(10):1716–1731, 2007.

[35] Wentao Fan, Nizar Bouguila, and Djemel Ziou. Variational learning for finite dirichlet mixture models and applications. *IEEE transactions on neural networks and learning systems*, 23(5):762–774, 2012.

[36] Nizar Bouguila and Djemel Ziou. Unsupervised selection of a finite dirichlet mixture model: an mml-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):993–1009, 2006.

[37] Henry YT Ngan, Nelson HC Yung, and Anthony GO Yeh. Outlier detection in traffic data based on the dirichlet process mixture model. *IET intelligent transport systems*, 9(7):773–781, 2015.

[38] Nour Moustafa, Gideon Creech, and Jill Slay. Big data analytics for intrusion detection system: Statistical decision-making using finite dirichlet mixture models. *Data Analytics and Decision Support for Cybersecurity: Trends, Methodologies and Applications*, pages 127–156, 2017.

[39] Nizar Bouguila. Hybrid generative/discriminative approaches for proportional data modeling and classification. *IEEE Transactions on Knowledge and Data Engineering*, 24(12):2184–2202, 2011.

[40] Nizar Bouguila. Bayesian hybrid generative discriminative learning based on finite liouville mixture models. *Pattern Recognition*, 44(6):1183–1200, 2011.

[41] Ali Sefidpour and Nizar Bouguila. Spatial color image segmentation based on finite non-gaussian mixture models. *Expert Systems with Applications*, 39(10):8993–9001, 2012.

[42] Izhak Golan and Ran El-Yaniv. Deep anomaly detection using geometric transformations. *Advances in neural information processing systems*, 31, 2018.

[43] Yu Zhang, Xiushan Nie, Rundong He, Meng Chen, and Yilong Yin. Normality learning in multispace for video anomaly detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(9):3694–3706, 2020.

[44] Mehdi Hosseinzadeh, Amir Masoud Rahmani, Bay Vo, Moazam Bidaki, Mohammad Masdari, and Mehran Zangakani. Improving security using svm-based anomaly detection: issues and challenges. *Soft Computing*, 25:3195–3223, 2021.

[45] Sarah M Erfani, Sutharshan Rajasegarar, Shanika Karunasekera, and Christopher Leckie. High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning. *Pattern Recognition*, 58:121–134, 2016.

[46] Ali Nedaie and Amir Abbas Najafi. Support vector machine with dirichlet feature mapping. *Neural Networks*, 98:87–101, 2018.

[47] Md Hafizur Rahman and Nizar Bouguila. Efficient feature mapping in classifying proportional data. *IEEE Access*, 9:3712–3724, 2020.

[48] Sghaier O Amayri M, Bouguila N. Dirichlet and liouville-based normality scores for deep anomaly detection using transformations: Applications to images and beyond images. *ieee transactions on neural networks and learning systems*, Manuscript Submitted.

[49] Sghaier O Amayri M, Bouguila N. Libby-novick beta-liouville distribution for enhanced anomaly detection in proportional data. *Transactions on Intelligent Systems and Technology*, Manuscript Submitted.

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[51] Thomas Minka. Estimating a dirichlet distribution, 2000.

[52] Divya Ankam and Nizar Bouguila. Generalized dirichlet regression and other compositional models with application to market-share data mining of information technology companies. In *ICEIS (1)*, pages 158–166, 2019.

[53] Elise Epaillard and Nizar Bouguila. Variational bayesian learning of generalized dirichlet-based hidden markov models applied to unusual events detection. *IEEE transactions on neural networks and learning systems*, 30(4):1034–1047, 2018.

[54] Bromensele Samuel Oboh and Nizar Bouguila. Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization. In *2017 IEEE international conference on industrial technology (ICIT)*, pages 1085–1090. IEEE, 2017.

[55] Robin KS Hankin. A generalization of the dirichlet distribution. *Journal of Statistical Software*, 33:1–18, 2010.

[56] Nuha Zamzami and Nizar Bouguila. Probabilistic modeling for frequency vectors using a flexible shifted-scaled dirichlet distribution prior. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 14(6):1–35, 2020.

[57] Rua Alsuroji, Nizar Bouguila, and Nuha Zamzami. Predicting defect-prone software modules using shifted-scaled dirichlet distribution. In *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*, pages 15–18. IEEE, 2018.

[58] Narges Manouchehri, Oumayma Dalhoumi, Manar Amayri, and Nizar Bouguila. Variational learning of a shifted scaled dirichlet model with component splitting approach. In *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, pages 75–78. IEEE, 2020.

[59] Sami Bourouis, Abdullah Alharbi, and Nizar Bouguila. Bayesian learning of shifted-scaled dirichlet mixture models and its application to early covid-19 detection in chest x-ray images. *Journal of Imaging*, 7(1):7, 2021.

[60] Can Hu, Wentao Fan, Ji-Xiang Du, and Nizar Bouguila. A novel statistical approach for clustering positive data based on finite inverted beta-liouville mixture models. *Neurocomputing*, 333:110–123, 2019.

[61] Nizar Bouguila and Tarek Elguebaly. A fully bayesian model based on reversible jump mcmc and finite beta mixtures for clustering. *Expert Systems with Applications*, 39(5):5946–5959, 2012.

[62] Nizar Bouguila. Infinite liouville mixture models with application to text and texture categorization. *Pattern Recognition Letters*, 33(2):103–110, 2012.

[63] Alex Krizhevsky and Geoff Hinton. Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9, 2010.

[64] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[65] Nicolas Wicker, Jean Muller, Ravi Kiran Reddy Kalathur, and Olivier Poch. A maximum likelihood approximation method for dirichlet's parameter estimation. *Computational statistics & data analysis*, 52(3):1315–1322, 2008.

[66] Lucas Deecke, Robert Vandermeulen, Lukas Ruff, Stephan Mandt, and Marius Kloft. Image anomaly detection with generative adversarial networks. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14, 2018, Proceedings, Part I 18*, pages 3–17. Springer, 2019.

[67] Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. Deep structured energy based models for anomaly detection. In *International conference on machine learning*, pages 1100–1109. PMLR, 2016.

[68] Danijela D Protić. Review of kdd cup '99, nsl-kdd and kyoto 2006+ datasets. *Vojnotehnički glasnik/Military Technical Courier*, 66(3):580–596, 2018.

[69] Nizar Bouguila, Djemel Ziou, and Ernest Monga. Practical bayesian estimation of a finite beta mixture through gibbs sampling and its applications. *Statistics and Computing*, 16:215–225, 2006.

[70] Zafar Iqbal, Muhammad Rashad, and Muhammad Hanif. Properties of the libby-novick beta distribution with application. *International Journal of Analysis and Applications*, 19(3):360–388, 2021.

[71] Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

[72] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[73] Xuchun Li, Lei Wang, and Eric Sung. Adaboost with svm-based component classifiers. *Engineering Applications of Artificial Intelligence*, 21(5):785–795, 2008.

[74] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[75] Alejandro Baldominos, Yago Saez, and Pedro Isasi. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169, 2019.

[76] https://www.kaggle.com/datasets/sahistapatel96/bankadditionalfullcsv.

[77] Franklin A Graybill. Matrices with applications in statistics. *(No Title)*, 1983.

[78] Jonathan Huang. Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique report*, 18, 2005.